Virtualization documentation

Virtualization in Windows Server is one of the foundational technologies required to create your software defined infrastructure. Along with networking and storage, virtualization features deliver the flexibility you need to power workloads for your customers.

Windows Containers

OVERVIEW

Windows Containers

About Window containers

Containers vs. virtual machines

Hyper-V on Windows

OVERVIEW

Hyper-V on Windows overview

About Hyper-V on Windows

GET STARTED

Install Hyper-V

Create a Virtual Machine

Hyper-V on Windows Server

OVERVIEW

Hyper-V overview

System requirements for Hyper-V

Supported Windows guest operating systems

GET STARTED

Install the Hyper-V role

Create a virtual machine

Hyper-V Virtual Switch

OVERVIEW

Hyper-V Virtual Switch

HOW-TO GUIDE

Manage Hyper-V Virtual Switch

Use a guarded fabric to provide a secure environment for VMs

OVERVIEW

Guarded Fabric and Shielded VMs

Guarded fabric and shielded VMs

Article • 06/09/2022

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

One of the most important goals of providing a hosted environment is to guarantee the security of the virtual machines running in the environment. As a cloud service provider or enterprise private cloud administrator, you can use a guarded fabric to provide a more secure environment for VMs. A guarded fabric consists of one Host Guardian Service (HGS) - typically, a cluster of three nodes - plus one or more guarded hosts, and a set of shielded virtual machines (VMs).

> **ⓘ Important**
>
> Ensure that you have installed the latest cumulative update before you deploy shielded virtual machines in production.

Videos, blog, and overview topic about guarded fabrics and shielded VMs

- Video: How to protect your virtualization fabric from insider threats with Windows Server 2019
- Video: Introduction to Shielded Virtual Machines in Windows Server 2016
- Video: Dive into Shielded VMs with Windows Server 2016 Hyper-V
- Video: Deploying Shielded VMs and a Guarded Fabric with Windows Server 2016
- Blog: Datacenter and Private Cloud Security Blog
- Overview: Guarded fabric and shielded VMs overview

Planning topics

- Planning guide for hosters
- Planning guide for tenants

Deployment topics

- Deployment Guide
  - Quick start

- Deploy HGS
- Deploy guarded hosts
  - Configuring the fabric DNS for hosts that will become guarded hosts
  - Deploy a guarded host using AD mode
  - Deploy a guarded host using TPM mode
  - Confirm guarded hosts can attest
  - Shielded VMs - Hosting service provider deploys guarded hosts in VMM
- Deploy shielded VMs
  - Create a shielded VM template
  - Prepare a VM Shielding helper VHD
  - Set up Windows Azure Pack
  - Create a shielding data file
  - Deploy a shielded VM by using Windows Azure Pack
  - Deploy a shielded VM by using Virtual Machine Manager

Operations and management topic

- Managing the Host Guardian Service

Hyper-V Technology Overview

Article • 08/13/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

> **ⓘ Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Hyper-V is Microsoft's hardware virtualization product. It lets you create and run a software version of a computer, called a *virtual machine*. Each virtual machine acts like a complete computer, running an operating system and programs. When you need computing resources, virtual machines give you more flexibility, help save time and money, and are a more efficient way to use hardware than just running one operating system on physical hardware.

Hyper-V runs each virtual machine in its own isolated space, which means you can run more than one virtual machine on the same hardware at the same time. You might want to do this to avoid problems such as a crash affecting the other workloads, or to give different people, groups or services access to different systems.

Some ways Hyper-V can help you

Hyper-V can help you:

- **Establish or expand a private cloud environment.** Provide more flexible, on-demand IT services by moving to or expanding your use of shared resources and adjust utilization as demand changes.
- **Use your hardware more effectively.** Consolidate servers and workloads onto fewer, more powerful physical computers to use less power and physical space.
- **Improve business continuity.** Minimize the impact of both scheduled and unscheduled downtime of your workloads.

- **Establish or expand a virtual desktop infrastructure (VDI).** Use a centralized desktop strategy with VDI can help you increase business agility and data security, as well as simplify regulatory compliance and manage desktop operating systems and applications. Deploy Hyper-V and Remote Desktop Virtualization Host (RD Virtualization Host) on the same server to make personal virtual desktops or virtual desktop pools available to your users.

- **Make development and test more efficient.** Reproduce different computing environments without having to buy or maintain all the hardware you'd need if you only used physical systems.

Hyper-V and other virtualization products

Hyper-V in Windows and Windows Server replaces older hardware virtualization products, such as Microsoft Virtual PC, Microsoft Virtual Server, and Windows Virtual PC. Hyper-V offers networking, performance, storage and security features not available in these older products.

Hyper-V and most third-party virtualization applications that require the same processor features aren't compatible. That's because the processor features, known as hardware virtualization extensions, are designed to not be shared. For details, see Virtualization applications do not work together with Hyper-V, Device Guard, and Credential Guard.

What features does Hyper-V have?

Hyper-V offers many features. This is an overview, grouped by what the features provide or help you do.

Computing environment - A Hyper-V virtual machine includes the same basic parts as a physical computer, such as memory, processor, storage, and networking. All these parts have features and options that you can configure different ways to meet different needs. Storage and networking can each be considered categories of their own, because of the many ways you can configure them.

Disaster recovery and backup - For disaster recovery, Hyper-V Replica creates copies of virtual machines, intended to be stored in another physical location, so you can restore the virtual machine from the copy. For backup, Hyper-V offers two types. One uses saved states and the other uses Volume Shadow Copy Service (VSS) so you can make application-consistent backups for programs that support VSS.

Optimization - Each supported guest operating system has a customized set of services and drivers, called *integration services*, that make it easier to use the operating system in

a Hyper-V virtual machine.

Portability - Features such as live migration, storage migration, and import/export make it easier to move or distribute a virtual machine.

Remote connectivity - Hyper-V includes Virtual Machine Connection, a remote connection tool for use with both Windows and Linux. Unlike Remote Desktop, this tool gives you console access, so you can see what's happening in the guest even when the operating system isn't booted yet.

Security - Secure boot and shielded virtual machines help protect against malware and other unauthorized access to a virtual machine and its data.

For a summary of the features introduced in this version, see What's new in Hyper-V on Windows Server. Some features or parts have a limit to how many can be configured. For details, see Plan for Hyper-V scalability in Windows Server 2016.

How to get Hyper-V

Hyper-V is available in Windows Server and Windows, as a server role available for x64 versions of Windows Server. For server instructions, see Install the Hyper-V role on Windows Server. On Windows, it's available as feature in some 64-bit versions of Windows. It's also available as a downloadable, standalone server product, Microsoft Hyper-V Server.

Supported operating systems

Many operating systems will run on virtual machines. In general, an operating system that uses an x86 architecture will run on a Hyper-V virtual machine. Not all operating systems that can be run are tested and supported by Microsoft, however. For lists of what's supported, see:

- Supported Linux and FreeBSD virtual machines for Hyper-V on Windows
- Supported Windows guest operating systems for Hyper-V on Windows Server

How Hyper-V works

Hyper-V is a hypervisor-based virtualization technology. Hyper-V uses the Windows hypervisor, which requires a physical processor with specific features. For hardware details, see System requirements for Hyper-V on Windows Server.

In most cases, the hypervisor manages the interactions between the hardware and the virtual machines. This hypervisor-controlled access to the hardware gives virtual machines the isolated environment in which they run. In some configurations, a virtual machine or the operating system running in the virtual machine has direct access to graphics, networking, or storage hardware.

What does Hyper-V consist of?

Hyper-V has required parts that work together so you can create and run virtual machines. Together, these parts are called the virtualization platform. They're installed as a set when you install the Hyper-V role. The required parts include Windows hypervisor, Hyper-V Virtual Machine Management Service, the virtualization WMI provider, the virtual machine bus (VMbus), virtualization service provider (VSP) and virtual infrastructure driver (VID).

Hyper-V also has tools for management and connectivity. You can install these on the same computer that Hyper-V role is installed on, and on computers without the Hyper-V role installed. These tools are:

- Hyper-V Manager
- Hyper-V module for Windows PowerShell
- Virtual Machine Connection (sometimes called VMConnect)
- Windows PowerShell Direct

Related technologies

These are some technologies from Microsoft that are often used with Hyper-V:

- Failover Clustering
- Remote Desktop Services
- System Center Virtual Machine Manager
- Client Hyper-V

Various storage technologies: cluster shared volumes, SMB 3.0, storage spaces direct

Windows containers offer another approach to virtualization. See the Windows Containers library on MSDN.

Feedback

Was this page helpful? Yes No

System requirements for Hyper-V on Windows Server

Article • 08/13/2024

> Applies to: Windows Server 2022 (preview), Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

> **Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Hyper-V has specific hardware requirements, and some Hyper-V features have additional requirements. Use the details in this article to decide what requirements your system must meet so you can use Hyper-V the way you plan to. Then, review the Windows Server catalog. Keep in mind that requirements for Hyper-V exceed the general minimum requirements for Windows Server because a virtualization environment requires more computing resources.

If you're already using Hyper-V, it's likely that you can use your existing hardware. The general hardware requirements haven't changed significantly from Windows Server 2012 R2. But, you'll need newer hardware to use shielded virtual machines or discrete device assignment. Those features rely on specific hardware support, as described below. Other than that, the main difference in hardware is that second-level address translation (SLAT) is now required instead of recommended.

For details about maximum supported configurations for Hyper-V, such as the number of running virtual machines, see Plan for Hyper-V scalability in Windows Server. The list of operating systems you can run in your virtual machines is covered in Supported Windows guest operating systems for Hyper-V on Windows Server.

General requirements

Regardless of the Hyper-V features you want to use, you'll need:

- A 64-bit processor with second-level address translation (SLAT). To install the Hyper-V virtualization components such as Windows hypervisor, the processor must have SLAT. However, it's not required to install Hyper-V management tools

like Virtual Machine Connection (VMConnect), Hyper-V Manager, and the Hyper-V cmdlets for Windows PowerShell. See "How to check for Hyper-V requirements," below, to find out if your processor has SLAT.

- VM Monitor Mode extensions
- Enough memory - plan for *at least* 4 GB of RAM. More memory is better. You'll need enough memory for the host and all virtual machines that you want to run at the same time.
- Virtualization support turned on in the BIOS or UEFI:
  - Hardware-assisted virtualization. This is available in processors that include a virtualization option - specifically processors with Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) technology.
  - Hardware-enforced Data Execution Prevention (DEP) must be available and enabled. For Intel systems, this is the XD bit (execute disable bit). For AMD systems, this is the NX bit (no execute bit).

How to check for Hyper-V requirements

Open Windows PowerShell or a command prompt and type:

Windows Command Prompt

```
Systeminfo.exe
```

Scroll to the Hyper-V Requirements section to review the report.

Requirements for specific features

This section lists the requirements for discrete device assignment and shielded virtual machines.

Discrete device assignment

Host requirements are similar to the existing requirements for the SR-IOV feature in Hyper-V.

- The processor must have either Intel's Extended Page Table (EPT) or AMD's Nested Page Table (NPT).

- The chipset must have:
  - Interrupt remapping - Intel's VT-d with the Interrupt Remapping capability (VT-d2) or any version of AMD I/O Memory Management Unit (I/O MMU).
  - DMA remapping - Intel's VT-d with Queued Invalidations or any AMD I/O MMU.
  - Access control services (ACS) on PCI Express root ports.
- The firmware tables must expose the I/O MMU to the Windows hypervisor. Note that this feature might be turned off in the UEFI or BIOS. For instructions, see the hardware documentation or contact your hardware manufacturer.

Devices need GPU or non-volatile memory express (NVMe). For GPU, only certain devices support discrete device assignment. To verify, see the hardware documentation or contact your hardware manufacturer. For details about this feature, including how to use it and considerations, see the post "Discrete Device Assignment -- Description and background " in the Virtualization blog.

Shielded virtual machines

These virtual machines rely on virtualization-based security and are available starting with Windows Server 2016.

Host requirements are:

- UEFI 2.3.1c - supports secure, measured boot

  The following two are optional for virtualization-based security in general, but required for the host if you want the protection these features provide:

- TPM v2.0 - protects platform security assets
- IOMMU (Intel VT-D) - so the hypervisor can provide direct memory access (DMA) protection

Virtual machine requirements are:

- Generation 2
- Windows Server 2012 or newer as the guest operating system

Feedback

Was this page helpful? Yes No

Supported Windows guest operating systems for Hyper-V on Windows Server and Azure Stack HCI

Article • 08/13/2024

Hyper-V supports several versions of Windows Server, Windows, and Linux distributions to run in virtual machines, as guest operating systems. This article covers supported Windows Server and Windows guest operating systems. For Linux and FreeBSD distributions, see Supported Linux and FreeBSD virtual machines for Hyper-V on Windows.

Some operating systems have the integration services built-in. Others require that you install or upgrade integration services as a separate step after you set up the operating system in the virtual machine. For more information, see the following sections and Integration Services.

Guest operating systems configurable components are confined based on the hosting operating system. To learn more about the maximum configurable components in Hyper-V, see Plan for Hyper-V scalability in Windows Server.

Supported Windows Server guest operating systems

Following are the versions of Windows Server that are supported as guest operating systems for Hyper-V on Windows Server.

Expand table

Guest operating system (server)	Maximum number of virtual processors	Integration Services	Notes
Windows Server 2025 (preview)	2,048 for generation 2; 64 for generation 1; 2,048 available to the host OS (root partition)	Built-in	Hosted on Windows Server 2025 and later

Guest operating system (server)	Maximum number of virtual processors	Integration Services	Notes
Windows Server 2022	1,024 for generation 2; 64 for generation 1; 1,024 available to the host OS (root partition)	Built-in	Hosted on Windows Server 2019 and later, Azure Stack HCI, version 20H2 and later.
Windows Server 2019	240 for generation 2; 64 for generation 1; 320 available to the host OS (root partition)	Built-in	
Windows Server 2016	240 for generation 2; 64 for generation 1; 320 available to the host OS (root partition)	Built-in	
Windows Server 2012 R2	64	Built-in	
Windows Server 2012	64	Built-in	
Windows Server 2008 R2 with Service Pack 1 (SP 1)	64	Install all critical Windows updates after you set up the guest operating system.	Datacenter, Enterprise, Standard, and Web editions.
Windows Server 2008 with Service Pack 2 (SP2)	8	Install all critical Windows updates after you set up the guest operating system.	Datacenter, Enterprise, Standard, and Web editions (32-bit and 64-bit).

Supported Windows client guest operating systems

Following are the versions of Windows client that are supported as guest operating systems for Hyper-V on Windows Server.

Expand table

Guest operating system (client)	Maximum number of virtual processors	Integration Services	Notes
Windows 11	32	Built-in	Generation 2 virtual machine hosted on Windows Server 2019 or above Azure Stack HCI, version 20H2 and later.
Windows 10	32	Built-in	
Windows 8.1	32	Built-in	
Windows 7 with Service Pack 1 (SP1)	4	Upgrade the integration services after you set up the guest operating system.	Ultimate, Enterprise, and Professional editions (32-bit and 64-bit).

Guest operating system support on other versions of Windows

The following table gives links to information about guest operating systems supported for Hyper-V on other versions of Windows.

Expand table

Host operating system	Article
Windows 10, 11	Supported Guest Operating Systems for Client Hyper-V in Windows 10
Windows Server 2012 R2 and Windows 8.1	- Supported Windows Guest Operating Systems for Hyper-V in Windows Server 2012 R2 and Windows 8.1 - Linux and FreeBSD Virtual Machines on Hyper-V
Windows Server 2012 and Windows 8	Supported Windows Guest Operating Systems for Hyper-V in Windows Server 2012 and Windows 8
Windows Server 2008 and Windows Server 2008 R2	About Virtual Machines and Guest Operating Systems

How Microsoft provides support for guest operating systems

Microsoft provides support for guest operating systems in the following manner:

- Issues found in Microsoft operating systems and in integration services are supported by Microsoft support.
- For issues found in other operating systems that have been certified by the operating system vendor to run on Hyper-V, support is provided by the vendor.
- For issues found in other operating systems, Microsoft submits the issue to the multi-vendor support community, TSANet.

Related links

- Linux and FreeBSD Virtual Machines on Hyper-V
- Supported Guest Operating Systems for Client Hyper-V in Windows

---

Feedback

Was this page helpful? Yes No

Supported Linux and FreeBSD virtual machines for Hyper-V on Windows Server and Windows

Article • 08/13/2024

> Applies to: Azure Stack HCI, Windows Server 2025 (preview), Windows Server 2022, Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

ⓘ Important

Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Hyper-V supports both emulated and Hyper-V-specific devices for Linux and FreeBSD virtual machines. When running with emulated devices, no additional software is required to be installed. However emulated devices do not provide high performance and cannot leverage the rich virtual machine management infrastructure that the Hyper-V technology offers. In order to make full use of all benefits that Hyper-V provides, it is best to use Hyper-V-specific devices for Linux and FreeBSD. The collection of drivers that are required to run Hyper-V-specific devices are known as Linux Integration Services (LIS) or FreeBSD Integration Services (BIS).

LIS has been added to the Linux kernel and is updated for new releases. But Linux distributions based on older kernels may not have the latest enhancements or fixes. Microsoft provides a download containing installable LIS drivers for some Linux installations based on these older kernels. Because distribution vendors include versions of Linux Integration Services, it is best to install the latest downloadable version of LIS, if applicable, for your installation.

For other Linux distributions LIS changes are regularly integrated into the operating system kernel and applications so no separate download or installation is required.

For older FreeBSD releases (before 10.0), Microsoft provides ports that contain the installable BIS drivers and corresponding daemons for FreeBSD virtual machines. For

newer FreeBSD releases, BIS is built in to the FreeBSD operating system, and no separate download or installation is required except for a KVP ports download that is needed for FreeBSD 10.0.

> **Tip**
>
> - Download **Windows Server** from the Evaluation Center.

The goal of this content is to provide information that helps facilitate your Linux or FreeBSD deployment on Hyper-V. Specific details include:

- Linux distributions or FreeBSD releases that require the download and installation of LIS or BIS drivers.
- Linux distributions or FreeBSD releases that contain built-in LIS or BIS drivers.
- Feature distribution maps that indicate the features in major Linux distributions or FreeBSD releases.
- Known issues and workarounds for each distribution or release.
- Feature description for each LIS or BIS feature.

In this section

- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V
- Best practices for running FreeBSD on Hyper-V

Feedback

Was this page helpful? Yes No

Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V

Article • 09/27/2023

> Applies To: Azure Stack HCI, Windows Server 2022, Windows Server Windows Server 2019, Hyper-V Server Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 11, Windows 10, Windows 8.1

The following feature distribution maps indicate the features that are present in built-in and downloadable versions of Linux Integration Services. The known issues and workarounds for each distribution are listed after the tables.

The built-in Red Hat Enterprise Linux Integration Services drivers for Hyper-V (available since Red Hat Enterprise Linux 6.4) are sufficient for Red Hat Enterprise Linux guests to run using the high performance synthetic devices on Hyper-V hosts. These built-in drivers are certified by Red Hat for this use. Certified configurations can be viewed on this Red Hat web page: Red Hat Certification Catalog. It isn't necessary to download and install Linux Integration Services packages from the Microsoft Download Center, and doing so may limit your Red Hat support as described in Red Hat Knowledgebase article 1067: Red Hat Knowledgebase 1067.

Because of potential conflicts between the built-in LIS support and the downloadable LIS support when you upgrade the kernel, disable automatic updates, uninstall the LIS downloadable packages, update the kernel, reboot, and then install the latest LIS release, and reboot again.

> **Note**
>
> Official Red Hat Enterprise Linux certification information is available through the **Red Hat Customer Portal**.

In this section:

- RHEL/CentOS 9.x Series
- RHEL/CentOS 8.x Series

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The kernel module version numbers for the built-in LIS (as shown by **lsmod**, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch does not indicate that the built-in LIS is out of date.
- ✓ - Feature available
- (*blank*) - Feature not available

RHEL/CentOS 9.x Series

Feature	Host OS	9.x
LIS Availability		Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
>256 vCPUs		✓
Networking		
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2

Feature	Host OS	9.x
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Storage		
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Memory		
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9,10
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Video		
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Miscellaneous		

Feature	Host OS	9.x
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Generation 2 virtual machines		
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14, 17
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓

RHEL/CentOS 8.x Series

Feature	Host OS	8.1-8.6+	8.0
LIS Availability		Built in	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
>256 vCPUs		✓	
Networking			
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓

Feature	Host OS	8.1-8.6+	8.0
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
Storage			
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5	✓ Note 5
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Memory			
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2	N/A	N/A

Feature	Host OS	8.1-8.6+	8.0
	Azure Stack HCI		
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10	✓ Note 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10	✓ Note 9, 10
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
Video			
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Miscellaneous			
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓

Feature	Host OS	8.1-8.6+	8.0
Generation 2 virtual machines			
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14, 17	✓ Note 14
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓

RHEL/CentOS 7.x Series

This series only has 64-bit kernels.

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
LIS Availability		LIS 4.3	LIS 4.3	LIS 4.3	Built in	Built in	Built in	Built in	Built in	Built in	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓		✓	✓	✓				
>256 vCPUs					✓ Note 16						
Networking											
Jumbo frames	Windows Server	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
	2022, 2019, 2016, 2012 R2 Azure Stack HCI										
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓		✓	✓	✓				
Storage											
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓		
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2	✓ Note 5	✓ Note 5	✓ Note 5	✓ Note 4,5	✓ Note 4,5	✓ Note 4, 5	✓ Note 4, 5	✓ Note 4, 5	✓ Note 4, 5	✓ Note 4, 5

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
	Azure Stack HCI										
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓		
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓						
Memory											
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Dynamic Memory - Hot-Add	Windows Server	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note	✓ Note

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
	2022, 2019, 2016, 2012 R2 Azure Stack HCI	8, 9, 10	8, 9, 10	8, 9, 10	8, 9, 10	9, 10	9, 10	9, 10	9, 10	9, 10	8, 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10	✓ Note 9, 10
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓	✓							
Video											
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Miscellaneous											
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure	✓	✓	✓	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
	Stack HCI										
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4	
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓							
Hyper-V Sockets	Windows Server 2022, 2019, 2016	✓	✓	✓							
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓		✓	✓	✓	✓			

Feature	Host OS	7.5-7.7	7.3-7.4	7.0-7.2	7.6-7.9	7.5	7.4	7.3	7.2	7.1	7.0
Generation 2 virtual machines											
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

RHEL/CentOS 6.x Series

The 32-bit kernel for this series is PAE enabled. There is no built-in LIS support for RHEL/CentOS 6.0-6.3.

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
LIS Availability		LIS 4.3	LIS 4.3	LIS 4.3	Built in	Built in	Built in	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016							

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
	Azure Stack HCI							
>256 vCPUs								
Networking								
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓		
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2	✓	✓	✓	✓	✓		

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
	Azure Stack HCI							
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI							
Storage								
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5	✓ Note 5	✓ Note 5	✓ Note 4, 5	✓ Note 4, 5	✓ Note 4, 5, 6	✓ Note 4, 5, 6
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓			
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓				
Memory								

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 8, 9, 10	✓ Note 7, 8, 9, 10	
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10, 11
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI							
Video								
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	
Miscellaneous								
Key-Value Pair	Windows Server 2022,	✓	✓	✓	✓ Note	✓ Note	✓ Note	✓ Note

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
	2019, 2016, 2012 R2 Azure Stack HCI				12	12	12, 13	12, 13
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓ Note 12	✓ Note 12		
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓				
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓	✓				
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓						
Generation 2 virtual machines								
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14			

Feature	Host OS	6.7-6.10	6.4-6.6	6.0-6.3	6.10, 6.9, 6.8	6.6, 6.7	6.5	6.4
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI							

RHEL/CentOS 5.x Series

This series has a supported 32-bit PAE kernel available. There is no built-in LIS support for RHEL/CentOS before 5.9.

Feature	Host OS	5.2 -5.11	5.2-5.11	5.9 - 5.11
LIS Availability		LIS 4.3	LIS 4.1	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI			
>256 vCPUs				
Networking				
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 1	✓ Note 1	✓ Note 1
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2			

Feature	Host OS	5.2 -5.11	5.2-5.11	5.9 - 5.11
	Azure Stack HCI			
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Storage				
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3	
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5, 15	✓ Note 5	✓ Note 4, 5, 6
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Memory				
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10, 11	✓ Note 7, 9, 10, 11	

Feature	Host OS	5.2 -5.11	5.2-5.11	5.9 - 5.11
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Video				
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Miscellaneous				
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI			
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Generation 2 virtual machines				
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI			

Notes

1. For this RHEL/CentOS release, VLAN tagging works but VLAN trunking does not.

2. Static IP injection may not work if Network Manager has been configured for a given synthetic network adapter on the virtual machine. For smooth functioning of static IP injection please make sure that either Network Manager is either turned off completely or has been turned off for a specific network adapter through its ifcfg-ethX file.

3. On Windows Server 2012 R2 while using virtual fibre channel devices, make sure that logical unit number 0 (LUN 0) has been populated. If LUN 0 has not been populated, a Linux virtual machine might not be able to mount fibre channel devices natively.

4. For built-in LIS, the "hyperv-daemons" package must be installed for this functionality.

5. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).

6. While the Linux Integration Services download is preferred, live backup support for RHEL/CentOS 5.9 - 5.11/6.4/6.5 is also available through Hyper-V Backup Essentials for Linux.

7. Dynamic memory support is only available on 64-bit virtual machines.

8. Hot-Add support is not enabled by default in this distribution. To enable Hot-Add support you need to add a udev rule under /etc/udev/rules.d/ as follows:

   a. Create a file **/etc/udev/rules.d/100-balloon.rules**. You may use any other desired name for the file.

   b. Add the following content to the file: `SUBSYSTEM=="memory", ACTION=="add", ATTR{state}="online"`

   c. Reboot the system to enable Hot-Add support.

   While the Linux Integration Services download creates this rule on installation, the rule is also removed when LIS is uninstalled, so the rule must be recreated if dynamic memory is needed after uninstallation.

9. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:

- Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
- Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

10. If you are using Dynamic Memory on a Windows Server 2016 or Windows Server 2012 R2 operating system, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to hot-add failures, and you may not see any memory increase in a guest operating system.

11. Certain distributions, including those using LIS 4.0 and 4.1, only provide Ballooning support and do not provide Hot-Add support. In such a scenario, the dynamic memory feature can be used by setting the Startup memory parameter to a value which is equal to the Maximum memory parameter. This results in all the requisite memory being Hot-Added to the virtual machine at boot time and then later depending upon the memory requirements of the host, Hyper-V can freely allocate or deallocate memory from the guest using Ballooning. Please configure **Startup Memory** and **Minimum Memory** at or above the recommended value for the distribution.

12. To enable key/value pair (KVP) infrastructure, install the hypervkvpd or hyperv-daemons rpm package from your RHEL ISO. Alternatively the package can be installed directly from RHEL repositories.

13. The key/value pair (KVP) infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.

14. On Windows Server 2012 R2 Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using PowerShell:

    ```Powershell
    Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
    ```

    The Linux Integration Services download can be applied to existing Generation 2 VMs but does not impart Generation 2 capability.

15. In Red Hat Enterprise Linux or CentOS 5.2, 5.3, and 5.4 the filesystem freeze functionality is not available, so Live Virtual Machine Backup is also not available.
16. For RHEL 7.6, support for >256 vcpus is available in kernel 3.10.0-957.38.1 or later and kernel 3.10.0-1062.4.1 or later is required for RHEL 7.7.
17. RHEL 8.5 requires Windows Server 2019 or newer, or Azure Stack HCI 20H2 or newer.

See Also

- Set-VMFirmware
- Supported Debian virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V
- Red Hat Hardware Certification

Supported Debian virtual machines on Hyper-V

Article • 09/26/2023

> Applies to: Azure Stack HCI, Windows Server 2022, Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

This article outlines the support offered for Debian virtual machines (VMs) on Hyper-V.

Table legend

The following feature distribution map indicates the features that are present in each version of Windows Server. The known issues and workarounds for each distribution are listed after the table.

- **Built in** - Linux integration services (LIS) are included as part of this Linux distribution. The Microsoft-provided LIS download package doesn't work for this distribution. Don't install the Microsoft package. The kernel module version numbers for the built-in LIS (as shown by **lsmod**, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built-in LIS is out of date.
- ✓ - Feature available
- (*blank*) - Feature not available

Feature	Windows Server version	11 (Bullseye)	10.0-10.3 (Buster)
Availability		Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓ Note 4	✓ Note 4
Networking			
Jumbo frames	2019, 2016, 2012 R2	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓
Live Migration	2019, 2016, 2012 R2	✓	✓

Feature	Windows Server version	11 (Bullseye)	10.0-10.3 (Buster)
Static IP Injection	2019, 2016, 2012 R2		
vRSS	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
SR-IOV	2019, 2016	✓ Note 4	✓ Note 4
Storage			
VHDX resize	2019, 2016, 2012 R2	✓ Note 1	✓ Note 1
Virtual Fibre Channel	2019, 2016, 2012 R2		
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
TRIM support	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
SCSI WWN	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Memory			
PAE Kernel Support	2019, 2016, 2012 R2	✓	✓
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Runtime Memory Resize	2019, 2016	✓ Note 4	✓ Note 4
Video			
Hyper-V-specific video device	2019, 2016, 2012 R2	✓	✓
Miscellaneous			
Key-Value Pair	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
lsvmbus command	2019, 2016, 2012 R2		
Hyper-V Sockets	2019, 2016	✓ Note 4	✓ Note 4
PCI Passthrough/DDA	2019, 2016	✓ Note 4	✓ Note 4

Feature	Windows Server version	11 (Bullseye)	10.0-10.3 (Buster)
Generation 2 virtual machines			
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3
Secure boot	2019, 2016	✓	✓

Notes

1. Creating file systems on VHDs larger than 2 TB isn't supported.
2. Starting with Debian 8.3, the manually installed Debian package "hyperv-daemons" contains the key-value pair, fcopy, and VSS daemons. On Debian 7.x and 8.0-8.2, the hyperv-daemons package must come from Debian backports.
3. On Windows Server 2012 R2, Generation 2 virtual machines have secure boot enabled by default, and some Linux virtual machines don't boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager**, or you can disable it by using PowerShell:

   ```powershell
   Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
   ```

4. The latest upstream kernel capabilities are only available by using the kernels available in the Debian backports repository.
5. While Debian 7.x is out of support and uses an older kernel, the kernel included in Debian backports for Debian 7.x has improved Hyper-V capabilities.

See also

- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE Linux Enterprise Server (SLES) virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V

- Feature descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best practices for running Linux on Hyper-V

Supported Oracle Linux virtual machines on Hyper-V

Article • 09/27/2023

> Applies to: Azure Stack HCI, Windows Server 2022, Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features that are present in each version. The known issues and workarounds for each distribution are listed after the table.

In this section:

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The kernel module version numbers for the built in LIS (as shown by **lsmod**, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
- ✓ - Feature available
- (*blank*) - Feature not available
- **RHCK** - Red Hat Compatible Kernel
- **UEK** - Unbreakable Enterprise Kernel (UEK)
  - UEK4 - built on upstream Linux Kernel release 4.1.12
  - UEK5 - built on upstream Linux Kernel release 4.14
  - UEK6 - built on upstream Linux Kernel release 5.4

Oracle Linux 9.x Series

Feature	Windows Server version	9.0 (RHCK)
Availability		
Core	2019, 2016, 2012 R2	✓
Windows Server 2016 Accurate Time	2019, 2016	✓
Networking		
Jumbo frames	2019, 2016, 2012 R2	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓
Live Migration	2019, 2016, 2012 R2	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓
SR-IOV	2019, 2016	✓
Storage		
VHDX resize	2019, 2016, 2012 R2	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓
SCSI WWN	2019, 2016, 2012 R2	✓
Memory		
PAE Kernel Support	2019, 2016, 2012 R2	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Runtime Memory Resize	2019, 2016	✓
Video		
Hyper-V-specific video device	2019, 2016, 2012 R2	✓
Miscellaneous		

Feature	Windows Server version	9.0 (RHCK)
Key-Value Pair	2019, 2016, 2012 R2	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓
File copy from host to guest	2019, 2016, 2012 R2	✓
lsvmbus command	2019, 2016, 2012 R2	✓
Hyper-V Sockets	2019, 2016	✓
PCI Passthrough/DDA	2019, 2016	✓
Generation 2 virtual machines		
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12
Secure boot	2019, 2016	✓

Oracle Linux 8.x Series

Feature	Windows Server version	8.0-8.5 (RHCK)
Availability		
Core	2019, 2016, 2012 R2	✓
Windows Server 2016 Accurate Time	2019, 2016	✓
Networking		
Jumbo frames	2019, 2016, 2012 R2	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓
Live Migration	2019, 2016, 2012 R2	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓
SR-IOV	2019, 2016	✓
Storage		
VHDX resize	2019, 2016, 2012 R2	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3

Feature	Windows Server version	8.0-8.5 (RHCK)
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓
SCSI WWN	2019, 2016, 2012 R2	✓
Memory		
PAE Kernel Support	2019, 2016, 2012 R2	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Runtime Memory Resize	2019, 2016	✓
Video		
Hyper-V-specific video device	2019, 2016, 2012 R2	✓
Miscellaneous		
Key-Value Pair	2019, 2016, 2012 R2	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓
File copy from host to guest	2019, 2016, 2012 R2	✓
lsvmbus command	2019, 2016, 2012 R2	✓
Hyper-V Sockets	2019, 2016	✓
PCI Passthrough/DDA	2019, 2016	✓
Generation 2 virtual machines		
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12
Secure boot	2019, 2016	✓

Oracle Linux 7.x Series

This series only has 64-bit kernels.

Feature	Windows Server Version	7.5-7.8	7.3-7.4

		RHCK		UEK5	RHCK		UEK4
Availability		LIS 4.3	Built in	Built in	LIS 4.3	Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓			✓		
Networking							
Jumbo frames	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Live Migration	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Static IP injection	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
SR-IOV	2019, 2016	✓	✓	✓	✓	✓	✓
Storage							
VHDX resize	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5	✓ Note 4,5	✓ Note 5	✓ Note 5	✓ Note 4,5	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
SCSI WWN	2019, 2016, 2012	✓	✓	✓	✓	✓	

	R2						
Memory							
PAE Kernel Support	2019, 2016, 2012 R2	N/A	N/A	N/A	N/A	N/A	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Dynamic Memory Hot-Add	2019, 2016, 2012 R2	✓ Note 7,8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9
Dynamic Memory Ballooning	2019, 2016, 2012 R2	✓ Note 7,8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9	✓ Note 8,9
Runtime Memory Resize	2019, 2016	✓		✓	✓		
Video							
Hyper-V specific video	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Miscellaneous							
Key-value pair	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
lsvmbus command	2019, 2016, 2012 R2	✓		✓	✓		✓
Hyper-V Sockets	2019, 2016	✓		✓	✓		✓
PCI Passthrough/DDA	2019, 2016	✓	✓	✓	✓	✓	✓
Generation 2 virtual machines							
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note

							12
Secure boot	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓

Oracle Linux 6.x Series

This series only has 64-bit kernels.

Feature	Windows Server version	6.8-6.10 (RHCK)	6.8-6.10 (UEK4)
Availability		LIS 4.3	Built in
Core	2019, 2016, 2012 R2	✓	✓
Windows Server 2016 Accurate Time	2019, 2016		
Networking			
Jumbo frames	2019, 2016, 2012 R2	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓ Note 1	✓ Note 1
Live Migration	2019, 2016, 2012 R2	✓	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2	✓
vRSS	2019, 2016, 2012 R2	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓
SR-IOV	2019, 2016		
Storage			
VHDX resize	2019, 2016, 2012 R2	✓	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓	✓
SCSI WWN	2019, 2016, 2012 R2	✓	✓
Memory			
PAE Kernel Support	2019, 2016, 2012 R2	N/A	N/A

Feature	Windows Server version	6.8-6.10 (RHCK)	6.8-6.10 (UEK4)
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 6, 8, 9	✓ Note 6, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 6, 8, 9	✓ Note 6, 8, 9
Runtime Memory Resize	2019, 2016		
Video			
Hyper-V-specific video device	2019, 2016, 2012 R2	✓	✓
Miscellaneous			
Key-Value Pair	2019, 2016, 2012 R2	✓ Note 10,11	✓ Note 10,11
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓	✓
lsvmbus command	2019, 2016, 2012 R2	✓	✓
Hyper-V Sockets	2019, 2016	✓	✓
PCI Passthrough/DDA	2019, 2016	✓	✓
Generation 2 virtual machines			
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12	✓ Note 12
Secure boot	2019, 2016		

Notes

1. For this Oracle Linux release, VLAN tagging works but VLAN trunking does not.
2. Static IP injection may not work if Network Manager has been configured for a given synthetic network adapter on the virtual machine. For smooth functioning of static IP injection please make sure that either Network Manager is either turned off completely or has been turned off for a specific network adapter through its ifcfg-ethX file.
3. On Windows Server 2012 R2 while using virtual fibre channel devices, make sure that logical unit number 0 (LUN 0) has been populated. If LUN 0 has not been populated, a Linux virtual machine might not be able to mount fibre channel devices natively.

4. For built-in LIS, the "hyperv-daemons" package must be installed for this functionality.

5. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).

6. Dynamic memory support is only available on 64-bit virtual machines.

7. Hot-Add support is not enabled by default in this distribution. To enable Hot-Add support you need to add a udev rule under /etc/udev/rules.d/ as follows:

   a. Create a file **/etc/udev/rules.d/100-balloon.rules**. You may use any other desired name for the file.

   b. Add the following content to the file: `SUBSYSTEM=="memory", ACTION=="add", ATTR{state}="online"`

   c. Reboot the system to enable Hot-Add support.

   While the Linux Integration Services download creates this rule on installation, the rule is also removed when LIS is uninstalled, so the rule must be recreated if dynamic memory is needed after uninstallation.

8. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:

   - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.

   - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

9. If you are using Dynamic Memory on a Windows Server 2016 or Windows Server 2012 R2 operating system, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to hot-add failures, and you may not see any memory increase in a guest operating system.

10. To enable key/value pair (KVP) infrastructure, install the hypervkvpd or hyperv-daemons rpm package from your Oracle Linux ISO. Alternatively the package can be installed directly from Oracle Linux Yum repositories.

11. The key/value pair (KVP) infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.

12. On Windows Server 2012 R2 Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using PowerShell:

    ```
    Powershell
    ```

    ```
    Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
    ```

    The Linux Integration Services download can be applied to existing Generation 2 VMs but does not impart Generation 2 capability.

See Also

- Set-VMFirmware
- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V

Supported SUSE Linux Enterprise Server (SLES) virtual machines on Hyper-V

Article • 11/27/2023

> Applies to: Azure Stack HCI, Windows Server 2022, Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following is a feature distribution map that indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

The built-in SUSE Linux Enterprise Service drivers for Hyper-V are certified by SUSE. An example configuration can be viewed in this bulletin: SUSE YES Certification Bulletin.

Table legend

- **Built in** - LIS are included as part of this Linux distribution.The Microsoft-provided LIS download package does not work for this distribution, so do not install it.The kernel module version numbers for the built in LIS (as shown by **lsmod**, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
- ✓ - Feature available
- (*blank*) - Feature not available

SLES12+ is 64-bit only.

Feature	Operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
Availability		Built-in	Built-in	Built-in	Built-in	Built-in	Built-in	Built-in
Core	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Windows Server 2016 Accurate Time	WS/Hyper-V 2022,2019,2016	✓	✓	✓	✓			

Feature	Operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
Networking								
Jumbo frames	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
VLAN tagging and trunking	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Live migration	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Static IP Injection	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓Note 1	✓Note 1	✓Note 1	✓Note 1	✓Note 1	✓Note 1	✓Note 1
vRSS	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓		
TCP Segmentation and Checksum Offloads	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	
SR-IOV	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓			
Storage								
VHDX resize	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Virtual Fibre Channel	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Live virtual machine backup	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓ Note 2,3,8	✓Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8
TRIM support	WS/Hyper-V 2022,2019,2016,2012	✓	✓	✓	✓	✓	✓	

Feature	Operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
	Azure Stack HCI							
SCSI WWN	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓			
Memory								
PAE Kernel Support	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	N/A	N/A	N/A	N/A	N/A	✓	✓
Configuration of MMIO gap	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Dynamic Memory - Hot-Add	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓ Note 6	✓Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 4,5,6	✓ Note 4,5,6
Dynamic Memory - Ballooning	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 4,5,6	✓ Note 4,5,6
Runtime Memory Resize	WS/Hyper-V 2022,2019,2016	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6			
Video								
Hyper-V-specific video device	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Miscellaneous								
Key/value pair	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓ Note 7	✓ Note 7
Non-Maskable Interrupt	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓	✓	✓	

Feature	Operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
lsvmbus command	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓	✓	✓	✓			
Hyper-V Sockets	WS/Hyper-V 2022,2019,2016	✓	✓	✓				
PCI Passthrough/DDA	WS/Hyper-V 2022,2019,2016	✓	✓	✓	✓	✓		
Generation 2 virtual machines								
Boot using UEFI	WS/Hyper-V 2022,2019,2016,2012 Azure Stack HCI	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	
Secure boot	WS/Hyper-V 2022,2019,2016	✓	✓	✓	✓	✓		

Notes

1. Static IP injection may not work if **NetworkManager** has been configured for a given Hyper-V-specific network adapter on the virtual machine as it can override static IP settings that have been manually configured. To ensure smooth functioning of static IP injection please ensure that Network Manager is turned off completely or has been turned off for a specific network adapter through its **ifcfg-ethX** file.

2. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore.

3. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).

4. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:

    - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.

    - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

5. Dynamic memory support is only available on 64-bit virtual machines.

6. If you are using Dynamic Memory on Windows Server 2016 or Windows Server 2012 operating systems, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to Hot-Add failures, and you may not see any memory increase in a guest operating system.

7. In Windows Server 2016 or Windows Server 2012 R2, the key/value pair infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.

8. VSS backup will fail if a single partition is mounted multiple times.

9. On Windows Server 2012 R2, Generation 2 virtual machines have secure boot enabled by default and Generation 2 Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in Hyper-V Manager or you can disable it using PowerShell:

```powershell
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

See Also

- Set-VMFirmware
- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V

Supported Ubuntu virtual machines on Hyper-V

Article • 10/26/2023

> Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

Table legend

- **Built in** - Linux Integration Services (LIS) is included as part of this Linux distribution. The Microsoft-provided LIS download package doesn't work for this distribution, so don't install it. The kernel module version numbers for the built in LIS (as shown by **lsmod**, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
- ✓ - Feature available
- (*blank*) - Feature not available

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
Availability		Built-in	Built-in	Built-in	Built-in
Core	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Windows Server 2016 Accurate Time	2022, 2019, 2016	✓	✓	✓	✓
Networking					
Jumbo frames	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
VLAN tagging and trunking	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
Live migration	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Static IP Injection	2022, 2019, 2016, 2012 R2	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1
vRSS	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
TCP Segmentation and Checksum Offloads	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
SR-IOV	2022, 2019, 2016	✓	✓	✓	✓
Storage					
VHDX resize	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Virtual Fibre Channel	2022, 2019, 2016, 2012 R2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
Live virtual machine backup	2022, 2019, 2016, 2012 R2	✓ Note 3, 4, 5	✓ Note 3, 4, 5	✓ Note 3, 4, 5	✓ Note 3, 4, 5
TRIM support	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
SCSI WWN	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Memory					
PAE Kernel Support	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Configuration of MMIO gap	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Dynamic Memory - Hot-Add	2022, 2019, 2016, 2012 R2	✓ Note 6, 7, 8	✓ Note 6, 7, 8	✓ Note 6, 7, 8	✓ Note 6, 7, 8
Dynamic Memory - Ballooning	2022, 2019, 2016, 2012 R2	✓ Note 6, 7, 8	✓ Note 6, 7, 8	✓ Note 6, 7, 8	✓ Note 6, 7, 8
Runtime Memory Resize	2022, 2019, 2016	✓	✓	✓	✓
Video					

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
Hyper-V specific video device	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Miscellaneous					
Key/value pair	2022, 2019, 2016, 2012 R2	✓ Note 5, 9	✓ Note 5, 9	✓ Note 5, 9	✓ Note 5, 9
Non-Maskable Interrupt	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
File copy from host to guest	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
lsvmbus command	2022, 2019, 2016, 2012 R2	✓	✓	✓	✓
Hyper-V Sockets	2022, 2019, 2016	✓	✓	✓	✓
PCI Passthrough/DDA	2022, 2019, 2016	✓	✓	✓	✓
Generation 2 virtual machines					
Boot using UEFI	2022, 2019, 2016, 2012 R2	✓	✓ Note 10, 11	✓ Note 10, 11	✓ Note 10, 11
Secure boot	2022, 2019, 2016	✓	✓	✓	✓

Notes

1. Static IP injection may not work if **NetworkManager** has been configured for a given Hyper-V-specific network adapter on the virtual machine as it can override static IP settings that have been manually configured. To ensure smooth functioning of static IP injection, ensure that Network Manager is turned off completely or has been turned off for a specific network adapter through its **ifcfg-ethX** file.

2. While using virtual fiber channel devices, ensure that logical unit number 0 (LUN 0) has been populated. If LUN 0 hasn't been populated, a Linux virtual machine might not be able to mount fiber channel devices natively.

3. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system

consistency check (`fsck`) on restore.

4. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).

5. On long term support (LTS) releases use latest virtual Hardware Enablement (HWE) kernel for up to date Linux Integration Services.

   To install the Azure-tuned kernel on 16.04, 18.04 and 20.04, run the following commands as root (or sudo):

   Bash

   ```
   # apt-get update
   # apt-get install linux-azure
   ```

6. Dynamic memory support is only available on 64-bit virtual machines.

7. Dynamic Memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:

   - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
   - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

8. If you're using Dynamic Memory on Windows Server 2019, Windows Server 2016 or Windows Server 2012/2012 R2 operating systems, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to Hot-Add failures, and you might not see any memory increase on a guest operating system.

9. In Windows Server 2019, Windows Server 2016 or Windows Server 2012 R2, the key/value pair infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.

10. On Windows Server 2012 R2, Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines won't boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using PowerShell:

    Powershell

```
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

11. Before attempting to copy the VHD of an existing Generation 2 VHD virtual machine to create new Generation 2 virtual machines, follow these steps:

    a. Log in to the existing Generation 2 virtual machine.

    b. Change directory to the boot EFI directory:

    ```
    Bash
    # cd /boot/efi/EFI
    ```

    c. Copy the ubuntu directory in to a new directory named boot:

    ```
    Bash
    # sudo cp -r ubuntu/ boot
    ```

    d. Change directory to the newly created boot directory:

    ```
    Bash
    # cd boot
    ```

    e. Rename the shimx64.efi file:

    ```
    Bash
    # sudo mv shimx64.efi bootx64.efi
    ```

12. In order to perform live migrations for VMs that are Generation 2 configured, the **Migrate to a physical computer with a different processor version** option must be enabled under *Processor* > *Compatibility* in your VM settings. To learn more, see Processor compatibility mode in Hyper-V.

See Also

- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V

- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V
- Set-VMFirmware
- Ubuntu 14.04 in a Generation 2 VM - Ben Armstrong's Virtualization Blog

Supported FreeBSD virtual machines on Hyper-V

Article • 09/27/2023

> Applies to: Azure Stack HCI, Windows Server 2022, Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

Table legend

- **Built in** - BIS (FreeBSD Integration Service) are included as part of this FreeBSD release.
- ✓ - Feature available
- (*blank*) - Feature not available

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
Availability		Built in	Built in	Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓	✓	✓	
Networking					
Jumbo frames	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓	✓	✓
Live migration	2019, 2016, 2012 R2	✓	✓	✓	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4
vRSS	2019, 2016, 2012 R2	✓	✓	✓	✓

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓	✓	✓
Large Receive Offload (LRO)	2019, 2016, 2012 R2	✓	✓	✓	✓
SR-IOV	2019, 2016	✓	✓	✓	✓
Storage		Note 1	Note1	Note 1	Note 1
VHDX resize	2019, 2016, 2012 R2	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6
Virtual Fibre Channel	2019, 2016, 2012 R2				
Live virtual machine backup	2019, 2016, 2012 R2	✓	✓	✓	
TRIM support	2019, 2016, 2012 R2	✓	✓	✓	
SCSI WWN	2019, 2016, 2012 R2				
Memory					
PAE Kernel Support	2019, 2016, 2012 R2				
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2				
Dynamic Memory - Ballooning	2019, 2016, 2012 R2				
Runtime Memory Resize	2019, 2016				
Video					
Hyper-V specific video device	2019, 2016, 2012 R2				
Miscellaneous					
Key/value pair	2019, 2016, 2012 R2	✓	✓	✓	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓	✓	✓
File copy from host to guest	2019, 2016, 2012 R2				
lsvmbus command	2019, 2016, 2012 R2				
Hyper-V Sockets	2019, 2016				
PCI Passthrough/DDA	2019, 2016	✓	✓	✓	

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
Generation 2 virtual machines					
Boot using UEFI	2019, 2016, 2012 R2	✓	✓	✓	
Secure boot	2019, 2016				

Notes

1. Suggest to Label Disk Devices to avoid ROOT MOUNT ERROR during startup.
2. The virtual DVD drive may not be recognized when BIS drivers are loaded on FreeBSD 8.x and 9.x unless you enable the legacy ATA driver through the following command.

   ```sh
   # echo 'hw.ata.disk_enable=1' >> /boot/loader.conf
   # shutdown -r now
   ```

3. 9126 is the maximum supported MTU size.
4. In a failover scenario, you cannot set a static IPv6 address in the replica server. Use an IPv4 address instead.
5. KVP is provided by ports on FreeBSD 10.0. See the FreeBSD 10.0 ports on FreeBSD.org for more information.
6. To make VHDX online resizing work properly in FreeBSD 11.0, a special manual step is required to work around a GEOM bug which is fixed in 11.0+, after the host resizes the VHDX disk - open the disk for write, and run "gpart recover" as the following.

   ```sh
   # dd if=/dev/da1 of=/dev/da1 count=0
   # gpart recover da1
   ```

Additional Notes: The feature matrix of 10 stable and 11 stable is same with FreeBSD 11.1 release. In addition, FreeBSD 10.2 and previous versions (10.1, 10.0, 9.x, 8.x) are end

of life. Please refer here for an up-to-date list of supported releases and the latest security advisories.

See Also

- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best practices for running FreeBSD on Hyper-V

Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V

Article • 09/27/2023

> Applies to: Azure Stack HCI, Windows Server 2022, Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This article describes features available in components such as core, networking, storage, and memory when using Linux and FreeBSD on a virtual machine.

Core

Feature	Description
Integrated shutdown	With this feature, an administrator can shut down virtual machines from the Hyper-V Manager. For more information, see Operating system shutdown.
Time synchronization	This feature ensures that the maintained time inside a virtual machine is kept synchronized with the maintained time on the host. For more information, see Time synchronization.
Windows Server 2016 Accurate Time	This feature allows the guest to use the Windows Server 2016 Accurate Time capability, which improves time synchronization with the host to 1ms accuracy. For more information, see Windows Server 2016 Accurate Time
Multiprocessing support	With this feature, a virtual machine can use multiple processors on the host by configuring multiple virtual CPUs.
Heartbeat	With this feature, the host to can track the state of the virtual machine. For more information, see Heartbeat.
Integrated mouse support	With this feature, you can use a mouse on a virtual machine's desktop and also use the mouse seamlessly between the Windows Server desktop and the Hyper-V console for the virtual machine.
Hyper-V specific Storage device	This feature grants high-performance access to storage devices that are attached to a virtual machine.
Hyper-V specific Network device	This feature grants high-performance access to network adapters that are attached to a virtual machine.

Networking

Feature	Description
Jumbo frames	With this feature, an administrator can increase the size of network frames beyond 1500 bytes, which leads to a significant increase in network performance.
VLAN tagging and trunking	This feature allows you to configure virtual LAN (VLAN) traffic for virtual machines.
Live Migration	With this feature, you can migrate a virtual machine from one host to another host. For more information, see Virtual Machine Live Migration Overview.
Static IP Injection	With this feature, you can replicate the static IP address of a virtual machine after it has been failed over to its replica on a different host. Such IP replication ensures that network workloads continue to work seamlessly after a failover event.
vRSS (Virtual Receive Side Scaling)	Spreads the load from a virtual network adapter across multiple virtual processors in a virtual machine.For more information, see Virtual Receive-side Scaling in Windows Server 2012 R2.
TCP Segmentation and Checksum Offloads	Transfers segmentation and checksum work from the guest CPU to the host virtual switch or network adapter during network data transfers.
Large Receive Offload (LRO)	Increases inbound throughput of high-bandwidth connections by aggregating multiple packets into a larger buffer, decreasing CPU overhead.
SR-IOV	Single Root I/O devices use DDA to allow guests access to portions of specific NIC cards allowing for reduced latency and increased throughput. SR-IOV requires up to date physical function (PF) drivers on the host and virtual function (VF) drivers on the guest.

Storage

Feature	Description
VHDX resize	With this feature, an administrator can resize a fixed-size .vhdx file that is attached to a virtual machine. For more information, see Online Virtual Hard Disk Resizing Overview.
Virtual Fibre Channel	With this feature, virtual machines can recognize a fiber channel device and mount it natively. For more information, see Hyper-V Virtual Fibre Channel Overview.

Feature	Description
Live virtual machine backup	This feature facilitates zero down time backup of live virtual machines. Note that the backup operation doesn't succeed if the virtual machine has virtual hard disks (VHDs) that are hosted on remote storage such as a Server Message Block (SMB) share or an iSCSI volume. Additionally, ensure that the backup target doesn't reside on the same volume as the volume that you back up.
TRIM support	TRIM hints notify the drive that certain sectors that were previously allocated are no longer required by the app and can be purged. This process is typically used when an app makes large space allocations via a file and then self-manages the allocations to the file, for example, to virtual hard disk files.
SCSI WWN	The storvsc driver extracts World Wide Name (WWN) information from the port and node of devices attached to the virtual machine and creates the appropriate sysfs files.

Memory

Feature	Description
PAE Kernel Support	Physical Address Extension (PAE) technology allows a 32-bit kernel to access a physical address space that is larger than 4GB. Older Linux distributions such as RHEL 5.x used to ship a separate kernel that was PAE enabled. Newer distributions such as RHEL 6.x have prebuilt PAE support.
Configuration of MMIO gap	With this feature, appliance manufacturers can configure the location of the Memory Mapped I/O (MMIO) gap. The MMIO gap is typically used to divide the available physical memory between an appliance's Just Enough Operating Systems (JeOS) and the actual software infrastructure that powers the appliance.
Dynamic Memory - Hot-Add	The host can dynamically increase or decrease the amount of memory available to a virtual machine while it's in operation. Before provisioning, the Administrator enables Dynamic Memory in the Virtual Machine Settings panel and specifies the Startup Memory, Minimum Memory, and Maximum Memory for the virtual machine. When the virtual machine is in operation Dynamic Memory can't be disabled and only the Minimum and Maximum settings can be changed. (It's a best practice to specify these memory sizes as multiples of 128MB.) When the virtual machine is first started available memory is equal to the **Startup Memory**. As Memory Demand increases due to application workloads Hyper-V may dynamically allocate more memory to the virtual machine via the Hot-Add mechanism, if supported by that version of the kernel. The maximum amount of memory allocated is capped by the value of the **Maximum Memory** parameter. The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine

Feature	Description
	will show the highest amount of memory allocated. For more information, see Hyper-V Dynamic Memory Overview.
Dynamic Memory - Ballooning	The host can dynamically increase or decrease the amount of memory available to a virtual machine while it's in operation. Before provisioning, the Administrator enables Dynamic Memory in the Virtual Machine Settings panel and specify the Startup Memory, Minimum Memory, and Maximum Memory for the virtual machine. When the virtual machine is in operation Dynamic Memory can't be disabled and only the Minimum and Maximum settings can be changed. (It is a best practice to specify these memory sizes as multiples of 128MB.) When the virtual machine is first started available memory is equal to the **Startup Memory**. As Memory Demand increases due to application workloads Hyper-V may dynamically allocate more memory to the virtual machine via the Hot-Add mechanism (above). As Memory Demand decreases Hyper-V may automatically deprovision memory from the virtual machine via the Balloon mechanism. Hyper-V won't deprovision memory below the **Minimum Memory** parameter. The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine will show the highest amount of memory allocated. For more information, see Hyper-V Dynamic Memory Overview.
Runtime Memory Resize	An administrator can set the amount of memory available to a virtual machine while it's in operation, either increasing memory ("Hot Add") or decreasing it ("Hot Remove"). Memory is returned to Hyper-V via the balloon driver (see "Dynamic Memory - Ballooning"). The balloon driver maintains a minimum amount of free memory after ballooning, called the "floor", so assigned memory can't be reduced below the current demand plus this floor amount. The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine will show the highest amount of memory allocated. (It's a best practice to specify memory values as multiples of 128MB.)

Video

Feature	Description
Hyper-V-specific video device	This feature provides high-performance graphics and superior resolution for virtual machines. This device doesn't provide Enhanced Session Mode or RemoteFX capabilities.

Miscellaneous

Feature	Description
KVP (Key-Value pair) exchange	This feature provides a key/value pair (KVP) exchange service for virtual machines. Typically administrators use the KVP mechanism to perform read and write custom data operations on a virtual machine. For more information, see Data Exchange: Using key-value pairs to share information between the host and guest on Hyper-V.
Non-Maskable Interrupt	With this feature, an administrator can issue Non-Maskable Interrupts (NMI) to a virtual machine. NMIs are useful in obtaining crash dumps of operating systems that have become unresponsive due to application bugs. These crash dumps can be diagnosed after you restart.
File copy from host to guest	This feature allows files to be copied from the host physical computer to the guest virtual machines without using the network adaptor. For more information, see Guest services.
lsvmbus command	This command gets information about devices on the Hyper-V virtual machine bus (VMBus) similar to information commands like lspci.
Hyper-V Sockets	This is an additional communication channel between the host and guest operating system. To load and use the Hyper-V Sockets kernel module, see Make your own integration services.
PCI Passthrough/DDA	With Windows Server 2016 administrators can pass through PCI Express devices via the Discrete Device Assignment mechanism. Common devices are network cards, graphics cards, and special storage devices. The virtual machine will require the appropriate driver to use the exposed hardware. The hardware must be assigned to the virtual machine for it to be used. For more information, see Discrete Device Assignment - Description and Background. DDA is a prerequisite for SR-IOV networking. Virtual ports will need to be assigned to the virtual machine and the virtual machine must use the correct Virtual Function (VF) drivers for device multiplexing.

Generation 2 virtual machines

Feature	Description
Boot using UEFI	This feature allows virtual machines to boot using Unified Extensible Firmware Interface (UEFI). For more information, see Generation 2 Virtual Machine Overview.
Secure boot	This feature allows virtual machines to use UEFI based secure boot mode. When a virtual machine is booted in secure mode, various operating system components are

Feature	Description
	verified using signatures present in the UEFI data store. For more information, see Secure Boot.

See Also

- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V
- Best practices for running FreeBSD on Hyper-V

Best Practices for running Linux on Hyper-V

Article • 07/11/2022

> Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This topic contains a list of recommendations for running Linux virtual machine on Hyper-V.

Tuning Linux File Systems on Dynamic VHDX Files

Some Linux file systems may consume significant amounts of real disk space even when the file system is mostly empty. To reduce the amount of real disk space usage of dynamic VHDX files, consider the following recommendations:

- When creating the VHDX, use 1MB BlockSizeBytes (from the default 32MB) in PowerShell, for example:

Powershell

```
PS > New-VHD -Path C:\MyVHDs\test.vhdx -SizeBytes 127GB -Dynamic -
BlockSizeBytes 1MB
```

- The ext4 format is preferred to ext3 because ext4 is more space efficient than ext3 when used with dynamic VHDX files.
- When creating the filesystem specify the number of groups to be 4096, for example:

Bash

```
# mkfs.ext4 -G 4096 /dev/sdX1
```

Grub Menu Timeout on Generation 2 Virtual Machines

Because of legacy hardware being removed from emulation in Generation 2 virtual machines, the grub menu countdown timer counts down too quickly for the grub menu to be displayed, immediately loading the default entry. Until grub is fixed to use the EFI-supported timer, modify **/boot/grub/grub.conf**, **/etc/default/grub**, or equivalent to have "timeout=100000" instead of the default "timeout=5".

PxE Boot on Generation 2 Virtual Machines

Because the PIT timer is not present in Generation 2 Virtual Machines, network connections to the PxE TFTP server can be prematurely terminated and prevent the bootloader from reading Grub configuration and loading a kernel from the server.

On RHEL 6.x, the legacy grub v0.97 EFI bootloader can be used instead of grub2 as described here: https://access.redhat.com/documentation/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s1-netboot-pxe-config-efi.html

On Linux distributions other than RHEL 6.x, similar steps can be followed to configure grub v0.97 to load Linux kernels from a PxE server.

Additionally, on RHEL/CentOS 6.6 keyboard and mouse input will not work with the pre-install kernel which prevents specifying installation options in the menu. A serial console must be configured to allow choosing installation options.

- In the **efidefault** file on the PxE server, add the following kernel parameter "**console=ttyS1**"
- On the VM in Hyper-V, set up a COM port using this PowerShell cmdlet:

```powershell
Set-VMComPort -VMName <Name> -Number 2 -Path \\.\pipe\dbg1
```

Specifying a kickstart file to the pre-install kernel would also avoid the need for keyboard and mouse input during installation.

Use static MAC addresses with failover clustering

Linux virtual machines that will be deployed using failover clustering should be configured with a static media access control (MAC) address for each virtual network adapter. In some versions of Linux, the networking configuration may be lost after failover because a new MAC address is assigned to the virtual network adapter. To avoid losing the network configuration, ensure that each virtual network adapter has a static MAC address. You can configure the MAC address by editing the settings of the virtual machine in Hyper-V Manager or Failover Cluster Manager.

Use Hyper-V-specific network adapters, not the legacy network adapter

Configure and use the virtual Ethernet adapter, which is a Hyper-V-specific network card with enhanced performance. If both legacy and Hyper-V-specific network adapters are attached to a virtual machine, the network names in the output of **ifconfig -a** might show random values such as **_tmp12000801310**. To avoid this issue, remove all legacy network adapters when using Hyper-V-specific network adapters in a Linux virtual machine.

Use I/O scheduler noop/none for better disk I/O performance

The Linux kernel offers two sets of disk I/O schedulers to reorder requests. One set is for the older 'blk' subsystem and one set is for the newer 'blk-mq' subsystem. In either case, with today's solid state disks it is recommended to use a scheduler that passes the scheduling decisions to the underlying Hyper-V hypervisor. For Linux kernels using the 'blk' subsystem, this is the "noop" scheduler. For Linux kernels using the 'blk-mq' subsystem, this is the "none" scheduler.

For a particular disk, the available schedulers can be seen at this file system location: /sys/class/block/`<diskname>`/queue/scheduler, with the currently selected scheduler in square brackets. You can change the scheduler by writing to this file system location. The change must be added to an initialization script in order to persist across reboots. Consult your Linux distro documentation for details.

NUMA

Linux kernel versions earlier than 2.6.37 don't support NUMA on Hyper-V with larger VM sizes. This issue primarily impacts older distributions using the upstream Red Hat 2.6.32 kernel, and was fixed in Red Hat Enterprise Linux (RHEL) 6.6 (kernel-2.6.32-504). Systems running custom kernels older than 2.6.37, or RHEL-based kernels older than 2.6.32-504 must set the boot parameter `numa=off` on the kernel command line in grub.conf. For more information, see Red Hat KB 436883.

Reserve more memory for kdump

In case the dump capture kernel ends up with a panic on boot, reserve more memory for the kernel. For example, change the parameter **crashkernel=384M-:128M** to **crashkernel=384M-:256M** in the Ubuntu grub configuration file.

Shrinking VHDX or expanding VHD and VHDX files can result in erroneous GPT partition tables

Hyper-V allows shrinking virtual disk (VHDX) files without regard for any partition, volume, or file system data structures that may exist on the disk. If the VHDX is shrunk to where the end of the VHDX comes before the end of a partition, data can be lost, that partition can become corrupted, or invalid data can be returned when the partition is read.

After resizing a VHD or VHDX, administrators should use a utility like fdisk or parted to update the partition, volume, and file system structures to reflect the change in the size of the disk. Shrinking or expanding the size of a VHD or VHDX that has a GUID Partition Table (GPT) will cause a warning when a partition management tool is used to check the partition layout, and the administrator will be warned to fix the first and secondary GPT headers. This manual step is safe to perform without data loss.

Additional References

- Supported Linux and FreeBSD virtual machines for Hyper-V on Windows
- Best practices for running FreeBSD on Hyper-V
- Deploy a Hyper-V Cluster
- Create Linux Images for Azure

- Optimize your Linux VM on Azure

Best practices for running FreeBSD on Hyper-V

Article • 07/29/2021

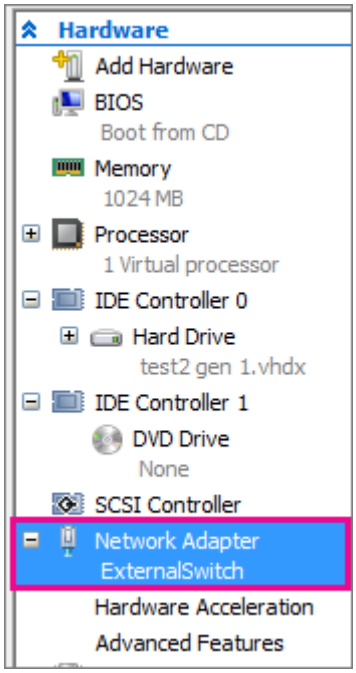
> Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This topic contains a list of recommendations for running FreeBSD as a guest operating system on a Hyper-V virtual machine.

Enable CARP in FreeBSD 10.2 on Hyper-V

The Common Address Redundancy Protocol (CARP) allows multiple hosts to share the same IP address and Virtual Host ID (VHID) to help provide high availability for one or more services. If one or more hosts fail, the other hosts transparently take over so users won't notice a service failure.To use CARP in FreeBSD 10.2, follow the instructions in the FreeBSD handbook and do the following in Hyper-V Manager.

- Verify the virtual machine has a Network Adapter and it's assigned a virtual switch. Select the virtual machine and select **Actions** > **Settings**.



- Enable MAC address spoofing. To do this,

1. Select the virtual machine and select **Actions** > **Settings**.
2. Expand **Network Adapter** and select **Advanced Features**.
3. Select **Enable MAC Address spoofing**.

Create labels for disk devices

During startup, device nodes are created as new devices are discovered. This can mean that device names can change when new devices are added. If you get a ROOT MOUNT ERROR during startup, you should create labels for each IDE partition to avoid conflicts and changes. To learn how, see Labeling Disk Devices. Below are examples.

> **ⓘ Important**
>
> Make a backup copy of your fstab before making any changes.

1. Reboot the system into single user mode. This can be accomplished by selecting boot menu option 2 for FreeBSD 10.3+ (option 4 for FreeBSD 8.x), or performing a 'boot -s' from the boot prompt.
2. In Single user mode, create GEOM labels for each of the IDE disk partitions listed in your fstab (both root and swap). Below is an example of FreeBSD 10.3.

   Bash

   ```
   # cat  /etc/fstab
   # Device           Mountpoint      FStype  Options   Dump   Pass#
   /dev/da0p2         /               ufs     rw        1       1
   /dev/da0p3         none            swap    sw        0       0

   # glabel  label rootfs  /dev/da0p2
   # glabel  label swap   /dev/da0p3
   # exit
   ```

   Additional information on GEOM labels can be found at: Labeling Disk Devices.
3. The system will continue with multi-user boot. After the boot completes, edit /etc/fstab and replace the conventional device names, with their respective labels. The final /etc/fstab will look like this:

   ```
   # Device                Mountpoint      FStype  Options         Dump
   Pass#
   ```

```
/dev/label/rootfs       /               ufs     rw              1
1
/dev/label/swap         none            swap    sw              0
0
```

4. The system can now be rebooted. If everything went well, it will come up normally and mount will show:

```
# mount
/dev/label/rootfs on / (ufs, local, journaled soft-updates)
devfs on /dev (devfs, local, mutilabel)
```

Use a wireless network adapter as the virtual switch

If the virtual switch on the host is based on wireless network adapter, reduce the ARP expiration time to 60 seconds by the following command. Otherwise the networking of the VM may stop working after a while.

```
# sysctl net.link.ether.inet.max_age=60
```

See also

- Supported FreeBSD virtual machines on Hyper-V

Hyper-V feature compatibility by generation and guest

Article • 08/13/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows Server 2019, Windows Server 2016

> **Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

The tables in this article show you the generations and operating systems that are compatible with some of the Hyper-V features, grouped by categories. In general, you'll get the best availability of features with a generation 2 virtual machine that runs the newest operating system.

Keep in mind that some features rely on hardware or other infrastructure. For hardware details, see System requirements for Hyper-V on Windows Server. In some cases, a feature can be used with any supported guest operating system. For details on which operating systems are supported, see:

- Supported Linux and FreeBSD virtual machines
- Supported Windows guest operating systems

Availability and backup

Expand table

Feature	Generation	Guest operating system
Checkpoints	1 and 2	Any supported guest
Guest clustering	1 and 2	Guests that run cluster-aware applications and have iSCSI target software installed
Replication	1 and 2	Any supported guest

Feature	Generation	Guest operating system
Domain Controller	1 and 2	Any supported Windows Server guest using only production checkpoints. See Supported Windows Server guest operating systems

Compute

Expand table

Feature	Generation	Guest operating system
Dynamic memory	1 and 2	Specific versions of supported guests. See Hyper-V Dynamic Memory Overview for versions older than Windows Server 2016 and Windows 10.
Hot add/removal of memory	1 and 2	Windows Server 2016, Windows 10
Virtual NUMA	1 and 2	Any supported guest

Development and test

Expand table

Feature	Generation	Guest operating system
COM/Serial ports	1 and 2 **Note:** For generation 2, use Windows PowerShell to configure. For details, see Add a COM port for kernel debugging.	Any supported guest

Mobility

Expand table

Feature	Generation	Guest operating system
Live migration	1 and 2	Any supported guest
Import/export	1 and 2	Any supported guest

Networking

Expand table

Feature	Generation	Guest operating system
Hot add/removal of virtual network adapter	2	Any supported guest
Legacy virtual network adapter	1	Any supported guest
Single root input/output virtualization (SR-IOV)	1 and 2	64-bit Windows guests, starting with Windows Server 2012 and Windows 8.
Virtual machine multi queue (VMMQ)	1 and 2	Any supported guest

Remote connection experience

Expand table

Feature	Generation	Guest operating system
Discrete device assignment (DDA)	1 and 2	Windows Server 2012 and later, Windows 10 and Windows 11
Enhanced session mode	1 and 2	Windows Server 2012 R2 and later, and Windows 8.1 and later, with Remote Desktop Services enabled **Note**: You might need to also configure the host. For details, see Use local resources on Hyper-V virtual machine with VMConnect.
RemoteFx	1 and 2	Generation 1 on 32-bit and 64-bit Windows versions starting with Windows 8. Generation 2 on 64-bit Windows 10 and Windows 11 versions

Security

Expand table

Feature	Generation	Guest operating system
Secure boot	2	**Linux**: Ubuntu 14.04 and later, SUSE Linux Enterprise Server 12 and later, Red Hat Enterprise Linux 7.0 and later, and CentOS 7.0

Feature	Generation	Guest operating system
		and later **Windows**: All supported versions that can run on a generation 2 virtual machine
Shielded virtual machines	2	**Windows**: All supported versions that can run on a generation 2 virtual machine

Storage

Expand table

Feature	Generation	Guest operating system
Shared virtual hard disks (VHDX only)	1 and 2	Windows Server 2012 and later
SMB3	1 and 2	All that support SMB3
Storage spaces direct	2	Windows Server 2016 and later
Virtual Fibre Channel	1 and 2	Windows Server 2012 and later
VHDX format	1 and 2	Any supported guest

Feedback

Was this page helpful? Yes No

Install the Hyper-V role on Windows Server

Article • 08/13/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows Server 2016, Windows Server 2019

> **Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

To create and run virtual machines, install the Hyper-V role on Windows Server by using Server Manager or the **Install-WindowsFeature** cmdlet in Windows PowerShell. For Windows 10 and Windows 11, see Install Hyper-V on Windows.

To learn more about Hyper-V, see the Hyper-V Technology Overview. To learn more about Hyper-V, see the Hyper-V Overview. To try out Windows Server 2025, you can download and install an evaluation copy. See the Evaluation Center.

Before you install Windows Server or add the Hyper-V role, make sure that:

- Your computer hardware is compatible. For more information, see System Requirements for Windows Server and System requirements for Hyper-V on Windows Server.
- You don't plan to use third-party virtualization apps that rely on the same processor features that Hyper-V requires. Examples include VMWare Workstation and VirtualBox. You can install Hyper-V without uninstalling these other apps. But, if you try to use them to manage virtual machines when the Hyper-V hypervisor is running, the virtual machines might not start or might run unreliably. For details and instructions for turning off the Hyper-V hypervisor if you need to use one of these apps, see Virtualization applications don't work together with Hyper-V, Device Guard, and Credential Guard.

If you want to install only the management tools, such as Hyper-V Manager, see Remotely manage Hyper-V hosts with Hyper-V Manager.

Install Hyper-V by using Server Manager

1. In **Server Manager**, on the **Manage** menu, select **Add Roles and Features**.
2. On the **Before you begin** page, verify that your destination server and network environment are prepared for the role and feature you want to install. Select **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**, and then select **Next**.
4. On the **Select destination server** page, select a server from the server pool, and then select **Next**.
5. On the **Select server roles** page, select **Hyper-V**. From the **Add Roles and Features Wizard** page, select **Add Features**, and then select **Next**.
6. On the **Select features** page, select **Next**, and then select **Next** again.
7. On the **Create Virtual Switches** page, **Virtual Machine Migration** page, and **Default Stores** page, select the options that suit your specific environment.
8. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**, and then select **Install**.
9. When installation is finished, verify that Hyper-V installed correctly. Open the **All Servers** page in Server Manager and select a server on which you installed Hyper-V. Check the **Roles and Features** tile on the page for the selected server.

Install Hyper-V by using the Install-WindowsFeature cmdlet

1. On the Windows desktop, select the Start button and type any part of the name **Windows PowerShell**.
2. Right-click Windows PowerShell and select **Run as Administrator**.
3. To install Hyper-V on a server you're connected to remotely, run the following command and replace `<computer_name>` with the name of server.

```
PowerShell

Install-WindowsFeature -Name Hyper-V -ComputerName <computer_name> -
IncludeManagementTools -Restart
```

If you're connected locally to the server, run the command without `-ComputerName <computer_name>`.

4. After the server restarts, you can see that the Hyper-V role is installed and see what other roles and features are installed by running the following command:

   ```
   PowerShell
   Get-WindowsFeature -ComputerName <computer_name>
   ```

   If you're connected locally to the server, run the command without `-ComputerName <computer_name>`.

> **Note**
>
> If you install this role on a server that runs the Server Core installation option of Windows Server 2016 and use the parameter `-IncludeManagementTools`, only the Hyper-V Module for Windows PowerShell is installed. You can use the GUI management tool, Hyper-V Manager, on another computer to remotely manage a Hyper-V host that runs on a Server Core installation. For instructions on connecting remotely, see Remotely manage Hyper-V hosts with Hyper-V Manager.

Additional References

- Install-WindowsFeature

Feedback

Create and configure a virtual switch with Hyper-V

Article • 08/13/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

> **ⓘ Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

This article shows you how to create and configure your virtual switch using Hyper-V Manager or PowerShell. A virtual switch allows virtual machines created on Hyper-V hosts to communicate with other computers. When you first install the Hyper-V role on Windows Server, you can optionally create a virtual switch at the same time. To learn more about virtual switches, see Hyper-V Virtual Switch.

For more information about how you can set up your networking infrastructure with Windows Server, review the Networking documentation.

Prerequisites

Before you can create and configure your virtual switch, your computer must meet the following prerequisites:

- The Hyper-V server role must be installed.
- Determine what type of virtual switch you need to create.
- Identify which network you'll connect your computer to. Review the Core network planning article for more information.
- Have administrative rights.

Create a virtual switch

Once you've completed the prerequisites, you'll be ready to create your virtual switch. In this section, we'll create a basic virtual switch by following these steps.

Hyper-V Manager

Here's how to create a virtual switch using Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.
3. Choose the type of virtual switch, then select **Create Virtual Switch**.
4. Enter a name for the virtual switch, then perform one of the following steps.
    a. If you selected *external*, choose the network adapter (NIC) that you want to use, then select **OK**.

    You'll be prompted with a warning that the change may disrupt your network connectivity; select **Yes** if you're happy to continue.

    b. If you selected *internal* or *private*, select **OK**.

Management Operating System sharing

An external virtual switch allows your virtual machines to connect to an external network. You can also allow the management operating system to share the same selected network adapter. To begin, follow these steps.

Hyper-V Manager

Here's how to allow the management operating system to share the selected network adapter switch using Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.
3. Select the virtual switch you wish to configure, check the **Allow management operating system to share this network adapter** and select **OK**.

    You'll be prompted with a warning that the change may disrupt your network connectivity; select **Yes** if you're happy to continue.

Virtual LAN (VLAN) identification

You can specify the VLAN identification (ID) used by virtual machines network adapters and virtual switches. For virtual switches connected to either an external or internal network you can specify the (VLAN) ID. The VLAN ID number is used by the management operating system and virtual machines communicating through this virtual switch.

You can also configure your virtual switch with other VLAN options, such port mode and the native VLAN ID. For these options, you'll need to use PowerShell and ensure the configuration is compatible with your networks configuration.

To configure the VLAN identification for the switch, follow these steps.

Hyper-V Manager

Here's how to specify the VLAN ID using the Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.
3. Select the virtual switch you wish to configure, check the **Enable virtual LAN identification for management operating system**.
    a. You can enter any VLAN ID number or leave the default, then select **OK**.

        You'll be prompted to warn you that the change may disrupt your network connectivity, select **Yes** if you're happy to continue.

VLAN identifiers should be consistent with your network to ensure compatibility between your computer, virtual machines, and other network devices.

Next step

Now you've created and configured your virtual switch, here are other articles to help you continue with Hyper-V.

- Learn about Switch Embedded Teaming (SET).
- Learn how to create a virtual machine in Hyper-V.
- Learn about other configuration options in the Set-VMSwitch and Set-VMNetworkAdapterVlan PowerShell reference articles.

Feedback

Was this page helpful? Yes No

Create a virtual machine in Hyper-V

Article • 08/13/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019, Windows 11, Windows 10

> **ⓘ Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Learn how to create a virtual machine by using Hyper-V Manager and Windows PowerShell and what options you have when you create a virtual machine in Hyper-V Manager.

Create a virtual machine

Hyper-V Manager

1. Open **Hyper-V Manager**.
2. From the **Action** pane, click **New**, and then click **Virtual Machine**.
3. From the **New Virtual Machine Wizard**, click **Next**.
4. Make the appropriate choices for your virtual machine on each of the pages. For more information, see New virtual machine options and defaults in Hyper-V Manager.
5. After verifying your choices in the **Summary** page, click **Finish**.
6. In Hyper-V Manager, right-click the virtual machine and select **connect**.
7. In the Virtual Machine Connection window, select **Action** > **Start**.

Options in Hyper-V Manager New Virtual Machine Wizard

The following table lists the options you can pick when you create a virtual machine in Hyper-V Manager and the defaults for each.

Expand table

Page	Default for Windows Server 2016, Windows 10, and later	Other options
Specify Name and Location	Name: New Virtual Machine. Location: **C:\ProgramData\Microsoft\Windows\Hyper-V**.	You can also enter your own name and choose another location for the virtual machine. This is where the virtual machine configuration files will be stored.
Specify Generation	Generation 1	You can also choose to create a Generation 2 virtual machine. For more information, see Should I create a generation 1 or 2 virtual machine in Hyper-V?.
Assign Memory	Startup memory: 1024 MB Dynamic memory: **not selected**	You can set the startup memory from 32 MB to 5902 MB. You can also choose to use Dynamic Memory. For more information, see Hyper-V Dynamic Memory Overview.
Configure Networking	Not connected	You can select a network connection for the virtual machine to use from a list of existing virtual switches. See Create a virtual switch for Hyper-V virtual machines.
Connect Virtual Hard Disk	Create a virtual hard disk Name: <*vmname*>.vhdx **Location**: **C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks** **Size**: 127 GB	You can also choose to use an existing virtual hard disk or wait and attach a virtual hard disk later.

Page	Default for Windows Server 2016, Windows 10, and later	Other options
Installation Options	Install an operating system later	These options change the boot order of the virtual machine so that you can install from an .iso file, bootable floppy disk or a network installation service, like Windows Deployment Services (WDS).
Summary	Displays the options that you have chosen, so that you can verify they are correct. - Name - Generation - Memory - Network - Hard Disk - Operating System	**Tip:** You can copy the summary from the page and paste it into e-mail or somewhere else to help you keep track of your virtual machines.

Additional References

- New-VM
- Supported virtual machine configuration versions
- Should I create a generation 1 or 2 virtual machine in Hyper-V?
- Create a virtual switch for Hyper-V virtual machines

Feedback

Guest operating system and application supportability on Hyper-V

Article • 11/24/2022

Hyper-V is a hypervisor that is broadly used in many Microsoft server products, including the Windows Server family (Datacenter, Standard, and Essentials editions) and Azure Stack HCI. Hyper-V provides a platform with broad ecosystem support and compatibility. This article clarifies which versions of Windows Server or Azure Stack HCI map to which Hyper-V build numbers. This helps understand supported scenarios where a guest operating system or application has been validated for Hyper-V.

Although the different products that include Hyper-V could contain variations in features, the common codebase provides a consistent platform for guest operating systems and applications running inside of a virtual machine to run on compatible products that share the same Hyper-V build number. This means that any support or compatibility statements for a guest operating system or application that is certified for specific builds of Hyper-V is compatible with all products that share the same build number for Hyper-V.

The table below shows which Hyper-V build numbers are available in which compatible products:

Hyper-V build	Compatible products
20348	Windows Server 2022 Datacenter Windows Server 2022 Standard Windows Server 2022 Essentials Azure Stack HCI version 21H2 Azure Stack HCI version 22H2
17763 and 17784	Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2019 Essentials Hyper-V Server 2019 Azure Stack HCI version 20H2
14393	Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2016 Essentials Hyper-V Server 2016

For more information, see:

- Windows Server release information

- Supported Windows guest operating systems on Hyper-V
- Supported Linux and FreeBSD guest operating systems on Hyper-V

Should I create a generation 1 or 2 virtual machine in Hyper-V?

Article • 06/20/2024

> Applies to: Windows 10, Windows 11, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019, Windows Server 2022, Azure Stack HCI

Creating a generation 1 or generation 2 virtual machine depends on which guest operating system you want to install and the boot method you want to use to deploy the virtual machine. We recommend you create a generation 2 virtual machines to take advantage of features like Secure Boot unless one of the following statements is true:

- You're using an existing, prebuilt virtual hard drive (VHD or VHDX files), which isn't compatible with UEFI.
- Generation 2 doesn't support the operating system you want to run on the virtual machine.
- Generation 2 doesn't support the boot method you want to use.

For more information about what features are available with generation 2 virtual machines, see Hyper-V feature compatibility by generation and guest.

You can't change a virtual machine's generation once it's created. We recommend that you review the considerations here, and choose the operating system, boot method, and features you want to use before you choose a generation.

What are the advantages of using generation 2 virtual machines?

Here are some of the advantages you get when you use a generation 2 virtual machine:

- **Secure Boot**

  Use Secure Boot to help prevent unauthorized firmware, operating systems, or UEFI drivers from running at boot time. Secure Boot verifies the boot loader is signed by a trusted authority in the UEFI database. Secure Boot is enabled by default for generation 2 virtual machines. If you need to run a guest operating system that Secure Boot doesn't support, you can disable it after you create the virtual machine. For more information, see Secure Boot.

To Secure Boot generation 2 Linux virtual machines, you need to choose the UEFI CA Secure Boot template when you create the virtual machine.

- **Larger boot volume** The maximum boot volume for generation 2 virtual machines is 64 TB. This maximum boot volume is the maximum disk size supported by a `.VHDX` For generation 1 virtual machines, the maximum boot volume is 2 TB for a `.VHDX` and 2040 GB for a `.VHD` For more information, see Hyper-V Virtual Hard Disk Format Overview.

  You might also see a slight improvement in virtual machine boot and installation times with generation 2 virtual machines.

Which guest operating systems are supported?

Generation 1 virtual machines support most guest operating systems. Generation 2 virtual machines support most 64-bit versions of Windows and more current versions of Linux and FreeBSD operating systems. Use the following sections to see which generation of virtual machine supports the guest operating system you want to install.

- Windows guest operating system support
- CentOS and Red Hat Enterprise Linux guest operating system support
- Debian guest operating system support
- FreeBSD guest operating system support
- Oracle Linux guest operating system support
- SUSE guest operating system support
- Ubuntu guest operating system support

Windows guest operating system support

The following table shows which 64-bit versions of Windows you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

64-bit versions of Windows	Generation 1	Generation 2
Windows Server 2025	✓	✓

64-bit versions of Windows	Generation 1	Generation 2
Windows Server 2022	✓	✓
Windows Server 2019	✓	✓
Windows Server 2016	✓	✓
Windows Server 2012 R2	✓	✓
Windows Server 2012	✓	✓
Windows Server 2008 R2	✓	✕
Windows Server 2008	✓	✕
Windows 11	✕	✓
Windows 10	✓	✓
Windows 8.1	✓	✓
Windows 8	✓	✓
Windows 7	✓	✕

The following table shows which 32-bit versions of Windows you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

32-bit versions of Windows	Generation 1	Generation 2
Windows 10	✓	✕
Windows 8.1	✓	✕
Windows 8	✓	✕
Windows 7	✓	✕

CentOS and Red Hat Enterprise Linux guest operating system support

The following table shows which versions of Red Hat Enterprise Linux (RHEL) and CentOS you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Operating system versions	Generation 1	Generation 2
RHEL/CentOS 8.x series	✓	✓
RHEL/CentOS 7.x series	✓	✓
RHEL/CentOS 6.x series	✓	✓ **Note:** Only supported on Windows Server 2016 and above.
RHEL/CentOS 5.x series	✓	✕

For more information, see CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V.

Debian guest operating system support

The following table shows which versions of Debian you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Operating system versions	Generation 1	Generation 2
Debian 10.x (buster) series	✓	✓
Debian 9.x (stretch) series	✓	✓
Debian 8.x (jessie) series	✓	✓
Debian 7.x (wheezy) series	✓	✕

For more information, see Debian virtual machines on Hyper-V.

FreeBSD guest operating system support

The following table shows which versions of FreeBSD you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Operating system versions	Generation 1	Generation 2
FreeBSD 12 to 12.1	✓	✓
FreeBSD 11.1 to 11.3	✓	✓
FreeBSD 11	✓	✕
FreeBSD 10 to 10.3	✓	✕
FreeBSD 9.1 and 9.3	✓	✕
FreeBSD 8.4	✓	✕

For more information, see FreeBSD virtual machines on Hyper-V.

Oracle Linux guest operating system support

The following table shows which versions of Red Hat Compatible Kernel Series you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Red Hat Compatible Kernel Series versions	Generation 1	Generation 2
Oracle Linux 8.x series	✓	✓
Oracle Linux 7.x series	✓	✓
Oracle Linux 6.x series	✓	✕

The following table shows which versions of Unbreakable Enterprise Kernel you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Unbreakable Enterprise Kernel (UEK) versions	Generation 1	Generation 2
Oracle Linux UEK R3 QU3	✓	✕
Oracle Linux UEK R3 QU2	✓	✕
Oracle Linux UEK R3 QU1	✓	✕

For more information, see Oracle Linux virtual machines on Hyper-V.

SUSE guest operating system support

The following table shows which versions of SUSE you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Operating system versions	Generation 1	Generation 2
SUSE Linux Enterprise Server 15 series	✓	✓
SUSE Linux Enterprise Server 12 series	✓	✓
SUSE Linux Enterprise Server 11 series	✓	✕
Open SUSE 12.3	✓	✕

For more information, see SUSE virtual machines on Hyper-V.

Ubuntu guest operating system support

The following table shows which versions of Ubuntu you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Expand table

Operating system versions	Generation 1	Generation 2
Ubuntu 20.04	✓	✓
Ubuntu 18.04	✓	✓
Ubuntu 16.04	✓	✓
Ubuntu 14.04	✓	✓
Ubuntu 12.04	✓	✕

For more information, see Ubuntu virtual machines on Hyper-V.

How can I boot the virtual machine?

Generation 1 and generation 2 VMs support different boot methods, these methods are shown in the following table.

Expand table

Boot method	Generation 1	Generation 2
PXE boot by using a standard network adapter	✕	✓
PXE boot by using a legacy network adapter	✓	✕
Boot from a SCSI virtual hard disk (`.VHDX`) or virtual DVD (.ISO)	✕	✓
Boot from IDE Controller virtual hard disk (`.VHD`), virtual DVD (`.ISO`) or a physical CD/DVD drive	✓	✕
Boot from virtual floppy (`.VFD`)	✓	✕

What's the difference in device support?

The following table compares the devices available between generation 1 and generation 2 virtual machines.

Expand table

Generation 1 Device	Generation 2 Replacement	Generation 2 Enhancements
IDE controller	Virtual SCSI controller	Boot from `.VHDX` (64 TB maximum size, and online resize capability)
IDE CD-ROM	Virtual SCSI CD-ROM	Support for up to 64 SCSI DVD devices per SCSI controller.
Legacy BIOS	UEFI firmware	Secure Boot
Legacy network adapter	Synthetic network adapter	Network boot with IPv4 and IPv6
Floppy controller and DMA controller	No floppy controller support	N/A
Universal asynchronous receiver/transmitter (UART) for COM ports	Optional UART for debugging	Faster and more reliable
i8042 keyboard controller	Software-based input	Uses fewer resources because there's no emulation. Also reduces the attack surface from the guest operating system.
PS/2 keyboard	Software-based keyboard	Uses fewer resources because there's no emulation. Also reduces the attack surface from the guest operating system.

Generation 1 Device	Generation 2 Replacement	Generation 2 Enhancements
PS/2 mouse	Software-based mouse	Uses fewer resources because there's no emulation. Also reduces the attack surface from the guest operating system.
S3 video	Software-based video	Uses fewer resources because there's no emulation. Also reduces the attack surface from the guest operating system.
PCI bus	No longer required	N/A
Programmable interrupt controller (PIC)	No longer required	N/A
Programmable interval timer (PIT)	No longer required	N/A
Super I/O device	No longer required	N/A

Considerations for using generation 1 and generation 2 virtual machines

Here are some more tips about using the different generations of virtual machines.

Creating VMs with more than 64 logical CPUs

Hyper-V manager might fail to create a new generation 1 VM on a system with more than 64 logical CPUs. Hyper-V manager doesn't let you specify the number of virtual processors at VM creation time. For hosts with more than 64 logical processors, specify the number of virtual processors at VM creation using Windows Admin Center, PowerShell, or another tool.

Uploading a virtual hard drive to Azure

Virtual hard drives created on generation 1 and generation 2 VMs can be uploaded to Azure as long as they use the VHD file format. The virtual hard drive must have a fixed (not dynamically expanding) sized disk. See Generation 2 VMs on Azure to learn more about generation 2 capabilities supported on Azure. For more information on uploading a Windows VHD or VHDX, see Prepare a Windows VHD or VHDX to upload to Azure.

Attach or add a DVD drive

- You can't attach a physical CD or DVD drive to a generation 2 virtual machine. The virtual DVD drive in generation 2 virtual machines only supports ISO image files. To create an ISO image file of a Windows environment, you can use the *OScdimg* command line tool. For more information, see Oscdimg Command-Line Options.
- When you create a new virtual machine with the `New-VM` Windows PowerShell cmdlet, the generation 2 virtual machine doesn't have a DVD drive. You can add a DVD drive while the virtual machine is running.

Use UEFI firmware

- Secure Boot or UEFI firmware isn't required on the physical Hyper-V host. For generation 2 VMs, Hyper-V provides virtual firmware to virtual machines that is independent of what's on the Hyper-V host.
- UEFI firmware in a generation 2 virtual machine doesn't support setup mode for Secure Boot.
- We don't support running a UEFI shell or other UEFI applications in a generation 2 virtual machine. Using a non-Microsoft UEFI shell or UEFI applications is technically possible if they're compiled directly from the sources. If these applications aren't digitally signed correctly, you must disable Secure Boot for the virtual machine.

Work with VHDX files

- You can resize a VHDX file that contains the boot volume for a generation 2 virtual machine while the virtual machine is running.
- We don't support or recommend that you create a single virtual disk (VHD or VHDX file) that is bootable to **both** generation 1 and generation 2 virtual machines. Instead, create bootable VHDX files that target only generation 1 **or** generation 2 virtual machines.
- The virtual machine generation is a property of the virtual machine, not a property of the virtual hard disk. You can't tell if a VHDX file was created as a generation 1 or a generation 2 virtual machine.
- A VHDX file created with a generation 2 virtual machine can be attached to the IDE controller or the SCSI controller of a generation 1 virtual machine. However, if the virtual hard drive is a bootable VHDX file, the generation 1 virtual machine fails to boot.

Use IPv6 instead of IPv4

When you boot from network with PXE, generation 2 virtual machines use IPv4 by default. To use IPv6 instead, run the Set-VMFirmware Windows PowerShell cmdlet. For example, the following command sets the preferred protocol to IPv6 for a virtual machine named TestVM:

```
PowerShell
Set-VMFirmware -VMName 'TestVM' -IPProtocolPreference IPv6
```

Add a COM port for kernel debugging

COM ports aren't available in generation 2 virtual machines until you add them. You can add COM ports with Windows PowerShell or Windows Management Instrumentation (WMI). These steps show you how to do it with Windows PowerShell.

To add a COM port:

1. Disable Secure Boot. Kernel debugging isn't compatible with Secure Boot. Make sure the virtual machine is in an Off state, then use the Set-VMFirmware cmdlet. For example, the following command disables Secure Boot on virtual machine TestVM:

   ```
   PowerShell
   Set-VMFirmware -VMName 'TestVM' -EnableSecureBoot Off
   ```

2. Add a COM port. Use the Set-VMComPort cmdlet to add a COM port. For example, the following command configures the first COM port on virtual machine, TestVM, to connect to the named pipe, TestPipe, on the local computer:

   ```
   PowerShell
   Set-VMComPort -VMName 'TestVM' -Number 1 -Path '\\.\pipe\TestPipe'
   ```

> **Note**
>
> Configured COM ports aren't listed in the settings of a virtual machine in Hyper-V Manager.

See Also

- Linux and FreeBSD Virtual Machines on Hyper-V
- Use local resources on Hyper-V virtual machine with VMConnect
- Plan for Hyper-V scalability in Windows Server 2016

Plan for Hyper-V networking in Windows Server

Article • 12/07/2022

> Applies to: Windows Server 2022, Microsoft Hyper-V Server 2016, Windows Server 2016, Microsoft Hyper-V Server 2019, Windows Server 2019

A basic understanding of networking in Hyper-V helps you plan networking for virtual machines. This article also covers some networking considerations when using live migration and when using Hyper-V with other server features and roles.

Hyper-V networking basics

Basic networking in Hyper-V is fairly simple. It uses two parts - a virtual switch and a virtual networking adapter. You'll need at least one of each to establish networking for a virtual machine. The virtual switch connects to any Ethernet-based network. The virtual network adapter connects to a port on the virtual switch, which makes it possible for a virtual machine to use a network.

The easiest way to establish basic networking is to create a virtual switch when you install Hyper-V. Then, when you create a virtual machine, you can connect it to the switch. Connecting to the switch automatically adds a virtual network adapter to the virtual machine. For instructions, see Create a virtual switch for Hyper-V virtual machines.

To handle different types of networking, you can add virtual switches and virtual network adapters. All switches are part of the Hyper-V host, but each virtual network adapter belongs to only one virtual machine.

The virtual switch is a software-based layer-2 Ethernet network switch. It provides built-in features for monitoring, controlling, and segmenting traffic, as well as security, and diagnostics. You can add to the set of built-in features by installing plug-ins, also called *extensions*. These are available from independent software vendors. For more information about the switch and extensions, see Hyper-V Virtual Switch.

Switch and network adapter choices

Hyper-V offers three types of virtual switches and two types of virtual network adapters. You'll choose which one of each you want when you create it. You can use Hyper-V

Manager or the Hyper-V module for Windows PowerShell to create and manage virtual switches and virtual network adapters. Some advanced networking capabilities, such as extended port access control lists (ACLs), can only be managed by using cmdlets in the Hyper-V module.

You can make some changes to a virtual switch or virtual network adapter after you create it. For example, it's possible to change an existing switch to a different type, but doing that affects the networking capabilities of all the virtual machines connected to that switch. So, you probably won't do this unless you made a mistake or need to test something. As another example, you can connect a virtual network adapter to a different switch, which you might do if you want to connect to a different network. But, you can't change a virtual network adapter from one type to another. Instead of changing the type, you'd add another virtual network adapter and choose the appropriate type.

Virtual switch types are:

- **External virtual switch** - Connects to a wired, physical network by binding to a physical network adapter.
- **Internal virtual switch** - Connects to a network that can be used only by the virtual machines running on the host that has the virtual switch, and between the host and the virtual machines.
- **Private virtual switch** - Connects to a network that can be used only by the virtual machines running on the host that has the virtual switch, but doesn't provide networking between the host and the virtual machines.

Virtual switch options:

Setting name	Description
Allow management operating system to share this network adapter	Allow the Hyper-V host to share the use of the virtual switch and NIC or NIC team with the virtual machine. With this enabled, the host can use any of the settings that you configure for the virtual switch, such as Quality of Service (QoS) settings, security settings, or other features of the Hyper-V virtual switch.
Enable single-root I/O virtualization (SR-IOV)	Allow virtual machine traffic to bypass the virtual machine switch and go directly to the physical NIC. SR-IOV is only available for virtual machines running Windows Server. For more information, see Single-Root I/O Virtualization in the Poster Companion Reference: Hyper-V Networking.

Virtual network adapter types are:

- **Hyper-V specific network adapter** - Available for both generation 1 and generation 2 virtual machines. It's designed specifically for Hyper-V and requires a driver that's included in Hyper-V integration services. This type of network adapter faster and is the recommended choice unless you need to boot to the network or are running an unsupported guest operating system. The required driver is provided only for supported guest operating systems. Note that in Hyper-V Manager and the networking cmdlets, this type is just referred to as a network adapter.

- **Legacy network adapter** - Available only in generation 1 virtual machines. Emulates an Intel 21140-based PCI Fast Ethernet Adapter and can be used to boot to a network so you can install an operating system from a service such as Windows Deployment Services.

Hyper-V networking and related technologies

Recent Windows Server releases introduced improvements that give you more options for configuring networking for Hyper-V. For example, Windows Server 2012 introduced support for converged networking. This lets you route network traffic through one external virtual switch. Windows Server 2016 builds on this by allowing Remote Direct Memory Access (RDMA) on network adapters bound to a Hyper-V virtual switch. You can use this configuration either with or without Switch Embedded Teaming (SET). For details, see Remote Direct Memory Access (RDMA) and Switch Embedded Teaming (SET)

Some features rely on specific networking configurations or do better under certain configurations. Consider these when planning or updating your network infrastructure.

Failover clustering - It's a best practice to isolate cluster traffic and use Hyper-V Quality of Service (QoS) on the virtual switch. For details, see Network Recommendations for a Hyper-V Cluster

Live migration - Use performance options to reduce network and CPU usage and the time it takes to complete a live migration. For instructions, see Set up hosts for live migration without Failover Clustering.

Storage Spaces Direct - This feature relies on the SMB3.0 network protocol and RDMA. For details, see Storage Spaces Direct in Windows Server 2016.

Plan for Hyper-V scalability in Windows Server

Article • 04/10/2024

> **Important**
>
> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

This article gives you details about the maximum configuration for components you can add and remove on a Hyper-V host or its virtual machines, such as virtual processors or checkpoints. As you plan your deployment, consider the maximums that apply to each virtual machine, and those that apply to the Hyper-V host. Maximums continue to grow in Windows Server versions, in response to requests to support newer scenarios such as machine learning and data analytics.

> **Note**
>
> For information about System Center Virtual Machine Manager (VMM), see Virtual Machine Manager. VMM is a Microsoft product for managing a virtualized data center that is sold separately.

Maximums for virtual machines

These maximums apply to each virtual machine when the host is run the selected product version. The guest operating system might support less than the virtual machine maximum. Not all components are available in both generations of virtual machines. For a comparison of the generations, see Should I create a generation 1 or 2 virtual machine in Hyper-V?

Expand table

Component	Maximum	Notes
Checkpoints	50	The actual number might be lower, depending on the available storage. Each checkpoint is stored as an .avhd file that uses physical storage.

Component	Maximum	Notes
Memory	• 240 TB for generation 2 • 1 TB for generation 1	Review the requirements for the specific operating system to determine the minimum and recommended amounts.
Serial (COM) ports	2	None.
Size of physical disks attached directly to a virtual machine	Varies	Maximum size is determined by the guest operating system.
Virtual Fibre Channel adapters	4	As a best practice, we recommended that you connect each virtual Fibre Channel Adapter to a different virtual SAN.
Virtual floppy devices	1 virtual floppy drive	None.
Virtual hard disk capacity	• 64 TB for VHDX format • 2,040 GB for VHD format	Each virtual hard disk is stored on physical media as either a .vhdx or a .vhd file, depending on the format used by the virtual hard disk.
Virtual IDE disks	4	The startup disk (sometimes called the boot disk) must be attached to one of the IDE devices. The startup disk can be either a virtual hard disk or a physical disk attached directly to a virtual machine.
Virtual processors	• 2048 for generation 2 • 64 for generation 1	The number of virtual processors supported by a guest operating system might be lower. For details, see the information published for the specific operating system.
Virtual SCSI controllers	4	Use of virtual SCSI devices requires integration services, which are available for supported guest operating systems. For details on which operating systems are supported, see Supported Linux and FreeBSD virtual machines and Supported Windows guest operating systems.
Virtual SCSI disks	256	Each SCSI controller supports up to 64 disks, which means that each virtual machine can be configured with as many as 256 virtual SCSI disks. (4 controllers x 64 disks per controller)

Component	Maximum	Notes
Virtual network adapters	68 adapters total: • 64 Hyper-V specific network adapters • 4 legacy network adapters;	The Hyper-V specific network adapter provides better performance and requires a driver included in integration services. For more information, see Plan for Hyper-V networking in Windows Server.

Maximums for Hyper-V hosts

These maximums apply to each Hyper-V host running the selected product version.

Expand table

Component	Maximum	Notes
Logical processors	2,048	Both of these features must be enabled in the firmware: • Hardware-assisted virtualization • Hardware-enforced Data Execution Prevention (DEP)
Memory	• 4 PB for hosts that support 5-level paging • 256 TB for hosts that support 4-level paging	None.
Network adapter teams (NIC Teaming)	No limits imposed by Hyper-V.	None.
Physical network adapters	No limits imposed by Hyper-V.	None.
Running virtual machines per server	1024	None.
Storage	Limited by what is supported by the host operating system. No limits imposed by Hyper-V.	**Note:** Microsoft supports network-attached storage (NAS) when using SMB 3.0. NFS-based storage isn't supported.

Component	Maximum	Notes
Virtual network switch ports per server	Varies; no limits imposed by Hyper-V.	The practical limit depends on the available computing resources.
Virtual processors available the host	2,048	The limit is applied to the host operating system (root partition)
Virtual processors per logical processor	No ratio imposed by Hyper-V.	None.
Virtual processors per server	2048	None.
Virtual storage area networks (SANs)	No limits imposed by Hyper-V.	None.
Virtual switches	Varies; no limits imposed by Hyper-V.	The practical limit depends on the available computing resources.

Failover Clusters and Hyper-V

This table lists the maximums that apply when using Hyper-V and Failover Clustering. It's important to do capacity planning to ensure that there's enough hardware resources to run all the virtual machines in a clustered environment.

Expand table

Component	Maximum	Notes
Nodes per cluster	64	Consider the number of nodes you want to reserve for failover, and maintenance tasks such as applying updates. We recommend that you plan for enough resources to allow for 1 node to be reserved for failover. Meaning it remains idle until another node is failed over to it, sometimes referred to as a passive node. You can increase this number if you want to reserve more nodes. There's no recommended ratio or multiplier of reserved nodes to active nodes; the only requirement is that the total number of nodes in a cluster can't exceed the maximum of 64.
Running virtual machines per cluster and per node	8,000 per cluster	Several factors can affect the real number of virtual machines you can run at the same time on one node, such as: - Amount of physical memory being used by each virtual machine.

Component	Maximum	Notes
		- Networking and storage bandwidth. - Number of disk spindles, which affects disk I/O performance.

Plan for Hyper-V security in Windows Server

Article • 08/17/2021

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Secure the Hyper-V host operating system, the virtual machines, configuration files, and virtual machine data. Use the following list of recommended practices as a checklist to help you secure your Hyper-V environment.

Secure the Hyper-V host

- **Keep the host OS secure.**
  - Minimize the attack surface by using the minimum Windows Server installation option that you need for the management operating system. For more information, see the Installation Options section of the Windows Server technical content library. We don't recommend that you run production workloads on Hyper-V on Windows 10.
  - Keep the Hyper-V host operating system, firmware, and device drivers up to date with the latest security updates. Check your vendor's recommendations to update firmware and drivers.
  - Don't use the Hyper-V host as a workstation or install any unnecessary software.
  - Remotely manage the Hyper-V host. If you must manage the Hyper-V host locally, use Credential Guard. For more information, see Protect derived domain credentials with Credential Guard.
  - Enable code integrity policies. Use virtualization-based security protected Code Integrity services. For more information, see Device Guard Deployment Guide.

- **Use a secure network.**
  - Use a separate network with a dedicated network adapter for the physical Hyper-V computer.
  - Use a private or secure network to access VM configurations and virtual hard disk files.
  - Use a private/dedicated network for your live migration traffic. Consider enabling IPSec on this network to use encryption and secure your VM's data going over the network during migration. For more information, see Set up hosts for live migration without Failover Clustering.

- **Secure storage migration traffic.**

  Use SMB 3.0 for end-to-end encryption of SMB data and data protection tampering or eavesdropping on untrusted networks. Use a private network to access the SMB share contents to prevent man-in-the-middle attacks. For more information, see SMB Security Enhancements.

- **Configure hosts to be part of a guarded fabric.**

  For more information, see Guarded fabric.

- **Secure devices.**

  Secure the storage devices where you keep virtual machine resource files.

- **Secure the hard drive.**

  Use BitLocker Drive Encryption to protect resources.

- **Harden the Hyper-V host operating system.**

  Use the baseline security setting recommendations described in the Windows Server Security Baseline.

- **Grant appropriate permissions.**
  - Add users that need to manage the Hyper-V host to the Hyper-V administrators group.
  - Don't grant virtual machine administrators permissions on the Hyper-V host operating system.

- **Configure anti-virus exclusions and options for Hyper-V.**

  Windows Defender already has automatic exclusions configured. For more information about exclusions, see Recommended antivirus exclusions for Hyper-V hosts.

- **Don't mount unknown VHDs.** This can expose the host to file system level attacks.

- **Don't enable nesting in your production environment unless it's required.**

  If you enable nesting, don't run unsupported hypervisors on a virtual machine.

For more secure environments:

- **Use hardware with a Trusted Platform Module (TPM) 2.0 chip to set up a guarded fabric.**

For more information, see System requirements for Hyper-V on Windows Server 2016.

Secure virtual machines

- **Create generation 2 virtual machines for supported guest operating systems.**

  For more information, see Generation 2 security settings.

- **Enable Secure Boot.**

  For more information, see Generation 2 security settings.

- **Keep the guest OS secure.**
  - Install the latest security updates before you turn on a virtual machine in a production environment.
  - Install integration services for the supported guest operating systems that need it and keep it up to date. Integration service updates for guests that run supported Windows versions are available through Windows Update.
  - Harden the operating system that runs in each virtual machine based on the role it performs. Use the baseline security setting recommendations that are described in the Windows Security Baseline.

- **Use a secure network.**

  Make sure virtual network adapters connect to the correct virtual switch and have the appropriate security setting and limits applied.

- **Store virtual hard disks and snapshot files in a secure location.**

- **Secure devices.**

  Configure only required devices for a virtual machine. Don't enable discrete device assignment in your production environment unless you need it for a specific scenario. If you do enable it, make sure to only expose devices from trusted vendors.

- **Configure antivirus, firewall, and intrusion detection software** within virtual machines as appropriate based on the virtual machine role.

- **Enable virtualization based security for guests that run Windows 10 or Windows Server 2016 or later.**

  For more information, see the Device Guard Deployment Guide.

- **Only enable Discrete Device Assignment if needed for a specific workload**.

  Due to the nature of passing through a physical device, work with the device manufacturer to understand if it should be used in a secure environment.

For more secure environments:

- **Deploy virtual machines with shielding enabled and deploy them to a guarded fabric.**

  For more information, see Generation 2 security settings and Guarded fabric.

Plan for GPU acceleration in Windows Server

Article • 05/16/2024

> Applies to: Windows Server 2025 (preview), Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article introduces the graphics virtualization capabilities available in Windows Server.

When to use GPU acceleration

Depending on your workload, you might want to consider GPU acceleration. Here's what you should consider before choosing GPU acceleration:

- **App and desktop remoting (VDI/DaaS) workloads**: If you're building an app or desktop remoting service with Windows Server, consider the catalog of apps you expect your users to run. Some types of apps, such as CAD/CAM apps, simulation apps, games, and rendering/visualization apps, rely heavily on 3D rendering to deliver smooth and responsive interactivity. Most customers consider GPUs a necessity for a reasonable user experience with these kinds of apps.
- **Remote rendering, encoding, and visualization workloads**: These graphics-oriented workloads tend to rely heavily on a GPU's specialized capabilities, such as efficient 3D rendering and frame encoding/decoding, in order to achieve cost-effectiveness and throughput goals. For this kind of workload, a single GPU-enabled virtual machine (VM) might be able to match the throughput of many CPU-only VMs.
- **HPC and ML workloads**: For highly data-parallel computational workloads, such as high-performance compute and machine learning model training or inference, GPUs can dramatically shorten time to result, time to inference, and training time. Alternatively, they might offer better cost-effectiveness than a CPU-only architecture at a comparable performance level. Many High Performance Compute (HPC) and machine learning frameworks can use GPU acceleration; consider whether GPU acceleration might benefit your specific workload.

GPU virtualization in Windows Server

GPU virtualization technologies enable GPU acceleration in a virtualized environment, typically within virtual machines. If your workload is virtualized with Hyper-V, then you need to employ graphics virtualization in order to provide GPU acceleration from the physical GPU to your virtualized apps or services. However, if your workload runs directly on physical Windows Server hosts, then you have no need for graphics virtualization; your apps and services already have access to the GPU capabilities and APIs natively supported in Windows Server.

The following graphics virtualization technologies are available to Hyper-V VMs in Windows Server:

- Discrete Device Assignment (DDA)
- GPU Partitioning (GPU-P)

In addition to VM workloads, Windows Server also supports GPU acceleration of containerized workloads within Windows Containers. For more information, see GPU Acceleration in Windows containers.

Discrete Device Assignment (DDA)

Discrete Device Assignment (DDA) allows you to dedicate one or more physical GPUs to a virtual machine. In a DDA deployment, virtualized workloads run on the native driver and typically have full access to the GPU's functionality. DDA offers the highest level of app compatibility and potential performance. DDA can also provide GPU acceleration to Linux VMs, subject to support.

A DDA deployment can accelerate only a limited number of virtual machines, since each physical GPU can provide acceleration to at most one VM. If you're developing a service whose architecture supports shared virtual machines, consider hosting multiple accelerated workloads per VM. For example, if you're building a Remote Desktop Services solution, you can improve user scale by using the multi-session capabilities of Windows Server to host multiple user desktops on each VM. These users share the benefits of GPU acceleration.

For more information, see these articles:

- Plan for deploying Discrete Device Assignment
- Deploy graphics devices using Discrete Device Assignment

GPU Partitioning (GPU-P)

> **Important**
>
> GPU partitioning in Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Beginning with Windows Server 2025, GPU partitioning allows you to share a physical GPU device with multiple virtual machines (VMs). With GPU partitioning or GPU virtualization, each VM gets a dedicated fraction of the GPU instead of the entire GPU.

The GPU partitioning uses the Single Root IO Virtualization (SR-IOV) interface, which provides a hardware-backed security boundary with predictable performance for each VM. Each VM can access only the GPU resources dedicated to them and the secure hardware partitioning prevents unauthorized access by other VMs.

To learn more about GPU partitioning, see these articles:

- GPU partitioning
- Partition and assign GPUs to a virtual machine

Comparing DDA and GPU partitioning

Consider the following functionality and support differences between graphics virtualization technologies when planning your deployment:

Expand table

Description	Discrete Device Assignment	GPU Partitioning
GPU resource model	Dedicated only	Partitioned
VM density	Low (one or more GPUs to one VM)	High (one or more GPUs to many VMs)
App compatibility	All GPU capabilities provided by vendor (DX 12, OpenGL, CUDA)	All GPU capabilities provided by vendor (DX 12, OpenGL, CUDA)
AVC444	Available through Group Policy	Available through Group Policy
GPU VRAM	Up to VRAM supported by the GPU	Up to VRAM supported by the GPU per partition

Description	Discrete Device Assignment	GPU Partitioning
GPU driver in guest	GPU vendor driver (NVIDIA, AMD, Intel)	GPU vendor driver (NVIDIA, AMD, Intel)

Plan for deploying devices by using Discrete Device Assignment

Article • 06/12/2023

> Applies to: Windows Server 2022, Microsoft Hyper-V Server 2019, Windows Server 2019, Microsoft Hyper-V Server 2016, Windows Server 2016

Discrete Device Assignment allows physical Peripheral Component Interconnect Express (PCIe) hardware to be directly accessible from within a virtual machine (VM). This article discusses the type of devices that can be used, host system requirements, limitations imposed on the VMs, and security implications.

For Discrete Device Assignment, Microsoft supports two device classes: Graphics Adapters and NVMe Storage devices. Other devices are likely to work, and hardware vendors are able to offer statements of support for those devices. For other devices, contact specific hardware vendors for support.

To learn about other methods of GPU virtualization, see Plan for GPU acceleration in Windows Server. If you're ready to try Discrete Device Assignment, you can go to Deploy graphics devices using Discrete Device Assignment or Deploy NVMe Storage Devices using Discrete Device Assignment.

Supported VMs and guest operating systems

Discrete Device Assignment is supported for Generation 1 or 2 VMs. The guests supported include:

- Windows 10 or later
- Windows Server 2016 or later
- Windows Server 2012 R2 with the Update to add Discrete Device Assignment support for Azure.

For more information, see Supported Linux and FreeBSD virtual machines for Hyper-V on Windows Server and Windows.

System requirements

Your system must meet the Hardware Requirements for Windows Server and System Requirements for Hyper-V on Windows Server. Discrete Device Assignment also requires

server class hardware that's capable of granting the operating system control over configuring the PCIe fabric (Native PCI Express Control). In addition, the PCIe Root Complex has to support Access Control Services (ACS), which enables Hyper-V to force all PCIe traffic through the Input-Output Memory Management Unit.

These capabilities usually aren't exposed directly in the BIOS of the server and are often hidden behind other settings. If the same capabilities are required for SR-IOV support and in the BIOS, you might need to set "Enable SR-IOV." Reach out to your system vendor if you're unable to identify the correct setting in your BIOS.

To help ensure the hardware is capable of Discrete Device Assignment, you can run the machine profile script on a Hyper-V enabled host. The script tests if your server is correctly set up and what devices are capable of Discrete Device Assignment.

Device requirements

Not every PCIe device can be used with Discrete Device Assignment. Older devices that use legacy (INTx) PCI Interrupts aren't supported. For more information, see Discrete Device Assignment - Machines and devices. You can also run the Machine Profile Script to display which devices are capable of being used for Discrete Device Assignment.

Device manufacturers can reach out to their Microsoft representative for more details.

Device driver

Discrete Device Assignment passes the entire PCIe device into the Guest VM. A host driver isn't required to be installed prior to the device being mounted within the VM. The only requirement on the host is that the device's PCIe Location Path can be determined. The device's driver can be installed to help in identifying the device. A GPU without its device driver installed on the host might appear as a Microsoft Basic Render Device. If the device driver is installed, its manufacturer and model is likely to be displayed.

When the device is mounted inside the guest, the Manufacturer's device driver can be installed like normal inside the guest VM.

VM limitations

Due to the nature of how Discrete Device Assignment is implemented, some features of a VM are restricted while a device is attached. The following features aren't available:

- VM Save/Restore
- Live migration of a VM
- The use of dynamic memory
- Adding the VM to a high availability (HA) cluster

Security

Discrete Device Assignment passes the entire device into the VM. This pass means all capabilities of that device are accessible from the guest operating system. Some capabilities, like firmware updating, might adversely affect the stability of the system. Numerous warnings are presented to the admin when dismounting the device from the host. You should only use Discrete Device Assignment where the tenants of the VMs are trusted.

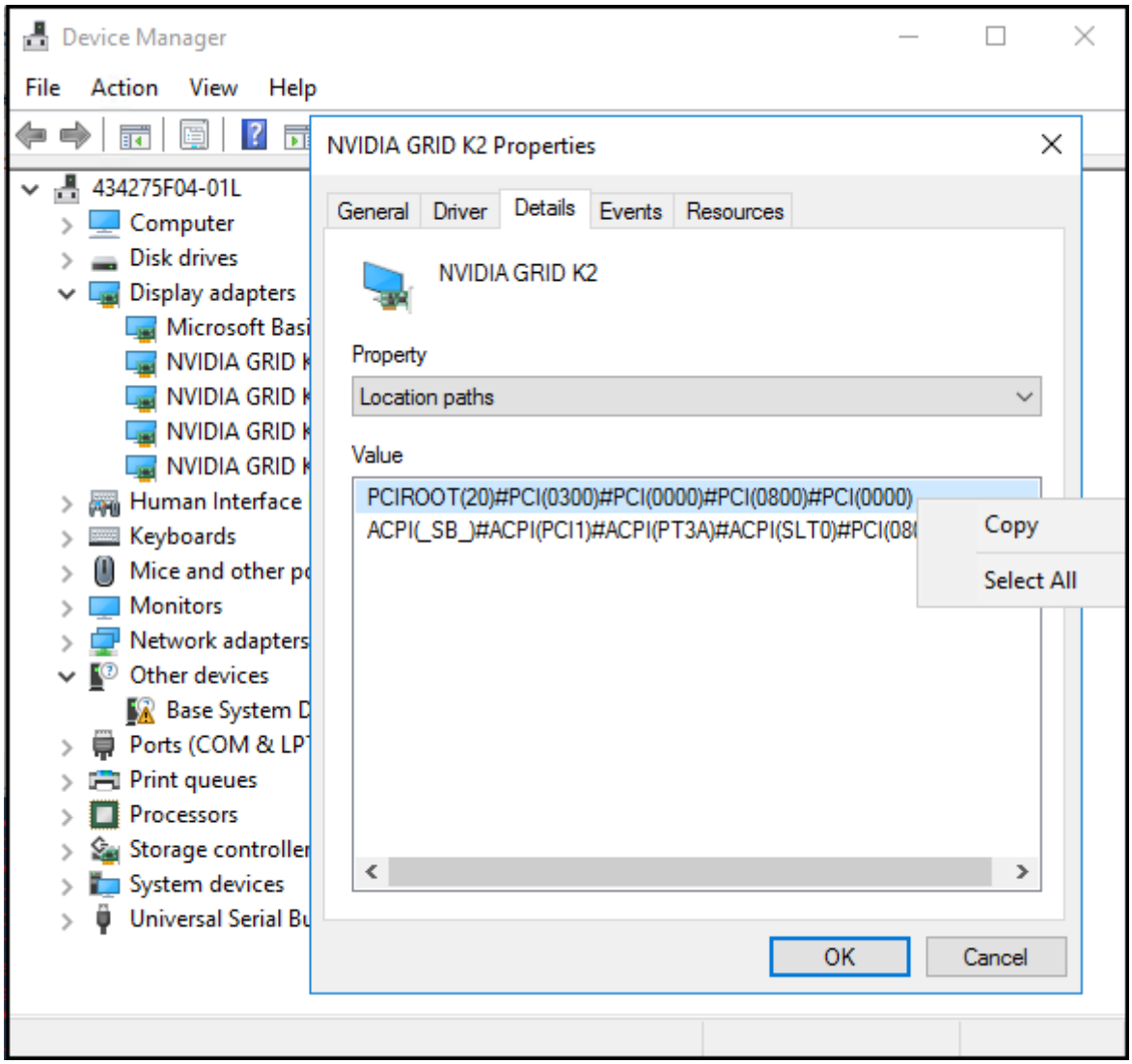
If the admin desires to use a device with an untrusted tenant, device manufactures can create a Device Mitigation driver that can be installed on the host. Contact the device manufacturer for details on whether they provide a Device Mitigation Driver.

If you would like to bypass the security checks for a device that doesn't have a Device Mitigation Driver, you have to pass the `-Force` parameter to the `Dismount-VMHostAssignableDevice` cmdlet. When you make this pass, you have changed the security profile of that system. You should only make this change during prototyping or trusted environments.

PCIe location path

The PCIe location path is required to dismount and mount the device from the Host. An example location path is `PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)`. The Machine Profile Script also returns the location path of the PCIe device.

Get the location path by using Device Manager



1. Open Device Manager and locate the device.
2. Right-click the device and select **Properties**.
3. On the **Details** tab, expand the **Property** drop-down menu and select **Location Paths**.
4. Right-click the entry that begins with **PCIROOT** and select **Copy** to get the location path for the device.

MMIO space

Some devices, especially GPUs, require more MMIO space to be allocated to the VM for the memory of that device to be accessible. By default, each VM starts off with 128 MB of low MMIO space and 512 MB of high MMIO space allocated to it. However, a device might require more MMIO space, or multiple devices might be passed through such that the combined requirements exceed these values. Changing MMIO Space is straightforward and can be performed in PowerShell by using the following commands:

PowerShell

```
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName $vm
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName $vm
```

The easiest way to determine how much MMIO space to allocate is to use the Machine Profile Script. To download and run the Machine Profile Script, run the following commands in a PowerShell console:

PowerShell

```
curl -o SurveyDDA.ps1
https://raw.githubusercontent.com/MicrosoftDocs/Virtualization-
Documentation/live/hyperv-tools/DiscreteDeviceAssignment/SurveyDDA.ps1
.\SurveyDDA.ps1
```

For devices that can be assigned, the script displays the MMIO requirements of a given device. The following script output is an example:

PowerShell

```
NVIDIA GRID K520
Express Endpoint -- more secure.
    ...
    And it requires at least: 176 MB of MMIO gap space
...
```

The low MMIO space is used only by 32-bit operating systems and devices that use 32-bit addresses. In most circumstances, setting the high MMIO space of a VM is enough since 32-bit configurations aren't common.

> **ⓘ Important**
>
> When you assign MMIO space to a VM, be sure to specify sufficient MMIO space. The MMIO space should be the sum of the requested MMIO space for all desired assigned devices plus a buffer for other virtual devices that require a few MB of MMIO space. Use the default MMIO values previously described as the buffer for low and high MMIO (128 MB and 512 MB, respectively).

Consider the previous example. If you assign a single K520 GPU, set the MMIO space of the VM to the value outputted by the machine profile script plus a buffer: 176 MB + 512 MB. If you assign three K520 GPUs, set the MMIO space to three times the base amount of 176 MB plus a buffer, or 528 MB + 512 MB.

For a more in-depth look at MMIO space, see Discrete Device Assignment - GPUs on the Tech Community blog.

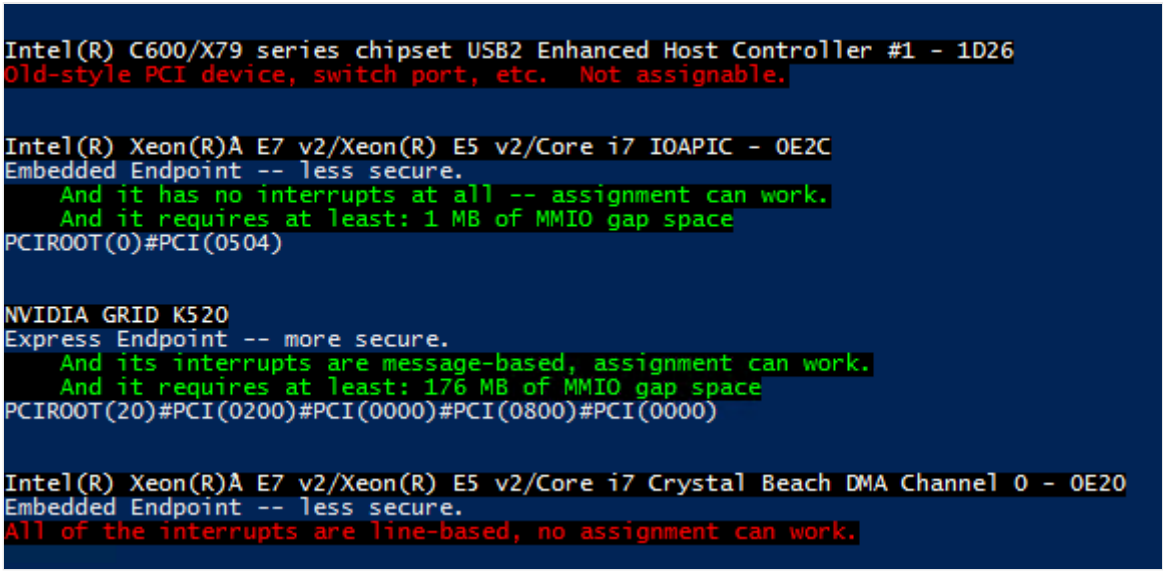
Machine profile script

To identify if the server is configured correctly, and what devices can be passed through by using Discrete Device Assignment, run the SurveyDDA.ps1. PowerShell script.

Before you use the script, ensure you have the Hyper-V role installed and you run the script from a PowerShell command window that has Administrator privileges.

If the system is incorrectly configured to support Discrete Device Assignment, the tool displays an error message with details about the issue. If the system is correctly configured, the tool enumerates all devices located on the PCIe Bus.

For each device it finds, the tool displays whether it's able to be used with Discrete Device Assignment. If a device is identified as being compatible with Discrete Device Assignment, the script provides a reason. When a device is successfully identified as being compatible, the device's Location Path is displayed. Additionally, if that device requires MMIO space, it's displayed as well.



What is Nested Virtualization?

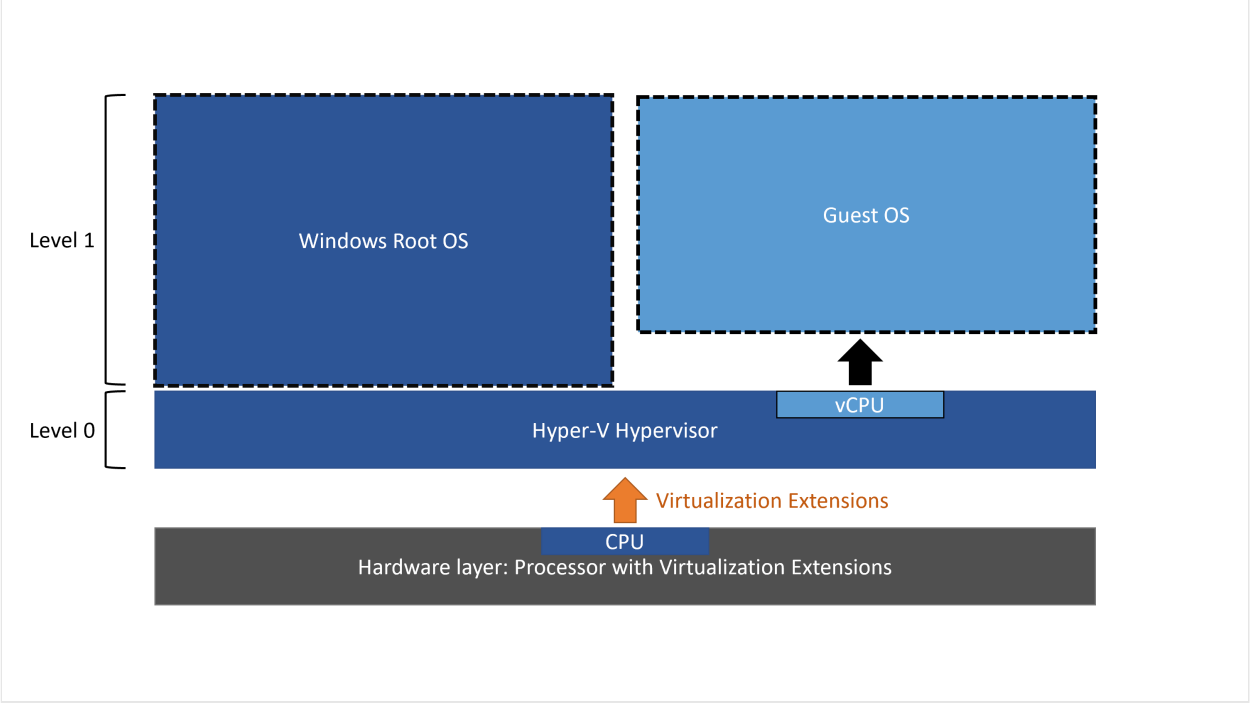
Article • 07/31/2024

Nested virtualization is a feature that lets you run Hyper-V inside a Hyper-V virtual machine (VM). Over the years hardware has improved and the use cases for Nested Virtualization have expanded. For example, Nested Virtualization can be useful for:

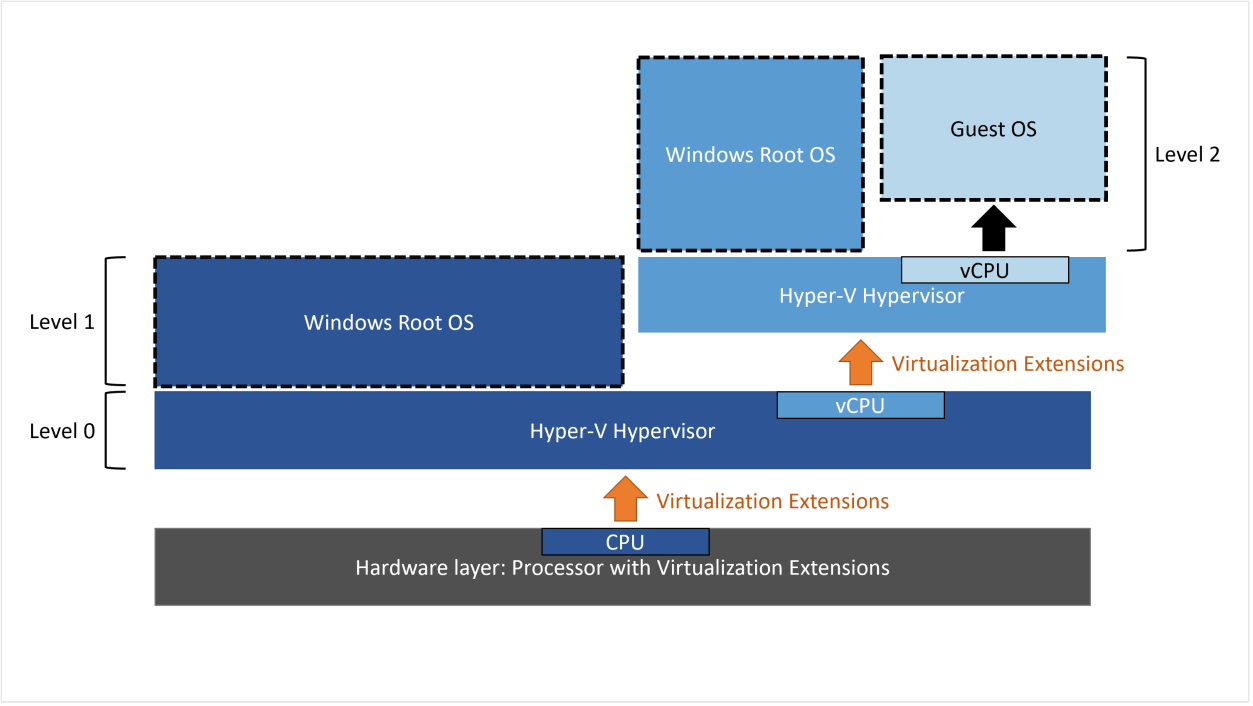
- Running applications or emulators in a nested VM
- Testing software releases on VMs
- Reducing deployment times for training environments
- Using Hyper-V isolation for containers

Modern processors include hardware features that make virtualization faster and more secure. Hyper-V relies on these processor extensions to run virtual machines, for example, Intel VT-x and AMD-V. Nested virtualization makes this hardware support available to guest virtual machines.

The following diagram shows Hyper-V without nesting. The Hyper-V hypervisor takes full control of the hardware virtualization capabilities (orange arrow), and doesn't expose them to the guest operating system.



In contrast, the following diagram shows Hyper-V with Nested Virtualization enabled. In this case, Hyper-V exposes the hardware virtualization extensions to its virtual machines. With nesting enabled, a guest virtual machine can install its own hypervisor and run its own guest VMs.



Dynamic Memory and Runtime Memory Resize

When Hyper-V is running inside a virtual machine, the virtual machine must be turned off to adjust its memory. Meaning that even if dynamic memory is enabled, the amount of memory doesn't fluctuate. Simply enabling nested virtualization has no effect on dynamic memory or runtime memory resize.

For virtual machines without dynamic memory enabled, attempting to adjust the amount of memory while running fails. The incompatibility only occurs while Hyper-V is running in the VM.

Third party virtualization apps

Virtualization applications other than Hyper-V aren't supported in Hyper-V virtual machines, and are likely to fail. Virtualization applications include any software that requires hardware virtualization extensions.

Supported scenarios

Using a nested Hyper-V VM in production is supported for both Azure and on-premises in the following scenarios. We also recommend you make sure that your services and applications are also supported.

Nested Virtualization isn't suitable for Windows Server Failover Clustering, and performance sensitive applications. We recommended you fully evaluate the services

and applications.

Hyper-V VMs on Hyper-V VMs

Running Hyper-V VMs nested on Hyper-V VMs is great for test labs and evaluation environments. Especially where configurations can be easily modified and saved states can be used to revert to specific configurations. Test labs don't typically require the same service level agreement (SLA) as production environments.

Production environments running Hyper-V VMs running on Hyper-V VMs are supported. However, it's recommended to make sure that your services and applications are also supported. If you use nested Hyper-V VM in production, it is recommended to fully evaluate whether the services or applications provide the expected behavior.

To learn more about setting up Nested Virtualization on Azure, see our Tech Community blog How to Setup Nested Virtualization for Azure VM/VHD.

Third party virtualization on Hyper-V virtualization

Whilst it might be possible for third party virtualization to run on Hyper-V, Microsoft doesn't test this scenario. Third party virtualization on Hyper-V virtualization isn't supported, ensure your hypervisor vendor supports this scenario.

Hyper-V virtualization on third party virtualization

Whilst it might be possible for Hyper-V virtualization to run on third party virtualization, Microsoft doesn't test this scenario. Hyper-V virtualization on third party virtualization isn't supported, ensure your hypervisor vendor supports this scenario.

Azure Stack HCI nested on Hyper-V VMs

Azure Stack HCI is designed and tested to run on validated physical hardware. Azure Stack HCI can run nested in a virtual machine for evaluation, but production environments in a nested configuration aren't supported.

To learn more about Azure Stack HCI nested on Hyper-V VMs, see Nested virtualization in Azure Stack HCI.

Hyper-V isolated containers running nested on Hyper-V

Microsoft offers Hyper-V isolation for containers. This isolation mode offers enhanced security and broader compatibility between host and container versions. With Hyper-V isolation, multiple container instances run concurrently on a host. Each container runs inside of a highly optimized virtual machine and effectively gets its own kernel. Since a Hyper-V isolated container offers isolation through a hypervisor layer between itself and the container host, when the container host is a Hyper-V based virtual machine, there's performance overhead. The associated performance overhead occurs in terms of container start-up time, storage, network, and CPU operations.

When a Hyper-V isolated container is run in a Hyper-V VM, it's running nested. Using a Hyper-V VM opens many useful scenarios but also increases latency, as there are two levels of hypervisors running above the physical host.

Running Hyper-V isolated containers nested on Hyper-V is supported. One level of nested virtualization is supported in production, which allows for isolated container deployments.

To learn more about Nested Hyper-V Containers, see Performance tuning Windows Server Containers.

Running WSL2 in a Hyper-V VM running nested on Hyper-V

Windows Subsystem for Linux (WSL) is a feature of the Windows operating system that enables you to run a Linux file system, along with Linux command-line tools and GUI apps, directly on Windows.

Running WSL2 in a Hyper-V VM running nested on Hyper-V is supported.

To learn more about how to enabled WSL 2 to run in a VM, see Frequently Asked Questions about Windows Subsystem for Linux.

Next step

- Run Hyper-V in a Virtual Machine with Nested Virtualization

Feedback

Hyper-V Virtual Fibre Channel

Article • 04/26/2024

Hyper-V provides Fibre Channel ports within guest operating systems (OSes) that let you connect to Fibre Channel directly from your virtual machines (VMs). This feature lets you virtualize workloads that use direct access to Fibre Channel storage, cluster guests over Fibre Channel, and gives you more storage options for servers hosted in your virtualization infrastructure. This article gives a brief overview of Fibre Channel on Hyper-V so you can understand how to implement the Virtual Fibre Channel into your Hyper-V planning.

Prerequisites

In order to use Virtual Fibre Channel on Hyper-V, your deployment must have the following:

- One or more installations of Windows Server 2012 or later with the Hyper-V role installed.
- A device with one or more Fibre Channel host bus adapters (HBAs) and an updated HBA driver that supports Virtual Fibre Channel.
    - HBA ports must also have a Fibre Channel topography that can support N_Port ID Virtualization (NPIV), a maximum transfer size of at least 0.5 MB, and data transfers of at least 128 physical pages.
    - The maximum transfer length of the adapter determines the LUN limit. For example, a maximum transfer length of 512k allows about 2,250 LUNs. You need to configure your LUNs and multi-storage paths to maintain these limits.
- An NPIV-enabled Storage Area Network (SAN).
- VMs that use Windows Server 2012 or later as their guest OS and can support virtual Fibre Channel adapters.
- Storage accessible through Virtual Fibre Channel support devices that present logical units. Virtual Fibre Channel logical units can't be used as boot media.

How Virtual Fibre Channel works

Virtual Fibre Channel for Hyper-V provides the guest OS with unmediated access to a SAN by using a standard World Wide Name (WWN) associated with a virtual machine. Hyper-V users can use Fibre Channel SANs to virtualize workloads that require direct access to SAN logical unit numbers (LUNs). Fibre Channel SANs also let you operate in new scenarios, such as running failover clustering inside the guest OS of a VM connected to shared Fibre Channel storage.

Physical deployments use the advanced storage functionality of mid-range and high-end storage arrays to offload certain management tasks from hosts to the Windows software virtual hard disk stack. In virtualized deployments, Virtual Fibre Channel plays a similar role, letting you use your SAN's functionality directly from Hyper-V VMs to offload functionality. For example, when you use Hyper-V to take a snapshot of a LUN, you can offload storage functionality on the SAN hardware by using a hardware Volume Shadow Copy Service (VSS) provider from within a Hyper-V VM.

The following sections explain which Virtual Fibre Channel features help it offload management and storage tasks.

NPIV support

Virtual Fibre Channel for Hyper-V guests uses the existing N_Port ID Virtualization (NPIV) T11 standard to map multiple virtual N_Port IDs to a single physical Fibre Channel N_port. A new NPIV port is created on the host every time you start a VM configured with a virtual HBA. When the VM stops running on the host, the system removes the NPIV port. You should configure any HBA ports that you use for Virtual Fibre Channel in a Fibre Channel topology that supports NPIV, and your SAN should also support NPIV ports.

Virtual SAN support

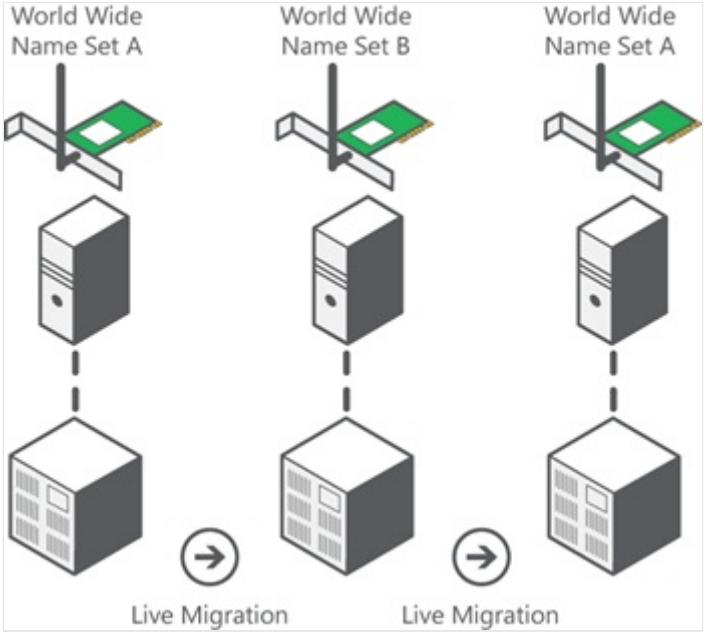
A virtual SAN defines a named group of physical Fibre Channel ports connected to the same physical SAN. Hyper-V lets you define virtual SANs on the host in scenarios where you connect a single Hyper-V host to different SANs through many Fibre Channel ports.

For example, if a Hyper-V host is connected to a production SAN and a test SAN, then the host is connected to each SAN through two physical Fibre Channel ports. In this scenario, you can configure two virtual SANs. The virtual production SAN has the two physical Fibre Channel ports connected to the physical production SAN. The virtual test SAN has two physical Fibre Channel ports connected to the test SAN. You can use the same technique to name two different paths to the same storage target.

You can configure up to four virtual Fibre Channel adapters on a VM and associate each one with a virtual SAN. Each virtual Fibre Channel adapter connects with one WWN address, and you can set each WWN address automatically or manually. However, in live migration scenarios, two WWN addresses are assigned to each adapter.

Live migration

To support live migration of virtual machines across Hyper-V hosts while maintaining Fibre Channel connectivity, two WWNs are configured for each virtual Fibre Channel adapter, as shown in the following diagram. Hyper-V automatically alternates between the Set A and Set B WWN addresses during a live migration. This ensures that all LUNs are available on the destination host before the migration and that no downtime occurs during the migration.



Tape library support

Windows Server only supports virtual tape libraries configured with virtual Fibre Channel adapters when using System Center Data Protection Manager 2012 R2 U3 or later with certified hardware. You can check if your virtual Fibre Channel adapter supports a tape library by contacting the tape library hardware vendor. You can also check compatibility by running the Data Protection Management (DPM) Tape Library Compatibility Test tool on the virtual tape library in a guest VM. For more information about the DPM Tape Library Compatibility Test, see Verify tape library compatibility.

MPIO functionality

Hyper-V in Windows Server can use the multipath I/O (MPIO) functionality to ensure continuous connectivity to Fibre Channel storage from within a VM. You can use MPIO functionality with Fibre Channel in the following ways:

- Use MPIO for host access. To use this functionality, you install multiple Fibre Channel ports on the host, then use MPIO to provide highly available connectivity to the LUNs the host can access.
- Virtualize workloads that use MPIO. To use this functionality, configure multiple virtual Fibre Channel adapters inside a VM, then use a separate copy of MPIO within the VM operating system to connect to the LUNs that the VM can access. This configuration can coexist with a host MPIO setup.
- Use different device-specific modules (DSMs) for the host or each VM. This approach allows live migration of the virtual machine configuration, including DSM, connectivity between hosts, and compatibility with existing server configurations and DSMs.

> **Note**
>
> Live migration scenarios where the guest VM uses MPIO don't support Asymmetric Logical Unit Assignment (ALUA).

Related content

- Plan for Hyper-V on Windows Server
- Plan for Hyper-V scalability on Windows Server

Configure virtual local area networks for Hyper-V

Article • 08/13/2024

Virtual local area networks (VLANs) offer one way to isolate network traffic. VLANs are configured in switches and routers that support 802.1q. If you configure multiple VLANs and want communication to occur between them, you need to configure the network devices to allow that.

You need the following to configure VLANs:

- A physical network adapter and driver that supports 802.1q VLAN tagging.
- A physical network switch that supports 802.1q VLAN tagging.

On the host, you configure the virtual switch to allow network traffic on the physical switch port. This is for the VLAN IDs that you want to use internally with virtual machines. Next, you configure the virtual machine to specify the VLAN that the virtual machine uses for all network communications.

To allow a virtual switch to use a VLAN

1. In Hyper-V Manager, select **Virtual Switch Manager** from the **Actions** pane on the right.
2. In **Virtual Switch Manager**, under **Virtual Switches** on the left, select a virtual switch connected to a physical network adapter that supports VLANs.
3. Under **VLAN ID** in the right pane, select **Enable virtual LAN identification for management operating system** and then type a number for the VLAN ID.
4. Select **OK**.

All traffic that goes through the physical network adapter connected to the virtual switch is tagged with the VLAN ID you set.

To allow a virtual machine to use a VLAN

1. In Hyper-V Manager, under **Virtual Machines**, right-click the appropriate virtual machine and select **Settings**. Or, select the machine and then select **Settings** under the machine name in the right pane.

2. On the **Settings** screen, under **Hardware** in the left pane, select a **Network Adapter** that has a virtual switch that's set up with a VLAN.
3. Under **VLAN ID** in the right pane, select **Enable virtual LAN identification**, and then type the same VLAN ID as the one you specified for the virtual switch.
4. Select **OK**.

If the virtual machine needs to use more VLANs, do one of the following:

- Connect more virtual network adapters to appropriate virtual switches and assign the VLAN IDs. Make sure to configure the IP addresses correctly, and that the traffic you want to route through the VLAN also uses the correct IP address.
- Configure the virtual network adapter in trunk mode by using the Set-VMNetworkAdapterVlan cmdlet.

See also

Hyper-V Virtual Switch

Feedback

Export and Import virtual machines

Article • 01/07/2022

> Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to export and import a virtual machine, which is a quick way to move or copy them. This article also discusses some of the choices to make when doing an export or import.

Export a Virtual Machine

An export gathers all required files into one unit--virtual hard disk files, virtual machine configuration files, and any checkpoint files. You can do this on a virtual machine that is in either a started or stopped state.

Using Hyper-V Manager

To create a virtual machine export:

1. In Hyper-V Manager, right-click the virtual machine and select **Export**.
2. Choose where to store the exported files, and click **Export**.

When the export is done, you can see all exported files under the export location.

Using PowerShell

Open a session as Administrator and run a command like the following, after replacing <vm name> and <path>:

PowerShell

```
Export-VM -Name \<vm name\> -Path \<path\>
```

For details, see Export-VM.

Import a Virtual Machine

Importing a virtual machine registers the virtual machine with the Hyper-V host. You can import back into the host, or new host. If you're importing to the same host, you don't need to export the virtual machine first, because Hyper-V tries to recreate the virtual machine from available files. Importing a virtual machine registers it so it can be used on the Hyper-V host.

> **ⓘ Important**
>
> Hyper-V virtual machine configurations have a specific version number. You can only import a virtual machine if the Hyper-V host supports that configuration version. Typically, this means that you can import a virtual machine to a Hyper-V host running a newer version of Hyper-V, but you cannot import a virtual machine created on a newer version of Hyper-V to an older version of Hyper-V. See **Supported virtual machine configuration versions** for more information.

The Import Virtual Machine wizard also helps you fix incompatibilities that can exist when moving from one host to another. This is commonly differences in physical hardware, such as memory, virtual switches, and virtual processors.

Import using Hyper-V Manager

To import a virtual machine:

1. From the **Actions** menu in Hyper-V Manager, click **Import Virtual Machine**.
2. Click **Next**.
3. Select the folder that contains the exported files, and click **Next**.
4. Select the virtual machine to import.
5. Choose the import type, and click **Next**. (For descriptions, see Import types, below.)
6. Click **Finish**.

Import using PowerShell

Use the **Import-VM** cmdlet, following the example for the type of import you want. For descriptions of the types, see Import types, below.

Register in place

This type of import uses the files where they are stored at the time of import and retains the virtual machine's ID. The following command shows an example of an import file. Run a similar command with your own values.

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-
29CED892A95A.vmcx'
```

Restore

To import the virtual machine specifying your own path for the virtual machine files, run a command like this, replacing the examples with your values:

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-
29CED892A95A.vmcx' -Copy -VhdDestinationPath 'D:\Virtual Machines\WIN10DOC'
-VirtualMachinePath 'D:\Virtual Machines\WIN10DOC'
```

Import as a copy

To complete a copy import and move the virtual machine files to the default Hyper-V location, run a command like this, replacing the examples with your values:

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-
29CED892A95A.vmcx' -Copy -GenerateNewId
```

For details, see Import-VM.

Import types

Hyper-V offers three import types:

- **Register in-place** – This type assumes export files are in the location where you'll store and run the virtual machine. The imported virtual machine has the same ID as it did at the time of export. Because of this, if the virtual machine is already registered with Hyper-V, it needs to be deleted before the import works. When the import has completed, the export files become the running state files and can't be removed.

- **Restore the virtual machine** – Restore the virtual machine to a location you choose, or use the default to Hyper-V. This import type creates a copy of the exported files and moves them to the selected location. When imported, the virtual machine has the same ID as it did at the time of export. Because of this, if the virtual machine is already running in Hyper-V, it needs to be deleted before the import can be completed. When the import has completed, the exported files remain intact and can be removed or imported again.
- **Copy the virtual machine** – This is similar to the Restore type in that you select a location for the files. The difference is that the imported virtual machine has a new unique ID, which means you can import the virtual machine to the same host multiple times.

Set up hosts for live migration without Failover Clustering

Article • 07/04/2024

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to set up hosts that aren't clustered so you can do live migrations between them. Use these instructions if you didn't set up live migration when you installed Hyper-V, or if you want to change the settings. To set up clustered hosts, use tools for Failover Clustering.

Requirements for setting up live migration

To set up non-clustered hosts for live migration, you'll need:

- A user account with permission to perform the various steps. Membership in the local Hyper-V Administrators group or the Administrators group on both the source and destination computers meets this requirement, unless you're configuring constrained delegation. Membership in the Domain Administrators group is required to configure constrained delegation.

- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.
  For version upgrade instructions, see Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016. For installation instructions, see Install the Hyper-V role on Windows Server.

- Source and destination computers that either belong to the same Active Directory domain, or belong to domains that trust each other.

- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run the tools from the server.

Consider options for authentication and networking

Consider how you want to set up the following:

- **Authentication**: Which protocol will be used to authenticate live migration traffic between the source and destination servers? The choice determines whether you'll need to sign on to the source server before starting a live migration:

  - Kerberos lets you avoid having to sign in to the server, but requires constrained delegation to be set up. See below for instructions.

  - CredSSP lets you avoid configuring constrained delegation, but requires you sign in to the source server. You can do this through a local console session, a Remote Desktop session, or a remote Windows PowerShell session.

    CredSPP requires signing in for situations that might not be obvious. For example, if you sign in to TestServer01 to move a virtual machine to TestServer02, and then want to move the virtual machine back to TestServer01, you'll need to sign in to TestServer02 before you try to move the virtual machine back to TestServer01. If you don't do this, the authentication attempt fails, an error occurs, and the following message is displayed:

    "Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

- **Performance**: Does it make sense to configure performance options? These options can reduce network and CPU usage, as well as make live migrations go faster. Consider your requirements and your infrastructure, and test different configurations to help you decide. The options are described at the end of step 2.

- **Network preference**: Will you allow live migration traffic through any available network, or isolate the traffic to specific networks? As a security best practice, we recommend that you isolate the traffic onto trusted, private networks because live migration traffic is not encrypted when it is sent over the network. Network isolation can be achieved through a physically isolated network or through another trusted networking technology such as VLANs.

Upgrading to Windows Server 2025 (preview)

> **ⓘ Important**

> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Starting with Windows Server 2025 (preview), Credential Guard is enabled by default on all domain-joined servers that aren't Domain Controllers. As a result you might not be able to use CredSSP-based Live Migration with Hyper-V after upgrading to Windows Server 2025. CredSSP-based delegation is the default for Windows Server 2022 and earlier for live migration. Instead use Kerberos constrained Delegation, as described in the following section. For more information, see Live migration with Hyper-V breaks when upgrading to Windows Server 2025.

Step 1: Configure constrained delegation (optional)

If you have decided to use Kerberos to authenticate live migration traffic, configure constrained delegation using an account that is a member of the Domain Administrators group.

Use the Users and Computers snap-in to configure constrained delegation

1. Open the Active Directory Users and Computers snap-in. (From Server Manager, select the server if it's not selected, click **Tools** >> **Active Directory Users and Computers**).
2. From the navigation pane in **Active Directory Users and Computers**, select the domain and double-click the **Computers** folder.
3. From the **Computers** folder, right-click the computer account of the source server and then click **Properties**.
4. From **Properties**, click the **Delegation** tab.
5. On the delegation tab, select **Trust this computer for delegation to the specified services only** and then select **Use any authentication protocol**.
6. Click **Add**.
7. From **Add Services**, click **Users or Computers**.

8. From **Select Users or Computers**, type the name of the destination server. Click **Check Names** to verify it, and then click **OK**.

9. From **Add Services**, in the list of available services, do the following and then click **OK**:

    - To move virtual machine storage, select **cifs**. This is required if you want to move the storage along with the virtual machine, as well as if you want to move only a virtual machine's storage. If the server is configured to use SMB storage for Hyper-V, this should already be selected.

    - To move virtual machines, select **Microsoft Virtual System Migration Service**.

10. On the **Delegation** tab of the Properties dialog box, verify that the services you selected in the previous step are listed as the services to which the destination computer can present delegated credentials. Click **OK**.

11. From the **Computers** folder, select the computer account of the destination server and repeat the process. In the **Select Users or Computers** dialog box, be sure to specify the name of the source server.

The configuration changes take effect after both of the following happen:

- The changes are replicated to the domain controllers that the servers running Hyper-V are logged into.
- The domain controller issues a new Kerberos ticket.

Step 2: Set up the source and destination computers for live migration

This step includes choosing options for authentication and networking. As a security best practice, we recommend that you select specific networks to use for live migration traffic, as discussed above. This step also shows you how to choose the performance option.

Use Hyper-V Manager to set up the source and destination computers for live migration

1. Open Hyper-V Manager. (From Server Manager, click **Tools** >>**Hyper-V Manager**.)

2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to

add more servers.)

3. In the **Action** pane, click **Hyper-V Settings** >>**Live Migrations**.
4. In the **Live Migrations** pane, check **Enable incoming and outgoing live migrations**.
5. Under **Simultaneous live migrations**, specify a different number if you don't want to use the default of 2.
6. Under **Incoming live migrations**, if you want to use specific network connections to accept live migration traffic, click **Add** to type the IP address information. Otherwise, click **Use any available network for live migration**. Click **OK**.
7. To choose Kerberos and performance options, expand **Live Migrations** and then select **Advanced Features**.
    - If you have configured constrained delegation, under **Authentication protocol**, select **Kerberos**.
    - Under **Performance options**, review the details and choose a different option if it's appropriate for your environment.
8. Click **OK**.
9. Select the other server in Hyper-V Manager and repeat the steps.

Use Windows PowerShell to set up the source and destination computers for live migration

Three cmdlets are available for configuring live migration on non-clustered hosts: Enable-VMMigration, Set-VMMigrationNetwork, and Set-VMHost. This example uses all three and does the following:

- Configures live migration on the local host
- Allows incoming migration traffic only on a specific network
- Chooses Kerberos as the authentication protocol

Each line represents a separate command.

PowerShell

```
PS C:\> Enable-VMMigration

PS C:\> Set-VMMigrationNetwork 192.168.10.1
```

```
PS C:\> Set-VMHost -VirtualMachineMigrationAuthenticationType Kerberos
```

Set-VMHost also lets you choose a performance option (and many other host settings). For example, to choose SMB but leave the authentication protocol set to the default of CredSSP, type:

PowerShell

```
PS C:\> Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
```

This table describes how the performance options work.

Expand table

Option	Description
TCP/IP	Copies the memory of the virtual machine to the destination server over a TCP/IP connection.
Compression	Compresses the memory content of the virtual machine before copying it to the destination server over a TCP/IP connection. **Note:** This is the **default** setting.
SMB	Copies the memory of the virtual machine to the destination server over an SMB 3.0 connection. - SMB Direct is used when the network adapters on the source and destination servers have Remote Direct Memory Access (RDMA) capabilities enabled. - SMB Multichannel automatically detects and uses multiple connections when a proper SMB Multichannel configuration is identified. For more information, see Improve Performance of a File Server with SMB Direct.

Next steps

After you set up the hosts, you're ready to do a live migration. For instructions, see Use live migration without Failover Clustering to move a virtual machine.

Feedback

Was this page helpful? Yes No

Upgrade virtual machine version in Hyper-V on Windows or Windows Server

Article • 01/07/2022

> Applies to: Windows Server 2022, Windows 10, Windows Server 2019, Windows Server 2016

Make the latest Hyper-V features available on your virtual machines by upgrading the configuration version. Don't do this until:

- You upgrade your Hyper-V hosts to the latest version of Windows or Windows Server.
- You upgrade the cluster functional level.
- You're sure that you won't need to move the virtual machine back to a Hyper-V host that runs a previous version of Windows or Windows Server.

For more information, see Cluster Operating System Rolling Upgrade and Perform a rolling upgrade of a Hyper-V host cluster in VMM.

Step 1: Check the virtual machine configuration versions

1. On the Windows desktop, click the Start button and type any part of the name **Windows PowerShell**.
2. Right-click Windows PowerShell and select **Run as Administrator**.
3. Use the Get-VMcmdlet. Run the following command to get the versions of your virtual machines.

PowerShell

```
Get-VM * | Format-Table Name, Version
```

You can also see the configuration version in Hyper-V Manager by selecting the virtual machine and looking at the **Summary** tab.

Step 2: Upgrade the virtual machine configuration version

1. Shut down the virtual machine in Hyper-V Manager.
2. Select Action > Upgrade Configuration Version. If this option isn't available for the virtual machine, then it's already at the highest configuration version supported by the Hyper-V host.

To upgrade the virtual machine configuration version by using Windows PowerShell, use the Update-VMVersion cmdlet. Run the following command where vmname is the name of the virtual machine.

PowerShell

```
Update-VMVersion <vmname>
```

Supported virtual machine configuration versions

Using the PowerShell cmdlet Get-VMHostSupportedVersion you can see what virtual machine configuration versions your Hyper-V Host supports. When you create a virtual machine, it's created with the default configuration version. To see which virtual machine configuration versions your Hyper-V Host supports and what the default is, run the following command.

PowerShell

```
Get-VMHostSupportedVersion
```

If you need to create a virtual machine that you can move to a Hyper-V Host that runs an older version of Windows, use the New-VM cmdlet with the `-Version` parameter. For example, to create a virtual machine named "WindowsCV5" with configuration version 5.0, run the following command:

PowerShell

```
New-VM -Name "WindowsCV5" -Version 5.0
```

ⓘ Note

You can only import a virtual machine if the Hyper-V host supports that configuration version. Typically, this means that you can import a virtual machine to a Hyper-V host running a newer version of Hyper-V, but you cannot import a virtual machine created on a newer version of Hyper-V to an older version of Hyper-V.

If the VM's configuration version is not listed as supported for your Hyper-V host OS in the table below, you either have to upgrade the VM configuration version to a newer version or create a new VM of the same generation using the existing virtual hard disks before you can start the VM.

Supported VM configuration versions for long-term servicing hosts

The following table lists the VM configuration versions for hosts running a long-term servicing version of Windows.

Hyper-V host Windows version	**10.0**	**9.3**	**9.2**	**9.1**	**9.0**	**8.3**	**8.2**	**8.1**	**8.0**	**7.1**	**7.0**	**6.2**	**5.0**
Windows Server 2022	✓	✓	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows 10 Enterprise LTSC 2021	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows Server 2019	✕	✕	✕	✕	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 Enterprise LTSC 2019	✕	✕	✕	✕	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server 2016	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓	✓	✓	✓
Windows 10 Enterprise 2016 LTSB	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓	✓	✓	✓
Windows 10 Enterprise 2015 LTSB	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
Windows Server 2012 R2	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓
Windows 8.1	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓

Supported VM configuration versions for Semi-Annual Channel hosts

The following table lists the VM configuration versions for hosts running a Semi-Annual Channel version of Windows. To get more information on Semi-Annual Channel versions of Windows, visit the following pages for Windows Server and Windows.

Hyper-V host Windows version	**10.0**	**9.3**	**9.2**	**9.1**	**9.0**	**8.3**	**8.2**	**8.1**	**8.0**	**7.1**	**7.0**	**6.2**	**5.0**
Windows 11 (version 21H2)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows 10 November 2021 Update (version 21H2)	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows 10 May 2021 Update (version 21H1)	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows Server, version 20H2	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows 10 October 2020 Update (version 20H2)	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows Server, version 2004	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows 10 May 2020 Update (version 2004)	✕	✕	✓	✓	✓	✓	✓	✓	✓	✕	✕	✕	✕
Windows Server, version 1909	✕	✕	✕	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 November 2019 Update (version 1909)	✕	✕	✕	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server, version 1903	✕	✕	✕	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Hyper-V host Windows version	10.0	9.3	9.2	9.1	9.0	8.3	8.2	8.1	8.0	7.1	7.0	6.2	5.0
Windows 10 May 2019 Update (version 1903)	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server, version 1809	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 October 2018 Update (version 1809)	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server, version 1803	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 April 2018 Update (version 1803)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 Fall Creators Update (version 1709)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Windows 10 Creators Update (version 1703)	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Windows 10 Anniversary Update (version 1607)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

Why should I upgrade the virtual machine configuration version?

When you move or import a virtual machine to a computer that runs Hyper-V on Windows Server 2019, Windows Server 2016, or Windows 10, the virtual machine's configuration isn't automatically updated. This means that you can move the virtual machine back to a Hyper-V host that runs a previous version of Windows or Windows Server. But, this also means that you can't use some of the new virtual machine features until you manually update the configuration version.

> **ⓘ Important**

You can't downgrade a virtual machine configuration version after you've upgraded it.

The virtual machine configuration version represents the compatibility of the virtual machine's configuration, saved state, and snapshot files with the version of Hyper-V. When you update the configuration version, you change the file structure that is used to store the virtual machines configuration and the checkpoint files. You also update the configuration version to the latest version supported by that Hyper-V host. Upgraded virtual machines use a new configuration file format, which is designed to increase the efficiency of reading and writing virtual machine configuration data. The upgrade also reduces the potential for data corruption in the event of a storage failure.

The following table lists descriptions, file name extensions, and default locations for each type of file that's used for new or upgraded virtual machines.

Virtual machine file types	**Description**
Configuration	Virtual machine configuration information that is stored in binary file format. File name extension: .vmcx Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines
Runtime state	Virtual machine runtime state information that is stored in binary file format. File name extension: .vmrs and .vmgs Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines
Virtual hard disk	Stores virtual hard disks for the virtual machine. File name extension: .vhd or .vhdx Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Hard Disks
Automatic virtual hard disk	Differencing disk files used for virtual machine checkpoints. File name extension: .avhdx Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Hard Disks
Checkpoint	Checkpoints are stored in multiple checkpoint files. Each checkpoint creates a configuration file and runtime state file. File name extensions: .vmrs and .vmcx Default location: C:\ProgramData\Microsoft\Windows\Snapshots

What happens if I don't upgrade the virtual machine configuration version?

If you have virtual machines that you created with an earlier version of Hyper-V, some features that are available on the newer host OS may not work with those virtual machines until you update the configuration version.

As a general guidance, we recommend updating the configuration version once you have successfully upgraded the virtualization hosts to a newer version of Windows and feel confident that you do not need to roll back. When you are using the cluster OS rolling upgrade feature, this would typically be after updating the cluster functional level. This way, you will benefit from new features and internal changes and optimizations as well.

> **Note**
>
> Once the VM configuration version is updated, the VM won't be able to start on hosts that do not support the updated configuration version.

The following table shows the minimum virtual machine configuration version required to use some Hyper-V features.

Feature	Minimum VM configuration version
Allow additional processor features for Perfmon	9.0
Automatically expose simultaneous multithreading configuration for VMs running on hosts using the Core Scheduler	9.0
Hibernation support	9.0
Increase the default maximum number for virtual devices to 64 per device (e.g. networking and assigned devices)	8.3
Guest Virtualization-Based Security support (VBS)	8.0
Key storage drive	8.0
Large memory VMs	8.0
Nested Virtualization	8.0
Virtual processor count	8.0
XSAVE support	8.0
Virtual Machine Multi Queues (VMMQ)	7.1
Virtual Trusted Platform Module (vTPM)	7.0

Feature	Minimum VM configuration version
Hot Add/Remove Memory	6.2
PowerShell Direct	6.2
Production Checkpoints	6.2
Secure Boot for Linux VMs	6.2
Virtual Machine Grouping	6.2

For more information about these features, see What's new in Hyper-V on Windows Server.

GPU partitioning

Article • 09/26/2024

> Applies to: Windows Server 2025 (preview)

> **Important**
>
> GPU partitioning in Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

GPU partitioning allows you to share a physical GPU device with multiple virtual machines (VMs). With GPU partitioning or GPU virtualization, each VM gets a dedicated fraction of the GPU instead of the entire GPU.

The GPU partitioning feature uses the Single Root IO Virtualization (SR-IOV) interface, which provides a hardware-backed security boundary with predictable performance for each VM. Each VM can access only the GPU resources dedicated to them and the secure hardware partitioning prevents unauthorized access by other VMs.

Windows Server introduces live migration with GPU partitioning. There are specific requirements to use GPU partitioning live migration. Aside from recommended live migration best practices, your cluster hosts need to have Input/Output Memory Management Unit (IOMMU) DMA bit tracking capable processors. For example, processors supporting Intel VT-D or AMD-Vi. If you use Windows Server and live migration without IOMMU enabled processors, the VMs are automatically restarted where GPU resources are available.

GPU partitioning is designed for standalone servers. You can live migrate VMs between standalone nodes for planned downtime; however, for customers that require clustering for unplanned downtime, you must use Windows Server 2025 Datacenter.

When to use GPU partitioning

Some workloads, such as virtual desktop infrastructure (VDI), Artificial Intelligent (AI) and Machine Learning (ML) inferencing require GPU acceleration, GPU partitioning can help reduce your total cost of ownership for your overall infrastructure.

For example:

- VDI applications: Distributed edge customers run basic productivity apps, such as Microsoft Office and graphics-heavy visualization workloads in their VDI environments, which require GPU acceleration. For such workloads, you can achieve the required GPU acceleration via DDA or GPU partitioning. With GPU partitioning, you can create multiple partitions and assign each partition to VM hosting a VDI environment. GPU partitioning helps you achieve the desired density and scale the number of supported users by an order of magnitude.

- Inference with ML: Customers in retail stores and manufacturing plants can run inference at the edge, which requires GPU support for their servers. Using GPU on your servers, you can run ML models to get quick results that can be acted on before the data is sent to the cloud. The full data set can optionally be transferred to continue to retrain and improve your ML models. Along with DDA where you assign an entire physical GPU to a VM, GPU partitioning allows you to run multiple inferencing applications in parallel on the same GPU, but in separate physical partitions, thereby utilizing the GPU to the maximum.

Supported guest operating systems

GPU partitioning on Windows Server 2025 and later supports these guest operating systems:

- Windows 10 or later
- Windows 10 Enterprise multi-session or later
- Windows Server 2019 or later
- Linux Ubuntu 18.04 LTS, Linux Ubuntu 20.04 LTS, Linux Ubuntu 22.04 LTS

Supported GPUs

The following GPUs support GPU partitioning:

- NVIDIA A2
- NVIDIA A10
- NVIDIA A16
- NVIDIA A40
- NVIDIA L2
- NVIDIA L4
- NVIDIA L40
- NVIDIA L40S

> **Note**

The NVIDIA driver doesn't currently support GPU partitioning for live migration.

We recommend that you work with your Original Equipment Manufacturer (OEM) partners and GPU Independent Hardware Vendors (IHVs) to plan, order, and set up the systems for your desired workloads with the appropriate configurations and necessary software. However, we support more GPUs if you want to use GPU acceleration via Discrete Device Assignment (DDA). Reach out to your OEM partners and IHVs to get a list of GPUs that support DDA. For more information about using GPU acceleration via DDA, see Discrete Device Assignment (DDA).

For best performance, we recommend that you create a homogeneous configuration for GPUs across all the servers in your cluster. A homogeneous configuration consists of installing the same make and model of the GPU, and configuring the same partition count in the GPUs across all the servers in the cluster. For example, in a cluster of two servers with one or more GPUs installed, all the GPUs must have the same make, model, and size. The partition count on each GPU must also match.

Limitations

Consider the following limitations when using the GPU partitioning feature:

- GPU partitioning is unsupported if your configuration isn't homogeneous. Here are some examples of unsupported configurations:
  - Mixing GPUs from different vendors in the same cluster.
  - Using different GPU models from different product families from the same vendor in the same cluster.
- You can't assign a physical GPU as both Discrete Device Assignment (DDA) or partitionable GPU. You can either assign it as DDA or as partitionable GPU, but not both.
- You can assign only a single GPU partition to a VM.
- Partitions are autoassigned to the VMs. You can't choose a specific partition for a specific VM.
- You can partition your GPU using Windows Admin Center or using PowerShell. We recommend that you use Windows Admin Center to configure and assign GPU partitions. Windows Admin Center automatically validates for a homogeneous configuration of the GPUs across all the servers in your cluster. It provides appropriate warnings and errors to take any corrective action needed.

- If using PowerShell to provision GPU partitioning, you must perform the provisioning steps on each server in the cluster. You must manually ensure that the homogeneous configuration is maintained for GPUs across all the servers in your cluster.
- When live migrating a virtual machine with a GPU partition assigned, Hyper-V live migration automatically falls back to using TCP/IP with compression. Migrating a virtual machine has the potential effect of increasing the CPU utilization of a host. In addition, live migrations could take longer than with virtual machines without GPU partitions attached.

Related content

For more information on using GPUs with your VMs and GPU partitioning, see:

- Partition and assign GPUs to a virtual machine
- Use GPUs with clustered VMs
- Accelerate your edge workloads with affordable NVIDIA GPU-powered Azure Stack HCI solutions blog

---

Feedback

Partition and assign GPUs to a virtual machine

Article • 09/25/2024

> Applies to: Windows Server 2025 (preview)

> **ⓘ Important**
>
> GPU partitioning in Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

This article describes how to configure graphics processing unit (GPU) partitions and assign a partition to a virtual machine (VM). It provides instructions on how to configure GPU partition count, assign GPU partitions, and unassign GPU partitions via Windows Admin Center and PowerShell.

To provision the GPU partitioning feature, you need to complete the following steps:

- Complete all the prerequisites.
- Verify GPU driver installation.
- Configure partition count.
- Assign GPU partition to a VM.
- If necessary, unassign a GPU partition from a VM.

Prerequisites

There are several requirements and things to consider before you begin to use the GPU partitioning feature:

Prerequisites for the host server

- You must have Windows Server 2025 installed on the host server. If clustering is required when live migrating a virtual machine host, Windows Server 2025 Datacenter must be installed.

- The Hyper-V role installed and configured on your server. See Install the Hyper-V role on Windows Server to found out how to get stared.

- Install the physical GPU device of the same make, model, and size on every server of the cluster. Refer to your OEM-provided documentation when installing the GPU device on your physical servers in the cluster.
- Install the GPU drivers on every server of the cluster by following instructions from your GPU IHVs. For NVIDIA GPU drivers, see the NVIDIA vGPU documentation.
- Ensure that the virtualization support and SR-IOV are enabled in the BIOS of each server in the cluster. Reach out to your system vendor if you're unable to identify the correct setting in your BIOS.
- Cluster hosts need to have Input/Output Memory Management Unit (IOMMU) DMA bit tracking capable processors. For example, processors supporting Intel VT-D or AMD-Vi.

> **Note**
>
> When live migrating a virtual machine with a GPU partition assigned, Hyper-V live migration will automatically fall back to using TCP/IP with compression. This has the potential effect of increasing the CPU utilization of a host. In addition, live migrations could take longer than with virtual machines without GPU partitions attached.

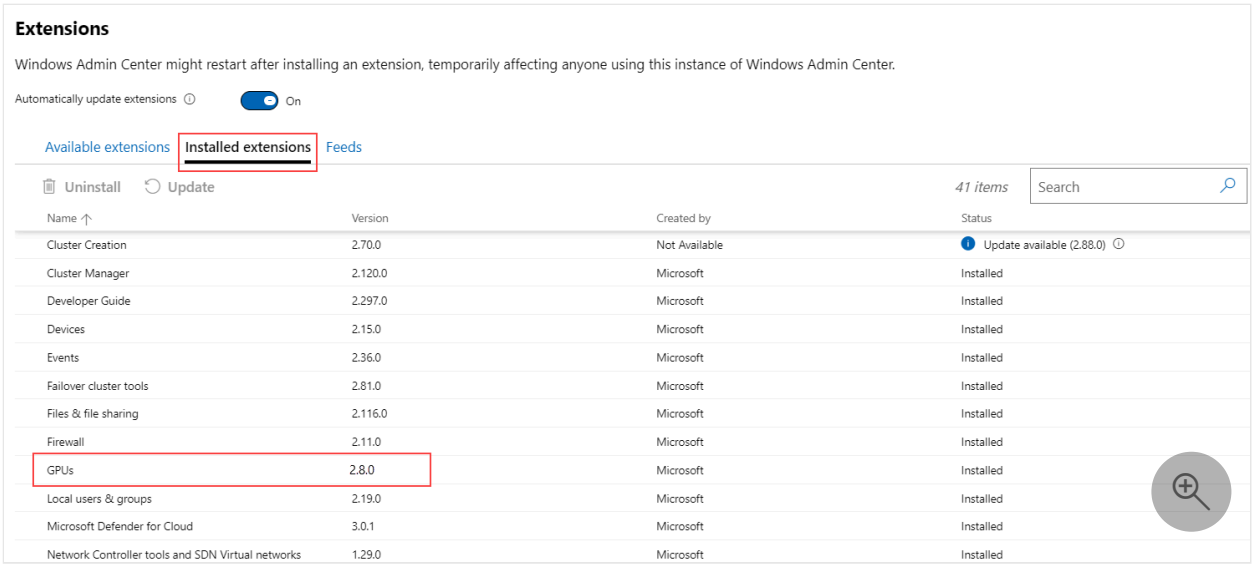
Prerequisites for the VMs

- Deploy a VM using a guest operating system from the Supported guest operating systems list.
- Install the GPU drivers on the VM by following instructions from your GPU IHVs. For NVIDIA GPU drivers, see the NVIDIA vGPU documentation.

Prerequisites for Windows Admin Center

If you're using Windows Admin Center to provision GPU partitioning, you must install the latest version of Windows Admin Center with the **GPUs** extension, version 2.8.0 or later. For instructions on how to install the **GPUs** extensions in Windows Admin Center, see Installing an extension.

After you install the extension, it appears under the **Installed extensions** tab as shown in the following screenshot. Make sure the version of the **GPUs** extension is 2.8.0 or later.



Prerequisites for PowerShell

If you're using PowerShell to provision GPU partitioning, you must run all PowerShell commands as the Administrator user.

For detailed information on how to use PowerShell commands for GPU partitioning, see the Add-VMGpuPartitionAdapter, Get-VMGpuPartitionAdapter, and Remove-VMGpuPartitionAdapter reference documentation.

Verify GPU driver installation

After you complete all the prerequisites, you must verify if the GPU driver is installed and partitionable.

Windows Admin Center

Follow these steps to verify if the GPU driver is installed and partitionable using Windows Admin Center:

1. Launch Windows Admin Center and make sure the **GPUs** extension is already installed.

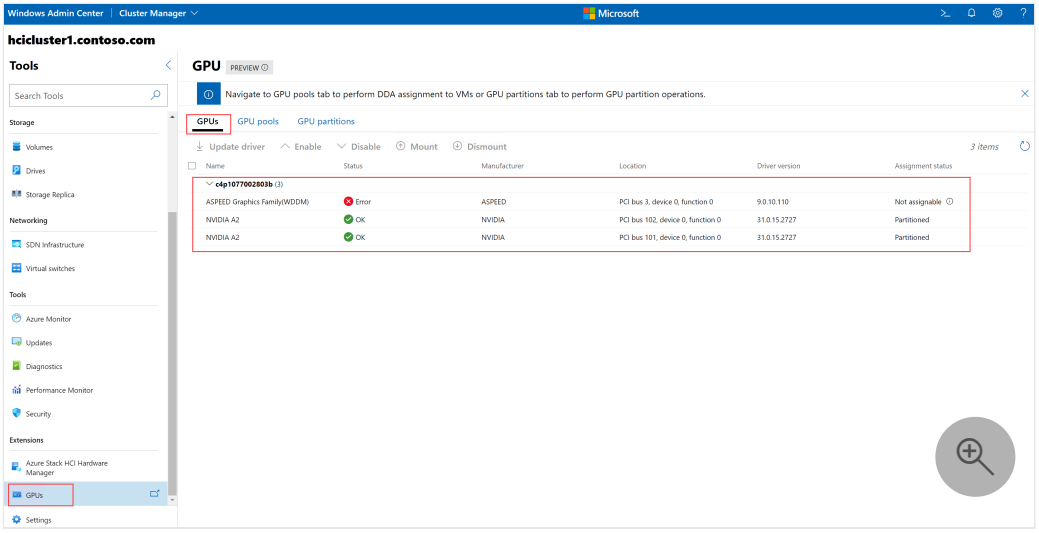
2. Select **Cluster Manager** from the top dropdown menu and connect to your cluster.

3. From the **Settings** menu, select **Extensions** > **GPUs**.

   The **GPUs** tab on the **GPU** page displays inventory of all the servers and the physical GPUs that are installed on each server.

4. Check the **Assigned status** column for each GPU for all the servers. The **Assigned status** column can have one of these statuses:

    - **Ready for DDA assignment**. Indicates that the GPU is available for DDA assignment. You can't use it for GPU partitioning.
    - **Partitioned**. Indicates that the GPU is partitionable.
    - **Paravirtualization**. Indicates that the GPU has the partitioned driver capability installed but SR-IOV on the server isn't enabled.
    - **Not assignable**. Indicates that the GPU isn't assignable because it's an older PCI-style device or switch port.

5. Proceed further in the GPU partitioning workflow only if the **Assigned status** column shows **Partitioned** for the GPUs in all the servers in your cluster.

Configure GPU partition count

Each partitionable GPU comes with a set of valid partition counts that's predefined by its OEM. You can't define the number of partitions a GPU can have. However, you can configure the partition count to any of the valid count from within the supported set.

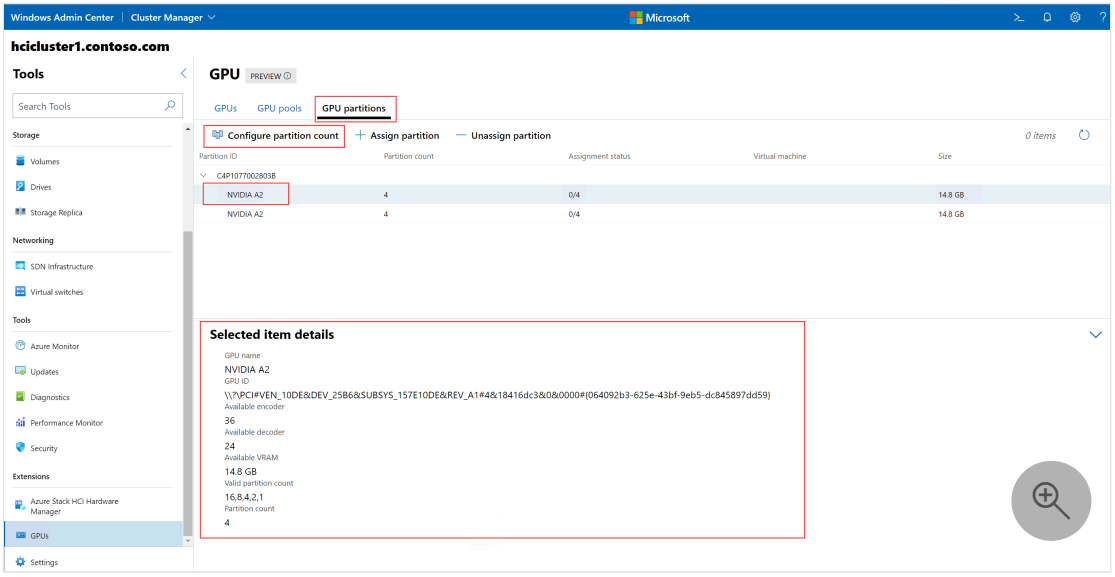
Windows Admin Center

Follow these steps to configure partition count via Windows Admin Center:

1. Select the **GPU partitions** tab to configure partition counts. You can also assign partition to VMs and unassign partitions from VMs using this tab.

> **Note**
>
> If there are no partitionable GPUs available in your cluster or the correct GPU partitioning driver isn't installed, the GPU partitions tab displays the following message:
>
> *No partitionable GPUs have been found. Please check that you have a GPU with the correct GPU-P driver to proceed.*

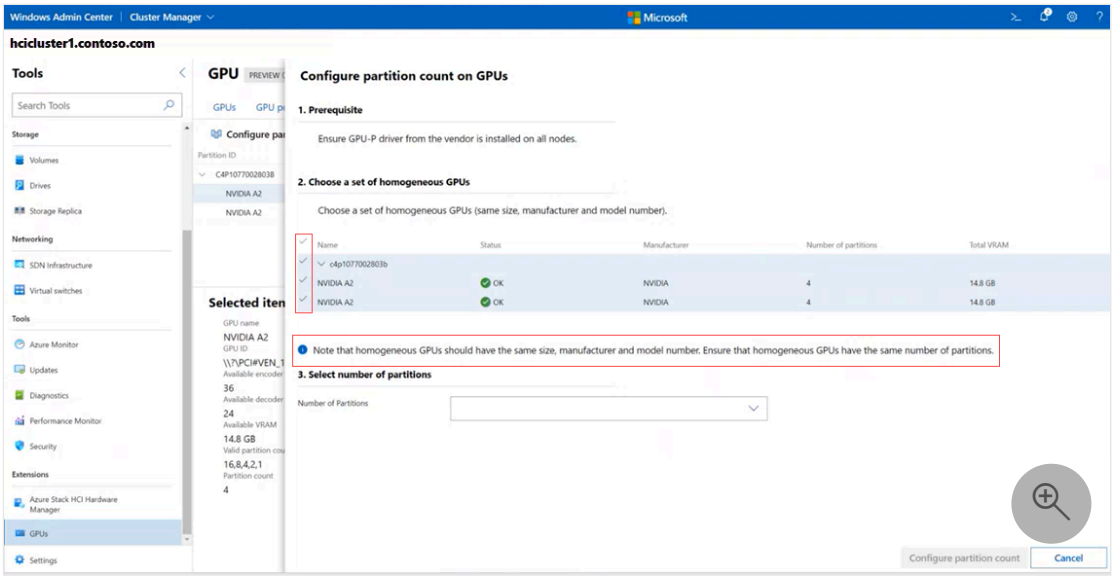
2. Select a GPU or a GPU partition to display its details in the bottom section of the page, under **Selected item details**. For example, if you select a GPU, it displays the GPU name, GPU ID, available encoder and decoder, available VRAM, valid partition count, and the current partition count. If you select a GPU partition, it displays the partition ID, VM ID, instance path, partition VRAM, partition encode, and partition decode.



3. Select **Configure partition count**.

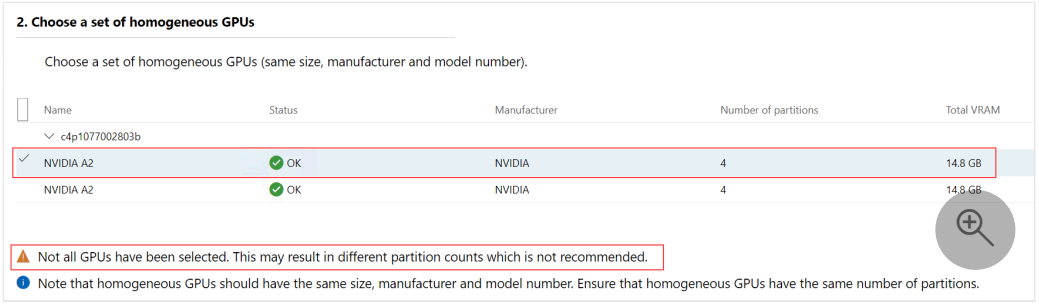
   The **Configure partition count on GPUs** page is displayed. For each server, it displays the GPU devices installed on them.

4. Select a set of homogeneous GPUs. A set of homogeneous GPUs is the one that has GPUs of the same size, manufacturer, model number, and number of partitions. By default, Windows Admin Center automatically selects a set of homogenous GPUs if it detects one, as shown in the following screenshot:

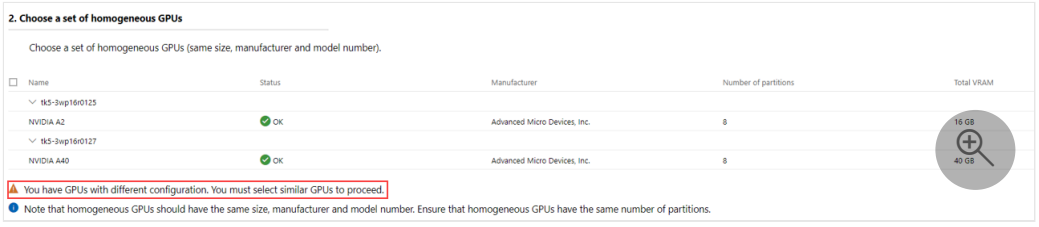


You might see a warning or an error depending on what selections you make:

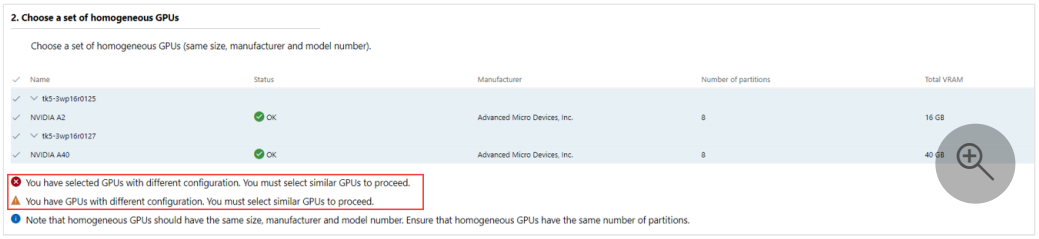
- **Warning.** If you deselect one or more GPUs from the homogeneous set of GPUs, Windows Admin Center gives you a warning, but doesn't stop you from proceeding further. Warning text indicates that you're not selecting all the GPUs and it might result in different partition count, which isn't recommended.



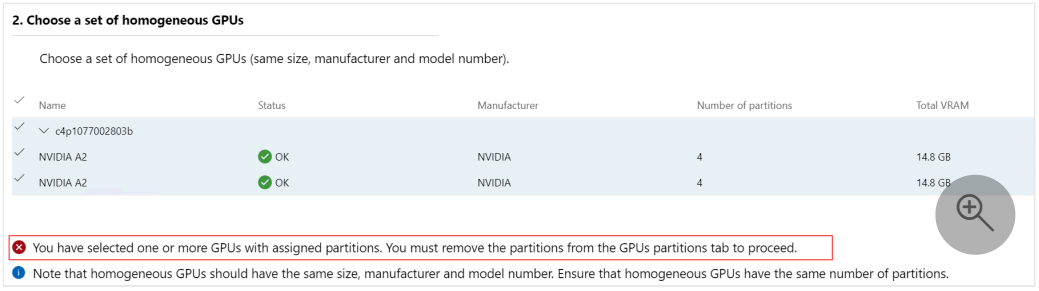
- **Warning.** If not all the GPUs across all the servers have the same configuration, Windows Admin Center gives a warning. You must manually select the GPUs with the same configuration to proceed further.



- **Error.** If you select GPUs with different configurations, Windows Admin Center gives you an error, and doesn't let you proceed.

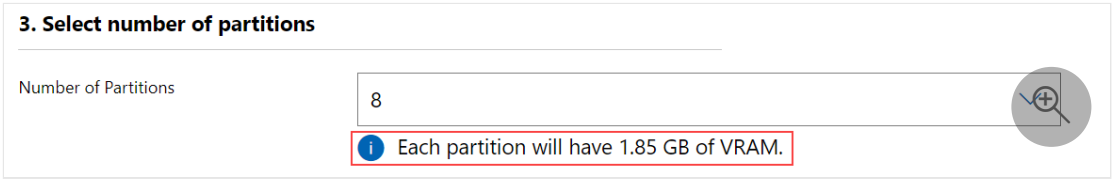


- **Error.** If you select a GPU partition that is already assigned to a VM, Windows Admin Center gives you an error, and doesn't let you proceed. You must first unassign the partition from the VM before proceeding further. See Unassign a partition from a VM.



5. After you select a homogeneous set of GPUs, select the partition count from the **Number of Partitions** dropdown list. This list automatically populates the partition counts configured by your GPU manufacturer. The counts displayed in the list can vary depending on the type of GPU you selected.

   As soon as you select a different partition count, a tooltip appears below the dropdown list, which dynamically displays the size of VRAM that each partition gets. For example, if the total VRAM is 16 GB for 16 partitions in the GPU, changing the partition count from 16 to 8 assigns each partition with 1.85 GB of VRAM.



6. Select **Configure partition count**.

   After the partition count is configured, Windows Admin Center notifies you that the partition count is successfully configured and displays the **GPU partitions** tab again. You can see the new partition count for the GPU partition under the **Partition count** column.

Assign GPU partition to a VM

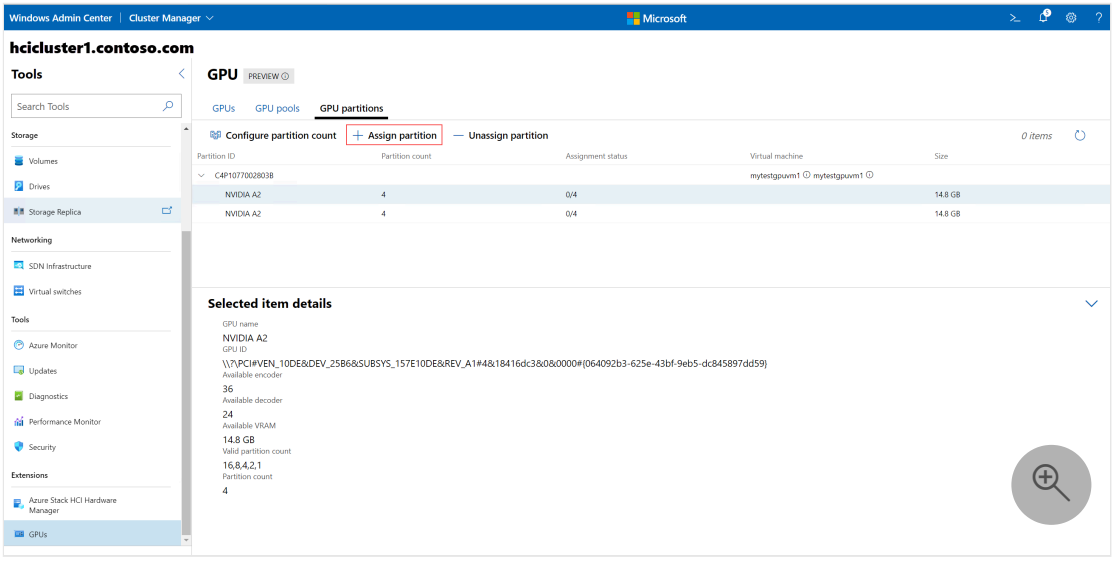
Save your workloads before assigning partition to the VM.

> **Note**
>
> Currently, you can assign only a single GPU partition to a VM. Both the VM and the GPU partition need to be on the same host machine. We recommend that you plan ahead and determine the GPU partition size based on your workload performance requirements.

Windows Admin Center

You must save your workloads before assigning partitions. If your VM is currently turned on or running, Windows Admin Center automatically turns it off, assigns the partition, and then automatically turns it on.

1. On the **GPU partitions** tab, select **+ Assign partition**.



   The **Assign GPU partition to VM** page is displayed.

2. From **Choose the server** list, select the server where the VM resides. This list displays all the servers in your cluster.

3. Search for and select the VM to assign the GPU partition to. The list automatically populates the VMs that reside on the server that you selected in step 2.

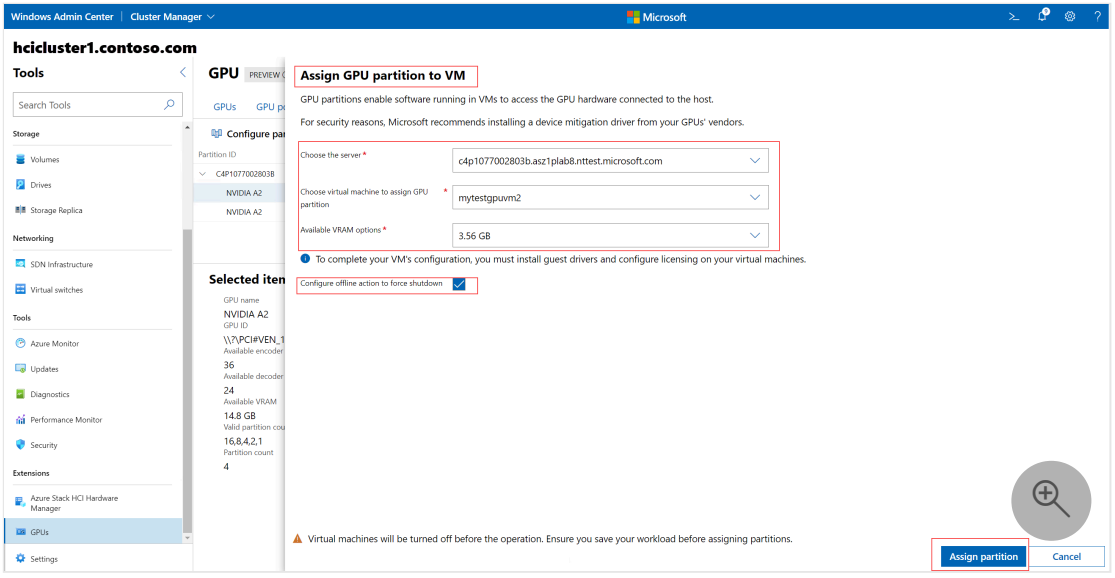
   - If a GPU partition is already assigned to a VM, that VM appears as grayed out.

   - Select all the VMs at once by selecting the **Select All** checkbox.

4. Select the available VRAM options. The value in this field must match the size of the partition count that you configured.

5. (Optional, but recommended) Select the **Configure offline action for force shutdown** checkbox if you want your VM to be highly available and fail over if its host server goes down.

6. Select **Assign partition**. This assigns partition of the selected VRAM size to the selected VM on the selected host server.



After the partition is assigned, Windows Admin Center notifies you that the partition is successfully assigned and displays the **GPU partitions** tab again. On the **GPU partitions** tab, the VM appears on the GPU partition row under the server it's installed on.

Unassign a partition from a VM

You can unassign a GPU partition from the VM if you no longer need it to run your workloads. Unassigning the partition frees up the GPU partition resource, which you can reassign to another VM later.

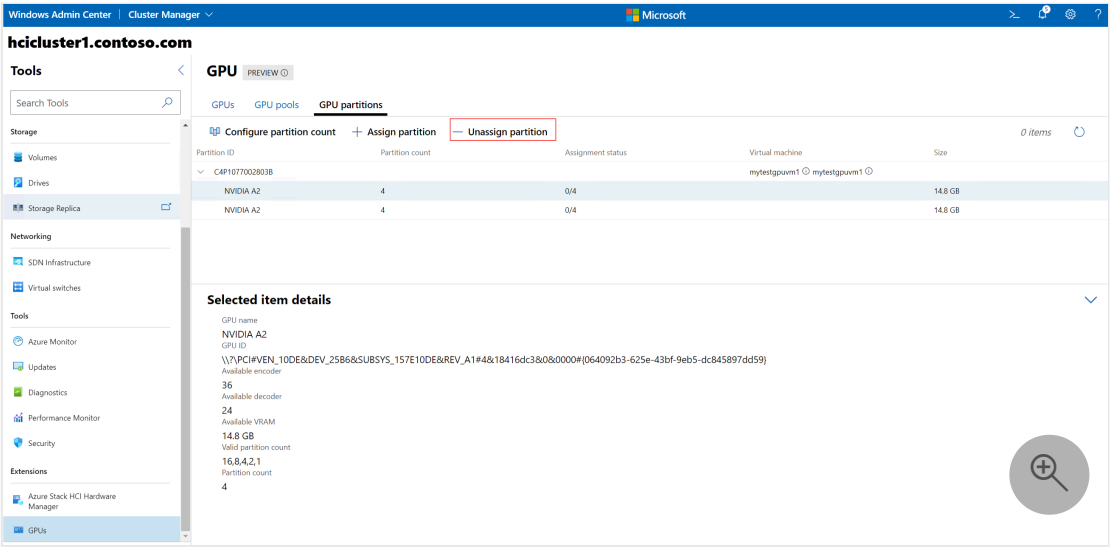
You must save your workloads before unassigning partitions.

Windows Admin Center

If your VM is currently turned on or running, Windows Admin Center automatically turns it off first, unassigns the partition, then automatically turns it on.

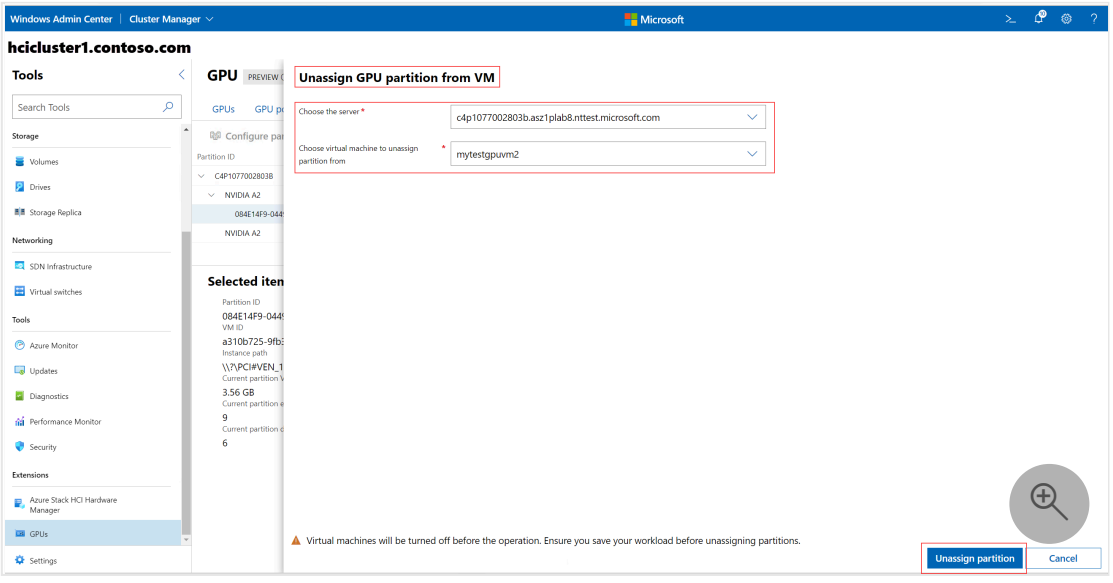
Follow these steps to unassign a partition from a VM:

1. On the **GPU partitions** tab, select the GPU partition that you want to unassign.
2. Select - **Unassign partition**.



The **Unassign GPU partition from VM** page is displayed.

3. From **Choose the server** list, select the server that has the GPU partition that you want to unassign.
4. From **Choose virtual machine to unassign partition from** list, search, or select the VM to unassign the partition from.
5. Select **Unassign partition**.



After the partition is unassigned, Windows Admin Center notifies you that the partition is successfully unassigned and displays the **GPU partitions** tab again. On the **GPU partitions** tab, the VM from which the partition is unassigned no longer shows on the GPU partition row.

Feedback

Was this page helpful? Yes No

Deploy graphics devices by using Discrete Device Assignment

Article • 05/16/2024

Learn how to use Discrete Device Assignment (DDA) to pass an entire PCIe device into a virtual machine (VM) with PowerShell. Doing so allows high performance access to devices like NVMe storage or graphics cards from within a VM while being able to apply the device's native drivers. For more information on devices that work and possible security implications, see Plan for Deploying Devices using Discrete Device Assignment.

This article takes you through the steps to use a device with DDA:

1. Configure the VM for DDA
2. Dismount the device from the host partition
3. Assign the device to the guest VM

Prerequisites

Before you can use DDA to deploy graphics devices, you need to have the following.

- A Hyper-V host running Windows Server 2016 or later.
- A VM running one of the following operating systems:
  - Windows Server 2016 or later.
  - Windows 10 or later.
- Review Plan for Deploying Devices using Discrete Device Assignment to ensure your hardware is compatible with DDA.
  - Run the SurveyDDA.ps1. PowerShell script to identify if the server is configured correctly. The script also displays which devices can be passed through by using Discrete Device Assignment.
- Administrative rights to the Hyper-V host.
- (Optional) Though not required, if Single Root I/O Virtualization (SR-IOV) isn't enabled or supported, you might encounter issues when you use DDA to deploy graphics devices.

Configure the VM for DDA

The first step in the solution is to address DDA restrictions to the VMs.

1. Sign in to the Hyper-V host as an administrator.
2. Open an elevated PowerShell prompt.
3. Configure the `Automatic Stop Action` of a VM to enable **TurnOff** with the following PowerShell cmdlet:

```PowerShell
Set-VM -Name VMName -AutomaticStopAction TurnOff
```

VM preparation for graphics devices

Some hardware performs better if the VM in configured in a certain way. For details on whether you need the following configurations for your hardware, reach out to the hardware vendor. For more information, see Plan for Deploying Devices using Discrete Device Assignment and on this blog post.

1. Enable Write-Combining on the CPU by using the following cmdlet:

```PowerShell
Set-VM -GuestControlledCacheTypes $true -VMName VMName
```

2. Configure the 32-bit memory mapped IO (MMIO) space by using the following cmdlet:

```PowerShell
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName VMName
```

3. Configure greater than 32-bit MMIO space by using the following cmdlet:

```PowerShell
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName VMName
```

> **Tip**
>
> The MMIO space values shown are reasonable values to set for experimenting with a single GPU. If after starting the VM the device is reporting an error

relating to not enough resources, you'll likely need to modify these values. For more information about how to precisely calculate MMIO requirements, see **Plan for Deploying Devices using Discrete Device Assignment**.

Dismount the device from the host partition

Follow the instructions in this section to dismount the device from the host partition.

Install the partitioning driver (optional)

DDA gives hardware vendors the ability to provide a security mitigation driver with their devices. This driver isn't the same as the device driver installed in the guest VM. It's up to the hardware vendor's discretion to provide this driver. But if they do provide a driver, install it before dismounting the device from the host partition. Reach out to the hardware vendor to see if they have a mitigation driver.

If no partitioning driver is provided, during dismount you must use the `-Force` option to bypass the security warning. For more information about the security implications, see Plan for Deploying Devices using Discrete Device Assignment.

Locate the device's location path

The PCI location path is required to dismount and mount the device from the host. An example location path looks like this: `PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)`. For more information about locating the location path, see Plan for Deploying Devices using Discrete Device Assignment.

Disable the device

Use Device Manager or PowerShell to ensure the device is **Disabled**.

Dismount the device

Depending on whether the vendor provided a mitigation driver, you must either use the `-Force` option or not, as shown here:

- If a mitigation driver was installed, use the following cmdlet:

```
PowerShell
```

```
Dismount-VMHostAssignableDevice -LocationPath $locationPath
```

- If a mitigation driver wasn't installed, use the following cmdlet:

```PowerShell
Dismount-VMHostAssignableDevice -Force -LocationPath $locationPath
```

Assign the device to the guest VM

The final step is to tell Hyper-V that a VM should have access to the device. Specify the location path and the name of the VM.

```PowerShell
Add-VMAssignableDevice -LocationPath $locationPath -VMName VMName
```

Complete tasks on the VM

After a device is successfully mounted in a VM, you're now able to start that VM and interact with the device as though you were running on a bare metal system. You're now able to install the hardware vendor's drivers in the VM, and applications are able to see the hardware. You can verify it by opening Device Manager in the guest VM and seeing that the hardware is available.

Remove a device and return it to the host

If you want to return the device back to its original state, you must stop the VM and issue this command:

```PowerShell
# Remove the device from the VM
Remove-VMAssignableDevice -LocationPath $locationPath -VMName VMName

# Mount the device back in the host
Mount-VMHostAssignableDevice -LocationPath $locationPath
```

You can then re-enable the device in Device Manager, and the host operating system is able to interact with the device again.

Example - Mount a GPU to a VM

This example uses PowerShell to configure a VM named **ddatest1** to take the first GPU available by the manufacturer NVIDIA and assign it into the VM.

PowerShell

```
# Configure the VM for a Discrete Device Assignment
$vm = "ddatest1"
# Set automatic stop action to TurnOff
Set-VM -Name $vm -AutomaticStopAction TurnOff
# Enable Write-Combining on the CPU
Set-VM -GuestControlledCacheTypes $true -VMName $vm
# Configure 32 bit MMIO space
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName $vm
# Configure Greater than 32 bit MMIO space
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName $vm

# Find the Location Path and disable the Device
# Enumerate all PNP Devices on the system
$pnpdevs = Get-PnpDevice -presentOnly
# Select only those devices that are Display devices manufactured by NVIDIA
$gpudevs = $pnpdevs | Where-Object {$_.Class -like "Display" -and
$_.Manufacturer -like "NVIDIA"}
# Select the location path of the first device that's available to be
dismounted by the host.
$locationPath = ($gpudevs | Get-PnpDeviceProperty
DEVPKEY_Device_LocationPaths).data[0]
# Disable the PNP Device
Disable-PnpDevice -InstanceId $gpudevs[0].InstanceId

# Dismount the Device from the Host
Dismount-VMHostAssignableDevice -Force -LocationPath $locationPath

# Assign the device to the guest VM.
Add-VMAssignableDevice -LocationPath $locationPath -VMName $vm
```

Troubleshoot issues with mounting a GPU

If you pass a GPU into a VM but Remote Desktop Services or an application isn't recognizing the GPU, check for the following common issues.

- Make sure you install the most recent version of the GPU vendor's supported driver, and that the driver isn't reporting errors. You can do so by checking the device state in Device Manager.
- Make sure your device has enough MMIO space allocated within the VM. For more information, see MMIO Space.

- Make sure you use a GPU that the vendor supports being used in this configuration. For example, some vendors prevent their consumer cards from working when passed through to a VM.

- Make sure the application supports running inside a VM, and that the application supports both the GPU and its associated drivers. Some applications have allowlists of GPUs and environments.

- If you use the Remote Desktop Session Host role or Windows Multipoint Services on the guest, you must make sure that a specific Group Policy entry is set to allow use of the default GPU. Use a Group Policy Object applied to the guest (or the Local Group Policy Editor on the guest). Navigate to the following Group Policy item:

  Computer Configuration\Administrator Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Remote Session Environment\Use hardware graphics adapters for all Remote Desktop Services sessions.

  Set the Group Policy value to **Enabled**, then reboot the VM after you apply the policy.

Deploy NVMe Storage Devices using Discrete Device Assignment

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Microsoft Hyper-V Server 2016, Windows Server 2016

Starting with Windows Server 2016, you can use Discrete Device Assignment, or DDA, to pass an entire PCIe Device into a VM. This will allow high performance access to devices like NVMe storage or Graphics Cards from within a VM while being able to leverage the devices native drivers. Please visit the Plan for Deploying Devices using Discrete Device Assignment for more details on which devices work, what are the possible security implications, etc. There are three steps to using a device with DDA:

- Configure the VM for DDA
- Dismount the Device from the Host Partition
- Assigning the Device to the Guest VM

All command can be executed on the Host on a Windows PowerShell console as an Administrator.

Configure the VM for DDA

Discrete Device Assignment imposes some restrictions to the VMs and the following step needs to be taken.

1. Configure the “Automatic Stop Action” of a VM to TurnOff by executing

```
Set-VM -Name VMName -AutomaticStopAction TurnOff
```

Dismount the Device from the Host Partition

Locating the Device's Location Path

The PCI Location path is required to dismount and mount the device from the Host. An example location path looks like the following:

`"PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)"`. More details on located the Location Path can be found here: Plan for Deploying Devices using Discrete Device Assignment.

Disable the Device

Using Device Manager or PowerShell, ensure the device is “disabled.”

Dismount the Device

```
Dismount-VMHostAssignableDevice -LocationPath $locationPath
```

Assigning the Device to the Guest VM

The final step is to tell Hyper-V that a VM should have access to the device. In addition to the location path found above, you'll need to know the name of the vm.

```
Add-VMAssignableDevice -LocationPath $locationPath -VMName VMName
```

What's Next

After a device is successfully mounted in a VM, you're now able to start that VM and interact with the device as you normally would if you were running on a bare metal system. You can verify this by opening device manager in the Guest VM and seeing that the hardware now shows up.

Removing a Device and Returning it to the Host

If you want to return he device back to its original state, you will need to stop the VM and issue the following:

```
```

```
#Remove the device from the VM
Remove-VMAssignableDevice -LocationPath $locationPath -VMName VMName
#Mount the device back in the host
Mount-VMHostAssignableDevice -LocationPath $locationPath
```

You can then re-enable the device in device manager and the host operating system will be able to interact with the device again.

Run Hyper-V in a Virtual Machine with Nested Virtualization

Article • 04/19/2024

Nested Virtualization is a feature that allows you to run Hyper-V inside of a Hyper-V virtual machine (VM). Nested Virtualization is helpful for running a Visual Studio phone emulator in a virtual machine, or testing configurations that ordinarily require several hosts.

To learn more about Nested Virtualization and supported scenarios, see What is Nested Virtualization for Hyper-V?.

Prerequisites

Intel processor with VT-x and EPT technology

- The Hyper-V host must be either Windows Server 2016 or later, or Windows 10 or later.
- VM configuration version 8.0 or higher.

AMD EPYC / Ryzen processor or later

- The Hyper-V host must be either Windows Server 2022 or later, or Windows 11 or later.
- VM configuration version 9.3 or higher.

> **Note**
>
> The guest can be any Windows supported guest operating system. Newer Windows operating systems may support enlightenments that improve performance. To enable Nested Virtualization in an Azure VM, make sure to set Security Type as "**Standard**".

Configure Nested Virtualization

1. Create a virtual machine. See the prerequisites for the required OS and VM versions.

2. While the virtual machine is in the OFF state, run the following command on the physical Hyper-V host to enable nested virtualization for the virtual machine.

```
PowerShell

Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions $true
```

3. Start the virtual machine.

4. Install Hyper-V within the virtual machine, just like you would for a physical server. For more information on installing Hyper-V, see, Install Hyper-V.

> **Note**
>
> When using Windows Server 2019 as the first level VM, the number of vCPUs should be 225 or less.

Disable Nested Virtualization

You can disable nested virtualization for a stopped virtual machine using the following PowerShell command:

```
PowerShell

Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions $false
```

Networking options

There are two options for networking with nested virtual machines:

1. MAC address spoofing
2. NAT networking

MAC address spoofing

In order for network packets to be routed through two virtual switches, MAC address spoofing must be enabled on the first (L1) level of virtual switch. To enable MAC address spoofing, run the following PowerShell command.

```
PowerShell
```

```
Get-VMNetworkAdapter -VMName <VMName> | Set-VMNetworkAdapter -
MacAddressSpoofing On
```

Network Address Translation (NAT)

The second option relies on network address translation (NAT). This approach is best suited for cases where MAC address spoofing isn't possible, like in a public cloud environment.

First, a virtual NAT switch must be created in the host virtual machine (the "middle" VM). The following example creates a new internal switch named `VmNAT` and creates a NAT object for all IP addresses in the `192.168.100.0/24` subnet.

PowerShell

```
New-VMSwitch -Name VmNAT -SwitchType Internal
New-NetNat –Name LocalNAT –InternalIPInterfaceAddressPrefix
“192.168.100.0/24”
```

Next, assign an IP address to the net adapter:

PowerShell

```
Get-NetAdapter "vEthernet (VmNat)" | New-NetIPAddress -IPAddress
192.168.100.1 -AddressFamily IPv4 -PrefixLength 24
```

Each nested virtual machine must have an IP address and gateway assigned to it. The gateway IP must point to the NAT adapter from the previous step. You may also want to assign a DNS server:

PowerShell

```
Get-NetAdapter "vEthernet (VmNat)" | New-NetIPAddress -IPAddress
192.168.100.2 -DefaultGateway 192.168.100.1 -AddressFamily IPv4 -
PrefixLength 24
Netsh interface ip add dnsserver “vEthernet (VmNat)” address=<my DNS server>
```

Next steps

- Remotely manage Hyper-V hosts with Hyper-V Manager

Feedback

Was this page helpful? Yes No

Physical storage architectures for Hyper-V

Article • 06/24/2024

Windows Server Hyper-V is a mature hypervisor platform that supports a wide range of physical storage architectures, from standalone systems with no resiliency to clustered systems with complex resiliency requirements. This article is an overview of some of the most widely used configuration options.

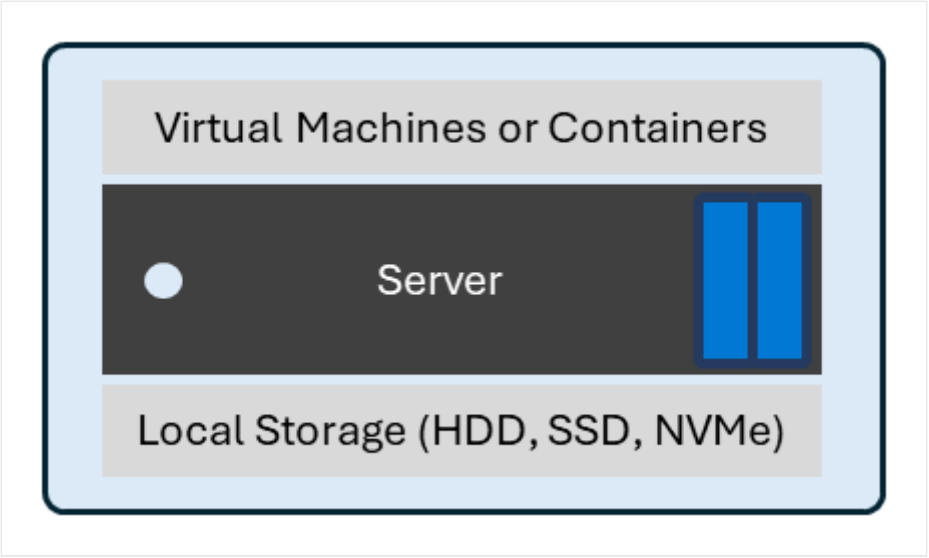
> **Note**
>
> This article doesn't describe every possible storage architecture configuration. It also doesn't describe less common types of configuration or edge cases.

Standalone Hyper-V with local disks

Hyper-V supports several nonclustered, or standalone, configurations where you install Hyper-V on a standalone server and use local storage.

When using this configuration:

- Standalone Hyper-V with local disks doesn't support automatic failover of virtual machine (VM) workloads due to the lack of shared storage and the compute (VM processing and memory) systems being nonclustered. For example, if the physical host loses power, the VMs running on it restart when you power the physical host back on. You can also live migrate VMs to other nodes or clusters using shared-nothing migration.
- You can optionally configure local disk resiliency for local drives using hardware or software RAID solutions. If you need more information, we recommend you contact your storage vendor.



Disaggregated Hyper-V with SAN or NAS Storage

In this configuration, you install Hyper-V is a cluster with VMs that access their storage over the network like hyperconverged storage, but Hyper-V uses SAN or NAS storage from a storage vendor instead.

When using this configuration:

- VMs are highly available to any node in the same compute cluster.
- SAN and NAS systems provide their own availability guarantees.

In this model, the compute and storage scales independently from one another. For example, if you need more processing or memory resources to host your VMs (CPU or RAM), you can add more compute nodes or clusters without adding more storage. Each compute cluster can contain between 1 and 64 nodes.

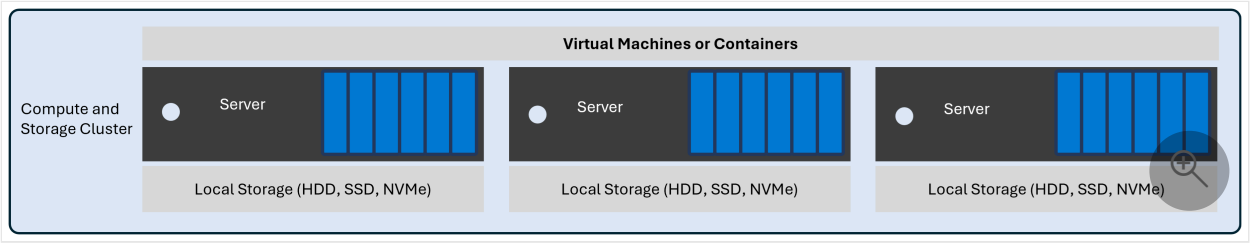
You can scale storage independently from the compute cluster. Storage cluster scaling is vendor-specific. Contact your vendors to understand how the storage solution they offer can scale.

Hyperconverged Hyper-V and Storage Spaces Direct

In this configuration, you install Hyper-V and Storage Spaces Direct on each node in a cluster. Each node contains local disks with data replicated to other nodes in the same cluster.

When using this configuration:

- You can move VMs to any node in the cluster using Live Migrate or automatically restart after a failure (failover).
- Data gets replicated to other nodes in the cluster to increase storage resilience. Storage Spaces Direct supports several storage resiliency models. For more information, see Fault tolerance and storage efficiency on Azure Stack HCI and Windows Server clusters.
- In this model, each physical host contains storage and compute resources. As a result, those physical resources scale symmetrically. Every new host automatically adds both compute and storage resources. Each cluster can contain between 1 and 16 nodes.



Disaggregated Hyper-V with hyperconverged storage

In this configuration, you install Hyper-V and Storage Spaces Direct in separate clusters and access the VMs' configuration and storage over the network.

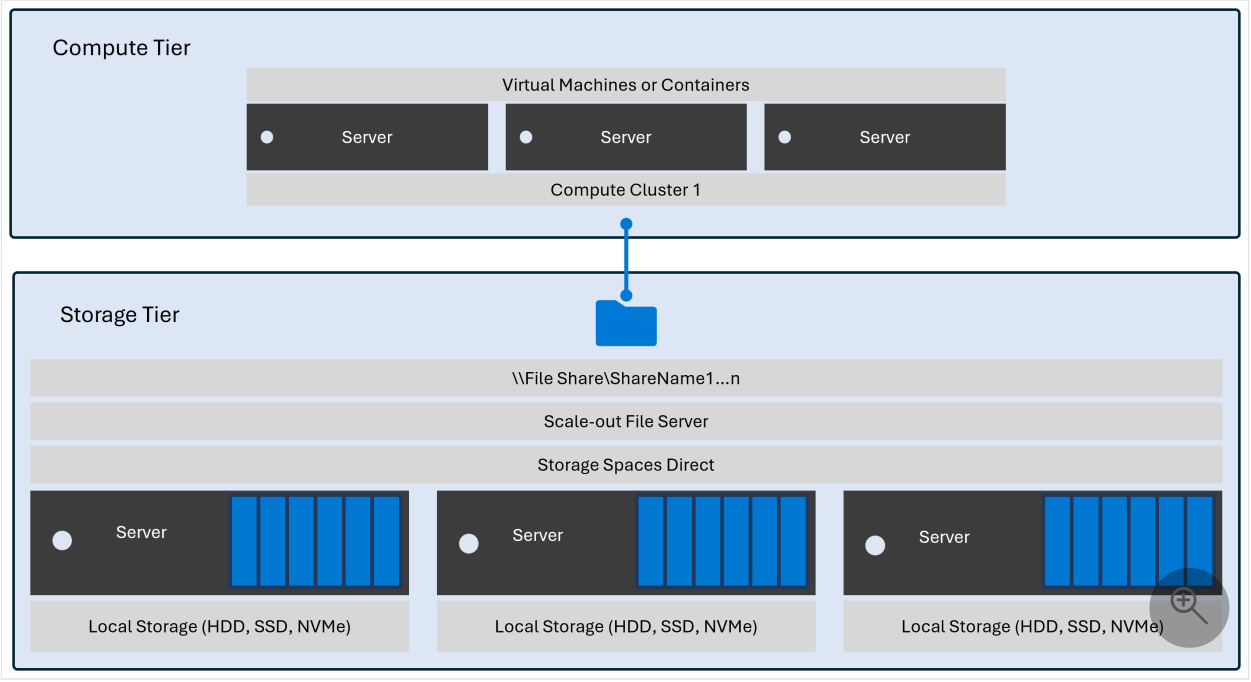
When using this configuration:

- VMs are highly available to any node in the same compute cluster.
- Data gets replicated to other nodes in the storage cluster to increase storage resilience. Storage Spaces Direct supports several storage resiliency models. For more information, see Fault tolerance and storage efficiency on Azure Stack HCI and Windows Server clusters.

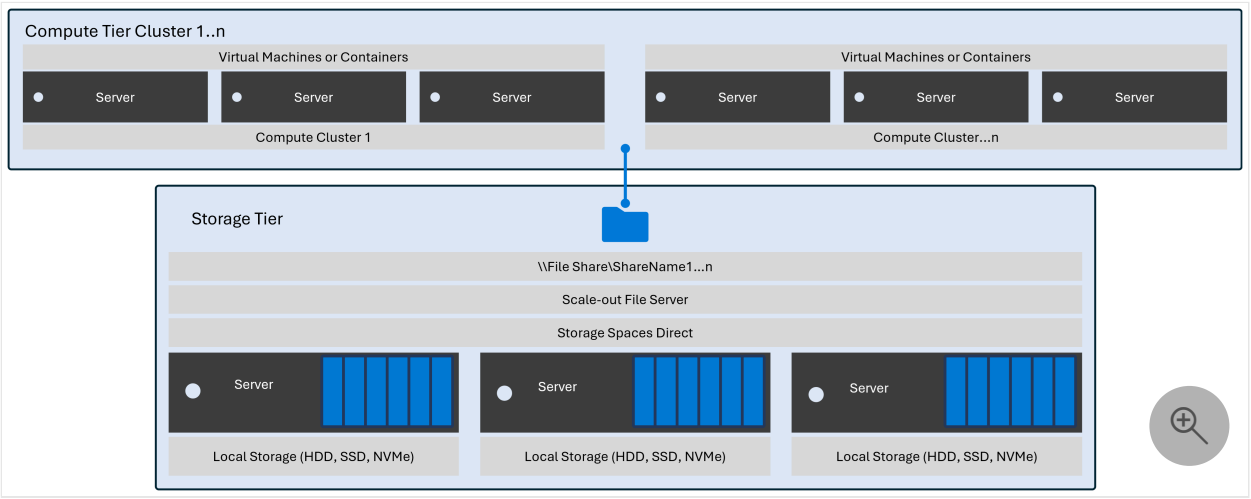
In this model, compute and storage scale independently from each other. This architecture is ideal for scenarios where your storage and compute requirements grow at different rates. For example, if you need more CPU or memory resources to host VMs, you can add extra cluster nodes without also adding more storage. If you have VMs that consume a lot of storage but don't consume many CPU resources, you can add another storage node or cluster without adding more compute resources.

- Each compute cluster can contain between 1 and 64 nodes.
- Each storage cluster can contain between 1 and 16 nodes.

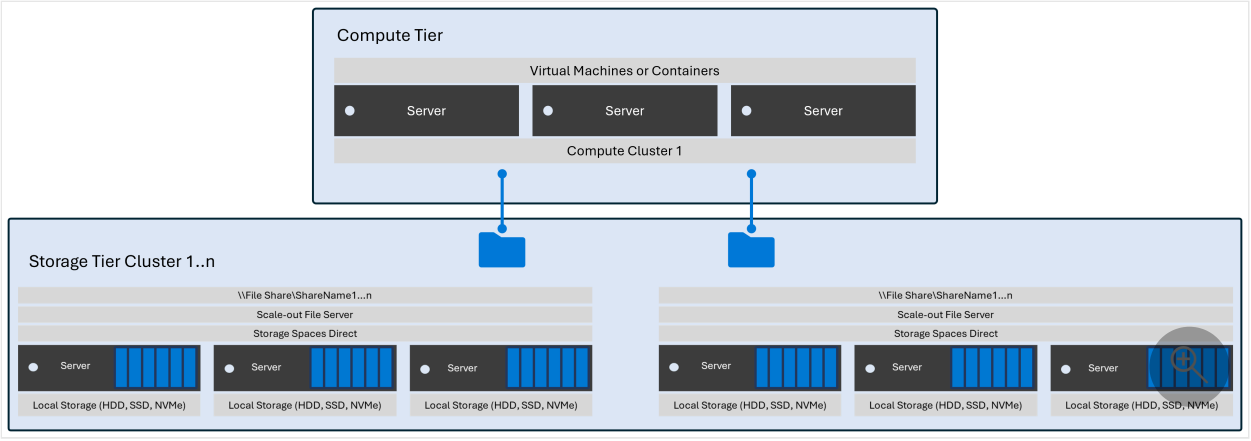
The following diagram shows a simple example deployment with one compute cluster and one storage cluster.



When you add more compute resources to run your VMs, you can either add a new node to the existing compute cluster or add a new cluster. The following diagram shows what happens to the simple deployment if you add a new cluster to it without adding more storage.



When you add more storage resources for hosting data without adding more compute resources, you can either add a new node to the existing storage cluster or add a new cluster. The following diagram shows what the simple deployment looks like when you add a new cluster without adding compute resources like CPU or RAM.

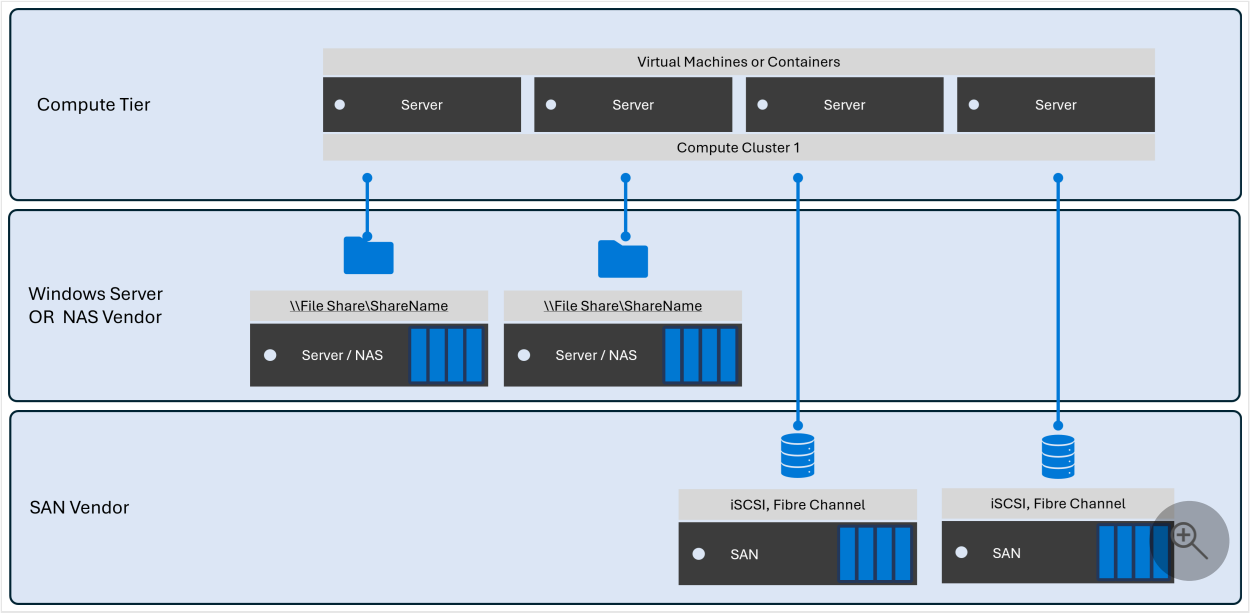


Mixed architecture support

Hyper-V can support the combining the following types of architectures in the same compute cluster:

- Disaggregated Hyper-V with hyperconverged storage
- Disaggregated Hyper-V with SAN
- Disaggregated Hyper-V with NAS

The following diagram shows an example of a deployment with a compute cluster containing a mix of disaggregated SAN and NAS storage.



Network Storage Protocols

Windows Server supports the following network file storage protocols:

- SMB over TCP

- SMB over QUIC
- SMB over RDMA (SMB Direct)

Windows Server also supports the following network block storage protocols:

- iSCSI
- Fibre Channel
- InfiniBand

> **Note**
>
> Configuration specifics ultimately determine whether your deployment can support these protocols. For example, deployments that use the Hyper-V virtual switch don't support InfiniBand. However, they can support InfiniBand devices when they aren't bound to the virtual switch.

Microsoft also provides an in-box software-based iSCSI initiator for network block storage.

You can also use a storage vendor client for any device available in the Windows Server catalog.

Feedback

Use GPUs with clustered VMs

Article • 08/08/2024

> Applies to: Windows Server 2025 (preview)

> **ⓘ Important**
>
> GPU with clustered VMs in Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

You can include GPUs in your clusters to provide GPU acceleration to workloads running in clustered VMs. GPU acceleration can be provided via Discrete Device Assignment (DDA), which allows you to dedicate one or more physical GPUs to a VM, or through GPU Partitioning. Clustered VMs can take advantage of GPU acceleration, and clustering capabilities such as high availability via failover. Live migration of virtual machines (VMs) isn't currently supported, but VMs can be automatically restarted and placed where GPU resources are available if there's a failure.

In this article, you will learn how to use graphics processing units (GPUs) with clustered VMs to provide GPU acceleration to workloads using Discrete Device Assignment. This article guides you through preparing the cluster, assigning a GPU to a cluster VM, and failing over that VM using Windows Admin Center and PowerShell.

Prerequisites

There are several requirements and things to consider before you begin to use GPUs with clustered VMs:

- You need a Windows Server Failover cluster running Windows Server 2025 or later.
- You must install the same make and model of the GPUs across all the servers in your cluster.
- Review and follow the instructions from your GPU manufacturer to install the necessary drivers and software on each server in the cluster.
- Depending on your hardware vendor, you might also need to configure any GPU licensing requirements.

- You need a machine with Windows Admin Center installed. This machine could be one of your cluster nodes.
- Create a VM to assign the GPU to. Prepare that VM for DDA by setting its cache behavior, stop action, and memory-mapped I/O (MMIO) properties according to the instructions in Deploy graphics devices using Discrete Device Assignment.
- Prepare the GPUs in each server by installing security mitigation drivers on each server, disabling the GPUs, and dismounting them from the host. To learn more about this process, see Deploy graphics devices by using Discrete Device Assignment.

Prepare the cluster

When the prerequisites are complete, you can prepare the cluster to use GPUs with clustered VMs.

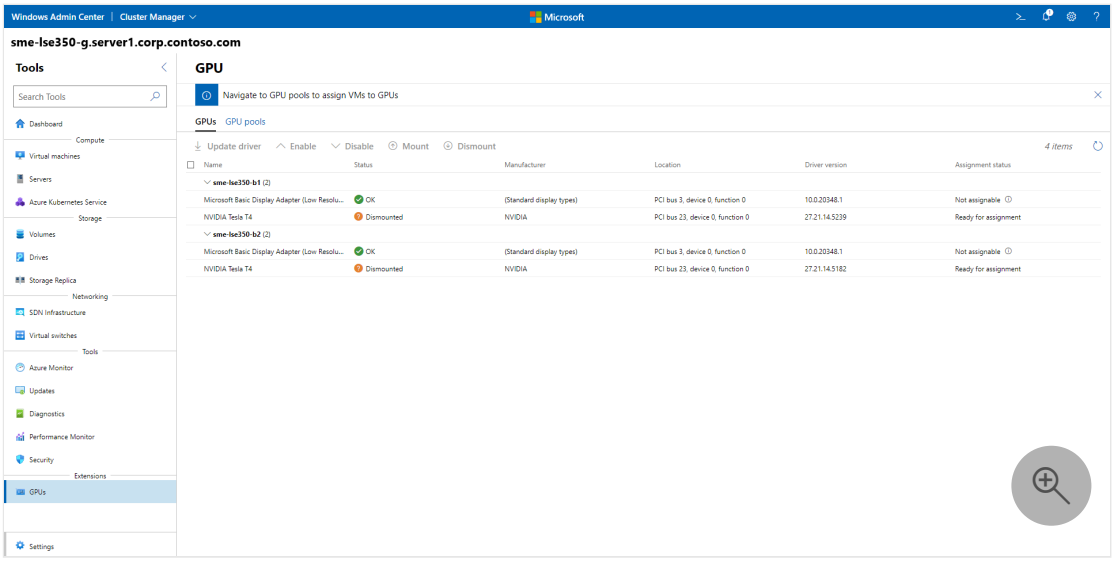
Preparing the cluster involves creating a resource pool that contains the GPUs that are available for assignment to VMs. The cluster uses this pool to determine VM placement for any started or moved VMs that are assigned to the GPU resource pool.

Windows Admin Center

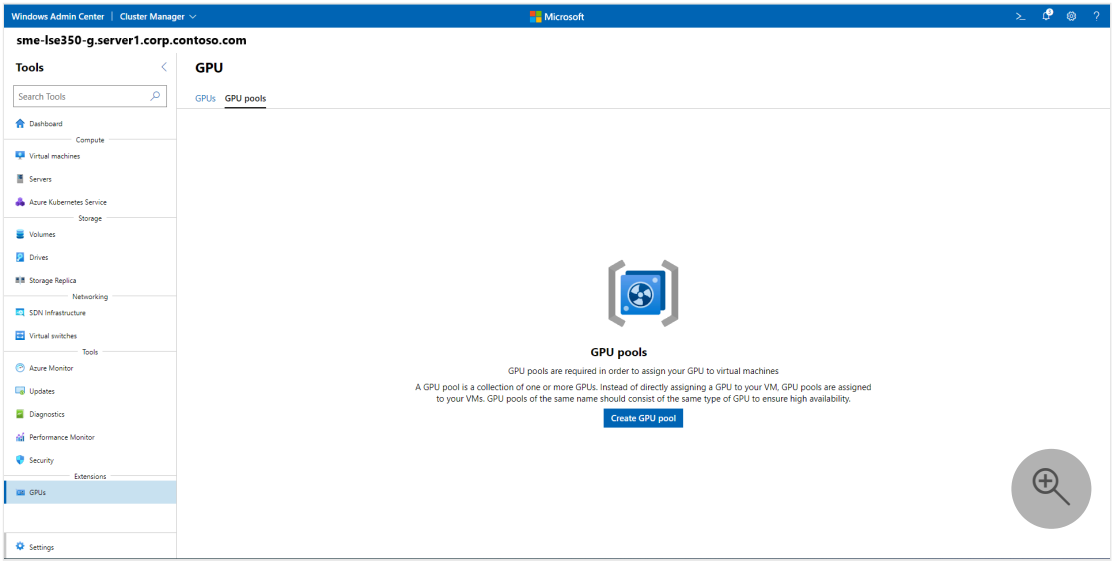
Using Windows Admin Center, follow these steps to prepare the cluster to use GPUs with clustered VMs.

To prepare the cluster and assign a VM to a GPU resource pool:

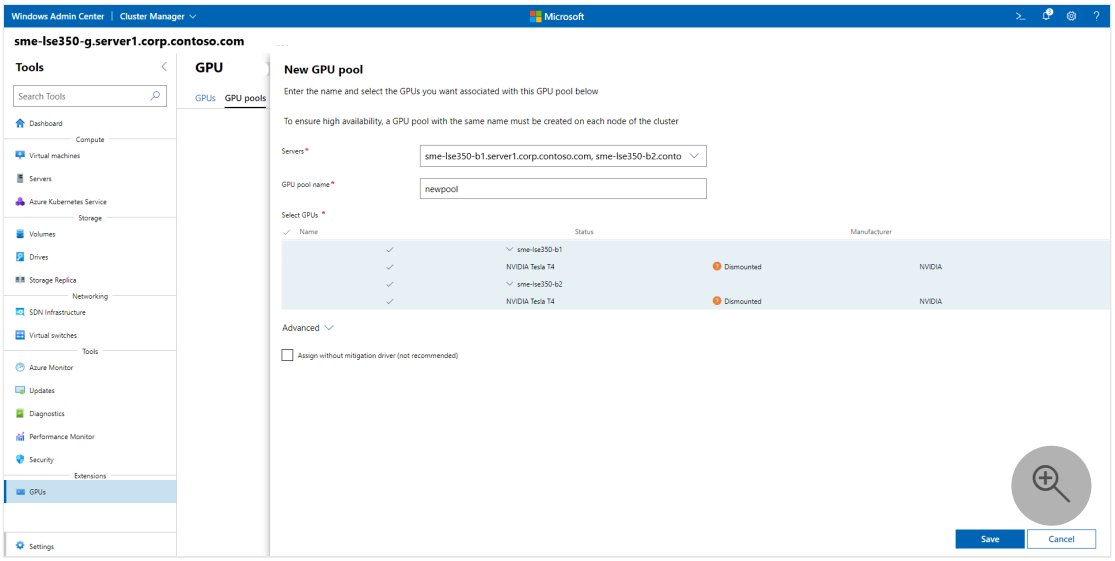
1. Launch Windows Admin Center and make sure the **GPUs** extension is already installed.
2. Select **Cluster Manager** from the top dropdown menu and connect to your cluster.
3. From the **Settings** menu, select **Extensions** > **GPUs**.
4. On the **Tools** menu, under **Extensions**, select **GPUs** to open the tool.



5. On tool's main page, select the **GPU pools** tab, and then select **Create GPU pool**.



6. On the **New GPU pool** page, specify the following and then select **Save**:
   a. **Server** name
   b. **GPU pool** name
   c. **GPUs** that you want to add to the pool



After the process completes, you'll receive a success prompt that shows the name of the new GPU pool and the host server.

Assign a VM to a GPU resource pool

You can now assign a VM to a GPU resource pool. You can assign one or more VMs to a clustered GPU resource pool, and remove a VM from a clustered GPU resource pool.

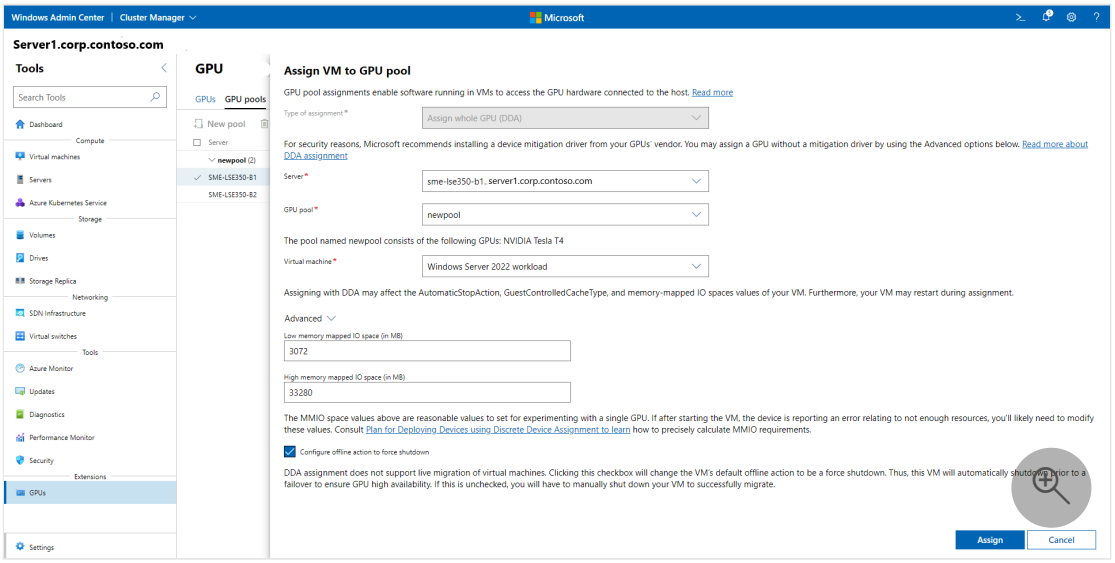
Windows Admin Center

Follow these steps to assign an existing VM to a GPU resource pool using Windows Admin Center.

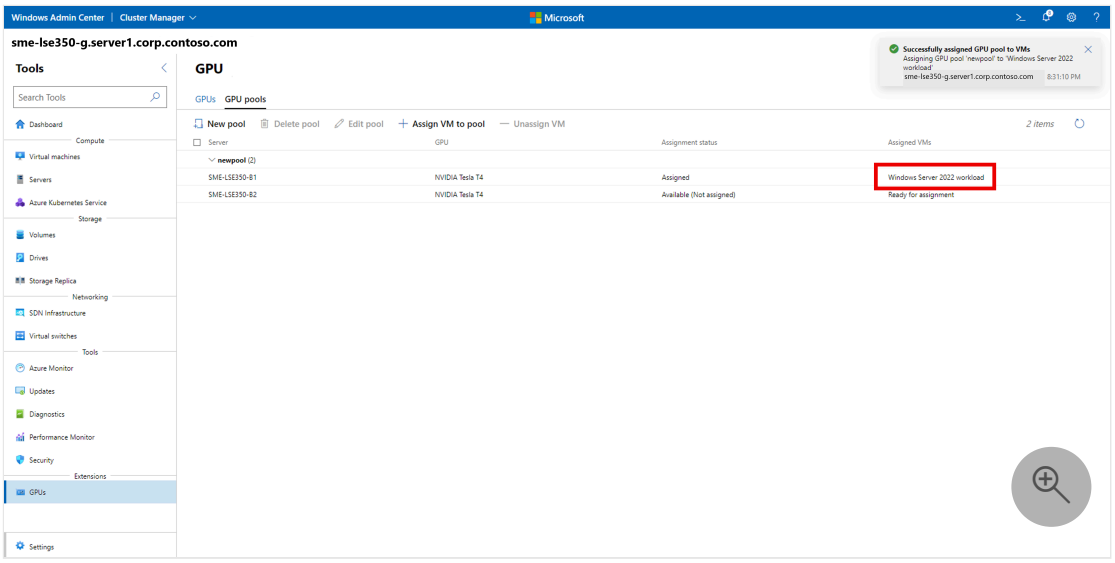
> **Note**
>
> You also need to install drivers from your GPU manufacturer inside the VM so that apps in the VM can take advantage of the GPU assigned to them.

1. On the **Assign VM to GPU pool** page, specify the following, then select **Assign**:
   a. **Server** name
   b. **GPU pool** name
   c. **Virtual machine** that you want to assign the GPU to from the GPU pool.

   You can also define advanced setting values for memory-mapped IO (MMIO) spaces to determine resource requirements for a single GPU.



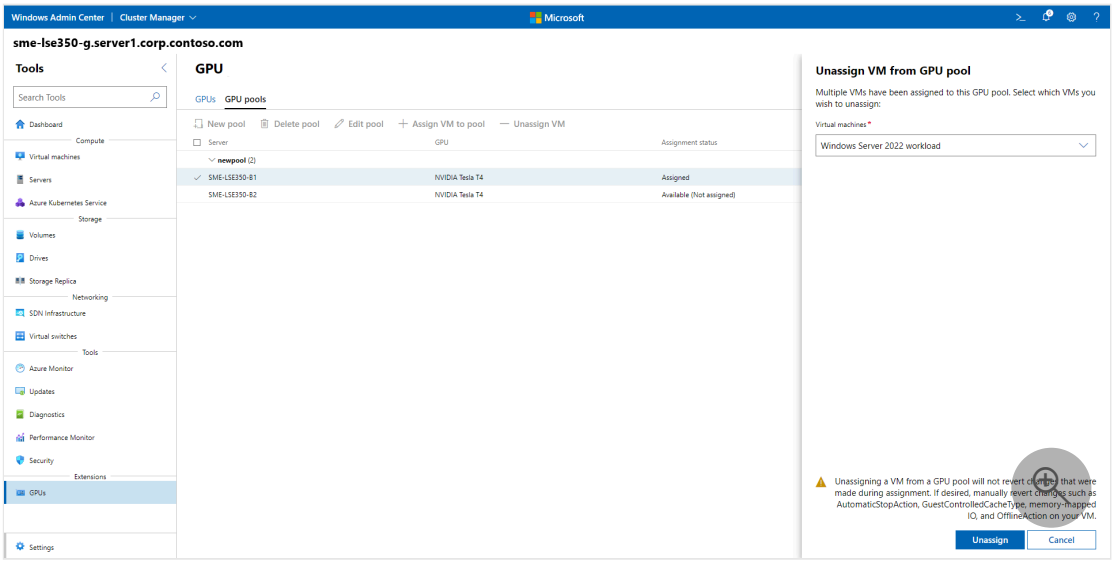
After the process completes, you'll receive a confirmation prompt that shows you successfully assigned the GPU from the GPU resource pool to the VM, which displays under **Assigned VMs**.



To unassign a VM from a GPU resource pool:

1. On the **GPU pools** tab, select the GPU that you want to unassign, and then select **Unassign VM**.

2. On the **Unassign VM from GPU pool** page, in the **Virtual machines** list box, specify the name of the VM, and then select **Unassign**.



After the process completes, you receive a success prompt that the VM has been unassigned from the GPU pool, and under **Assignment status** the GPU shows **Available (Not assigned)**.

When you start the VM, the cluster ensures that the VM is placed on a server with available GPU resources from this cluster-wide pool. The cluster also assigns the GPU to the VM through DDA, which allows the GPU to be accessed from workloads inside the VM.

Fail over a VM with an assigned GPU

To test the cluster's ability to keep your GPU workload available, perform a drain operation on the server where the VM is running with an assigned GPU. To drain the server, follow the instructions in Failover cluster maintenance procedures. The cluster restarts the VM on another server in the cluster, as long as another server has sufficient available GPU resources in the pool that you created.

Related content

For more information on using GPUs with your clustered VMs, see:

- Manage VMs with Windows Admin Center
- Plan for deploying devices by using Discrete Device Assignment

---

Feedback

Was this page helpful? Yes No

About dump encryption

Article • 09/17/2020

Dump encryption can be used to encrypt crash dumps and live dumps generated for a system. The dumps are encrypted using a symmetric encryption key which is generated for each dump. This key itself is then encrypted using the public key specified by the trusted administrator of the host (crash dump encryption key protector). This ensures that only someone having the matching private key can decrypt and therefore access contents of the dump. This capability is leveraged in a guarded fabric. Note: If you configure dump encryption, also disable Windows Error Reporting. WER cannot read encrypted crash dumps.

Configuring dump encryption

Manual configuration

To turn on dump encryption using the registry, configure the following registry values under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl`

Expand table

Value Name	Type	Value
DumpEncryptionEnabled	DWORD	1 to enable dump encryption, 0 to disable dump encryption
EncryptionCertificates\Certificate.1::PublicKey	Binary	Public key (RSA, 2048 Bit) that should be used for encrypting dumps. This has to be formatted as BCRYPT_RSAKEY_BLOB.
EncryptionCertificates\Certificate.1::Thumbprint	String	Certificate thumbprint to allow automatic lookup of private key in the local certificate store when decrypting a crash dump.

Configuration using script

To simplify configuration, a sample script is available to enable dump encryption based on a public key from a certificate.

1. In a trusted environment: Create a certificate with a 2048 Bit RSA key and export the public certificate
2. On target hosts: Import the public certificate to the local certificate store
3. Run the sample configuration script

```
.\Set-DumpEncryptionConfiguration.ps1 -Certificate
(Cert:\CurrentUser\My\093568AB328DF385544FAFD57EE53D73EFAAF519) -Force
```

Decrypting encrypted dumps

To decrypt an existing encrypted dump file, you need to download and install the Debugging Tools for Windows. This tool set contains KernelDumpDecrypt.exe which can be used to decrypt an encrypted dump file. If the certificate including the private key is present in the current user's certificate store, the dump file can be decrypted by calling

```
KernelDumpDecrypt.exe memory.dmp memory_decr.dmp
```

After decryption, tools like WinDbg can open the decrypted dump file.

Troubleshooting dump encryption

If dump encryption is enabled on a system but no dumps are being generated, please check the system's `System` event log for `Kernel-IO` event 1207. When dump encryption cannot be initialized, this event is created and dumps are disabled.

Expand table

Detailed error message	Steps to mitigate
Public Key or Thumbprint registry missing	Check if both registry values exist in the expected location
Invalid Public Key	Make sure that the public key stored in the PublicKey registry value is stored as BCRYPT_RSAKEY_BLOB.
Unsupported Public Key Size	Currently, only 2048 Bit RSA keys are supported. Configure a key that matches this requirement

Also check if the value `GuardedHost` under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl\ForceDumpsDisabled` is set to a value other than 0. This disables crash dumps completely. If this is the case, set it to 0.

Feedback

Was this page helpful? Yes No

Cmdlets for configuring persistent memory devices for Hyper-V VMs

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019

This article provides system administrators and IT Pros with information about configuring Hyper-V VMs with persistent memory (aka storage class memory or NVDIMM). JDEC-compliant NVDIMM-N persistent memory devices are supported in Windows Server 2016 and Windows 10 and provide byte-level access to very low latency non-volatile devices. VM persistent memory devices are supported in Windows Server 2019.

Create a persistent memory device for a VM

Use the **New-VHD** cmdlet to create a persistent memory device for a VM. The device must be created on an existing NTFS DAX volume. The new filename extension (.vhdpmem) is used to specify that the device is a persistent memory device. Only the fixed VHD file format is supported.

Example: `New-VHD d:\VMPMEMDevice1.vhdpmem -Fixed -SizeBytes 4GB`

Create a VM with a persistent memory controller

Use the **New-VM cmdlet** to create a Generation 2 VM with specified memory size and path to a VHDX image. Then, use **Add-VMPmemController** to add a persistent memory controller to a VM.

Example:

PowerShell

```
New-VM -Name "ProductionVM1" -MemoryStartupBytes 1GB -VHDPath
c:\vhd\BaseImage.vhdx

Add-VMPmemController ProductionVM1x
```

Attach a persistent memory device to a VM

Use **Add-VMHardDiskDrive** to attach a persistent memory device to a VM

Example: `Add-VMHardDiskDrive ProductionVM1 PMEM -ControllerLocation 1 -Path D:\VPMEMDevice1.vhdpmem`

Persistent memory devices within a Hyper-V VM appear as a persistent memory device to be consumed and managed by the guest operating system. Guest operating systems can use the device as a block or DAX volume. When persistent memory devices within a VM are used as a DAX volume, they benefit from low latency byte-level address-ability of the host device (no I/O virtualization on the code path).

> **Note**
>
> Persistent memory is only supported for Hyper-V Gen2 VMs. Live Migration and Storage Migration are not supported for VMs with persistent memory. Production checkpoints of VMs do not include persistent memory state.

Choose between standard or production checkpoints in Hyper-V

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Starting with Windows Server 2016 and Windows 10, you can choose between standard and production checkpoints for each virtual machine. Production checkpoints are the default for new virtual machines.

- Production checkpoints are "point in time" images of a virtual machine, which can be restored later on in a way that is completely supported for all production workloads. This is achieved by using backup technology inside the guest to create the checkpoint, instead of using saved state technology.
- Standard checkpoints capture the state, data, and hardware configuration of a running virtual machine and are intended for use in development and test scenarios. Standard checkpoints can be useful if you need to recreate a specific state or condition of a running virtual machine so that you can troubleshoot a problem.

Change checkpoints to production or standard checkpoints

1. In **Hyper-V Manager**, right-click the virtual machine and click **Settings**.
2. Under the **Management** section, select **Checkpoints**.
3. Select either production checkpoints or standard checkpoints.

   If you choose production checkpoints, you can also specify whether the host should take a standard checkpoint if a production checkpoint can't be taken. If you clear this checkbox and a production checkpoint can't be taken, then no checkpoint is taken.
4. If you want to store the checkpoint configuration files in a different place, change it in the **Checkpoint File Location** section.
5. Click **Apply** to save your changes. If you're done, click **OK** to close the dialog box.

> **Note**
>
> Only **Production Checkpoints** are supported on guests that run Active Directory Domain Services role (Domain Controller) or Active Directory Lightweight Directory Services role.

Additional References

- Production checkpoints
- Enable or disable checkpoints

Create Hyper-V VHD Set files

Article • 12/08/2020

VHD Set files are a new shared Virtual Disk model for guest clusters in Windows Server 2016. VHD Set files support online resizing of shared virtual disks, support Hyper-V Replica, and can be included in application-consistent checkpoints.

VHD Set files use a new VHD file type, .VHDS. VHD Set files store checkpoint information about the group virtual disk used in guest clusters, in the form of metadata.

Hyper-V handles all aspects of managing the checkpoint chains and merging the shared VHD set. Management software can run disk operations like online resizing on VHD Set files in the same way it does for .VHDX files. This means that management software doesn't need to know about the VHD Set file format.

> ⓘ **Note**
>
> It's important to evaluate the impact of VHD Set files before deployment into production. Make sure that there is no performance or functional degradation in your environment, such as disk latency.

Create a VHD Set file from Hyper-V Manager

1. Open Hyper-V Manager. Click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the Action pane, click **New**, and then click **Hard Disk**.
3. On the **Choose Disk Format** page, select **VHD Set** as the format of the virtual hard disk.
4. Continue through the pages of the wizard to customize the virtual hard disk. You can click **Next** to move through each page of the wizard, or you can click the name of a page in the left pane to move directly to that page.
5. After you have finished configuring the virtual hard disk, click **Finish**.

Create a VHD Set file from Windows PowerShell

Use the New-VHD cmdlet, with the file type .VHDS in the file path. This example creates a VHD Set file named base.vhds that is 10 Gigabytes in size.

PowerShell

```
PS c:\>New-VHD -Path c:\base.vhds -SizeBytes 10GB
```

Migrate a shared VHDX file to a VHD Set file

Migrating an existing shared VHDX to a VHDS requires taking the VM offline. This is the recommended process using Windows PowerShell:

1. Remove the VHDX from the VM. For example, run:

   PowerShell

   ```
   PS c:\>Remove-VMHardDiskDrive existing.vhdx
   ```

2. Convert the VHDX to a VHDS. For example, run:

   PowerShell

   ```
   PS c:\>Convert-VHD existing.vhdx new.vhds
   ```

3. Add the VHDS to the VM. For example, run:

   PowerShell

   ```
   PS c:\>Add-VMHardDiskDrive new.vhds
   ```

Enable or disable checkpoints in Hyper-V

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

You can choose to enable or disable checkpoints for each virtual machine.

1. In **Hyper-V Manager**, right-click the virtual machine and click **Settings**.
2. Under the **Management** section, select **Checkpoints**.
3. To allow checkpoints to be taken of this virtual machine, make sure **Enable checkpoints** is selected. To disable checkpoints, clear the **Enable checkpoints** check box.
4. Click **Apply** to apply your changes. If you are done, click **OK** to close the dialog box.

Additional References

Choose between standard or production checkpoints in Hyper-V

Remotely manage Hyper-V hosts with Hyper-V Manager

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows 10, Windows 8.1

This article lists the supported combinations of Hyper-V hosts and Hyper-V Manager versions and describes how to connect to remote and local Hyper-V hosts so you can manage them.

Hyper-V Manager lets you manage a small number of Hyper-V hosts, both remote and local. It's installed when you install the Hyper-V Management Tools, which you can do either through a full Hyper-V installation or a tools-only installation. Doing a tools-only installation means you can use the tools on computers that don't meet the hardware requirements to host Hyper-V. For details about hardware for Hyper-V hosts, see System requirements.

If Hyper-V Manager isn't installed, see the instructions below.

Supported combinations of Hyper-V Manager and Hyper-V host versions

In some cases you can use a different version of Hyper-V Manager than the Hyper-V version on the host, as shown in the table. When you do this, Hyper-V Manager provides the features available for the version of Hyper-V on the host you're managing. For example, if you use the version of Hyper-V Manager in Windows Server 2012 R2 to remotely manage a host running Hyper-V in Windows Server 2012, you won't be able to use features available in Windows Server 2012 R2 on that Hyper-V host.

The following table shows which versions of a Hyper-V host you can manage from a particular version of Hyper-V Manager. Only supported operating system versions are listed. For details about the support status of a particular operating system version, use the **Search product lifecycle** button on the Microsoft Lifecycle Policy page. In general, older versions of Hyper-V Manager can only manage a Hyper-V host running the same version or the comparable Windows Server version.

Hyper-V Manager version	Hyper-V host version

Hyper-V Manager version	Hyper-V host version
Windows Server 2016, Windows 10	- Windows Server 2016—all editions and installation options, including Nano Server, and corresponding version of Hyper-V Server - Windows Server 2012 R2—all editions and installation options, and corresponding version of Hyper-V Server - Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server - Windows 10 - Windows 8.1
Windows Server 2012 R2, Windows 8.1	- Windows Server 2012 R2—all editions and installation options, and corresponding version of Hyper-V Server - Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server - Windows 8.1
Windows Server 2012	- Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server
Windows Server 2008 R2 Service Pack 1, Windows 7 Service Pack 1	- Windows Server 2008 R2—all editions and installation options, and corresponding version of Hyper-V Server
Windows Server 2008, Windows Vista Service Pack 2	- Windows Server 2008—all editions and installation options, and corresponding version of Hyper-V Server

> **Note**
>
> Service pack support ended for Windows 8 on January 12, 2016. For more information, see the **Windows 8.1 FAQ**.

Connect to a Hyper-V host

To connect to a Hyper-V host from Hyper-V Manager, right-click **Hyper-V Manager** in the left pane, and then click **Connect to Server**.

Manage Hyper-V on a local computer

Hyper-V Manager doesn't list any computers that host Hyper-V until you add the computer, including a local computer. To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.

3. From **Select Computer**, click **Local computer** and then click **OK**.

If you can't connect:

- It's possible that only the Hyper-V tools are installed. To check that Hyper-V platform is installed, look for the Virtual Machine Management service. /(Open the Services desktop app: click **Start**, click the **Start Search** box, type **services.msc**, and then press **Enter**. If the Virtual Machine Management service isn't listed, install the Hyper-V platform by following the instructions in Install Hyper-V.
- Check that your hardware meets the requirements. See System requirements.
- Check that your user account belongs to the Administrators group or the Hyper-V Administrators group.

Manage Hyper-V hosts remotely

To manage remote Hyper-V hosts, enable remote management on both the local computer and remote host.

On Windows Server, open Server Manager >**Local Server** >**Remote management** and then click **Allow remote connections to this computer**.

Or, from either operating system, open Windows PowerShell as Administrator and run:

PowerShell

```
Enable-PSRemoting
```

Connect to hosts in the same domain

For Windows 8.1 and earlier, remote management works only when the host is in the same domain and your local user account is also on the remote host.

To add a remote Hyper-V host to Hyper-V Manager, select **Another computer** in the **Select Computer** dialogue box and type the remote host's hostname, NetBIOS name, or fully qualified domain name (FQDN).

Hyper-V Manager in Windows Server 2016 and Windows 10 offers more types of remote connection than previous versions, described in the following sections.

Connect to a Windows Server 2016 or Windows 10 remote host as a different user

This lets you connect to the Hyper-V host when you're not running on the local computer as a user that's a member of either the Hyper-V Administrators group or the Administrators group on the Hyper-V host. To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.
3. Select **Connect as another user** in the **Select Computer** dialogue box.
4. Select **Set User**.

> ⓘ **Note**
>
> This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

Connect to a Windows Server 2016 or Windows 10 remote host using IP address

To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.
3. Type the IP address into the **Another Computer** text field.

> ⓘ **Note**
>
> This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

Connect to a Windows Server 2016 or Windows 10 remote host outside your domain, or with no domain

To do this:

1. On the Hyper-V host to be managed, open a Windows PowerShell session as Administrator.

2. Create the necessary firewall rules for private network zones:

```PowerShell
Enable-PSRemoting
```

3. To allow remote access on public zones, enable firewall rules for CredSSP and WinRM:

   ```
   PowerShell

   Enable-WSManCredSSP -Role server
   ```

   For details, see Enable-PSRemoting and Enable-WSManCredSSP.

Next, configure the computer you'll use to manage the Hyper-V host.

1. Open a Windows PowerShell session as Administrator.
2. Run these commands:

   ```
   PowerShell

   Set-Item WSMan:\localhost\Client\TrustedHosts -Value "fqdn-of-hyper-v-host"
   ```

   ```
   PowerShell

   Enable-WSManCredSSP -Role client -DelegateComputer "fqdn-of-hyper-v-host"
   ```

3. You might also need to configure the following group policy:

   - **Computer Configuration** > **Administrative Templates** > **System** > **Credentials Delegation** > **Allow delegating fresh credentials with NTLM-only server authentication**
   - Click **Enable** and add *wsman/fqdn-of-hyper-v-host*.

4. Open **Hyper-V Manager**.
5. In the left pane, right-click **Hyper-V Manager**.
6. Click **Connect to Server**.

> **Note**
>
> This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

For cmdlet details, see Set-Item and Enable-WSManCredSSP.

Install Hyper-V Manager

To use a UI tool, choose the one appropriate for the operating system on the computer where you'll run Hyper-V Manager:

On Windows Server, open Server Manager > **Manage** > **Add roles and features**. Move to the **Features** page and expand **Remote server administration tools** > **Role administration tools** > **Hyper-V management tools**.

On Windows, Hyper-V Manager is available on any Windows operating system that includes Hyper-V.

1. On the Windows desktop, click the Start button and begin typing **Programs and features**.
2. In search results, click **Programs and Features**.
3. In the left pane, click **Turn Windows features on or off**.
4. Expand the Hyper-V folder, and **click Hyper-V Management Tools**.
5. To install Hyper-V Manager, click **Hyper-V Management Tools**. If you want to also install the Hyper-V module, click that option.

To use Windows PowerShell, run the following command as Administrator:

PowerShell

```
add-windowsfeature rsat-hyper-v-tools
```

Additional References

Install Hyper-V

Hyper-V Host CPU Resource Management

Article • 09/17/2020

Hyper-V host CPU resource controls introduced in Windows Server 2016 or later allow Hyper-V administrators to better manage and allocate host server CPU resources between the "root", or management partition, and guest VMs. Using these controls, administrators can dedicate a subset of the processors of a host system to the root partition. This can segregate the work done in a Hyper-V host from the workloads running in guest virtual machines by running them on separate subsets of the system processors.

For details about hardware for Hyper-V hosts, see Windows 10 Hyper-V System Requirements.

Background

Before setting controls for Hyper-V host CPU resources, it's helpful to review the basics of the Hyper-V architecture. You can find a general summary in the Hyper-V Architecture section. These are important concepts for this article:

- Hyper-V creates and manages virtual machine partitions, across which compute resources are allocated and shared, under control of the hypervisor. Partitions provide strong isolation boundaries between all guest virtual machines, and between guest VMs and the root partition.
- The root partition is itself a virtual machine partition, although it has unique properties and much greater privileges than guest virtual machines. The root partition provides the management services that control all guest virtual machines, provides virtual device support for guests, and manages all device I/O for guest virtual machines. Microsoft strongly recommends not running any application workloads in a host partition.
- Each virtual processor (VP) of the root partition is mapped 1:1 to an underlying logical processor (LP). A host VP will always run on the same underlying LP – there is no migration of the root partition's VPs.
- By default, the LPs on which host VPs run can also run guest VPs.
- A guest VP may be scheduled by the hypervisor to run on any available logical processor. While the hypervisor scheduler takes care to consider temporal cache

locality, NUMA topology, and many other factors when scheduling a guest VP, ultimately the VP could be scheduled on any host LP.

The Minimum Root, or "Minroot" Configuration

Early versions of Hyper-V had an architectural maximum limit of 64 VPs per partition. This applied to both the root and guest partitions. As systems with more than 64 logical processors appeared on high-end servers, Hyper-V also evolved its host scale limits to support these larger systems, at one point supporting a host with up to 320 LPs. However, breaking the 64 VP per partition limit at that time presented several challenges and introduced complexities that made supporting more than 64 VPs per partition prohibitive. To address this, Hyper-V limited the number of VPs given to the root partition to 64, even if the underlying machine had many more logical processors available. The hypervisor would continue to utilize all available LPs for running guest VPs, but artificially capped the root partition at 64. This configuration became known as the "minimum root", or "minroot" configuration. Performance testing confirmed that, even on large scale systems with more than 64 LPs, the root did not need more than 64 root VPs to provide sufficient support to a large number of guest VMs and guest VPs – in fact, much less than 64 root VPs was often adequate, depending of course on the number and size of the guest VMs, the specific workloads being run, etc.

This "minroot" concept continues to be utilized today. In fact, even as Windows Server 2016 Hyper-V increased its maximum architectural support limit for host LPs to 512 LPs, the root partition will still be limited to a maximum of 320 LPs.

Using Minroot to Constrain and Isolate Host Compute Resources

With the high default threshold of 320 LPs in Windows Server 2016 Hyper-V, the minroot configuration will only be utilized on the very largest server systems. However, this capability can be configured to a much lower threshold by the Hyper-V host administrator, and thus leveraged to greatly restrict the amount of host CPU resources available to the root partition. The specific number of root LPs to utilize must of course be chosen carefully to support the maximum demands of the VMs and workloads allocated to the host. However, reasonable values for the number of host LPs can be determined through careful assessment and monitoring of production workloads, and validated in non-production environments before broad deployment.

Enabling and Configuring Minroot

The minroot configuration is controlled via hypervisor BCD entries. To enable minroot, from a cmd prompt with administrator privileges:

```
bcdedit /set hypervisorrootproc n
```

Where n is the number of root VPs.

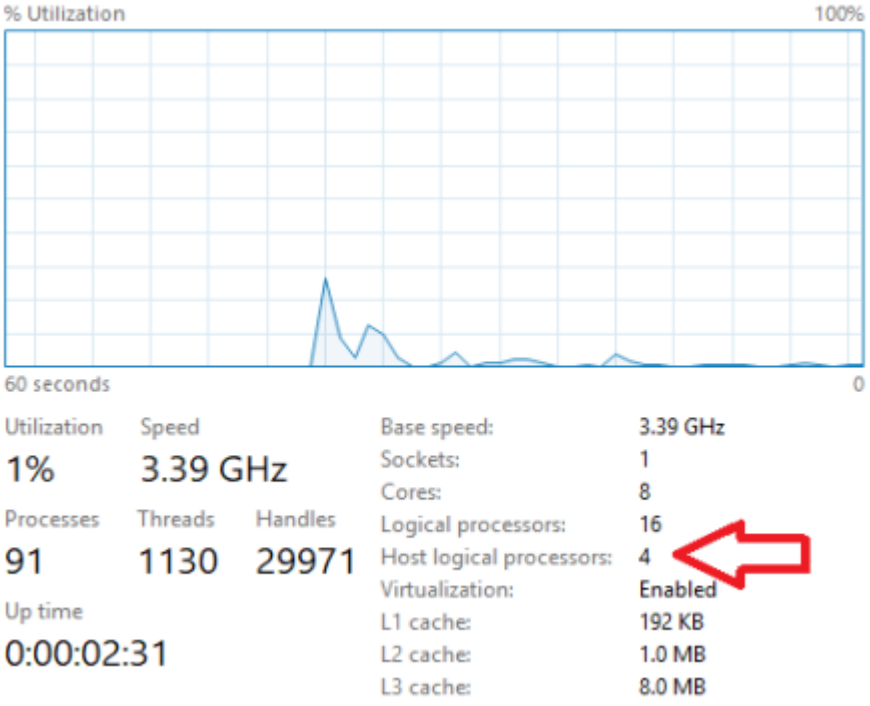
The system must be rebooted, and the new number of root processors will persist for the lifetime of the OS boot. The minroot configuration cannot be changed dynamically at runtime.

If there are multiple NUMA nodes, each node will get `n/NumaNodeCount` processors.

Note that with multiple NUMA nodes, you must ensure the VM's topology is such that there are enough free LPs (i.e., LPs without root VPs) on each NUMA node to run the corresponding VM's NUMA node VPs.

Verifying the Minroot Configuration

You can verify the host's minroot configuration using Task Manager, as shown below.



When Minroot is active, Task Manager will display the number of logical processors currently allotted to the host, in addition to the total number of logical processors in the

system.

Virtual Machine Resource Controls

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article describes Hyper-V resource and isolation controls for virtual machines. These capabilities, which we'll refer to as Virtual Machine CPU Groups, or just "CPU groups", were introduced in Windows Server 2016. CPU groups allow Hyper-V administrators to better manage and allocate the host's CPU resources across guest virtual machines. Using CPU groups, Hyper-V administrators can:

- Create groups of virtual machines, with each group having different allocations of the virtualization host's total CPU resources, shared across the entire group. This allows the host administrator to implement classes of service for different types of VMs.
- Set CPU resource limits to specific groups. This "group cap" sets the upper bound for host CPU resources that the entire group may consume, effectively enforcing the desired class of service for that group.
- Constrain a CPU group to run only on a specific set of the host system's processors. This can be used to isolate VMs belonging to different CPU groups from each other.

Managing CPU Groups

CPU groups are managed through the Hyper-V Host Compute Service, or HCS. A great description of the HCS, its genesis, links to the HCS APIs, and more is available on the Microsoft Virtualization team's blog in the posting Introducing the Host Compute Service (HCS).

> **Note**
>
> Only the HCS may be used to create and manage CPU groups; the Hyper-V Manager applet, WMI and PowerShell management interfaces don't support CPU groups.

Microsoft provides a command line utility, cpugroups.exe, on the Microsoft Download Center which uses the HCS interface to manage CPU groups. This utility can also

display the CPU topology of a host.

How CPU Groups Work

Allocation of host compute resources across CPU groups is enforced by the Hyper-V hypervisor, using a computed CPU group cap. The CPU group cap is a fraction of the total CPU capacity for a CPU group. The value of the group cap depends on the group class, or priority level assigned. The computed group cap can be thought of as "a number of LP's worth of CPU time". This group budget is shared, so if only a single VM were active, it could use the entire group's CPU allocation for itself.

The CPU group cap is calculated as G = *n* x *C*, where:

- *G* is the amount of host LP we'd like to assign to the group
- *n* is the total number of logical processors (LPs) in the group
- *C* is the maximum CPU allocation — that is, the class of service desired for the group, expressed as a percentage of the system's total compute capacity

For example, consider a CPU group configured with 4 logical processors (LPs), and a cap of 50%.

- G = n * C
- G = 4 * 50%
- G = 2 LP's worth of CPU time for the entire group

In this example, the CPU group G is allocated 2 LP's worth of CPU time.

Note that the group cap applies regardless of the number of virtual machines or virtual processors bound to the group, and regardless of the state (e.g., shutdown or started) of the virtual machines assigned to the CPU group. Therefore, each VM bound to the same CPU group will receive a fraction of the group's total CPU allocation, and this will change with the number of VMs bound to the CPU group. Therefore, as VMs are bound or unbound VMs from a CPU group, the overall CPU group cap must be readjusted and set to maintain the resulting per-VM cap desired. The VM host administrator or virtualization management software layer is responsible for managing group caps as necessary to achieve the desired per-VM CPU resource allocation.

Example Classes of Service

Let's look at some simple examples. To start with, assume the Hyper-V host administrator would like to support two tiers of service for guest VMs:

1. A low-end "C" tier. We'll give this tier 10% of the entire host's compute resources.
2. A mid-range "B" tier. This tier is allocated 50% of the entire host's compute resources.

At this point in our example we'll assert that no other CPU resource controls are in use, such as individual VM caps, weights, and reserves. However, individual VM caps are important, as we'll see a bit later.

For simplicity's sake, let's assume each VM has 1 VP, and that our host has 8 LPs. We'll start with an empty host.

To create the "B" tier, the host adminstartor sets the group cap to 50%:

- G = n * C
- G = 8 * 50%
- G = 4 LP's worth of CPU time for the entire group

The host administrator adds a single "B" tier VM. At this point, our "B" tier VM can use at most 50% worth of the host's CPU, or the equivalent of 4 LPs in our example system.

Now, the admin adds a second "Tier B" VM. The CPU group's allocation—is divided evenly among all the VMs. We've got a total of 2 VMs in Group B, so each VM now gets half of Group B's total of 50%, 25% each, or the equivalent of 2 LPs worth of compute time.

Setting CPU Caps on Individual VMs

In addition to the group cap, each VM can also have an individual "VM cap". Per-VM CPU resource controls, including a CPU cap, weight, and reserve, have been a part of Hyper-V since its introduction. When combined with a group cap, a VM cap specifies the maximum amount of CPU that each VP can get, even if the group has CPU resources available.

For example, the host administrator might want to place a 10% VM cap on "C" VMs. That way, even if most "C" VPs are idle, each VP could never get more than 10%. Without a VM cap, "C" VMs could opportunistically achieve performance beyond levels allowed by their tier.

Isolating VM Groups to Specific Host Processors

Hyper-V host administrators may also want the ability to dedicate compute resources to a VM. For example, imagine the administrator wanted to offer a premium "A" VM that has a class cap of 100%. These premium VMs also require the lowest scheduling latency and jitter possible; that is, they may not be de-scheduled by any other VM. To achieve this separation, a CPU group can also be configured with a specific LP affinity mapping.

For example, to fit an "A" VM on the host in our example, the administrator would create a new CPU group, and set the group's processor affinity to a subset of the host's LPs. Groups B and C would be affinitized to the remaining LPs. The administrator could create a single VM in Group A, which would then have exclusive access to all LPs in Group A, while the presumably lower tier groups B and C would share the remaining LPs.

Segregating Root VPs from Guest VPs

By default, Hyper-V will create a root VP on each underlying physical LP. These root VPs are strictly mapped 1:1 with the system LPs, and do not migrate — that is, each root VP will always execute on the same physical LP. Guest VPs may be run on any available LP, and will share execution with root VPs.

However, it may be desirable to completely separate root VP activity from guest VPs. Consider our example above where we implement a premium "A" tier VM. To ensure our "A" VM's VPs have the lowest possible latency and "jitter", or scheduling variation, we'd like to run them on a dedicated set of LPs and ensure the root does not run on these LPs.

This can be accomplished using a combination of the "minroot" configuration, which limits the host OS partition to running on a subset of the total system logical processors, along with one or more affinitized CPU groups.

The virtualization host can be configured to restrict the host partition to specific LPs, with one or more CPU groups affinitized to the remaining LPs. In this manner, the root and guest partitions can run on dedicated CPU resources, and completely isolated, with no CPU sharing.

For more information about the "minroot" configuration, see Hyper-V Host CPU Resource Management.

Using the CpuGroups Tool

Let's look at some examples of how to use the CpuGroups tool.

> **Note**
>
> Command line parameters for the CpuGroups tool are passed using only spaces as delimiters. No '/' or '-' characters should proceed the desired command line switch.

Discovering the CPU Topology

Executing CpuGroups with the GetCpuTopology returns information about the current system, as shown below, including the LP Index, the NUMA node to which the LP belongs, the Package and Core IDs, and the ROOT VP index.

The following example shows a system with 2 CPU sockets and NUMA nodes, a total of 32 LPs, and multi threading enabled, and configured to enable Minroot with 8 root VPs, 4 from each NUMA node. The LPs that have root VPs have a RootVpIndex >= 0; LPs with a RootVpIndex of -1 are not available to the root partition, but are still managed by the hypervisor and will run guest VPs as allowed by other configuration settings.

Console

```
C:\vm\tools>CpuGroups.exe GetCpuTopology

LpIndex NodeNumber PackageId CoreId RootVpIndex
------- ---------- --------- ------ -----------
      0          0         0      0           0
      1          0         0      0           1
      2          0         0      1           2
      3          0         0      1           3
      4          0         0      2          -1
      5          0         0      2          -1
      6          0         0      3          -1
      7          0         0      3          -1
      8          0         0      4          -1
      9          0         0      4          -1
     10          0         0      5          -1
     11          0         0      5          -1
     12          0         0      6          -1
     13          0         0      6          -1
     14          0         0      7          -1
     15          0         0      7          -1
     16          1         1     16           4
     17          1         1     16           5
     18          1         1     17           6
     19          1         1     17           7
     20          1         1     18          -1
     21          1         1     18          -1
     22          1         1     19          -1
     23          1         1     19          -1
     24          1         1     20          -1
     25          1         1     20          -1
```

```
      26          1          1      21          -1
      27          1          1      21          -1
      28          1          1      22          -1
      29          1          1      22          -1
      30          1          1      23          -1
      31          1          1      23          -1
```

Example 2 – Print all CPU groups on the host

Here, we'll list all CPU groups on the current host, their GroupId, the group's CPU cap, and the indicies of LPs assigned to that group.

Note that valid CPU cap values are in the range [0, 65536], and these values express the group cap in percent (e.g., 32768 = 50%).

Console

```
C:\vm\tools>CpuGroups.exe GetGroups

CpuGroupId                           CpuCap  LpIndexes
------------------------------------ ------ --------
36AB08CB-3A76-4B38-992E-000000000002 32768  4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003 65536  12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004 65536  24,25,26,27,28,29,30,31
```

Example 3 – Print a single CPU group

In this example, we'll query a single CPU Group using the GroupId as a filter.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups /GroupId:36AB08CB-3A76-4B38-992E-
000000000003
CpuGroupId                           CpuCap   LpIndexes
------------------------------------ ------ ----------
36AB08CB-3A76-4B38-992E-000000000003 65536  12,13,14,15
```

Example 4 – Create a new CPU group

Here, we'll create a new CPU group, specifying the Group ID and the set of LPs to assign to the group.

Console

```
C:\vm\tools>CpuGroups.exe CreateGroup /GroupId:36AB08CB-3A76-4B38-992E-
000000000001 /GroupAffinity:0,1,16,17
```

Now display our newly added group.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups
CpuGroupId                           CpuCap LpIndexes
------------------------------------ ------ ---------
36AB08CB-3A76-4B38-992E-000000000001 65536 0,1,16,17
36AB08CB-3A76-4B38-992E-000000000002 32768 4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003 65536 12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004 65536 24,25,26,27,28,29,30,31
```

Example 5 – Set the CPU group cap to 50%

Here, we'll set the CPU group cap to 50%.

Console

```
C:\vm\tools>CpuGroups.exe SetGroupProperty /GroupId:36AB08CB-3A76-4B38-992E-
000000000001 /CpuCap:32768
```

Now let's confirm our setting by displaying the group we just updated.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups /GroupId:36AB08CB-3A76-4B38-992E-
000000000001

CpuGroupId                           CpuCap LpIndexes
------------------------------------ ------ ---------
36AB08CB-3A76-4B38-992E-000000000001 32768 0,1,16,17
```

Example 6 – Print CPU group ids for all VMs on the host

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup

VmName                               VmId
CpuGroupId
------ ------------------------------------ --------------------------------
----
```

```
    G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-
000000000002
    P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-
000000000003
    P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-
000000000004
    G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-
000000000002
    G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 36ab08cb-3a76-4b38-992e-
000000000002
```

Example 7 – Unbind a VM from the CPU group

To remove a VM from a CPU group, set to VM's CpuGroupId to the NULL GUID. This unbinds the VM from the CPU group.

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:00000000-0000-0000-
0000-000000000000

C:\vm\tools>CpuGroups.exe GetVmGroup
VmName                                VmId
CpuGroupId
------ ------------------------------------ ------------------------------
----
    G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-
000000000002
    P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-
000000000003
    P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-
000000000004
    G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-
000000000002
    G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 00000000-0000-0000-0000-
000000000000
```

Example 8 – Bind a VM to an existing CPU group

Here, we'll add a VM to an existing CPU group. Note that the VM must not be bound to any existing CPU group, or setting CPU group id will fail.

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:36AB08CB-3A76-4B38-
992E-000000000001
```

Now, confirm the VM G1 is in the desired CPU group.

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup
VmName                                  VmId
CpuGroupId
------ ------------------------------------ --------------------------------
----
    G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-
000000000002
    P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-
000000000003
    P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-
000000000004
    G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-
000000000002
    G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 36AB08CB-3A76-4B38-992E-
000000000001
```

Example 9 – Print all VMs grouped by CPU group id

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms
CpuGroupId                           VmName
VmId
------------------------------------ ------ --------------------------------
----
36AB08CB-3A76-4B38-992E-000000000001     G1 F699B50F-86F2-4E48-8BA5-
EB06883C1FDC
36ab08cb-3a76-4b38-992e-000000000002     G2 4ABCFC2F-6C22-498C-BB38-
7151CE678758
36ab08cb-3a76-4b38-992e-000000000002     G3 B0F3FCD5-FECF-4A21-A4A2-
DE4102787200
36ab08cb-3a76-4b38-992e-000000000003     P1 973B9426-0711-4742-AD3B-
D8C39D6A0DEC
36ab08cb-3a76-4b38-992e-000000000004     P2 A593D93A-3A5F-48AB-8862-
A4350E3459E8
```

Example 10 – Print all VMs for a single CPU group

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms /GroupId:36ab08cb-3a76-4b38-992e-
000000000002

CpuGroupId                           VmName
VmId
```

```
---------------------------------- ------ ------------------------------
----
36ab08cb-3a76-4b38-992e-000000000002     G2 4ABCFC2F-6C22-498C-BB38-
7151CE678758
36ab08cb-3a76-4b38-992e-000000000002     G3 B0F3FCD5-FECF-4A21-A4A2-
DE4102787200
```

Example 11 – Attempting to delete a non-empty CPU Group

Only empty CPU groups—that is, CPU groups with no bound VMs—can be deleted. Attempting to delete a non-empty CPU group will fail.

Console

```
C:\vm\tools>CpuGroups.exe DeleteGroup /GroupId:36ab08cb-3a76-4b38-992e-
000000000001
(null)
Failed with error 0xc0350070
```

Example 12 – Unbind the only VM from a CPU group and delete the group

In this example, we'll use several commands to examine a CPU group, remove the single VM belonging to that group, then delete the group.

First, let's enumerate the VMs in our group.

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms /GroupId:36AB08CB-3A76-4B38-992E-
000000000001
CpuGroupId                           VmName
VmId
---------------------------------- ------ ------------------------------
----
36AB08CB-3A76-4B38-992E-000000000001     G1 F699B50F-86F2-4E48-8BA5-
EB06883C1FDC
```

We see that only a single VM, named G1, belongs to this group. Let's remove the G1 VM from our group by setting the VM's group ID to NULL.

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:00000000-0000-0000-
0000-000000000000
```

And verify our change...

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup /VmName:g1
VmName                                VmId
CpuGroupId
------ ------------------------------------ ------------------------------
----
    G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 00000000-0000-0000-0000-
000000000000
```

Now that the group is empty, we can safely delete it.

Console

```
C:\vm\tools>CpuGroups.exe DeleteGroup /GroupId:36ab08cb-3a76-4b38-992e-
000000000001
```

And confirm our group is gone.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups
CpuGroupId                           CpuCap                      LpIndexes
------------------------------------ ------ ----------------------------
36AB08CB-3A76-4B38-992E-000000000002 32768  4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003 65536  12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004 65536 24,25,26,27,28,29,30,31
```

Example 13 – Bind a VM back to its original CPU group

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:36AB08CB-3A76-4B38-
992E-000000000002

C:\vm\tools>CpuGroups.exe GetGroupVms
CpuGroupId VmName VmId
------------------------------------ ------------------------------ ------
------------------------------
36ab08cb-3a76-4b38-992e-000000000002 G2 4ABCFC2F-6C22-498C-BB38-7151CE678758
36ab08cb-3a76-4b38-992e-000000000002 G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200
```

```
36AB08CB-3A76-4B38-992E-000000000002 G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC
36ab08cb-3a76-4b38-992e-000000000003 P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC
36ab08cb-3a76-4b38-992e-000000000004 P2 A593D93A-3A5F-48AB-8862-A4350E3459E8
```

Manage Hyper-V hypervisor scheduler types

Article • 08/04/2023

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server version 1803, Windows Server version 1709, Windows 10

This article describes new modes of virtual processor scheduling logic introduced in Windows Server 2016. These modes, or *scheduler types*, determine how the Hyper-V hypervisor allocates and manages work across guest virtual processors. A Hyper-V host administrator can:

- Select hypervisor scheduler types best suited for guest virtual machines (VMs).
- Configure VMs to take advantage of scheduling logic.

Prerequisites

You must install the following updates in order to use the hypervisor scheduler features described later in this article. These updates include changes to support the new `hypervisorschedulertype` BCD option, which is necessary for host configuration.

Version	Release	Update Required	KB Article
Windows Server 2016	1607	2018.07 C	KB4338822
Windows Server 2016	1703	2018.07 C	KB4338827
Windows Server 2016	1709	2018.07 C	KB4338817
Windows Server 2019	1804	None	None

Background

Before discussing the logic and controls behind Hyper-V virtual processor scheduling, it's important to understand certain concepts like simultaneous multithreading and how Hyper-V virtualizes processors.

Understand SMT

Simultaneous multithreading (SMT) is a technique in modern processor designs that lets separate, independent execution threads share processor resources. SMT usually gives a modest performance boost to most workloads. It parallelizes computations when possible, increasing instruction throughput. However, there are also times where there's no noticeable improvement in performance or even a slight loss when threads compete with each other for shared processor resources.

In order to use SMT with Windows Server, you must have a compatible processor. For example, a processor with Intel Hyper-Threading technology or Advanced Micro Devices (AMD) Multithreading (SMT).

For the purposes of this article, the descriptions of SMT and how it's used by Hyper-V apply equally to both Intel and AMD systems.

- For more information on Intel HT Technology, see Intel Hyper-Threading Technology.
- For more information on AMD SMT, see The "Zen" Core Architecture.

Understand how Hyper-V virtualizes processors

Before considering hypervisor scheduler types, you should understand Hyper-V architecture. You can find a more detailed summary of how this architecture works in the Hyper-V technology overview, but for now, you should keep the following concepts in mind:

- Hyper-V creates and manages VM partitions, allocating and sharing compute resources across them, under control of the hypervisor. Partitions provide strong isolation boundaries between all guest VMs and between guest VMs and the root partition.
- The root partition is itself a VM partition, although it has unique properties and greater privileges than guest VMs. The root partition:
  - Provides management services that control all guest VMs.
  - Provides virtual device support for guests.
  - Manages all device input and output for guest VMs.

  We recommend not running any application workloads in the root partition.
- Each virtual processor (VP) of the root partition is mapped one-to-one to an underlying logical processor (LP). A host VP always runs on the same underlying LP. There's no migration of the root partition's VPs.
- By default, the LPs which host the root partition's VPs run can also run guest VPs.

- Hypervisor might schedule the guest VP to run on any available logical processor. While the hypervisor scheduler tries to consider temporal cache locality, non-uniform memory access (NUMA) topology, and many other factors when scheduling a guest VP, ultimately the VP can be scheduled on any host LP.

Hypervisor scheduler types

In Windows Server 2016, the Hyper-V hypervisor supports several modes of scheduler logic, which determine how the hypervisor schedules virtual processors on the underlying logical processors. These scheduler types are:

- The classic scheduler.
- The core scheduler.
- The root scheduler.

The classic scheduler

The classic scheduler has been the default for all versions of the Windows Hyper-V hypervisor since its inception, including Windows Server 2016 Hyper-V. The classic scheduler provides a fair share, preemptive, round-robin scheduling model for guest virtual processors.

The classic scheduler type is the most appropriate for most traditional Hyper-V uses, such as private clouds, hosting providers, and so on. The performance characteristics of the classic scheduler type are best optimized to support a wide range of virtualization scenarios, such as:

- Over-subscribing of VPs to LPs.
- Running many heterogeneous VMs and workloads at the same time.
- Running larger scale high-performance VMs.
- Supporting the full feature set of Hyper-V without restrictions and other scenarios.

The core scheduler

The hypervisor core scheduler is an alternative to the classic scheduler logic introduced in Windows Server 2016 and Windows 10, version 1607. The core scheduler offers a strong security boundary for guest workload isolation. It also reduces performance variability for workloads inside of VMs running on an SMT-enabled virtualization host. The core scheduler supports running both SMT and non-SMT VMs at the same time on the same SMT-enabled virtualization host.

The core scheduler:

- Uses the virtualization host's SMT topology.
- Optionally exposes SMT pairs to guest VMs.
- Schedules groups of guest virtual processors from the same VM onto groups of SMT logical processors.

This work happens symmetrically. If LPs are in groups of two, VPs are scheduled in groups of two, and a core is never shared between VMs. When you schedule the VP for a VM without SMT enabled, that VP consumes the entire core when it runs. The overall result of the core scheduler is that:

- It creates a strong security boundary exists for guest workload isolation. Guest VPs can only run on underlying physical core pairs, reducing vulnerability to side-channel snooping attacks.
- It reduces variability in throughput.
- It can potentially reduce performance. If only one VP in a group can run, only one of the instruction streams in the core launches while the other is left idle.
- The OS and applications running in the guest VM can use SMT behavior and programming interfaces (APIs) to control and distribute work across SMT threads, just like they would with a physical machine.

As of Windows Server 2019, Hyper-V uses the core scheduler by default. In earlier versions like Windows Server 2016, the scheduler was optional and the classic scheduler is the default option.

Core scheduler behavior with host SMT disabled

In some cases, you might configure the hypervisor to use the core scheduler type, but the SMT capability is disabled or isn't present on the virtualization host. In these cases, Hyper-V uses the classic scheduler behavior regardless of the hypervisor scheduler type setting.

The root scheduler

The root scheduler arrived with Windows 10, version 1803. When you enable the root scheduler type, the hypervisor gives the root partition control of work scheduling. The NT scheduler in the root partition's OS instance manages all aspects of scheduling work to system LPs.

The root scheduler addresses the unique requirements to support a utility partition and provide strong workload isolation, as used with Windows Defender Application Guard

(WDAG). In this scenario, leaving scheduling responsibilities to the root OS offers several advantages:

- You can use CPU resource controls applicable to container scenarios with the utility partition, simplifying management and deployment.
- The root OS scheduler can readily gather metrics about workload CPU use inside the container. It can use this data as input to the same scheduling policy applicable to all other workloads in the system.
- These same metrics also help attribute work done in an application container to the host system. Tracking these metrics is more difficult with traditional VM workloads, where some work on behalf of all running VMs takes place in the root partition.

Root scheduler use on client systems

Starting with Windows 10, version 1803, the root scheduler is used by default on client systems only, which means:

- You can enable the hypervisor to support virtualization-based security and WDAG workload isolation.
- It's important to properly operate future systems with heterogenous core architectures.

This configuration is the only supported hypervisor scheduler configuration for client systems. Administrators shouldn't attempt to override the default hypervisor scheduler type on Windows client systems.

Virtual Machine CPU resource controls and the root scheduler

Hyper-V's provided VM processor resource controls aren't supported when you enable the hypervisor root scheduler. The root operating system's scheduler logic manages host resources on a global basis and isn't managing a single VM's guest resources. The Hyper-V per-VM processor resource controls, such as caps, weights, and reserves, can only apply where the hypervisor directly controls VP scheduling, such as with the classic and core scheduler types.

Root scheduler use on server systems

We don't recommend using the root scheduler with Hyper-V on servers. Its performance characteristics haven't yet been fully characterized and tuned to accommodate the wide range of workloads typical of many server virtualization deployments.

Enable SMT in guest VMs

Once you configure the virtualization host's hypervisor to use the core scheduler type, you can also configure guest VMs to use SMT. Exposing the fact that VPs are hyperthreaded to a guest VM lets the scheduler in the guest operating system and workloads running in the VM detect and use the SMT topology in their own work scheduling.

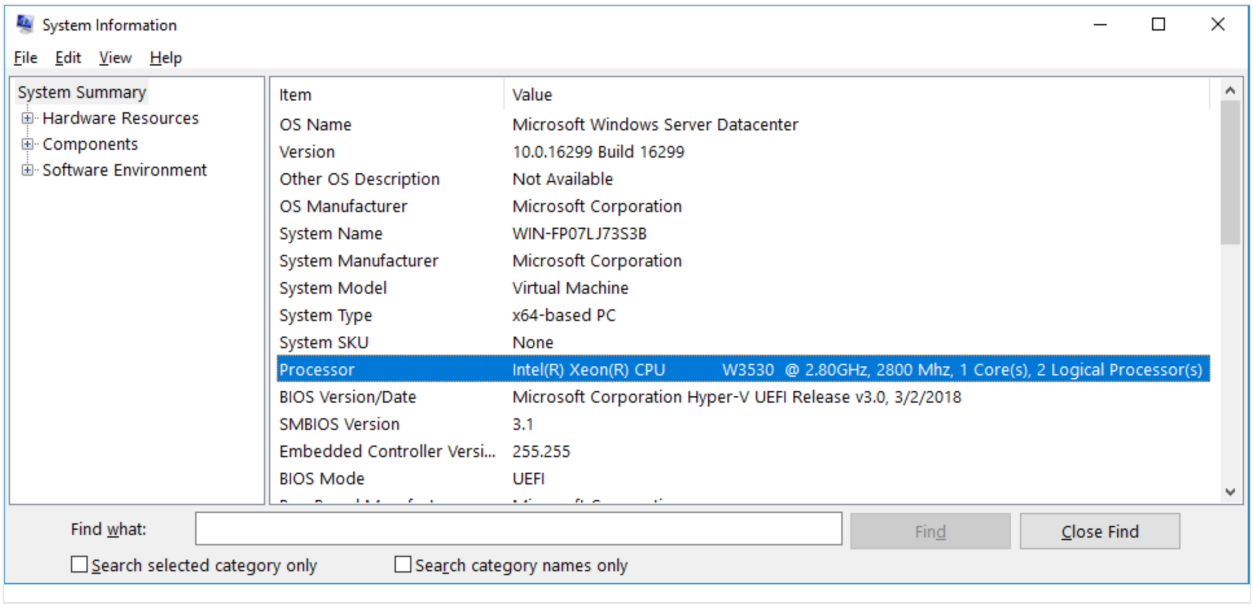
- In Windows Server 2016, guest SMT isn't configured by default. A Hyper-V host administrator must explicitly enable it.
- Starting with Windows Server 2019, new VMs you create on the host inherit the host SMT topology by default. For example, a version 9.0 VM that you create on a host with two SMT threads per core would also have two SMT threads per core.

You must use PowerShell to enable SMT in a guest VM. There's no user interface provided in Hyper-V Manager. To enable SMT in a guest VM:

1. Open a PowerShell window using an account that is a member of the Hyper-V Administrators group, or equivalent.
2. Run `Set-VMProcessor -VMName <VMName> -HwThreadCountPerCore <n>`, where `<n>` is the number of SMT threads per core the guest VM sees. `<n> = 0` sets the `HwThreadCountPerCore` value to match the host's SMT thread count per core value.

> ⓘ **Note**
>
> In Windows Server 2019 and later, you can set `HwThreadCountPerCore = 0` instead of matching the host SMT thread count.

The following screenshot shows system information taken from the guest operating system running in a VM. There are two virtual processors and SMT enabled. The guest operating system is detecting two logical processors belonging to the same core.



Configure the hypervisor scheduler type on Windows Server 2016 Hyper-V

Windows Server 2016 Hyper-V uses the classic hypervisor scheduler model by default. You can optionally configure the hypervisor to use the core scheduler. The core scheduler increases security by restricting guest VPs to run on corresponding physical SMT pairs. This configuration supports the use of VMs with SMT scheduling for their guest VPs.

> **Note**
>
> We recommend that all customers running Windows Server 2016 Hyper-V select the core scheduler to ensure their virtualization hosts are optimally protected against potentially malicious guest VMs.

Windows Server 2019 Hyper-V defaults to using the core scheduler

To ensure Hyper-V hosts are deployed in the optimal security configuration, Windows Server 2019 Hyper-V now uses the core hypervisor scheduler model by default. The host administrator might optionally configure the host to use the legacy classic scheduler. Prior to overriding the default settings, administrators should carefully read, understand, and consider the impacts each scheduler type has on the security and performance of virtualization hosts. For more information, see About Hyper-V hypervisor scheduler type selection.

Select the hypervisor scheduler type on Windows Server

The hypervisor scheduler configuration is controlled by the `hypervisorschedulertype` BCD entry.

To select a scheduler type:

1. Open a command prompt with administrator privileges.
2. Enter `bcdedit /set hypervisorschedulertype type`, where `type` is one of these options:

- `Classic`
- `Core`
- `Root`

You must reboot the system for any changes you make to the hypervisor scheduler type to take effect.

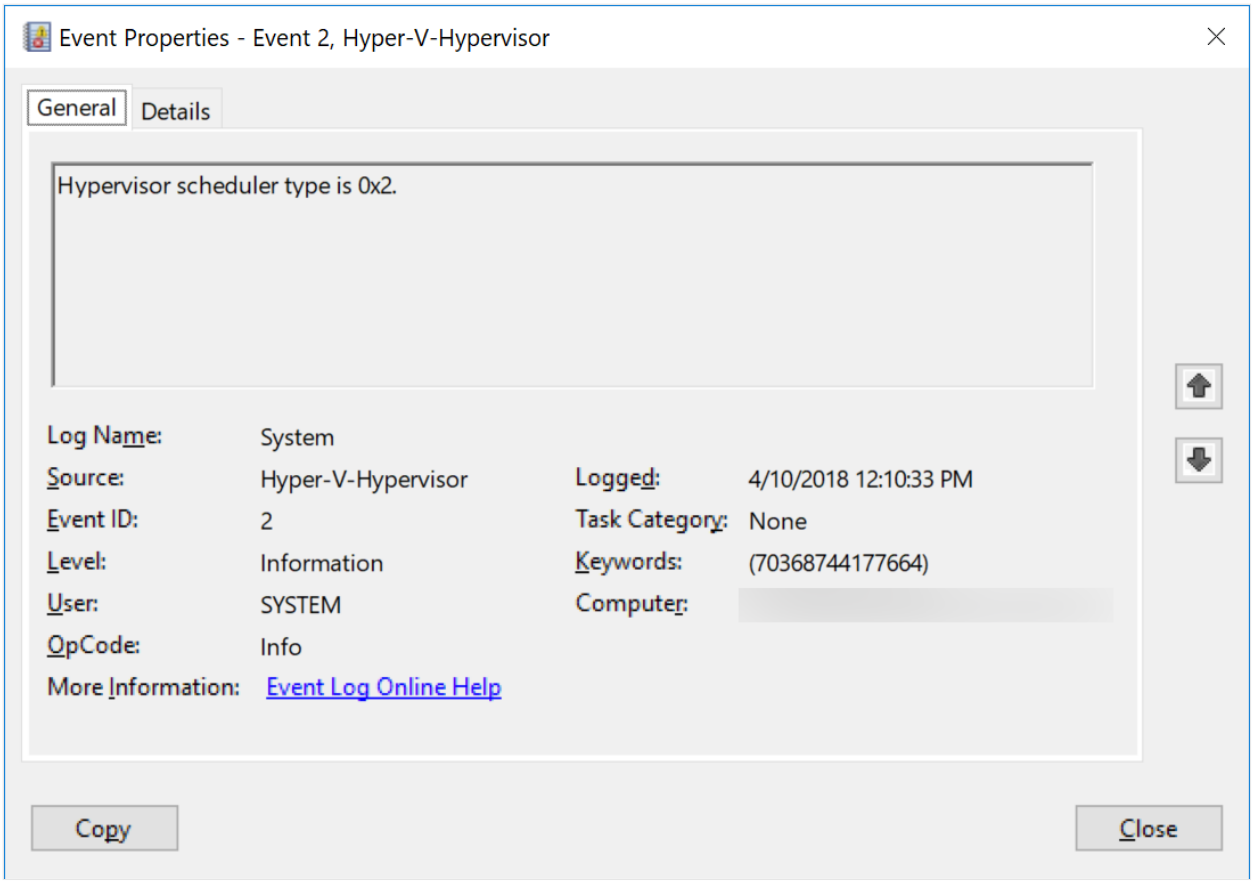
> **Note**
>
> The hypervisor root scheduler isn't supported on Windows Server Hyper-V at this time. Hyper-V administrators shouldn't attempt to configure the root scheduler for use with server virtualization scenarios.

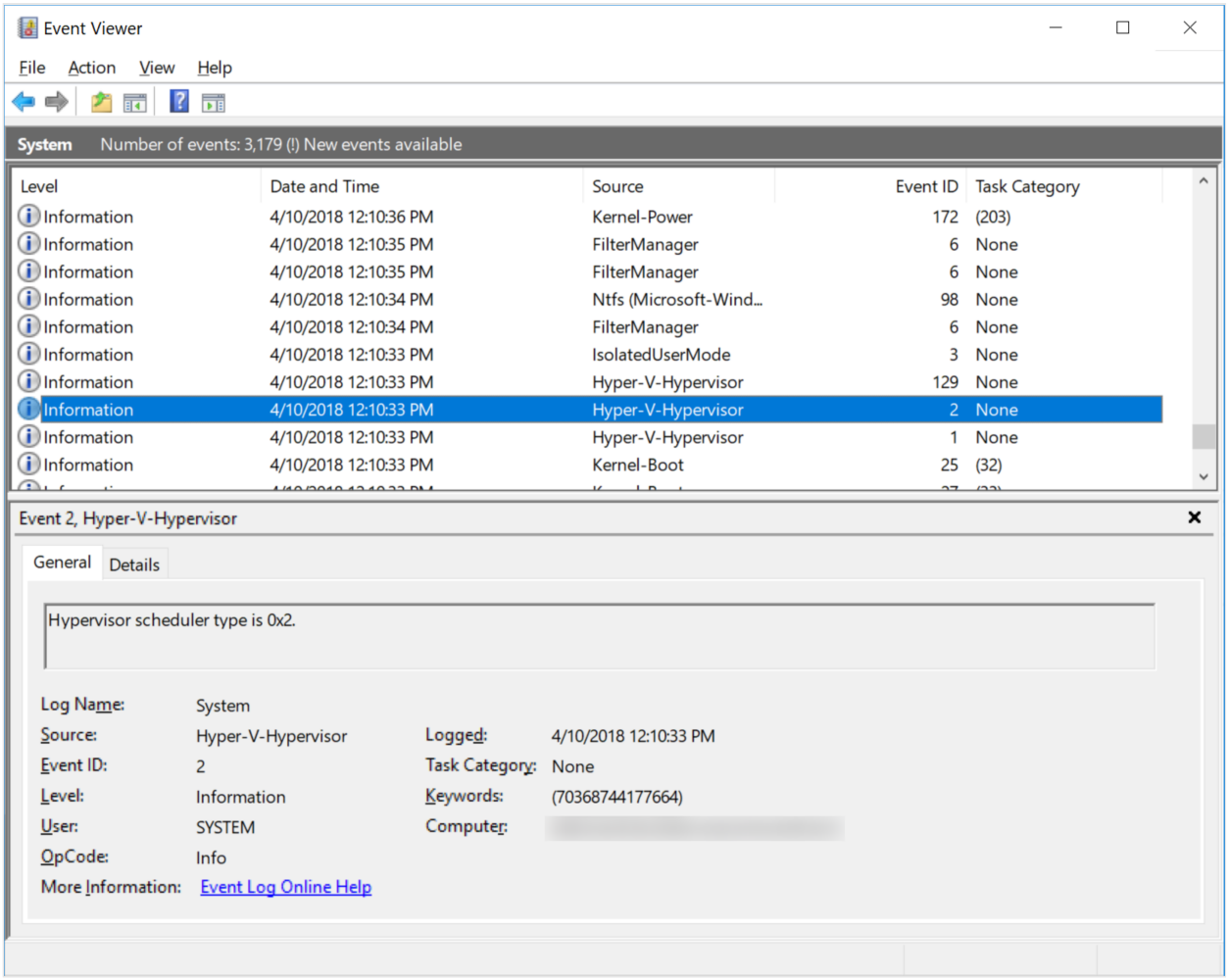
Determine the current scheduler type

You can determine which hypervisor scheduler type Hyper-V is currently using by examining the Event Viewer system log. You can see the most recent hypervisor launch event ID 2, which reports the hypervisor scheduler type configured at hypervisor launch. You can get the hypervisor launch events from the Windows Event Viewer or in PowerShell.

Hypervisor launch event ID 2 denotes the hypervisor scheduler type, where:

- 1 = Classic scheduler, SMT disabled
- 2 = Classic scheduler
- 3 = Core scheduler
- 4 = Root scheduler


Event Properties - Event 2, Hyper-V-Hypervisor
General
Details
Hypervisor scheduler type is 0x2.
Log Name: System
Source: Hyper-V-Hypervisor
Logged: 4/10/2018 12:10:33 PM
Event ID: 2
Task Category: None
Level: Information
Keywords: (70368744177664)
User: SYSTEM
Computer:
OpCode: Info
More Information: Event Log Online Help
Copy
Close

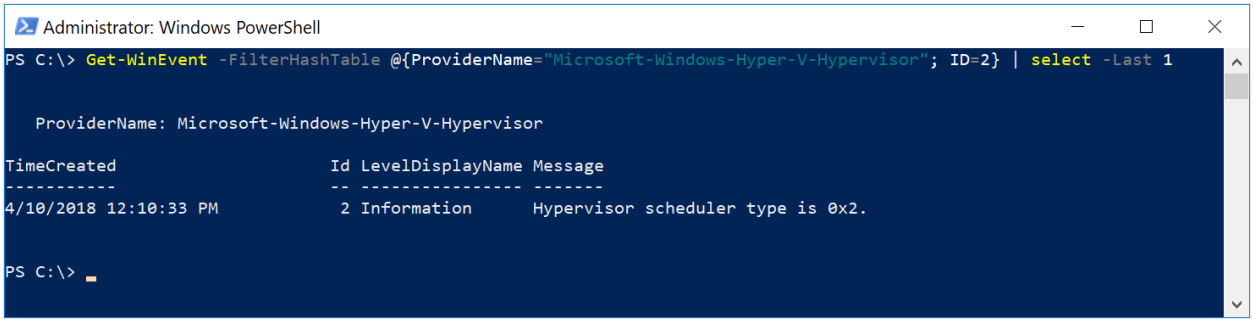

Event Viewer
File Action View Help
System Number of events: 3,179 (!) New events available
Level Date and Time Source Event ID Task Category
Information 4/10/2018 12:10:36 PM Kernel-Power 172 (203)
Information 4/10/2018 12:10:35 PM FilterManager 6 None
Information 4/10/2018 12:10:35 PM FilterManager 6 None
Information 4/10/2018 12:10:34 PM Ntfs (Microsoft-Wind... 98 None
Information 4/10/2018 12:10:34 PM FilterManager 6 None
Information 4/10/2018 12:10:33 PM IsolatedUserMode 3 None
Information 4/10/2018 12:10:33 PM Hyper-V-Hypervisor 129 None
Information 4/10/2018 12:10:33 PM Hyper-V-Hypervisor 2 None
Information 4/10/2018 12:10:33 PM Hyper-V-Hypervisor 1 None
Information 4/10/2018 12:10:33 PM Kernel-Boot 25 (32)
Event 2, Hyper-V-Hypervisor
General
Details
Hypervisor scheduler type is 0x2.
Log Name: System
Source: Hyper-V-Hypervisor
Logged: 4/10/2018 12:10:33 PM
Event ID: 2
Task Category: None
Level: Information
Keywords: (70368744177664)
User: SYSTEM
Computer:
OpCode: Info
More Information: Event Log Online Help

Query the Hyper-V hypervisor scheduler type launch event using PowerShell

To query for hypervisor event ID 2 using PowerShell, run the following commands from a PowerShell prompt:

PowerShell

```
Get-WinEvent -FilterHashTable @{ProviderName="Microsoft-Windows-Hyper-V-Hypervisor"; ID=2} -MaxEvents 1
```



About Hyper-V hypervisor scheduler type selection

Article • 12/23/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server, version 1709, Windows Server, version 1803

This document describes important changes to Hyper-V's default and recommended use of hypervisor scheduler types. These changes impact both system security and virtualization performance. Virtualization host administrators should review and understand the changes and implications described in this document, and carefully evaluate the impacts, suggested deployment guidance and risk factors involved to best understand how to deploy and manage Hyper-V hosts in the face of the rapidly changing security landscape.

ⓘ Important

Currently known side-channel security vulnerabilities sighted in multiple processor architectures could be exploited by a malicious guest VM through the scheduling behavior of the legacy hypervisor classic scheduler type when run on hosts with Simultaneous Multithreading (SMT) enabled. If successfully exploited, a malicious workload could observe data outside its partition boundary. This class of attacks can be mitigated by configuring the Hyper-V hypervisor to utilize the hypervisor core scheduler type and reconfiguring guest VMs. With the core scheduler, the hypervisor restricts a guest VM's VPs to run on the same physical processor core, therefore strongly isolating the VM's ability to access data to the boundaries of the physical core on which it runs. This is a highly effective mitigation against these side-channel attacks, which prevents the VM from observing any artifacts from other partitions, whether the root or another guest partition. Therefore, Microsoft is changing the default and recommended configuration settings for virtualization hosts and guest VMs.

Background

Starting with Windows Server 2016, Hyper-V supports several methods of scheduling and managing virtual processors, referred to as hypervisor scheduler types. A detailed

description of all hypervisor scheduler types can be found in Understanding and using Hyper-V hypervisor scheduler types.

> **Note**
>
> New hypervisor scheduler types were first introduced with Windows Server 2016, and are not available in prior releases. All versions of Hyper-V prior to Windows Server 2016 support only the classic scheduler. Support for the core scheduler was only recently published.

About hypervisor scheduler types

This article focuses specifically on the use of the new hypervisor core scheduler type versus the legacy "classic" scheduler, and how these scheduler types intersect with the use of Symmetric Multi-Threading, or SMT. It is important to understand the differences between the core and classic schedulers and how each places work from guest VMs on the underlying system processors.

The classic scheduler

The classic scheduler refers to the fair-share, round robin method of scheduling work on virtual processors (VPs) across the system - including root VPs as well as VPs belonging to guest VMs. The classic scheduler has been the default scheduler type used on all versions of Hyper-V (until Windows Server 2019, as described herein). The performance characteristics of the classic scheduler are well understood, and the classic scheduler is demonstrated to ably support over-subscription of workloads - that is, over-subscription of the host's VP:LP ratio by a reasonable margin (depending on the types of workloads being virtualized, overall resource utilization, etc.).

When run on a virtualization host with SMT enabled, the classic scheduler will schedule guest VPs from any VM on each SMT thread belonging to a core independently. Therefore, different VMs can be running on the same core at the same time (one VM running on one thread of a core while another VM is running on the other thread).

The core scheduler

The core scheduler leverages the properties of SMT to provide isolation of guest workloads, which impacts both security and system performance. The core scheduler ensures that VPs from a VM are scheduled on sibling SMT threads. This is done

symmetrically so that if LPs are in groups of two, VPs are scheduled in groups of two, and a system CPU core is never shared between VMs.

By scheduling guest VPs on underlying SMT pairs, the core scheduler offers a strong security boundary for workload isolation, and can also be used to reduce performance variability for latency sensitive workloads.

Note that when the VP is scheduled for a virtual machine without SMT enabled, that VP will consume the entire core when it runs, and the core's sibling SMT thread will be left idle. This is necessary to provide the correct workload isolation, but impacts overall system performance, especially as the system LPs become over-subscribed - that is, when total VP:LP ratio exceeds 1:1. Therefore, running guest VMs configured without multiple threads per core is a sub-optimal configuration.

Benefits of the using the core scheduler

The core scheduler offers the following benefits:

- A strong security boundary for guest workload isolation - Guest VPs are constrained to run on underlying physical core pairs, reducing vulnerability to side-channel snooping attacks.
- Reduced workload variability - Guest workload throughput variability is significantly reduced, offering greater workload consistency.
- Use of SMT in guest VMs - The OS and applications running in the guest virtual machine can utilize SMT behavior and programming interfaces (APIs) to control and distribute work across SMT threads, just as they would when run non-virtualized.

The core scheduler is currently used on Azure virtualization hosts, specifically to take advantage of the strong security boundary and low workload variabilty. Microsoft believes that the core scheduler type should be and will continue to be the default hypervisor scheduling type for the majority of virtualization scenarios. Therefore, to ensure our customers are secure by default, Microsoft is making this change for Windows Server 2019 now.

Core scheduler performance impacts on guest workloads

While required to effectively mitigate certain classes of vulnerabilities, the core scheduler may also potentially reduce performance. Customers may see a difference in the performance characteristics with their VMs and impacts to the overall workload capacity of their virtualization hosts. In cases where the core scheduler must run a non-

SMT VP, only one of the instruction streams in the underlying logical core executes while the other must be left idle. This will limit the total host capacity for guest workloads.

These performance impacts can be minimized by following the deployment guidance in this document. Host administrators must carefully consider their specific virtualization deployment scenarios and balance their tolerance for security risk against the need for maximum workload density, over-consolidation of virtualization hosts, etc.

Changes to the default and recommended configurations for Windows Server 2016 and Windows Server 2019

Deploying Hyper-V hosts with the maximum security posture requires use of the hypervisor core scheduler type. To ensure our customers are secure by default, Microsoft is changing the following default and recommended settings.

> ⓘ **Note**
>
> While the hypervisor's internal support for the scheduler types was included in the initial release of Windows Server 2016, Windows Server 1709, and Windows Server 1803, updates are required in order to access the configuration control which allows selecting the hypervisor scheduler type. Please refer to **Understanding and using Hyper-V hypervisor scheduler types** for details on these updates.

Virtualization host changes

- The hypervisor will use the core scheduler by default beginning with Windows Server 2019.
- Microsoft reccommends configuring the core scheduler on Windows Server 2016. The hypervisor core scheduler type is supported in Windows Server 2016, however the default is the classic scheduler. The core scheduler is optional and must be explicitly enabled by the Hyper-V host administrator.

Virtual machine configuration changes

- On Windows Server 2019, new virtual machines created using the default VM version 9.0 will automatically inherit the SMT properties (enabled or disabled) of

the virtualization host. That is, if SMT is enabled on the physical host, newly created VMs will also have SMT enabled, and will inherit the SMT topology of the host by default, with the VM having the same number of hardware threads per core as the underlying system. This will be reflected in the VM's configuration with HwThreadCountPerCore = 0, where 0 indicates the VM should inherit the host's SMT settings.

- Existing virtual machines with a VM version of 8.2 or earlier will retain their original VM processor setting for HwThreadCountPerCore, and the default for 8.2 VM version guests is HwThreadCountPerCore = 1. When these guests run on a Windows Server 2019 host, they will be treated as follows:

    1. If the VM has a VP count that is less than or equal to the count of LP cores, the VM will be treated as a non-SMT VM by the core scheduler. When the guest VP runs on a single SMT thread, the core's sibling SMT thread will be idled. This is non-optimal, and will result in overall loss of performance.

    2. If the VM has more VPs than LP cores, the VM will be treated as an SMT VM by the core scheduler. However, the VM will not observe other indications that it is an SMT VM. For example, use of the CPUID instruction or Windows APIs to query CPU toplogy by the OS or applications will not indicate that SMT is enabled.

- When an existing VM is explicitly updated from eariler VM versions to version 9.0 through the Update-VM operation, the VM will retain its current value for HwThreadCountPerCore. The VM will not have SMT force-enabled.

- On Windows Server 2016, Microsoft recommends enabling SMT for guest VMs. By default, VMs created on Windows Server 2016 would have SMT disabled, that is HwThreadCountPerCore is set to 1, unless explicitly changed.

> ⓘ **Note**
>
> Windows Server 2016 does not support setting HwThreadCountPerCore to 0.

Managing virtual machine SMT configuration

The guest virtual machine SMT configuration is set on a per-VM basis. The host administrator can inspect and configure a VM's SMT configuration to select from the following options:

1. Configure VMs to run as SMT-enabled, optionally inheriting the host SMT topology automatically
2. Configure VMs to run as non-SMT

The SMT configuaration for a VM is displayed in the Summary panes in the Hyper-V Manager console. Configuring a VM's SMT settings may be done by using the VM Settings or PowerShell.

Configuring VM SMT settings using PowerShell

To configure the SMT settings for a guest virtual machine, open a PowerShell window with sufficient permissions, and type:

```
PowerShell

Set-VMProcessor -VMName <VMName> -HwThreadCountPerCore <0, 1, 2>
```

Where:

- 0 = Inherit SMT topology from the host (this setting of HwThreadCountPerCore=0 is not supported on Windows Server 2016)
- 1 = Non-SMT
- Values > 1 = the desired number of SMT threads per core. May not exceed the number of physical SMT threads per core.

To read the SMT settings for a guest virtual machine, open a PowerShell window with sufficient permissions, and type:

```
PowerShell

(Get-VMProcessor -VMName <VMName>).HwThreadCountPerCore
```

Note that guest VMs configured with HwThreadCountPerCore = 0 indicates that SMT will be enabled for the guest, and will expose the same number of SMT threads to the guest as they are on the underlying virtualization host, typically 2.

Guest VMs may observe changes to CPU topology across VM mobility scenarios

The OS and applications in a VM may see changes to both host and VM settings before and after VM lifecycle events such as live migration or save and restore operations. During an operation in which VM state is saved and restored, both the VM's HwThreadCountPerCore setting and the realized value (that is, the computed combination of the VM's setting and source host's configuration) are migrated. The VM will continue running with these settings on the destination host. At the point the VM is shutdown and re-started, it's possible that the realized value observed by the VM will change. This should be benign, as OS and application layer software should look for CPU topology information as part of their normal startup and initialization code flows. However, because these boot time initialization sequences are skipped during live migration or save/restore operations, VMs that undergo these state transitions could observe the originally computed realized value until they are shut down and re-started.

Alerts regarding non-optimal VM configurations

Virtual machines configured with more VPs than there are physical cores on the host result in a non-optimal configuration. The hypervisor scheduler will treat these VMs as if they are SMT-aware. However, OS and application software in the VM will be presented a CPU topology showing SMT is disabled. When this condition is detected, the Hyper-V Worker Process will log an event on the virtualization host warning the host administrator that the VM's configuration is non-optimal, and recommending SMT be enabled for the VM.

How to identify non-optimally configured VMs

You can identify non-SMT VMs by examining the System Log in Event Viewer for Hyper-V Worker Process event ID 3498, which will be triggered for a VM whenever the number of VPs in the VM is greater than the physical core count. Worker process events can be obtained from Event Viewer, or via PowerShell.

Querying the Hyper-V worker process VM event using PowerShell

To query for Hyper-V worker process event ID 3498 using PowerShell, enter the following commands from a PowerShell prompt.

PowerShell

```
Get-WinEvent -FilterHashTable @{ProviderName="Microsoft-Windows-Hyper-V-Worker"; ID=3498}
```

Impacts of guest SMT configuaration on the use of hypervisor enlightenments for guest operating systems

The Microsoft hypervisor offers multiple enlightenments, or hints, which the OS running in a guest VM may query and use to trigger optimizations, such as those that might benefit performance or otherwise improve handling of various conditions when running virtualized. One recently introduced enlightenment concerns the handling of virtual processor scheduling and the use of OS mitigations for side-channel attacks that exploit SMT.

> ⓘ **Note**
>
> Microsoft recommends that host administrators enable SMT for guest VMs to optimize workload performance.

The details of this guest enlightenment are provided below, however the key takeaway for virtualization host administrators is that virtual machines should have HwThreadCountPerCore configured to match the host's physical SMT configuration. This allows the hypervisor to report that there is no non-architectural core sharing. Therefore, any guest OS supporting optimizations that require the enlightenment may be enabled. On Windows Server 2019, create new VMs and leave the default value of HwThreadCountPerCore (0). Older VMs migrated from Windows Server 2016 hosts can be updated to the Windows Server 2019 configuration version. After doing so, Microsoft recommends setting HwThreadCountPerCore = 0. On Windows Server 2016, Microsoft recommends setting HwThreadCountPerCore to match the host configuration (typically 2).

NoNonArchitecturalCoreSharing enlightenment details

Starting in Windows Server 2016, the hypervisor defines a new enlightenment to describe its handling of VP scheduling and placement to the guest OS. This enlightenment is defined in the Hypervisor Top Level Functional Specification v5.0c.

Hypervisor synthetic CPUID leaf CPUID.0x40000004.EAX:18[NoNonArchitecturalCoreSharing = 1] indicates that a virtual processor will never share a physical core with another virtual processor, except for virtual processors that are reported as sibling SMT threads. For example, a guest VP will never run on an SMT thread alongside a root VP running simultaneously on a sibling SMT thread on the same processor core. This condition is only possible when running virtualized, and so represents a non-architectural SMT behavior that also has serious

security implications. The guest OS can use NoNonArchitecturalCoreSharing = 1 as an indication that it is safe to enable optimizations, which may help it avoid the performance overhead of setting STIBP.

In certain configurations, the hypervisor will not indicate that NoNonArchitecturalCoreSharing = 1. As an example, if a host has SMT enabled and is configured to use the hypervisor classic scheduler, NoNonArchitecturalCoreSharing will be 0. This may prevent enlightened guests from enabling certain optimizations. Therefore, Microsoft recommends that host administrators using SMT rely on the hypervisor core scheduler and ensure that virtual machines are configured to inherit their SMT configuration from the host to ensure optimal workload performance.

Summary

The security threat landscape continues to evolve. To ensure our customers are secure by default, Microsoft is changing the default configuration for the hypervisor and virtual machines starting in Windows Server 2019 Hyper-V, and providing updated guidance and recommendations for customers running Windows Server 2016 Hyper-V. Virtualization host administrators should:

- Read and understand the guidance provided in this document
- Carefully evaluate and adjust their virtualization deployments to ensure they meet the security, performance, virtualization density, and workload responsiveness goals for their unique requirements
- Consider re-configuring existing Windows Server 2016 Hyper-V hosts to leverage the strong security benefits offered by the hypervisor core scheduler
- Update existing non-SMT VMs to reduce the performance impacts from scheduling constraints imposed by VP isolation that addresses hardware security vulnerabilities

Manage Hyper-V Integration Services

Article • 01/18/2022

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows 11. Windows 10

Hyper-V Integration Services enhance virtual machine performance and provide convenience features by leveraging two-way communication with the Hyper-V host. Many of these services are conveniences, such as guest file copy, while others are important to the virtual machine's functionality, such as synthetic device drivers. This set of services and drivers are sometimes referred to as *integration components*. You can control whether or not individual convenience services operate for any given virtual machine. The driver components are not intended to be serviced manually.

For details about each integration service, see Hyper-V Integration Services.

> **ⓘ Important**
>
> Each service you want to use must be enabled in both the host and guest in order to function. All integration services except *Hyper-V Guest Service Interface* are on by default on Windows guest operating systems. The services can be turned on and off individually. The next sections show you how.

Turn an integration service on or off using Hyper-V Manager

1. From the center pane, right-click the virtual machine and select **Settings**.
2. From the left pane of the **Settings** window, under **Management**, select **Integration Services**.

The Integration Services pane lists all integration services available on the Hyper-V host, and whether the host has enabled the virtual machine to use them.

Turn an integration service on or off using PowerShell

To do this in PowerShell, use Enable-VMIntegrationService and Disable-VMIntegrationService.

The following examples demonstrate turning the guest file copy integration service on and off for a virtual machine named *DemoVM*.

1. Get a list of running integration services:

```PowerShell
Get-VMIntegrationService -VMName "DemoVM"
```

2. The output should look like this:

```PowerShell
VMName      Name                    Enabled PrimaryStatusDescription
SecondaryStatusDescription
------      ----                    ------- ----------------------- --
------------------------
DemoVM      Guest Service Interface False   OK
DemoVM      Heartbeat               True    OK                      OK
DemoVM      Key-Value Pair Exchange True    OK
DemoVM      Shutdown                True    OK
DemoVM      Time Synchronization    True    OK
DemoVM      VSS                     True    OK
```

3. Turn on Guest Service Interface:

```PowerShell
Enable-VMIntegrationService -VMName "DemoVM" -Name "Guest Service
Interface"
```

4. Verify that Guest Service Interface is enabled:

```PowerShell
Get-VMIntegrationService -VMName "DemoVM"
```

5. Turn off Guest Service Interface:

```PowerShell
Disable-VMIntegrationService -VMName "DemoVM" -Name "Guest Service
Interface"
```

Checking the guest's integration services version

Some features may not work correctly or at all if the guest's integration services are not current. To get the version information for Windows, sign in to the guest operating system, open a command prompt, and run this command:

```
REG QUERY "HKLM\Software\Microsoft\Virtual Machine\Auto" /v
IntegrationServicesVersion
```

Earlier guest operating systems will not have all available services. For example, Windows Server 2008 R2 guests cannot have the Hyper-V Guest Service Interface.

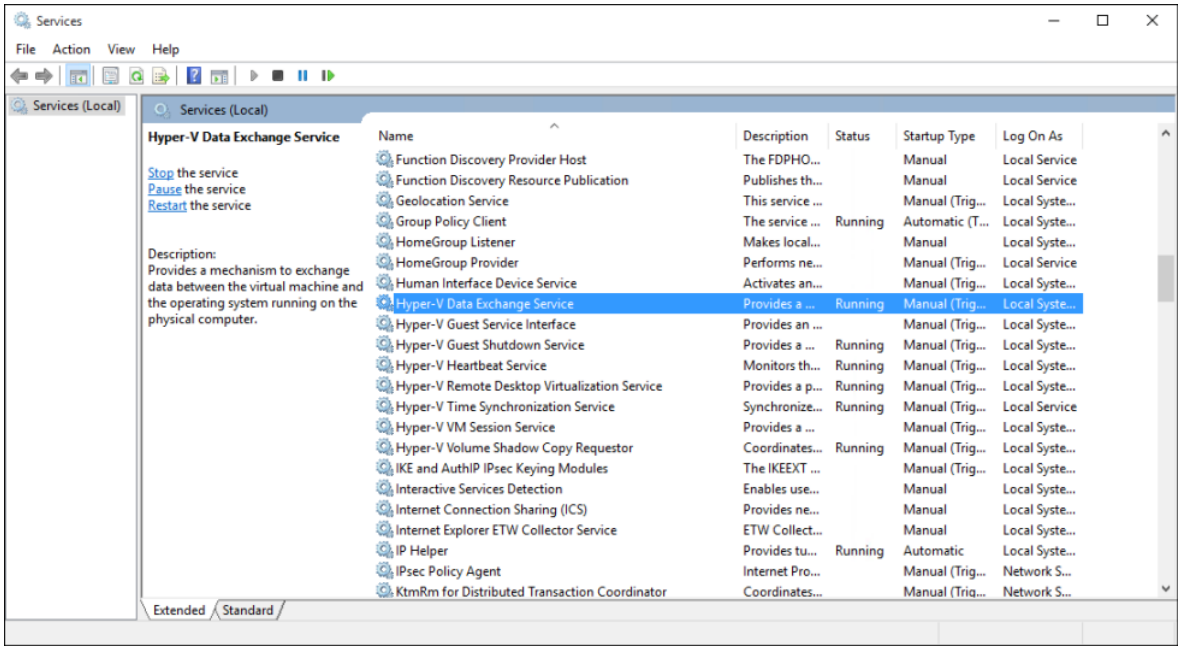
Start and stop an integration service from a Windows guest

In order for an integration service to be fully functional, its corresponding service must be running within the guest in addition to being enabled on the host. In Windows guests, each integration service is listed as a standard Windows service. You can use the Services applet in Control Panel or PowerShell to stop and start these services.

> **ⓘ Important**
>
> Stopping an integration service may severely affect the host's ability to manage your virtual machine. To work correctly, each integration service you want to use must be enabled on both the host and guest. As a best practice, you should only control integration services from Hyper-V using the instructions above. The matching service in the guest operating system will stop or start automatically when you change its status in Hyper-V. If you start a service in the guest operating system but it is disabled in Hyper-V, the service will stop. If you stop a service in the guest operating system that is enabled in Hyper-V, Hyper-V will eventually start it again. If you disable the service in the guest, Hyper-V will be unable to start it.

Use Windows Services to start or stop an integration service within a Windows guest

1. Open Services manager by running `services.msc` as an Administrator or by double-clicking the Services icon in Control Panel.



2. Find the services that start with **Hyper-V**.

3. Right-click the service you want start or stop. Select the desired action.

Use PowerShell to start or stop an integration service within a Windows guest

1. To get a list of integration services, run:

```PowerShell
Get-Service -Name vmic* | FT -AutoSize
```

2. The output should look similar to this:

```PowerShell
Status  Name               DisplayName
------  ----               -----------
Running vmicguestinterface Hyper-V Guest Service Interface
Running vmicheartbeat      Hyper-V Heartbeat Service
Running vmickvpexchange    Hyper-V Data Exchange Service
Running vmicrdv            Hyper-V Remote Desktop Virtualization
Service
Running vmicshutdown       Hyper-V Guest Shutdown Service
Running vmictimesync       Hyper-V Time Synchronization Service
Stopped vmicvmsession      Hyper-V PowerShell Direct Service
Running vmicvss            Hyper-V Volume Shadow Copy Requestor
```

3. Run either Start-Service or Stop-Service. For example, to turn off Windows PowerShell Direct, run:

```PowerShell
Stop-Service -Name vmicvmsession
```

Start and stop an integration service from a Linux guest

Linux integration services are generally provided through the Linux kernel. The Linux integration services driver is named **hv_utils**.

1. To find out if **hv_utils** is loaded, use this command:

```Bash
lsmod | grep hv_utils
```

2. The output should look similar to this:

```Bash
Module                  Size   Used by
hv_utils               20480   0
hv_vmbus               61440   8
hv_balloon,hyperv_keyboard,hv_netvsc,hid_hyperv,hv_utils,hyperv_fb,hv_s
torvsc
```

3. To find out if the required daemons are running, use this command.

```Bash
ps -ef | grep hv
```

4. The output should look similar to this:

```Bash
root       236     2  0 Jul11 ?        00:00:00 [hv_vmbus_con]
root       237     2  0 Jul11 ?        00:00:00 [hv_vmbus_ctl]
...
root       252     2  0 Jul11 ?        00:00:00 [hv_vmbus_ctl]
```

```
root      1286     1  0 Jul11 ?        00:01:11 /usr/lib/linux-
tools/3.13.0-32-generic/hv_kvp_daemon
root      9333     1  0 Oct12 ?        00:00:00 /usr/lib/linux-
tools/3.13.0-32-generic/hv_kvp_daemon
root      9365     1  0 Oct12 ?        00:00:00 /usr/lib/linux-
tools/3.13.0-32-generic/hv_vss_daemon
user     43774 43755  0 21:20 pts/0    00:00:00 grep --color=auto hv
```

5. To see what daemons are available, run:

```
Bash

compgen -c hv_
```

6. The output should look similar to this:

```
Bash

hv_vss_daemon
hv_get_dhcp_info
hv_get_dns_info
hv_set_ifconfig
hv_kvp_daemon
hv_fcopy_daemon
```

   Integration service daemons that might be listed include the following. If any are missing, they might not be supported on your system or they might not be installed. Find details, see Supported Linux and FreeBSD virtual machines for Hyper-V on Windows.

   - **hv_vss_daemon**: This daemon is required to create live Linux virtual machine backups.
   - **hv_kvp_daemon**: This daemon allows setting and querying intrinsic and extrinsic key value pairs.
   - **hv_fcopy_daemon**: This daemon implements a file copying service between the host and guest.

Examples

These examples demonstrate stopping and starting the KVP daemon, named `hv_kvp_daemon`.

1. Use the process ID (PID) to stop the daemon's process. To find the PID, look at the second column of the output, or use `pidof`. Hyper-V daemons run as root, so you'll need root permissions.

```
Bash

sudo kill -15 `pidof hv_kvp_daemon`
```

2. To verify that all `hv_kvp_daemon` processes are gone, run:

```
Bash

ps -ef | hv
```

3. To start the daemon again, run the daemon as root:

```
Bash

sudo hv_kvp_daemon
```

4. To verify that the `hv_kvp_daemon` process is listed with a new process ID, run:

```
Bash

ps -ef | hv
```

Keep integration services up to date

We recommend that you keep integration services up to date to get the best performance and most recent features for your virtual machines. This happens for Windows guests by default if they are set up to get important updates from Windows Update. Linux guests using current kernels contain integration services built in, but there may be optional updates available. You will receive the latest integration components when you update the kernel. For more information about Linux guests, see Supported Linux and FreeBSD virtual machines for Hyper-V on Windows.

> **Note**
>
> The image file *Integration Services disk* (vmguest.iso) isn't included with Hyper-V starting with Windows Server 2016 and Windows 10 because it's no longer needed. Windows Server 2012 and older require the Data Exchange integration service. If the Data Exchange integration service can't be enabled, integration services for these guests are available from the **Download Center** as a cabinet (cab) file. Instructions for applying a cab are available in this **Microsoft TechCommunity blog**

post. If your Hyper-V host is running Windows Server 2012 R2 and older, see the next section for how to install or update integration services.

Install or update integration services for Hyper-V hosts earlier than Windows Server 2016 and Windows 10

> **Note**
>
> This isn't required for Windows Server 2016 and Windows 10 or newer.

For Hyper-V hosts earlier than Windows Server 2016 and Windows 10, you'll need to **manually install or update** the integration services in the guest operating systems.

To manually install or update the integration services:

1. Open Hyper-V Manager.
2. Connect to the virtual machine. Right-click the virtual machine and select **Connect**.
3. From the Action menu of Virtual Machine Connection, select **Insert Integration Services Setup Disk**. This action loads the setup disk in the virtual DVD drive. Depending on the guest operating system, you might need to start the installation manually from File Explorer.
4. After the installation finishes, integration services are available for use.

Manage Windows virtual machines with PowerShell Direct

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Windows Server 2019

You can use PowerShell Direct to remotely manage a Windows 10, Windows Server 2016, or Windows Server 2019 virtual machine from a Windows 10, Windows Server 2016, or Windows Server 2019 Hyper-V host. PowerShell Direct allows Windows PowerShell management inside a virtual machine regardless of the network configuration or remote management settings on either the Hyper-V host or the virtual machine. This makes it easier for Hyper-V Administrators to automate and script virtual machine management and configuration.

There are two ways to run PowerShell Direct:

- Create and exit a PowerShell Direct session using PSSession cmdlets
- Run script or command with the Invoke-Command cmdlet

If you're managing older virtual machines, use Virtual Machine Connection (VMConnect) or configure a virtual network for the virtual machine.

Create and exit a PowerShell Direct session using PSSession cmdlets

1. On the Hyper-V host, open Windows PowerShell as Administrator.
2. Use the Enter-PSSession cmdlet to connect to the virtual machine. Run one of the following commands to create a session by using the virtual machine name or GUID:

```
Enter-PSSession -VMName <VMName>
```

```
Enter-PSSession -VMGUID <VMGUID>
```

3. Type your credentials for the virtual machine.
4. Run whatever commands you need to. These commands run on the virtual machine that you created the session with.
5. When you're done, use the Exit-PSSession to close the session.

```
Exit-PSSession
```

Run script or command with Invoke-Command cmdlet

You can use the Invoke-Command cmdlet to run a pre-determined set of commands on the virtual machine. Here is an example of how you can use the Invoke-Command cmdlet where PSTest is the virtual machine name and the script to run (foo.ps1) is in the script folder on the C:/ drive:

```
Invoke-Command -VMName PSTest  -FilePath C:\script\foo.ps1
```

To run a single command, use the **-ScriptBlock** parameter:

```
Invoke-Command -VMName PSTest  -ScriptBlock { cmdlet }
```

What's required to use PowerShell Direct?

To create a PowerShell Direct session on a virtual machine,

- The virtual machine must be running locally on the host and booted.
- You must be logged into the host computer as a Hyper-V administrator.
- You must supply valid user credentials for the virtual machine.
- The host operating system must run at least Windows 10 or Windows Server 2016.
- The virtual machine must run at least Windows 10 or Windows Server 2016.

You can use the Get-VM cmdlet to check that the credentials you're using have the Hyper-V administrator role and to get a list of the virtual machines running locally on the host and booted.

See Also

Enter-PSSession Exit-PSSession Invoke-Command

Set up Hyper-V Replica

Article • 08/28/2024

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016; Azure Stack HCI, version 23H2 (only for Hyper-V virtual machines) and Azure Stack HCI, version 22H2

Hyper-V Replica is an integral part of the Hyper-V role. It contributes to your disaster recovery strategy by replicating virtual machines from one Hyper-V host server to another to keep your workloads available. Hyper-V Replica creates a copy of a live virtual machine to a replica offline virtual machine. Note the following:

- **Hyper-V hosts**: Primary and secondary host servers can be physically co-located or in separate geographical locations with replication over a WAN link. Hyper-V hosts can be standalone, clustered, or a mixture of both. There's no Active Directory dependency between the servers and they don't need to be domain members.
- **Replication and change tracking**: When you enable Hyper-V Replica for a specific virtual machine, initial replication creates an identical replica virtual machine on a secondary host server. After that happens, Hyper-V Replica change tracking creates and maintains a log file that captures changes on a virtual machine VHD. The log file is played in reverse order to the replica VHD based on replication frequency settings. This means that the latest changes are stored and replicated asynchronously. Replication can be over HTTP or HTTPS.
- **Extended (chained) replication**: This lets you replicate a virtual machine from a primary host to a secondary host, and then replicate the secondary host to a third host. Note that you can't replicate from the primary host directly to the second and the third.

  This feature makes Hyper-V Replica more robust for disaster recovery because if an outage occurs you can recover from both the primary and extended replica. You can fail over to the extended replica if your primary and secondary locations go down. Note that the extended replica doesn't support application-consistent replication and must use the same VHDs that the secondary replica is using.
- **Failover**: If an outage occurs in your primary (or secondary if extended) location, you can manually initiate a test, planned, or unplanned failover.

Expand table

Question	Test	Planned	Unplanned
When should I run this?	Verify that a virtual machine can fail over and start in the secondary site Useful for testing and training	During planned downtime and outages	During unexpected events
Is a duplicate virtual machine created?	Yes	No	No
Where is it initiated?	On the replica virtual machine	Initiated on primary and completed on secondary	On the replica virtual machine
How often should I run?	We recommend once a month for testing	Once every six months or in accordance with compliance requirements	Only in case of disaster when the primary virtual machine is unavailable
Does the primary virtual machine continue to replicate?	Yes	Yes. When the outage is resolve reverse replication replicates the changes back to the primary site so that primary and secondary are synchronized.	No
Is there any data loss?	None	None. After failover Hyper-V Replica replicates the last set of tracked changes back to the primary to ensure zero data loss.	Depends on the event and recovery points
Is there any downtime?	None. It doesn't impact your production environment. It creates a duplicate test virtual machine during failover. After failover finishes you select **Failover** on the replica virtual machine and it's automatically cleaned up and deleted.	The duration of the planned outage	The duration of the unplanned outage

- **Recovery points**: When you configure replication settings for a virtual machine, you specify the recovery points you want to store from it. Recovery points

represent a snapshot in time from which you can recover a virtual machine. Obviously less data is lost if you recover from a very recent recovery point. You could access recovery points up to 24 hours ago.

Deployment prerequisites

Here's what you should verify before you begin:

- **Figure out which VHDs need to be replicated**. In particular, VHDs that contain data that is rapidly changing and not used by the Replica server after failover, such as page file disks, should be excluded from replication to conserve network bandwidth. Make a note of which VHDs can be excluded.

- **Decide how often you need to synchronize data**: The data on the Replica server is synchronized updated according to the replication frequency you configure (30 seconds, 5 minutes, or 15 minutes). The frequency you choose should consider the following: Are the virtual machines running critical data with a low RPO? What are your bandwidth considerations? Your highly critical virtual machines will obviously need more frequent replication.

- **Decide how to recover data**: By default Hyper-V Replica only stores a single recovery point that will be the latest replication sent from the primary to the secondary. However if you want the option to recover data to an earlier point in time you can specify that additional recovery points should be stored (to a maximum of 24 hourly points). If you do need additional recovery points you should note that this requires more overhead on processing and storage resources.

- **Figure out which workloads you'll replicate**: Standard Hyper-V Replica replication maintains consistency in the state of the virtual machine operating system after a failover, but not the state of applications that running on the virtual machine. If you want to be able to recovery your workload state you can create app-consistent recovery points. Note that app-consistent recovery isn't available on the extended replica site if you're using extended (chained) replication.

- **Decide how to do the initial replication of virtual machine data**: Replication starts by transferring the needs to transfer the current state of the virtual machines. This initial state can be transmitted directly over the existing network, either immediately or at a later time that you configure. You can also use a pre-existing restored virtual machine (for example, if you have restored an earlier backup of the virtual machine on the Replica server) as the initial copy. Or, you can save network bandwidth by copying the initial copy to external media and then physically

delivering the media to the Replica site. If you want to use a pre-existing virtual machine delete all previous snapshots associated with it.

Deployment steps

Step 1: Set up the Hyper-V hosts

You'll need at least two Hyper-V hosts with one or more virtual machines on each server. Get started and Install the Hyper-V role on Windows Server. The host server that you'll replicate virtual machines to will need to be set up as the replica server.

1. In the Hyper-V settings for the server you'll replicate virtual machines to, in **Replication Configuration**, select **Enable this computer as a Replica server**.

2. You can replicate over HTTP or encrypted HTTPS. Select **Use Kerberos (HTTP)** or **Use certificate-based Authentication (HTTPS**). By default HTTP 80 and HTTPS 443 are enabled as firewall exceptions on the replica Hyper-V server. If you change the default port settings you'll need to also change the firewall exception. If you're replicating over HTTPS, you'll need to select a certificate and you should have certificate authentication set up.

3. For authorization, select **Allow replication from any authenticated server** to allow the replica server to accept virtual machine replication traffic from any primary server that authenticates successfully. Select **Allow replication from the specified servers** to accept traffic only from the primary servers you specifically select.

   For both options you can specify where the replicated VHDs should be stored on the replica Hyper-V server.

4. Click **OK** or **Apply**.

Step 2: Set up the firewall

To allow replication between the primary and secondary servers, traffic must get through the Windows Firewall (or any other third-party firewalls). When you installed the Hyper-V role on the servers by default exceptions for HTTP (80) and HTTPS (443) are created. If you're using these standard ports, you'll just need to enable the rules:

- To enable the rules on a standalone host server:

    1. Open Windows Firewall with Advance Security and click **Inbound Rules**.

2. To enable HTTP (Kerberos) authentication, right-click **Hyper-V Replica HTTP Listener (TCP-In)** > **Enable Rule**. To enable HTTPS certificate-based authentication, right-click **Hyper-V Replica HTTPS Listener (TCP-In)** > **Enable Rule**.

- To enable rules on a Hyper-V cluster, open a Windows PowerShell session using **Run as Administrator**, then run one of these commands:
  - For HTTP:

    ```
    get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTP Listener (TCP-In)"}}
    ```

  - For HTTPS:

    ```
    get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTPS Listener (TCP-In)"}}
    ```

Enable virtual machine replication

Do the following on each virtual machine you want to replicate:

1. In the **Details** pane of Hyper-V Manager, select a virtual machine by clicking it. Right-click the selected virtual machine and click **Enable Replication** to open the Enable Replication wizard.

2. On the **Before you Begin** page, click **Next**.

3. On the **Specify Replica Server** page, in the Replica Server box, enter either the NetBIOS or FQDN of the Replica server. If the Replica server is part of a failover cluster, enter the name of the Hyper-V Replica Broker. Click **Next**.

4. On the **Specify Connection Parameters** page, Hyper-V Replica automatically retrieves the authentication and port settings you configured for the replica server. If values aren't being retrieved check that the server is configured as a replica server, and that it's registered in DNS. If required type in the setting manually.

5. On the **Choose Replication VHDs** page, make sure the VHDs you want to replicate are selected, and clear the checkboxes for any VHDs that you want to exclude from replication. Then click **Next**.

6. On the **Configure Replication Frequency** page, specify how often changes should be synchronized from primary to secondary. Then click **Next**.

7. On the **Configure Additional Recovery Points** page, select whether you want to maintain only the latest recovery point or to create additional points. If you want to consistently recover applications and workloads that have their own VSS writers we recommend you select **Volume Shadow Copy Service (VSS) frequency** and specify how often to create app-consistent snapshots. Note that the Hyper-V VMM Requestor Service must be running on both the primary and secondary Hyper-V servers. Then click **Next**.

8. On the **Choose Initial Replication** page, select the initial replication method to use. The default setting to send initial copy over the network will copy the primary virtual machine configuration file (VMCX) and the virtual hard disk files (VHDX and VHD) you selected over your network connection. Verify network bandwidth availability if you're going to use this option. If the primary virtual machine is already configured on the secondary site as a replicate virtual machine it can be useful to select **Use an existing virtual machine on the replication server as the initial copy**. You can use Hyper-V export to export the primary virtual machine and import it as a replica virtual machine on the secondary server. For larger virtual machines or limited bandwidth you can it choose to have initial replication over the network occur at a later time, and then configure off-peak hours, or to send the initial replication information as offline media.

   If you do offline replication you'll transport the initial copy to the secondary server using an external storage medium such as a hard disk or USB drive. To do this, you need to connect the external storage to the primary server (or owner node in a cluster) and then when you select Send initial copy using external media you can specify a location locally or on your external media where the initial copy can be stored. A placeholder virtual machine is created on the replica site. After initial replication completes the external storage can be shipped to the replica site. There you'll connect the external media to the secondary server or to the owner node of the secondary cluster. Then you'll import the initial replica to a specified location and merge it into the placeholder virtual machine.

9. On the **Completing the Enable Replication** page, review the information in the Summary, and then click **Finish.**. The virtual machine data will be transferred in accordance with your chosen settings. A dialog box appears to indicate that replication was successfully enabled.

10. If you want to configure extended (chained) replication, open the replica server, and right-click the virtual machine you want to replicate. Click **Replication** >

Extend Replication and specify the replication settings.

Run a failover

After completing these deployment steps your replicated environment is up and running. Now you can run failovers as needed.

Test failover: If you want to run a test failover right-click the replica virtual machine and select **Replication** > **Test Failover**. Pick the latest or other recovery point if configured. A new test virtual machine will be created and started on the secondary site. After you've finished testing, select **Stop Test Failover** on the replica virtual machine to clean it up. Note that for a virtual machine you can only run one test failover at a time. For more information, see Test failover in Hyper-V Replica.

Planned failover: To run a planned failover, right-click the primary virtual machine and select **Replication** > **Planned Failover**. Planned failover performs prerequisites checks to ensure zero data loss. It checks that the primary virtual machine is shut down before beginning the failover. After the virtual machine is failed over, it starts replicating the changes back to the primary site when it's available. Note that for this to work, the primary server should be configured to receive replication from the secondary server, or from the Hyper-V Replica Broker in the case of a primary cluster. Planned failover sends the last set of tracked changes. For more information, see Planned failover in Hyper-V Replica.

Unplanned failover: To run an unplanned failover, right-click on the replica virtual machine and select **Replication** > **Unplanned Failover** from Hyper-V Manager or Failover Clustering Manager. You can recover from the latest recovery point or from previous recovery points if this option is enabled. After failover, check that everything is working as expected on the failed over virtual machine, then click **Complete** on the replica virtual machine. Read more.

Feedback

Enable Intel Performance Monitoring Hardware in a Hyper-V virtual machine

Article • 05/17/2021

Intel processors contain features collectively called performance monitoring hardware (e.g. PMU, PEBS, LBR). These features are used by performance tuning software like Intel VTune Amplifier to analyze software performance. Prior to Windows Server 2019 and Windows 10 Version 1809, neither the host operating system nor Hyper-V guest virtual machines could use performance monitoring hardware when Hyper-V was enabled. Starting with Windows Server 2019 and Windows 10 Version 1809, the host operating system has access to performance monitoring hardware by default. Hyper-V guest virtual machines do not have access by default, but Hyper-V administrators may choose to grant access to one or more guest virtual machines. This document describes the steps required to expose performance monitoring hardware to guest virtual machines.

Requirements

To enable performance monitoring hardware in a virtual machine, you'll need:

- An Intel processor with performance monitoring hardware (i.e. PMU, PEBS, LBR). Refer to this document from Intel to determine which performance monitoring hardware your system supports.
- Windows Server 2019 or Windows 10 Version 1809 (October 2018 Update) or later
- A Hyper-V virtual machine *without* nested virtualization that is also in the stopped state

To enable upcoming Intel Processor Trace (IPT) performance monitoring hardware in a virtual machine, you'll need:

- An Intel processor that supports IPT and the PT2GPA feature. [^1] Refer to this document from Intel to determine which performance monitoring hardware your system supports.
- Windows Server version 1903 (SAC) or Windows 10 Version 1903 (May 2019 Update) or later
- A Hyper-V virtual machine *without* nested virtualization that is also in the stopped state
- PMU needs to be enabled via the command line using the command seen below.

[^1]: PT2GPA refers to the "Intel PT uses guest physical addresses" bit. This is described in 25.5.4.1 of the Intel SDM.

Enabling performance monitoring components in a virtual machine

To enable different performance monitoring components for a specific guest virtual machine, use the `Set-VMProcessor` PowerShell cmdlet while running as Administrator:

> **Note**
>
> The virtual machine generation must be 9.1 or higher. If nested virtualization is offered to the guest as well, then this needs 9.3 and up.

Powershell

```
# Enable IPT
Set-VMProcessor MyVMName -Perfmon @("ipt", "pmu")
```

Powershell

```
# Enable all components
Set-VMProcessor MyVMName -Perfmon @("ipt", "pmu", "lbr", "pebs")
```

Powershell

```
# Disable all components
Set-VMProcessor MyVMName -Perfmon @()
```

> **Note**
>
> When enabling the performance monitoring components, if `"pebs"` is specified, then `"pmu"` must also be specified.
> PEBS is only supported on hardware that has a PMU Version >= 4.
> Also, any command that attempts to enable `"ipt"` must also specify `"pmu"`.
>
> Enabling a component that is not supported by the host's physical processors will result in a virtual machine start failure.

Effects of enabling performance monitoring hardware on save/restore, export, and live

migration

Microsoft does not recommend live migrating or saving/restoring virtual machines with performance monitoring hardware between systems with different Intel hardware. The specific behavior of performance monitoring hardware is often non-architectural and changes between Intel hardware systems. Moving a running virtual machine between different systems can result in unpredictable behavior of the non-architectural counters.

Dynamic processor compatibility mode

Article • 07/10/2024

> Applies to: Windows Server 2025

The dynamic processor compatibility mode is updated to take advantage of new processor capabilities in a clustered environment. Processor compatibility works by determining the supported processor features for each individual node in the cluster and calculating the common denominator across all processors. Virtual machines (VMs) are configured to use the maximum number of features available across all servers in the cluster. This improves performance compared to the previous version of processor compatibility that defaulted to a minimal, fixed set of processor capabilities.

When to use processor compatibility mode

Processor compatibility mode allows you to move a live VM (live migrating) or move a VM that is saved between nodes with different process capability sets. However, even when processor compatibility is enabled, you can't move VMs between hosts with different processor manufacturers. For example, you can't move running VMs or saved state VMs from a host with Intel processors to a host with AMD processors. If you must move a VM in this manner, shut down the VM first, then restart it on the new host.

> **Important**
>
> Only Hyper-V VMs with the latest configuration version (10.0) benefit from the dynamic configuration. VMs with older versions don't benefit from the dynamic configuration and won't continue to use fixed processor capabilities from the previous version.

> **Note**
>
> You don't need to use processor compatibility mode if you plan to stop and restart the VMs. Any time a VM is restarted, the guest operating system enumerates the processor compatibilities that are available on the new host computer.

Why processor compatibility mode is needed

Processor manufacturers often introduce optimizations and capabilities in their processors. These capabilities often improve performance or security by using specialized hardware for a particular task. For example, many media applications use processor capabilities to speed up vector calculations. These features are rarely required for applications to run; they boost performance.

The capability set that's available on a processor varies depending on its make, model, and age. Operating systems and application software typically enumerate the system's processor capability set when they're first launched. Software doesn't expect the available processor capabilities to change during their lifetime, which never happens when running on a physical computer, because processor capabilities are static unless the processor upgrades.

However, VM mobility features allow a running VM to be migrated to a new virtualization host. If software in the VM detects and starts using a particular processor capability, and then the VM is moved to a new virtualization host without that capability, the software will likely fail. This could result in the application or VM crashing.

To avoid failures, Hyper-V performs "preflight" checks whenever a VM live migration or save/restore operation is initiated. These checks compare the set of processor features that are available to the VM on the source host against the set of features that are available on the target host. If these feature sets don't match, the migration or restore operation is canceled.

What's new in processor compatibility mode

In the past, all new processor instructions sets were hidden, meaning that the guest operating system and application software couldn't use processor instruction set enhancements to help applications and VMs stay performant.

To overcome this limitation, processor compatibility mode now provides enhanced, dynamic capabilities on processors capable of second-level address translation (SLAT). This new functionality calculates the common denominator of the CPU features supported by the nodes in the cluster and updates the existing processor compatibility mode on a VM to use this dynamically calculated feature set instead of the old hard-coded feature set.

The new processor compatibility mode ensures that the set of processor features available to VMs across virtualization hosts match by presenting a common capability set across all servers in the cluster. Each VM receives the maximum number of processor instruction sets that are present across all servers in the cluster. This process occurs

automatically and is always enabled and replicated across the cluster, so there's no command to enable or disable the process.

Using processor compatibility mode

There are important concepts to understand when using processor compatibility mode in Hyper-V:

- Running VMs can only be migrated between virtualization hosts that use processors from the same manufacturer.
- You must shut down the VM before you can enable or disable processor compatibility mode.
- Processor compatibility mode isn't needed for VM moves that involve a stop and restart of the VM.
- Anytime a VM is restarted, the guest operating system enumerates the processor features that are available on the new host computer.

> ⓘ **Note**
>
> In Windows Server, Microsoft recommends turning on processor compatibility mode only before VM migration scenarios, and then turning it off when the migration is complete.

Migrating running VMs between clusters

Assuming that all servers in each cluster are running the same hardware, it's possible to live migrate running VMs between clusters. There are three common scenarios.

- **Live migrating a VM from a cluster with new processors to a cluster with the same processors.** The VM capabilities are transferred to the destination cluster. This scenario doesn't require processor compatibility mode to be enabled; however, leaving it enabled won't cause any problems.
- **Live migrating a VM from a cluster with older processors to a cluster with newer processors.** The VM capabilities are transferred to the destination cluster. In this scenario, if the VM is restarted, it receives the latest calculated capability of the destination cluster.

- **Live migrating a VM from a cluster with newer processors to a cluster with older processors.** You'll need to set the VM processor to use the `MinimumFeatureSet` for the `CompatibilityForMigrationMode` parameter in PowerShell, or select **Compatible across other hosts with the same CPU manufacturer** in Windows Admin Center under **Virtual machines** > **Settings** > **Processors**. This setting assigns the VM to the minimum processor capabilities offered on the server. Once the compatibility is moved to **Compatible across the cluster (Recommended)** and the VM is restarted, it receives the latest calculated capability of the destination cluster.

Ramifications of using processor compatibility mode

It's difficult to quantify the overall performance effects of processor compatibility mode. The performance loss is primarily dependent on the workload running in the VM. Some workloads may be unaffected, while others show a noticeable difference. Software that heavily relies on hardware optimizations (such as encryption, compression, or intensive floating-point calculations) is impacted the most.

Applications that encrypt or decrypt a large amount of data benefit from this processor feature, so turning it off by enabling processor compatibility mode impacts the performance of these specific operations.

If you're concerned about the performance impact of processor compatibility mode, it's best to compare VM workload performance with processor compatibility mode enabled and with it disabled.

Configure a VM to use processor compatibility mode

This section explains how to configure a VM to use processor compatibility mode using either Hyper-V manager or PowerShell. It's possible to run VMs with and without compatibility mode in the same cluster.

> **ⓘ Important**
>
> You must shut down the VM before you can enable or disable processor compatibility mode.

Enable processor compatibility mode using Hyper-V Manager

To enable processor compatibility mode for a VM using Hyper-V Manager:

1. Shut down the VM.
2. Select **Start**, point to **Administrative Tools**, and then select **Hyper-V Manager**.
3. Select the server running Hyper-V and the desired VM.
4. If the VM is running, you must shut down the VM to enable the processor compatibility mode setting.
5. In the Action pane, select **Settings**, and then select **Processor**.
6. Expand **Processor**, and select **Compatibility**.
7. Select **Migrate to a physical computer with a different processor**, and then select **OK**.
8. Restart the VM.

Disable processor compatibility mode using Hyper-V Manager

To disable processor compatibility mode for a VM using Hyper-V Manager:

1. Shut down the VM.
2. Select **Start**, point to **Administrative Tools**, and then select **Hyper-V Manager**.
3. Select the server running Hyper-V and the desired VM.
4. If the VM is running, you must shut down the VM to disable the processor compatibility mode setting.
5. In the Action pane, select **Settings**, and then select **Processor**.
6. Expand **Processor**, and select **Compatibility**.
7. De-select the **Migrate to a physical computer with a different processor** checkbox, and then select **OK**.
8. Restart the VM.

Enable processor compatibility mode using PowerShell

To enable processor compatibility mode, run the following cmdlet:

```PowerShell
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-VMProcessor -CompatibilityForMigrationEnabled $true
```

We recommend setting the VM's CPU features to the maximum level supported by all servers in the cluster. This maximizes VM performance while preserving the ability to move the running VM to other servers in the cluster.

To enable the VM to use the cluster node common features, run the following cmdlet:

```PowerShell
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-VMProcessor -CompatibilityForMigrationEnabled $true -CompatibilityForMigrationMode CommonClusterFeatureSet
```

Alternatively, you can set the VM's CPU features to minimum, ensuring that you can move the running VM to other Hyper-V hosts outside the cluster if they have the same CPU manufacturer.

To enable the VM to use the default minimum features to migrate across clusters, run the following cmdlet:

```PowerShell
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-VMProcessor -CompatibilityForMigrationEnabled $true -CompatibilityForMigrationMode MinimumFeatureSet
```

Disable processor compatibility mode using PowerShell

To disable processor compatibility mode for a VM using PowerShell, shut down the VM and run the `Set-VMProcessor` cmdlet, setting `CompatibilityForMigrationEnabled` to **$false**:

```PowerShell
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-VMProcessor -CompatibilityForMigrationEnabled $false
```

Then restart the VM.

Feedback

Was this page helpful? Yes No

Live Migration Overview

Article • 09/17/2020

Live migration is a Hyper-V feature in Windows Server. It allows you to transparently move running Virtual Machines from one Hyper-V host to another without perceived downtime. The primary benefit of live migration is flexibility; running Virtual Machines are not tied to a single host machine. This allows actions like draining a specific host of Virtual Machines before decommissioning or upgrading it. When paired with Windows Failover Clustering, live migration allows the creation of highly available and fault tolerant systems.

Related Technologies and Documentation

Live migration is often used in conjunction with a few related technologies like Failover Clustering and System Center Virtual Machine Manager. If you're using Live Migration via these technologies, here are pointers to their latest documentation:

- Failover Clustering (Windows Server 2016)
- System Center Virtual Machine Manager (System Center 2016)

If you're using older versions of Windows Server, or need details on features introduced in older versions of Windows Server, here are pointers to historical documentation:

- Live Migration (Windows Server 2008 R2)
- Live Migration (Windows Server 2012 R2)
- Failover Clustering (Windows Server 2012 R2)
- Failover Clustering (Windows Server 2008 R2)
- System Center Virtual Machine Manager (System Center 2012 R2)
- System Center Virtual Machine Manager (System Center 2008 R2)

Live Migration in Windows Server 2016

In Windows Server 2016, there are fewer restrictions on live migration deployment. It now works without Failover Clustering. Other functionality remains unchanged from previous releases of Live Migration. For more details on configuring and using live migration without Failover Clustering:

- Set up hosts for live migration without Failover Clustering
- Use live migration without Failover Clustering to move a virtual machine

Live Migration Overview

Article • 09/17/2020

Live migration is a Hyper-V feature in Windows Server. It allows you to transparently move running Virtual Machines from one Hyper-V host to another without perceived downtime. The primary benefit of live migration is flexibility; running Virtual Machines are not tied to a single host machine. This allows actions like draining a specific host of Virtual Machines before decommissioning or upgrading it. When paired with Windows Failover Clustering, live migration allows the creation of highly available and fault tolerant systems.

Related Technologies and Documentation

Live migration is often used in conjunction with a few related technologies like Failover Clustering and System Center Virtual Machine Manager. If you're using Live Migration via these technologies, here are pointers to their latest documentation:

- Failover Clustering (Windows Server 2016)
- System Center Virtual Machine Manager (System Center 2016)

If you're using older versions of Windows Server, or need details on features introduced in older versions of Windows Server, here are pointers to historical documentation:

- Live Migration (Windows Server 2008 R2)
- Live Migration (Windows Server 2012 R2)
- Failover Clustering (Windows Server 2012 R2)
- Failover Clustering (Windows Server 2008 R2)
- System Center Virtual Machine Manager (System Center 2012 R2)
- System Center Virtual Machine Manager (System Center 2008 R2)

Live Migration in Windows Server 2016

In Windows Server 2016, there are fewer restrictions on live migration deployment. It now works without Failover Clustering. Other functionality remains unchanged from previous releases of Live Migration. For more details on configuring and using live migration without Failover Clustering:

- Set up hosts for live migration without Failover Clustering
- Use live migration without Failover Clustering to move a virtual machine

Set up hosts for live migration without Failover Clustering

Article • 07/04/2024

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to set up hosts that aren't clustered so you can do live migrations between them. Use these instructions if you didn't set up live migration when you installed Hyper-V, or if you want to change the settings. To set up clustered hosts, use tools for Failover Clustering.

Requirements for setting up live migration

To set up non-clustered hosts for live migration, you'll need:

- A user account with permission to perform the various steps. Membership in the local Hyper-V Administrators group or the Administrators group on both the source and destination computers meets this requirement, unless you're configuring constrained delegation. Membership in the Domain Administrators group is required to configure constrained delegation.

- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.
  For version upgrade instructions, see Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016. For installation instructions, see Install the Hyper-V role on Windows Server.

- Source and destination computers that either belong to the same Active Directory domain, or belong to domains that trust each other.

- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run the tools from the server.

Consider options for authentication and networking

Consider how you want to set up the following:

- **Authentication**: Which protocol will be used to authenticate live migration traffic between the source and destination servers? The choice determines whether you'll need to sign on to the source server before starting a live migration:

  - Kerberos lets you avoid having to sign in to the server, but requires constrained delegation to be set up. See below for instructions.

  - CredSSP lets you avoid configuring constrained delegation, but requires you sign in to the source server. You can do this through a local console session, a Remote Desktop session, or a remote Windows PowerShell session.

    CredSPP requires signing in for situations that might not be obvious. For example, if you sign in to TestServer01 to move a virtual machine to TestServer02, and then want to move the virtual machine back to TestServer01, you'll need to sign in to TestServer02 before you try to move the virtual machine back to TestServer01. If you don't do this, the authentication attempt fails, an error occurs, and the following message is displayed:

    "Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

- **Performance**: Does it make sense to configure performance options? These options can reduce network and CPU usage, as well as make live migrations go faster. Consider your requirements and your infrastructure, and test different configurations to help you decide. The options are described at the end of step 2.

- **Network preference**: Will you allow live migration traffic through any available network, or isolate the traffic to specific networks? As a security best practice, we recommend that you isolate the traffic onto trusted, private networks because live migration traffic is not encrypted when it is sent over the network. Network isolation can be achieved through a physically isolated network or through another trusted networking technology such as VLANs.

Upgrading to Windows Server 2025 (preview)

> **ⓘ Important**

> Windows Server 2025 is in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Starting with Windows Server 2025 (preview), Credential Guard is enabled by default on all domain-joined servers that aren't Domain Controllers. As a result you might not be able to use CredSSP-based Live Migration with Hyper-V after upgrading to Windows Server 2025. CredSSP-based delegation is the default for Windows Server 2022 and earlier for live migration. Instead use Kerberos constrained Delegation, as described in the following section. For more information, see Live migration with Hyper-V breaks when upgrading to Windows Server 2025.

Step 1: Configure constrained delegation (optional)

If you have decided to use Kerberos to authenticate live migration traffic, configure constrained delegation using an account that is a member of the Domain Administrators group.

Use the Users and Computers snap-in to configure constrained delegation

1. Open the Active Directory Users and Computers snap-in. (From Server Manager, select the server if it's not selected, click **Tools** >> **Active Directory Users and Computers**).
2. From the navigation pane in **Active Directory Users and Computers**, select the domain and double-click the **Computers** folder.
3. From the **Computers** folder, right-click the computer account of the source server and then click **Properties**.
4. From **Properties**, click the **Delegation** tab.
5. On the delegation tab, select **Trust this computer for delegation to the specified services only** and then select **Use any authentication protocol**.
6. Click **Add**.
7. From **Add Services**, click **Users or Computers**.

8. From **Select Users or Computers**, type the name of the destination server. Click **Check Names** to verify it, and then click **OK**.

9. From **Add Services**, in the list of available services, do the following and then click **OK**:

    - To move virtual machine storage, select **cifs**. This is required if you want to move the storage along with the virtual machine, as well as if you want to move only a virtual machine's storage. If the server is configured to use SMB storage for Hyper-V, this should already be selected.

    - To move virtual machines, select **Microsoft Virtual System Migration Service**.

10. On the **Delegation** tab of the Properties dialog box, verify that the services you selected in the previous step are listed as the services to which the destination computer can present delegated credentials. Click **OK**.

11. From the **Computers** folder, select the computer account of the destination server and repeat the process. In the **Select Users or Computers** dialog box, be sure to specify the name of the source server.

The configuration changes take effect after both of the following happen:

- The changes are replicated to the domain controllers that the servers running Hyper-V are logged into.
- The domain controller issues a new Kerberos ticket.

Step 2: Set up the source and destination computers for live migration

This step includes choosing options for authentication and networking. As a security best practice, we recommend that you select specific networks to use for live migration traffic, as discussed above. This step also shows you how to choose the performance option.

Use Hyper-V Manager to set up the source and destination computers for live migration

1. Open Hyper-V Manager. (From Server Manager, click **Tools** >>**Hyper-V Manager**.)

2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to

add more servers.)

3. In the **Action** pane, click **Hyper-V Settings** >>**Live Migrations**.
4. In the **Live Migrations** pane, check **Enable incoming and outgoing live migrations**.
5. Under **Simultaneous live migrations**, specify a different number if you don't want to use the default of 2.
6. Under **Incoming live migrations**, if you want to use specific network connections to accept live migration traffic, click **Add** to type the IP address information. Otherwise, click **Use any available network for live migration**. Click **OK**.
7. To choose Kerberos and performance options, expand **Live Migrations** and then select **Advanced Features**.
    - If you have configured constrained delegation, under **Authentication protocol**, select **Kerberos**.
    - Under **Performance options**, review the details and choose a different option if it's appropriate for your environment.
8. Click **OK**.
9. Select the other server in Hyper-V Manager and repeat the steps.

Use Windows PowerShell to set up the source and destination computers for live migration

Three cmdlets are available for configuring live migration on non-clustered hosts: Enable-VMMigration, Set-VMMigrationNetwork, and Set-VMHost. This example uses all three and does the following:

- Configures live migration on the local host
- Allows incoming migration traffic only on a specific network
- Chooses Kerberos as the authentication protocol

Each line represents a separate command.

PowerShell

```
PS C:\> Enable-VMMigration

PS C:\> Set-VMMigrationNetwork 192.168.10.1
```

```
PS C:\> Set-VMHost -VirtualMachineMigrationAuthenticationType Kerberos
```

Set-VMHost also lets you choose a performance option (and many other host settings). For example, to choose SMB but leave the authentication protocol set to the default of CredSSP, type:

PowerShell

```
PS C:\> Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
```

This table describes how the performance options work.

Expand table

Option	Description
TCP/IP	Copies the memory of the virtual machine to the destination server over a TCP/IP connection.
Compression	Compresses the memory content of the virtual machine before copying it to the destination server over a TCP/IP connection. **Note:** This is the **default** setting.
SMB	Copies the memory of the virtual machine to the destination server over an SMB 3.0 connection. - SMB Direct is used when the network adapters on the source and destination servers have Remote Direct Memory Access (RDMA) capabilities enabled. - SMB Multichannel automatically detects and uses multiple connections when a proper SMB Multichannel configuration is identified. For more information, see Improve Performance of a File Server with SMB Direct.

Next steps

After you set up the hosts, you're ready to do a live migration. For instructions, see Use live migration without Failover Clustering to move a virtual machine.

Feedback

Was this page helpful? Yes No

Use live migration without Failover Clustering to move a virtual machine

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This article shows you how to move a virtual machine by doing a live migration without using Failover Clustering. A live migration moves running virtual machines between Hyper-V hosts without any noticeable downtime.

To be able to do this, you'll need:

- A user account that's a member of the local Hyper-V Administrators group or the Administrators group on both the source and destination computers.
- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers and set up for live migrations. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.

  For version upgrade instructions, see Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016. For installation instructions, see Set up hosts for live migration.
- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run them from there.

Use Hyper-V Manager to move a running virtual machine

1. Open Hyper-V Manager. (From Server Manager, click **Tools** >>**Hyper-V Manager**.)
2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to add more servers.)
3. From the **Virtual Machines** pane, right-click the virtual machine and then click **Move**. This opens the Move Wizard.
4. Use the wizard pages to choose the type of move, destination server, and options.

5. On the **Summary** page, review your choices and then click **Finish**.

Use Windows PowerShell to move a running virtual machine

The following example uses the Move-VM cmdlet to move a virtual machine named *LMTest* to a destination server named *TestServer02* and moves the virtual hard disks and other file, such checkpoints and Smart Paging files, to the *D:\LMTest* directory on the destination server.

```
PS C:\> Move-VM LMTest TestServer02 -IncludeStorage -DestinationStoragePath
D:\LMTest
```

Troubleshooting

Failed to establish a connection

If you haven't set up constrained delegation, you must sign in to source server before you can move a virtual machine. If you don't do this, the authentication attempt fails, an error occurs, and this message is displayed:

"Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

To fix this problem, sign in to the source server and try the move again. To avoid having to sign in to a source server before doing a live migration, set up constrained delegation. You'll need domain administrator credentials to set up constrained delegation. For instructions, see Set up hosts for live migration.

Failed because the host hardware isn't compatible

If a virtual machine doesn't have processor compatibility turned on and has one or more snapshots, the move fails if the hosts have different processor versions. An error occurs and this message is displayed:

The virtual machine cannot be moved to the destination computer. The hardware on the destination computer is not compatible with the hardware requirements of this

virtual machine.

To fix this problem, shut down the virtual machine and turn on the processor compatibility setting.

1. From Hyper-V Manager, in the **Virtual Machines** pane, right-click the virtual machine and click Settings.
2. In the navigation pane, expand **Processors** and click **Compatibility**.
3. Check **Migrate to a computer with a different processor version**.
4. Click **OK**.

   To use Windows PowerShell, use the Set-VMProcessor cmdlet:

```
PS C:\> Set-VMProcessor TestVM -CompatibilityForMigrationEnabled $true
```

Use live migration with workgroup clusters

Article • 07/29/2024

> Applies to: Windows Server 2025

This article describes how to move a virtual machine by doing a live migration between hosts using workgroup clusters. Workgroup clusters are a type of Failover Cluster that doesn't use an on-premises domain controller or Active Directory forest. Instead, workgroup clusters are joined by workgroup. Workgroup clusters were introduced in Windows Server 2016. However, live migration for workgroup clusters wasn't supported until now, in Windows Server 2025. Benefit from the flexibility of live migration combined with the high availability of workgroup clusters.

Follow the steps in this article to perform your own live migration.

Prerequisites

The following prerequisites must be met in order to do a live migration of hosts using workgroup clusters:

- A workgroup cluster with two or more nodes is up and running. To learn more about creating a workgroup cluster, see Create a workgroup cluster.
- A local user account exists on each server node with an identical username and password.

Consider options for authentication and performance

When setting up live migrations for workgroup clusters, it's important to consider how authentication and performance.

- **Authentication**: Local accounts with an identical username and password on each node are used to create and configure the workgroup cluster. The cluster uses self-signing PKU2U certificates to authenticate and be able to move a virtual machine from one host node to another host node without Kerberos. The local accounts on each node are the only way to successfully authenticate workgroup clusters and allow for live migration between source and destination servers.

- **Performance**: Configuring performance options can reduce network and CPU usage. Different options like simultaneous migrations can also make live migrations go faster. Consider your requirements and your infrastructure, and test different configurations to help you decide.

Do a live migration with Hyper-V workgroup clusters

In the next sections, you'll complete the following steps:

- **Install the Hyper-V role and Hyper-V management tools.** Each virtual machine needs Hyper-V installed in order to be able to connect to other hosts and do a live migration.
- **Create a new virtual machine and add it to the workgroup cluster.** Add in a virtual machine role to facilitate the live migration.
- **Set up source and destination servers.** Set up each server node to enable live migrations.
- **Move a running virtual machine with live migration.** Complete a live migration by moving a running virtual machine between Hyper-V hosts without any noticeable downtime.

Step 1: Install the Hyper-V role

The Hyper-V role must be installed on the source and destination servers and set up for live migrations. Install this role before continuing.

Hyper-V provides the services that you can use to create and manage virtual machines. Hyper-V may be installed in many ways. This section describes how to install the role using either PowerShell or the Server Manager.

PowerShell

1. Open a PowerShell session as an Administrator.
2. Use the Install-WindowsFeature cmdlet to install the Hyper-V role with the following command:

```
PowerShell
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -
Restart
```

3. Wait for the role to be installed, and then restart your virtual machine.
4. Repeat the installation for the other virtual machines in the cluster.

> **Note**
>
> If you're unable to install the Hyper-V role, you might need to enable Nested Virtualization. See how to enable Nested Virtualization to allow Hyper-V to run inside of a Hyper-V virtual machine.

Step 2: Create a new virtual machine and add it to the workgroup cluster

Add a new Hyper-V virtual machine as a role to your workgroup cluster in order to do a live migration between hosts.

PowerShell

1. Connect to one of your server nodes.
2. Open a PowerShell session as an Administrator.
3. Run the following New-VM command to create a new virtual machine that has 10GB of memory and uses an existing VHDX image on the server node. Change the parameters and values as needed to customize your setup.

   PowerShell

   ```
   New-VM -Name "<VM_NAME>" -MemoryStartupBytes 10GB -VHDPath
   <PATH_TO_VHDX_FILE>
   ```

4. Add in the virtual machine as a **Virtual Machine** role in the workgroup cluster enable automatic failover.

   PowerShell

   ```
   Add-ClusterVirtualMachineRole -VirtualMachine <VM_NAME>
   ```

Step 3: Set up the source and destination computers for live migration

In this step, set up your source and host destination virtual machines to enable live migrations. Here, you can also specify live migration settings, such as how many live and storage migrations to allow at the same time.

PowerShell

1. Connect to one of your server nodes.
2. Open a PowerShell session as an Administrator.
3. First, use the Enable-VMMigration cmdlet to configure live migration on the Hyper-V virtual machine host.

   PowerShell

   ```
   Enable-VMMigration
   ```

4. Use the Set-VMHost cmdlet to configure the local Hyper-V host. The following command configures the virtual machine to allow 10 simultaneous live migrations and storage migrations. Change these values to the number of simultaneous live and storage migrations your setup allows. You might need to test different configurations to help you decide.

   PowerShell

   ```
   Set-VMHost -MaximumVirtualMachineMigrations 10 -
   MaximumStorageMigrations 10
   ```

   Set-VMHost also lets you specify a performance option, and other host settings. Consider using a parameter like `-VMMigrationPerformance` to choose more settings for your virtual machine.

5. Repeat the steps for the other server node.

Step 4: Move a running virtual machine with live migration

Finally, do a live migration to move a running virtual machine.

1. Connect to the current owner node in your workgroup cluster.
2. Open Failover Cluster Manager.
3. In the **Roles** section, select the virtual machine role, and right-click.
4. Select **Move**, then **Live Migration**, and then **Best Possible Node**.
5. In the **Information** column, see a status appear with the message **Live Migrating, X% completed**.
6. Once complete, confirm that the Owner Node column updates with the other node in your workgroup cluster.

Next steps

After completing a live migration, it's important to confirm that the migration works, and all virtual machines migrated were migrated successfully. If you notice any issues during or after a migration, it might be necessary to revisit the simultaneous migrations allowed or configure live migration performance options.

To learn more about Live Migration performance options, see Virtual Machine Live Migration Overview

You can also read more about live migration performance in Hyper-V Network I/O Performance.

Feedback

Generation 1 virtual machine security settings

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Use the generation 1 virtual machine security settings in Hyper-V Manager to help protect the data and state of a virtual machine.

Encryption support settings in Hyper-V Manager

You can help protect the data and state of the virtual machine by selecting the following encryption support option.

- **Encrypt state and virtual machine migration traffic** - Encrypts the virtual machine saved state when it's written to disk and the live migration traffic.

To enable this option, you must add a key storage drive for the virtual machine.

Key storage drive in Hyper-V Manager

A key storage drive provides a small drive to the virtual machine for a BitLocker key to be stored. This allows the virtual machine to encrypt its operating system disk without requiring a virtualized Trusted Platform Module (TPM) chip. The contents of the key storage drive are encrypted by using a Key Protector. The Key Protector authories the Hyper-V host to run the virtual machine. Both the contents of the key storage drive and the Key Protector are stored as part of the virtual machine's runtime state.

To decrypt the contents of the key storage drive and start the virtual machine, the Hyper-V host needs to be either:

- Part of an authorized guarded fabric for this virtual machine, or
- Have the private key from one of the virtual machine's guardians.

To learn more about guarded fabrics, please see the Introducing Shielded VMs section in Security and Assurance.

You can add a key storage drive to an empty slot on one of the virtual machine's IDE controllers. To do this, click **Add Key Storage Drive** to add a key storage drive to the first free IDE controller slot of this virtual machine.

Additional References

- Generation 2 virtual machine security settings in Hyper-V manager
- Security and Assurance

Feedback

Generation 2 virtual machine security settings for Hyper-V

Article • 12/08/2021

> Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Use the virtual machine security settings in Hyper-V Manager to help protect the data and state of a virtual machine. You can protect virtual machines from inspection, theft, and tampering from both malware that may run on the host, and datacenter administrators. The level of security you get depends on the host hardware you run, the virtual machine generation, and whether you set up the service, called the Host Guardian Service, that authorizes hosts to start shielded virtual machines.

The Host Guardian Service is a new role in Windows Server 2016. It identifies legitimate Hyper-V hosts and allows them to run a given virtual machine. You'd most commonly set up the Host Guardian Service for a datacenter. But you can create a shielded virtual machine to run it locally without setting up a Host Guardian Service. You can later distribute the shielded virtual machine to a Host Guardian Fabric.

If you haven't set up the Host Guardian Service or are running it in local mode on the Hyper-V host and the host has the virtual machine owner's guardian key, you can change the settings described in this topic. An owner of a guardian key is an organization that creates and shares a private or public key to own all virtual machines created with that key.

To learn how you can make your virtual machines more secure with the Host Guardian Service, see the following resources.

- Harden the Fabric: Protecting Tenant Secrets in Hyper-V (Ignite video)
- Guarded Fabric and Shielded VMs

Secure Boot setting in Hyper-V Manager

Secure Boot is a feature available with generation 2 virtual machines that helps prevent unauthorized firmware, operating systems, or Unified Extensible Firmware Interface (UEFI) drivers (also known as option ROMs) from running at boot time. Secure Boot is enabled by default. You can use secure boot with generation 2 virtual machines that run Windows or Linux distribution operating systems.

The templates described in the following table refer to the certificates that you need to verify the integrity of the boot process.

Expand table

Template name	Description
Microsoft Windows	Select to secure boot the virtual machine for a Windows operating system.
Microsoft UEFI Certificate Authority	Select to secure boot the virtual machine for a Linux distribution operating system.
Open Source Shielded VM	This template is leveraged to secure boot for Linux-based shielded VMs.

For more information, see the following topics.

- Windows 10 Security Overview
- Should I create a generation 1 or 2 virtual machine in Hyper-V?
- Linux and FreeBSD Virtual Machines on Hyper-V

Encryption support settings in Hyper-V Manager

You can help protect the data and state of the virtual machine by selecting the following encryption support options.

- **Enable Trusted Platform Module** - This setting makes a virtualized Trusted Platform Module (TPM) chip available to your virtual machine. This allows the guest to encrypt the virtual machine disk by using BitLocker. You can enable this by opening the VM settings, click on **Security**, then in the *Encryption Support* section, tick the box to **Enable Trusted Platform Module**. You can also use the Enable-VMTPM PowerShell cmdlet.
  - If your Hyper-V host is running Windows 10 1511, you have to enable Isolated User Mode.
- **Encrypt State and VM migration traffic** - Encrypts the virtual machine saved state and live migration traffic.

Enable Isolated User Mode

If you select **Enable Trusted Platform Module** on Hyper-V hosts that run versions of Windows earlier than Windows 10 Anniversary Update, you must enable Isolated User

Mode. You don't need to do this for Hyper-V hosts that run Windows Server 2016 or Windows 10 Anniversary Update or later.

Isolated User Mode is the runtime environment that hosts security applications inside Virtual Secure Mode on the Hyper-V host. Virtual Secure Mode is used to secure and protect the state of the virtual TPM chip.

To enable Isolated User Mode on the Hyper-V host that run earlier versions of Windows 10,

1. Open Windows PowerShell as an administrator.
2. Run the following commands:

```
PowerShell

Enable-WindowsOptionalFeature -Feature IsolatedUserMode -Online
New-Item -Path HKLM:\SYSTEM\CurrentControlSet\Control\DeviceGuard -
Force
New-ItemProperty -Path
HKLM:\SYSTEM\CurrentControlSet\Control\DeviceGuard -Name
EnableVirtualizationBasedSecurity -Value 1 -PropertyType DWord -Force
```

You can migrate a virtual machine with virtual TPM enabled to any host that runs Windows Server 2016, Windows 10 build 10586 or higher versions. But if you migrate it to another host, you may not be able to start it. You must update the Key Protector for that virtual machine to authorize the new host to run the virtual machine. For more information, see Guarded Fabric and Shielded VMs and System requirements for Hyper-V on Windows Server.

Security Policy in Hyper-V Manager

For more virtual machine security, use the **Enable Shielding** option to disable management features like console connection, PowerShell Direct, and some integration components. If you select this option, **Secure Boot**, **Enable Trusted Platform Module**, and **Encrypt State and VM migration traffic** options are selected and enforced.

You can run the shielded virtual machine locally without setting up a Host Guardian Service. But if you migrate it to another host, you may not be able to start it. You must update the Key Protector for that virtual machine to authorize the new host to run the virtual machine. For more information, see Guarded Fabric and Shielded VMs.

For more information about security in Windows Server, see Security and Assurance.

Feedback

Was this page helpful? Yes No

Hyper-V Virtual Machine Connection

Article • 02/11/2021

> **Applies to:** Windows Server 2019, Windows Server 2016, Windows 10, Windows 8.1, Windows Server 2012 R2, Windows Server 2012, Windows 8

Virtual Machine Connection (VMConnect) is a tool you can use to connect to a virtual machine to install or interact with the guest operating system in a virtual machine. Some of the tasks you can perform by using VMConnect include the following:

- Start and shut down a virtual machine
- Connect to a DVD image (.iso file) or a USB flash drive
- Create a checkpoint
- Modify the settings of a virtual machine

Tips for using VMConnect

You may find the following information helpful for using VMConnect:

Expand table

To do this...	Do this...
Send mouse clicks or keyboard input to the virtual machine	Click anywhere in the virtual machine window. The mouse pointer may appear as a small dot when you connect to a running virtual machine.
Return mouse clicks or keyboard input to the physical computer	Press CTRL+ALT+LEFT arrow and then move the mouse pointer outside of the virtual machine window. This mouse release key combination can be changed in the Hyper-V settings in Hyper-V Manager.
Send CTRL+ALT+DELETE key combination to a virtual machine	Select **Action** > **Ctrl+Alt+Delete** or use the key combination CTRL+ALT+END.
Switch from a window mode to a full-screen mode	Select **View** > **Full Screen Mode**. To switch back to window mode, press CTRL+ALT+BREAK.
Create a checkpoint to capture the current state of	Select **Action** > **Checkpoint** or use the key combination CTRL+N.

To do this...	Do this...
the machine for troubleshooting	
Change the settings of the virtual machine	Select **File** > **Settings**.
Connect to a DVD image (.iso file) or a virtual floppy disk (.vfd file)	Select **Media**. Virtual floppy disks are not supported for generation 2 virtual machines. For more information, see Should I create a generation 1 or 2 virtual machine in Hyper-V?.
Use a host's local resources on Hyper-V virtual machine like a USB flash drive	Turn on enhanced session mode on the Hyper-V host, use VMConnect to connect to the virtual machine, and before you connect, choose the local resource that you want to use. For the specific steps, see Use local resources on Hyper-V virtual machine with VMConnect.
Change saved VMConnect settings for a virtual machine	Run the following command in Windows PowerShell or the command prompt: `VMConnect.exe <ServerName> <VMName> /edit`
Prevent a VMConnect user from taking over another user's VMConnect session	Turn on enhanced session mode on Hyper-V host. Not having enhanced session mode turned on may pose a security and privacy risk. If a user is connected and logged on to a virtual machine through VMConnect and another authorized user connects to the same virtual machine, the session will be taken over by the second user and the first user will lose the session. The second user will be able to view the first user's desktop, documents, and applications.
Manage integration services or components that allow the VM to communicate with the Hyper-V host	On Hyper-V hosts that run Windows 10 or Windows Server 2016, you can't manage integration services with VMConnect. See these topics: - Turn on/turn off integration services from the Hyper-V host - Turn on/turn off integration services from a Windows virtual machine - Turn on/turn off integration services from a Linux virtual machine - Keep integration services updated for the virtual machine For hosts that run Windows Server 2012 or Windows Server 2012 R2, see Integration Services.
Resize the VMConnect window	You can change the size of the VMConnect window for generation 2 virtual machines that run a Windows operating system. To do this, you may need to turn on enhanced session mode on the Hyper-V host. For more information, see Turn on enhanced session mode on Hyper-V host. For virtual machines that run

To do this...	Do this...
	Ubuntu, see Changing Ubuntu Screen Resolution in a Hyper-V VM.

Keyboard shortcuts

By default, the keyboard input and mouse clicks are sent to the virtual machine. So you may need to press CTRL + ALT + LEFT arrow before you use the following shortcut keys.

Expand table

Key combination	Description
CTRL+ALT+LEFT arrow	Mouse release
CTRL+ALT+END	Equivalent of CTRL+ALT+DELETE in the virtual machine
CTRL+ALT+BREAK	Switch from full-screen mode back to windowed mode
CTRL+O	Opens the settings for the virtual machine
CTRL+S	Starts the virtual machine
CTRL+N	Create a checkpoint
CTRL+E	Revert to a checkpoint
CTRL+C	Do a screen capture

See Also

- Use local resources on Hyper-V virtual machine with VMConnect
- Hyper-V on Windows Server 2016
- Hyper-V on Windows 10

Feedback

Was this page helpful? Yes No

Use local resources on Hyper-V virtual machine with VMConnect

Article • 02/16/2023

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows 11, Windows 10, Windows 8.1

Virtual Machine Connection (VMConnect) lets you use a computer's local resources in a virtual machine, like a removable USB flash drive or a printer. Enhanced session mode also lets you resize the VMConnect window. This article shows you how to configure the host and then give the virtual machine access to a local resource.

Enhanced session mode and Type clipboard text are available only for virtual machines that run recent Windows operating systems. See Requirements for using local resources.

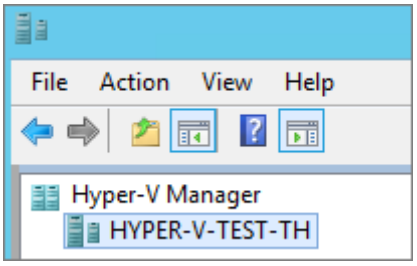
For virtual machines that run Ubuntu, see Changing Ubuntu Screen Resolution in a Hyper-V VM.

Turn on enhanced session mode on a Hyper-V host

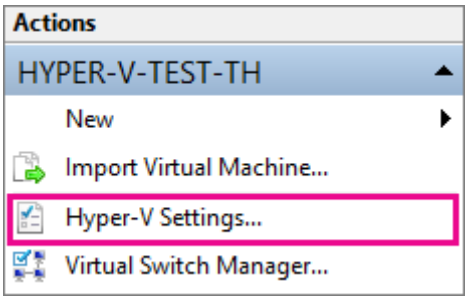
If your Hyper-V host runs Windows 10 or Windows 8.1, enhanced session mode is on by default, so you can skip this and move to the next section. But if your host runs Windows Server 2016 or Windows Server 2012 R2, do this first.

Turn on enhanced session mode:

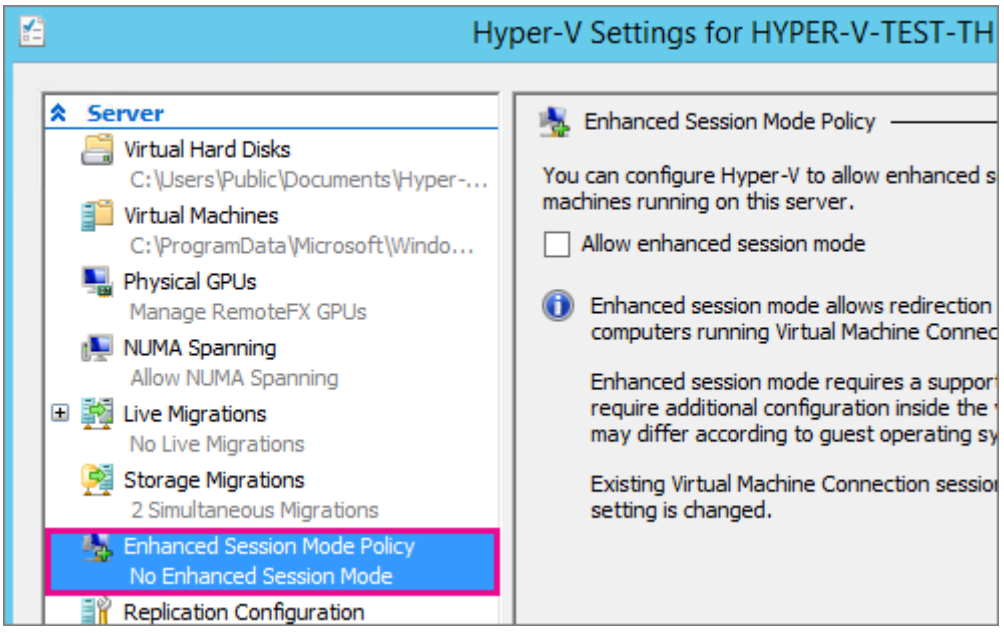
1. Connect to the computer that hosts the virtual machine.
2. In Hyper-V Manager, select the host's computer name.



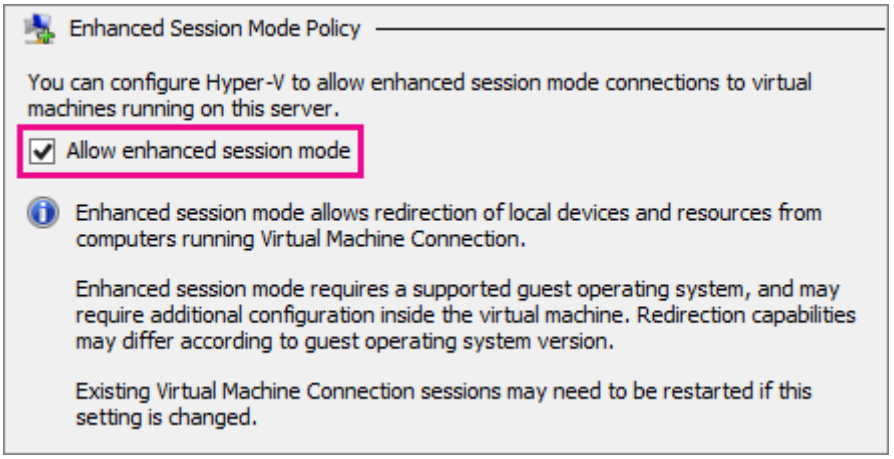
3. Select **Hyper-V settings**.



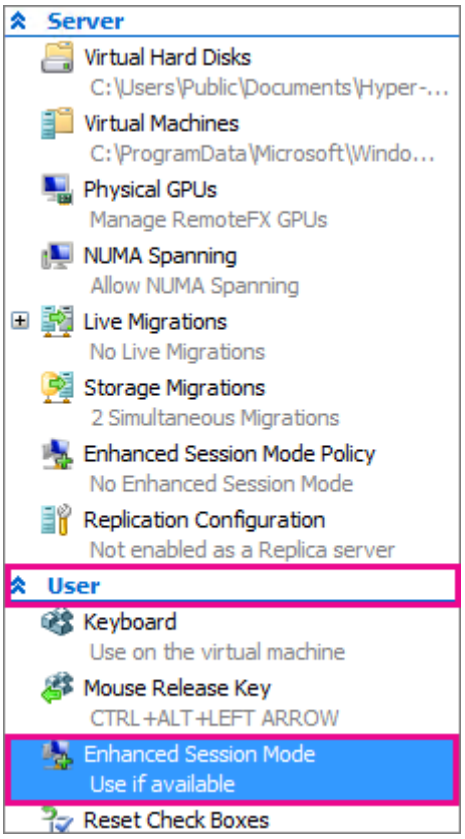
4. Under **Server**, select **Enhanced session mode policy**.



5. Select the **Allow enhanced session mode** check box.



6. Under **User**, select **Enhanced session mode**.



7. Select the **Allow enhanced session mode** check box.

8. Click **Ok**.

Choose a local resource

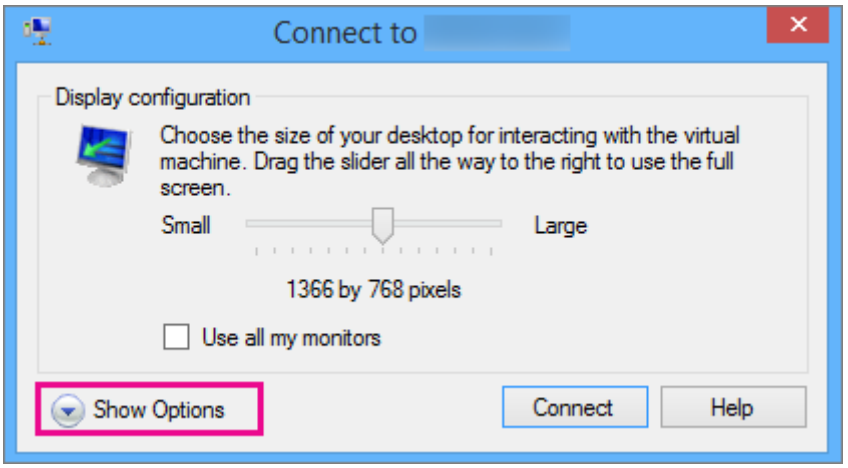
Local resources include printers, the clipboard, and local drive(s) on the computer where you're running VMConnect. For more information, see Requirements for using local resources.

To choose a local resource:

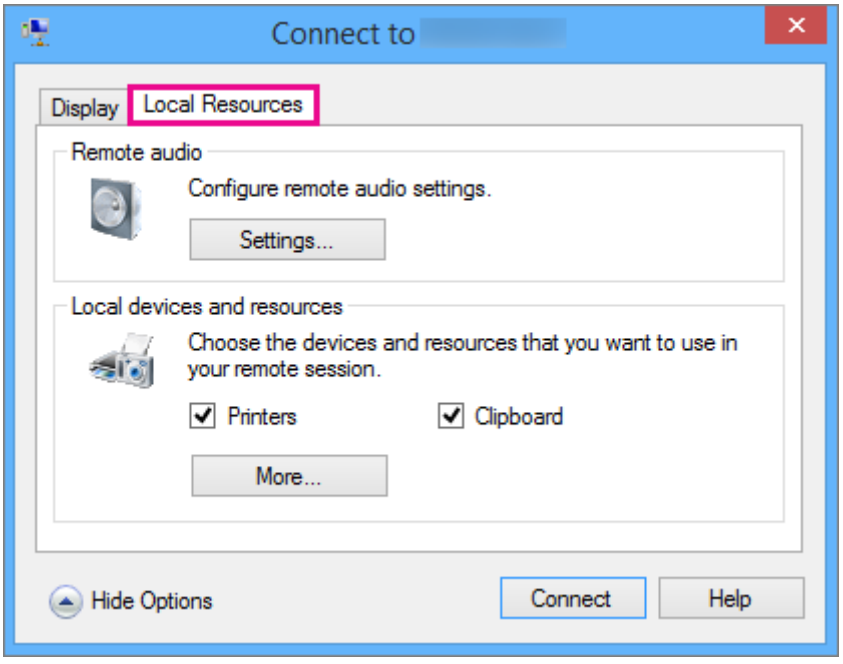
1. Open VMConnect.

2. Select the virtual machine that you want to connect to.

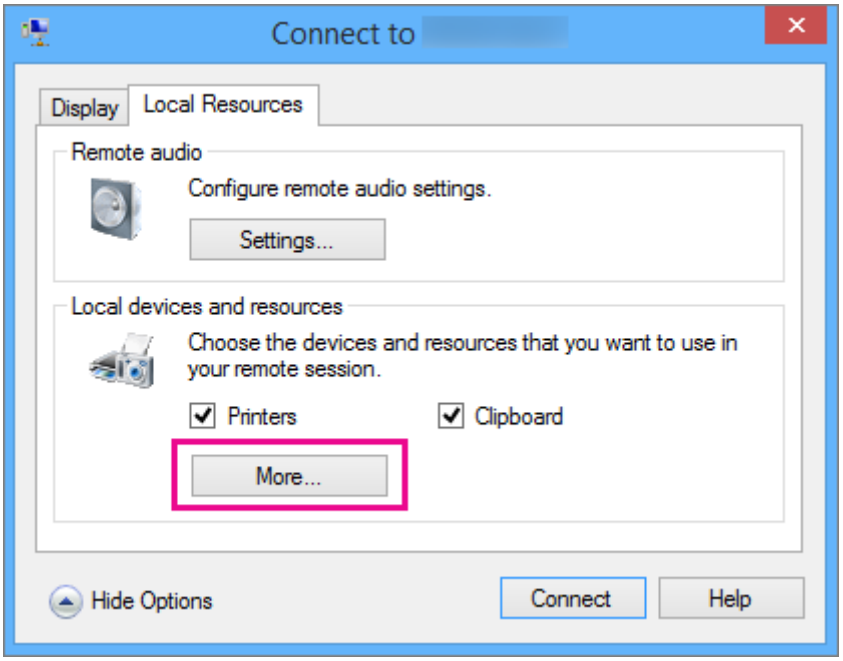
3. Click **Show options**.



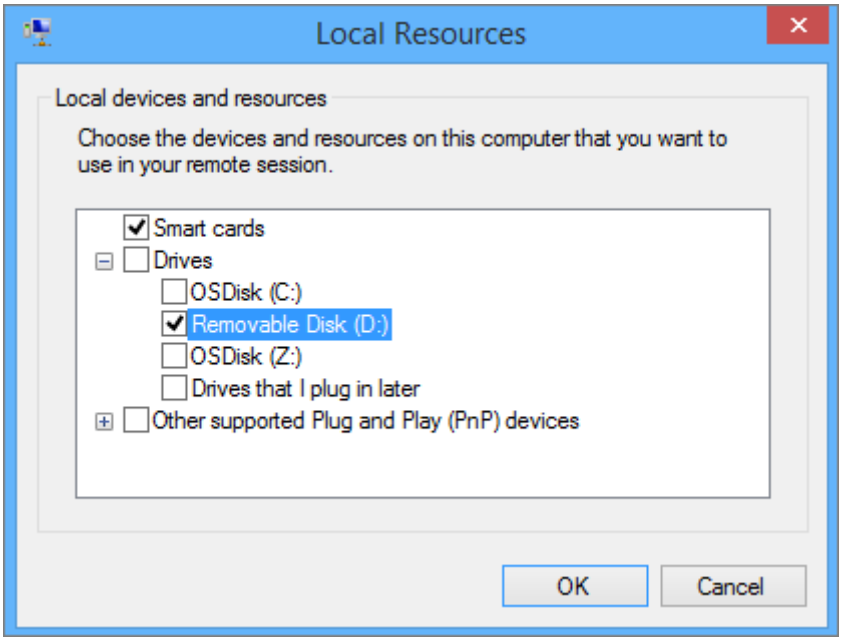
4. Select **Local resources**.



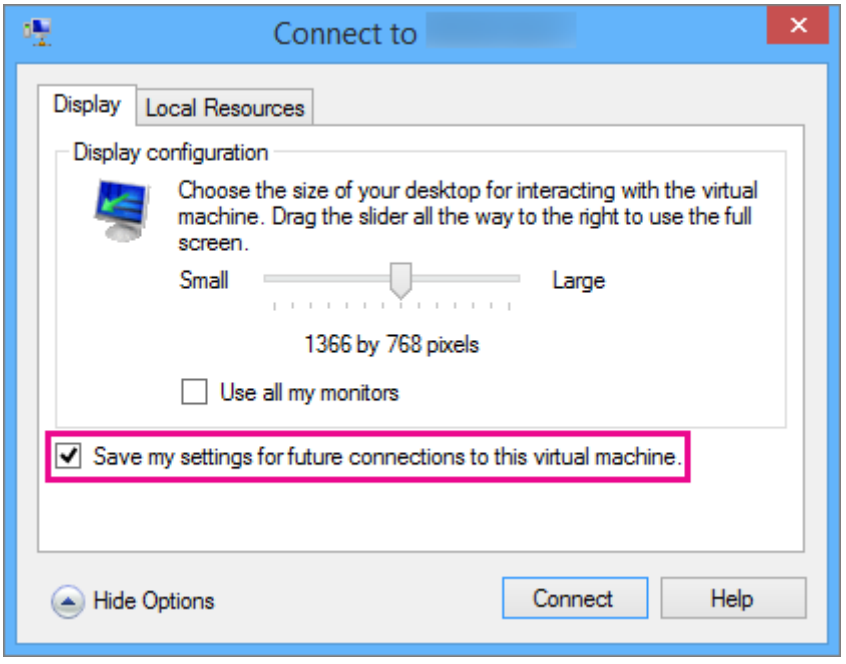
5. Click **More**.



6. Select the drive that you want to use on the virtual machine and click **Ok**.



7. Select **Save my settings for future connections to this virtual machine**.



8. Click **Connect**.

The path to the local drive shared to the virtual machine, in a Windows-based guest VM, is located at **This PC** under **Redirected drives and folders**. The path for a Linux-based guest VM is located at **/home/shared-drives**.

Edit VMConnect settings

You can easily edit your connection settings for VMConnect by running the following command in Windows PowerShell or the command prompt:

```
VMConnect.exe <ServerName> <VMName> /edit
```

> **Note**
>
> An elevated command prompt may be required.

Requirements for using local resources

To be able to use a computer's local resources on a virtual machine:

- The Hyper-V host must have **Enhanced session mode policy** and **Enhanced session mode** settings turned on.
- The computer on which you use VMConnect must run Windows 10, Windows 8.1, Windows Server 2016, or Windows Server 2012 R2.
- The virtual machine must have Remote Desktop Services enabled, and run Windows 10, Windows 8.1, Windows Server 2016, or Windows Server 2012 R2 as the guest operating system.

If the computer running VMConnect and the virtual machine both meet the requirements, you can use any of the following local resources if they're available:

- Display configuration
- Audio
- Printers
- Clipboards for copy and paste
- Smart cards
- USB devices
- Drives
- Supported plug and play devices

Why use a computer's local resources?

You may want to use the computer's local resources to:

- Troubleshoot a virtual machine without a network connection to the virtual machine.
- Copy and paste files to and from the virtual machine in the same way you copy and paste using a Remote Desktop Connection (RDP).
- Sign in to the virtual machine by using a smart card.
- Print from a virtual machine to a local printer.
- Test and troubleshoot developer applications that require USB and sound redirection without using RDP.

See also

Connect to a Virtual Machine

Should I create a generation 1 or 2 virtual machine in Hyper-V?

Feedback

Performance Tuning Hyper-V Servers

Article • 03/08/2024

Hyper-V is the virtualization server role in Windows Server and Azure Stack HCI. Virtualization servers can host multiple virtual machines that are isolated from each other but share the underlying hardware resources by virtualizing the processors, memory, and I/O devices. By consolidating servers onto a single machine, virtualization can improve resource usage and energy efficiency and reduce the operational and maintenance costs of servers. In addition, virtual machines and the management APIs offer more flexibility for managing resources, balancing load, and provisioning systems.

Additional References

- Hyper-V terminology
- Hyper-V architecture
- Hyper-V server configuration
- Hyper-V processor performance
- Hyper-V memory performance
- Hyper-V storage I/O performance
- Hyper-V network I/O performance
- Detecting bottlenecks in a virtualized environment
- Linux Virtual Machines

Hyper-V Virtual Switch

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This topic provides an overview of Hyper-V Virtual Switch, which provides you with the ability to connect virtual machines (VMs) to networks that are external to the Hyper-V host, including your organization's intranet and the Internet.

You can also connect to virtual networks on the server that is running Hyper-V when you deploy Software Defined Networking (SDN).

> **Note**
>
> In addition to this topic, the following Hyper-V Virtual Switch documentation is available.
>
> - **Manage Hyper-V Virtual Switch**
> - **Remote Direct Memory Access (RDMA) and Switch Embedded Teaming (SET)**
> - **Network Switch Team Cmdlets in Windows PowerShell**
> - **What's New in VMM 2016**
> - **Set up the VMM networking fabric**
> - **Hyper-V forum**
> - **Hyper-V: The WFP virtual switch extension should be enabled if it is required by third party extensions**
>
> For more information about other networking technologies, see **Networking in Windows Server 2016**.

Hyper-V Virtual Switch is a software-based layer-2 Ethernet network switch that is available in Hyper-V Manager when you install the Hyper-V server role.

Hyper-V Virtual Switch includes programmatically managed and extensible capabilities to connect VMs to both virtual networks and the physical network. In addition, Hyper-V Virtual Switch provides policy enforcement for security, isolation, and service levels.

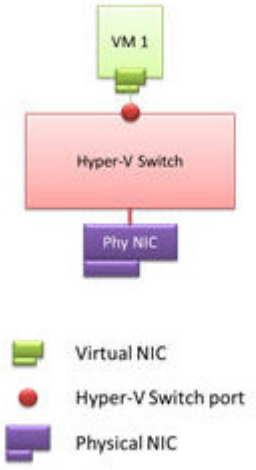
> **Note**

Hyper-V Virtual Switch only supports Ethernet, and does not support any other wired local area network (LAN) technologies, such as Infiniband and Fibre Channel.

Hyper-V Virtual Switch includes tenant isolation capabilities, traffic shaping, protection against malicious virtual machines, and simplified troubleshooting.

With built-in support for Network Device Interface Specification (NDIS) filter drivers and Windows Filtering Platform (WFP) callout drivers, the Hyper-V Virtual Switch enables independent software vendors (ISVs) to create extensible plug-ins, called Virtual Switch Extensions, that can provide enhanced networking and security capabilities. Virtual Switch Extensions that you add to the Hyper-V Virtual Switch are listed in the Virtual Switch Manager feature of Hyper-V Manager.

In the following illustration, a VM has a virtual NIC that is connected to the Hyper-V Virtual Switch through a switch port.



Hyper-V Virtual Switch capabilities provide you with more options for enforcing tenant isolation, shaping and controlling network traffic, and employing protective measures against malicious VMs.

> **Note**
>
> In Windows Server 2016, a VM with a virtual NIC accurately displays the maximum throughput for the virtual NIC. To view the virtual NIC speed in **Network Connections**, right-click the desired virtual NIC icon and then click **Status**. The virtual NIC **Status** dialog box opens. In **Connection**, the value of **Speed** matches the speed of the physical NIC installed in the server.

Uses for Hyper-V Virtual Switch

Following are some use case scenarios for Hyper-V Virtual Switch.

Displaying statistics: A developer at a hosted cloud vendor implements a management package that displays the current state of the Hyper-V virtual switch. The management package can query switch-wide current capabilities, configuration settings, and individual port network statistics using WMI. The status of the switch is then displayed to give administrators a quick view of the state of the switch.

Resource tracking: A hosting company is selling hosting services priced according to the level of membership. Various membership levels include different network performance levels. The administrator allocates resources to meet the SLAs in a manner that balances network availability. The administrator programmatically tracks information such as the current usage of bandwidth assigned, and the number of virtual machine (VM) assigned virtual machine queue (VMQ) or IOV channels. The same program also periodically logs the resources in use in addition to the per-VM resources assigned for double entry tracking or resources.

Managing the order of switch extensions: An enterprise has installed extensions on their Hyper-V host to both monitor traffic and report intrusion detection. During maintenance, some extensions may be updated causing the order of extensions to change. A simple script program is run to reorder the extensions after updates.

Forwarding extension manages VLAN ID: A major switch company is building a forwarding extension that applies all policies for networking. One element that is managed is virtual local area network (VLAN) IDs. The virtual switch cedes control of the VLAN to a forwarding extension. The switch company's installation programmatically call a Windows Management Instrumentation (WMI) application programming interface (API) that turns on the transparency, telling the Hyper-V Virtual Switch to pass and take no action on VLAN tags.

Hyper-V Virtual Switch Functionality

Some of the principal features that are included in the Hyper-V Virtual Switch are:

- **ARP/ND Poisoning (spoofing) protection**: Provides protection against a malicious VM using Address Resolution Protocol (ARP) spoofing to steal IP addresses from other VMs. Provides protection against attacks that can be launched for IPv6 using Neighbor Discovery (ND) spoofing.
- **DHCP Guard protection**: Protects against a malicious VM representing itself as a Dynamic Host Configuration Protocol (DHCP) server for man-in-the-middle attacks.

- **Port ACLs**: Provides traffic filtering based on Media Access Control (MAC) or Internet Protocol (IP) addresses/ranges, which enables you to set up virtual network isolation.
- **Trunk mode to a VM**: Enables administrators to set up a specific VM as a virtual appliance, and then direct traffic from various VLANs to that VM.
- **Network traffic monitoring**: Enables administrators to review traffic that is traversing the network switch.
- **Isolated (private) VLAN**: Enables administrators to segregate traffic on multiple vlans, to more easily establish isolated tenant communities.

Following is a list of capabilities that enhance Hyper-V Virtual Switch usability:

- **Bandwidth limit and burst support**: Bandwidth minimum guarantees amount of bandwidth reserved. Bandwidth maximum caps the amount of bandwidth a VM can consume.
- **Explicit Congestion Notification (ECN) marking support**: ECN marking, also known as Data CenterTCP (DCTCP), enables the physical switch and operating system to regulate traffic flow such that the buffer resources of the switch are not flooded, which results in increased traffic throughput.
- **Diagnostics**: Diagnostics allow easy tracing and monitoring of events and packets through the virtual switch.

Host network requirements for Azure Stack HCI

Article • 06/24/2024

> Applies to: Azure Stack HCI, versions 23H2 and 22H2

This topic discusses host networking considerations and requirements for Azure Stack HCI. For information on datacenter architectures and the physical connections between servers, see Physical network requirements.

For information on how to simplify host networking using Network ATC, see Simplify host networking with Network ATC.

Network traffic types

Azure Stack HCI network traffic can be classified by its intended purpose:

- **Management traffic:** Traffic to or from outside the local cluster. For example, storage replica traffic or traffic used by the administrator for management of the cluster like Remote Desktop, Windows Admin Center, Active Directory, etc.
- **Compute traffic:** Traffic originating from or destined to a virtual machine (VM).
- **Storage traffic:** Traffic using Server Message Block (SMB), for example Storage Spaces Direct or SMB-based live migration. This traffic is layer-2 traffic and is not routable.

> **Important**
>
> Storage replica uses non-RDMA based SMB traffic. This and the directional nature of the traffic (North-South) makes it closely aligned to that of "management" traffic listed above, similar to that of a traditional file share.

Select a network adapter

Network adapters are qualified by the **network traffic types** (see above) they are supported for use with. As you review the Windows Server Catalog, the Windows Server 2022 certification now indicates one or more of the following roles. Before purchasing a server for Azure Stack HCI, you must minimally have *at least* one adapter that is qualified for management, compute, and storage as all three traffic types are required on Azure Stack HCI. You can then use Network ATC to configure your adapters for the appropriate traffic types.

For more information about this role-based NIC qualification, please see this link.

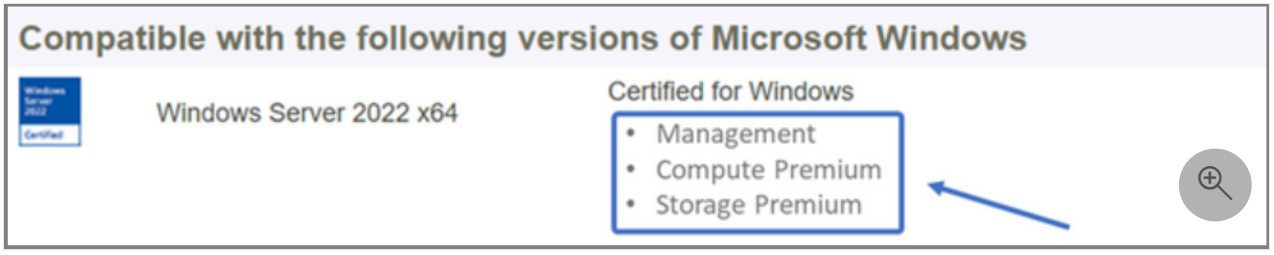
> **Important**

Using an adapter outside of its qualified traffic type is not supported.

Expand table

Level	Management Role	Compute Role	Storage Role
Role-based distinction	Management	Compute Standard	Storage Standard
Maximum Award	Not Applicable	Compute Premium	Storage Premium

> **Note**
>
> The highest qualification for any adapter in our ecosystem will contain the **Management**, **Compute Premium**, and **Storage Premium** qualifications.



Driver Requirements

Inbox drivers are not supported for use with Azure Stack HCI. To identify if your adapter is using an inbox driver, run the following cmdlet. An adapter is using an inbox driver if the **DriverProvider** property is **Microsoft.**

Powershell

```
Get-NetAdapter -Name <AdapterName> | Select *Driver*
```

Overview of key network adapter capabilities

Important network adapter capabilities used by Azure Stack HCI include:

- Dynamic Virtual Machine Multi-Queue (Dynamic VMMQ or d.VMMQ)
- Remote Direct Memory Access (RDMA)
- Guest RDMA
- Switch Embedded Teaming (SET)

Dynamic VMMQ

All network adapters with the Compute (Premium) qualification support Dynamic VMMQ. Dynamic VMMQ requires the use of Switch Embedded Teaming.

Applicable traffic types: compute

Certifications required: Compute (Premium)

Dynamic VMMQ is an intelligent, receive-side technology. It builds upon its predecessors of Virtual Machine Queue (VMQ), Virtual Receive Side Scaling (vRSS), and VMMQ, to provide three primary improvements:

- Optimizes host efficiency by using fewer CPU cores.
- Automatic tuning of network traffic processing to CPU cores, thus enabling VMs to meet and maintain expected throughput.
- Enables "bursty" workloads to receive the expected amount of traffic.

For more information on Dynamic VMMQ, see the blog post Synthetic accelerations.

RDMA

RDMA is a network stack offload to the network adapter. It allows SMB storage traffic to bypass the operating system for processing.

RDMA enables high-throughput, low-latency networking, using minimal host CPU resources. These host CPU resources can then be used to run additional VMs or containers.

Applicable traffic types: host storage

Certifications required: Storage (Standard)

All adapters with Storage (Standard) or Storage (Premium) qualification support host-side RDMA. For more information on using RDMA with guest workloads, see the "Guest RDMA" section later in this article.

Azure Stack HCI supports RDMA with either the Internet Wide Area RDMA Protocol (iWARP) or RDMA over Converged Ethernet (RoCE) protocol implementations.

> **Important**
>
> RDMA adapters only work with other RDMA adapters that implement the same RDMA protocol (iWARP or RoCE).

Not all network adapters from vendors support RDMA. The following table lists those vendors (in alphabetical order) that offer certified RDMA adapters. However, there are hardware vendors not included in this list that also support RDMA. See the Windows Server Catalog to find adapters with the Storage (Standard) or Storage (Premium) qualification which require RDMA support.

> **Note**
>
> InfiniBand (IB) is not supported with Azure Stack HCI.

Expand table

NIC vendor	iWARP	RoCE
Broadcom	No	Yes
Intel	Yes	Yes (some models)
Marvell (Qlogic)	Yes	Yes
Nvidia	No	Yes

For more information on deploying RDMA for the host, we highly recommend you use Network ATC. For information on manual deployment see the SDN GitHub repo.

iWARP

iWARP uses Transmission Control Protocol (TCP), and can be optionally enhanced with Priority-based Flow Control (PFC) and Enhanced Transmission Service (ETS).

Use iWARP if:

- You don't have experience managing RDMA networks.
- You don't manage or are uncomfortable managing your top-of-rack (ToR) switches.
- You won't be managing the solution after deployment.
- You already have deployments that use iWARP.
- You're unsure which option to choose.

RoCE

RoCE uses User Datagram Protocol (UDP), and requires PFC and ETS to provide reliability.

Use RoCE if:

- You already have deployments with RoCE in your datacenter.
- You're comfortable managing the DCB network requirements.

Guest RDMA

Guest RDMA enables SMB workloads for VMs to gain the same benefits of using RDMA on hosts.

Applicable traffic types: Guest-based storage

Certifications required: Compute (Premium)

The primary benefits of using Guest RDMA are:

- CPU offload to the NIC for network traffic processing.
- Extremely low latency.
- High throughput.

For more information, download the document from the SDN GitHub repo.

Switch Embedded Teaming (SET)

SET is a software-based teaming technology that has been included in the Windows Server operating system since Windows Server 2016. SET is the only teaming technology supported by Azure Stack HCI. SET works well with compute, storage, and management traffic and is supported with up to eight adapters in the same team.

Applicable traffic types: compute, storage, and management

Certifications required: Compute (Standard) or Compute (Premium)

SET is the only teaming technology supported by Azure Stack HCI. SET works well with compute, storage, and management traffic.

> **Important**
>
> Azure Stack HCI doesn't support NIC teaming with the older Load Balancing/Failover (LBFO). See the blog post **Teaming in Azure Stack HCI** for more information on LBFO in Azure Stack HCI.

SET is important for Azure Stack HCI because it's the only teaming technology that enables:

- Teaming of RDMA adapters (if needed).
- Guest RDMA.
- Dynamic VMMQ.
- Other key Azure Stack HCI features (see Teaming in Azure Stack HCI).

SET requires the use of symmetric (identical) adapters. Symmetric network adapters are those that have the same:

- make (vendor)
- model (version)
- speed (throughput)
- configuration

In 22H2, Network ATC will automatically detect and inform you if the adapters you've chosen are asymmetric. The easiest way to manually identify if adapters are symmetric is if the speeds and interface descriptions are **exact** matches. They can deviate only in the numeral listed in the description. Use the Get-NetAdapterAdvancedProperty cmdlet to ensure the configuration reported lists the same property values.

See the following table for an example of the interface descriptions deviating only by numeral (#):

Expand table

Name	Interface description	Link speed
NIC1	Network Adapter #1	25 Gbps
NIC2	Network Adapter #2	25 Gbps
NIC3	Network Adapter #3	25 Gbps
NIC4	Network Adapter #4	25 Gbps

> **Note**
>
> SET supports only switch-independent configuration by using either Dynamic or Hyper-V Port load-balancing algorithms. For best performance, Hyper-V Port is recommended for use on all NICs that operate at or above 10 Gbps. Network ATC makes all the required configurations for SET.

RDMA traffic considerations

If you implement DCB, you must ensure that the PFC and ETS configuration is implemented properly across every network port, including network switches. DCB is required for RoCE and optional for iWARP.

For detailed information on how to deploy RDMA, download the document from the SDN GitHub repo.

RoCE-based Azure Stack HCI implementations require the configuration of three PFC traffic classes, including the default traffic class, across the fabric and all hosts.

Cluster traffic class

This traffic class ensures that there's enough bandwidth reserved for cluster heartbeats:

- Required: Yes
- PFC-enabled: No
- Recommended traffic priority: Priority 7
- Recommended bandwidth reservation:
  - 10 GbE or lower RDMA networks = 2 percent
  - 25 GbE or higher RDMA networks = 1 percent

RDMA traffic class

This traffic class ensures that there's enough bandwidth reserved for lossless RDMA communications by using SMB Direct:

- Required: Yes
- PFC-enabled: Yes

- Recommended traffic priority: Priority 3 or 4
- Recommended bandwidth reservation: 50 percent

Default traffic class

This traffic class carries all other traffic not defined in the cluster or RDMA traffic classes, including VM traffic and management traffic:

- Required: By default (no configuration necessary on the host)
- Flow control (PFC)-enabled: No
- Recommended traffic class: By default (Priority 0)
- Recommended bandwidth reservation: By default (no host configuration required)

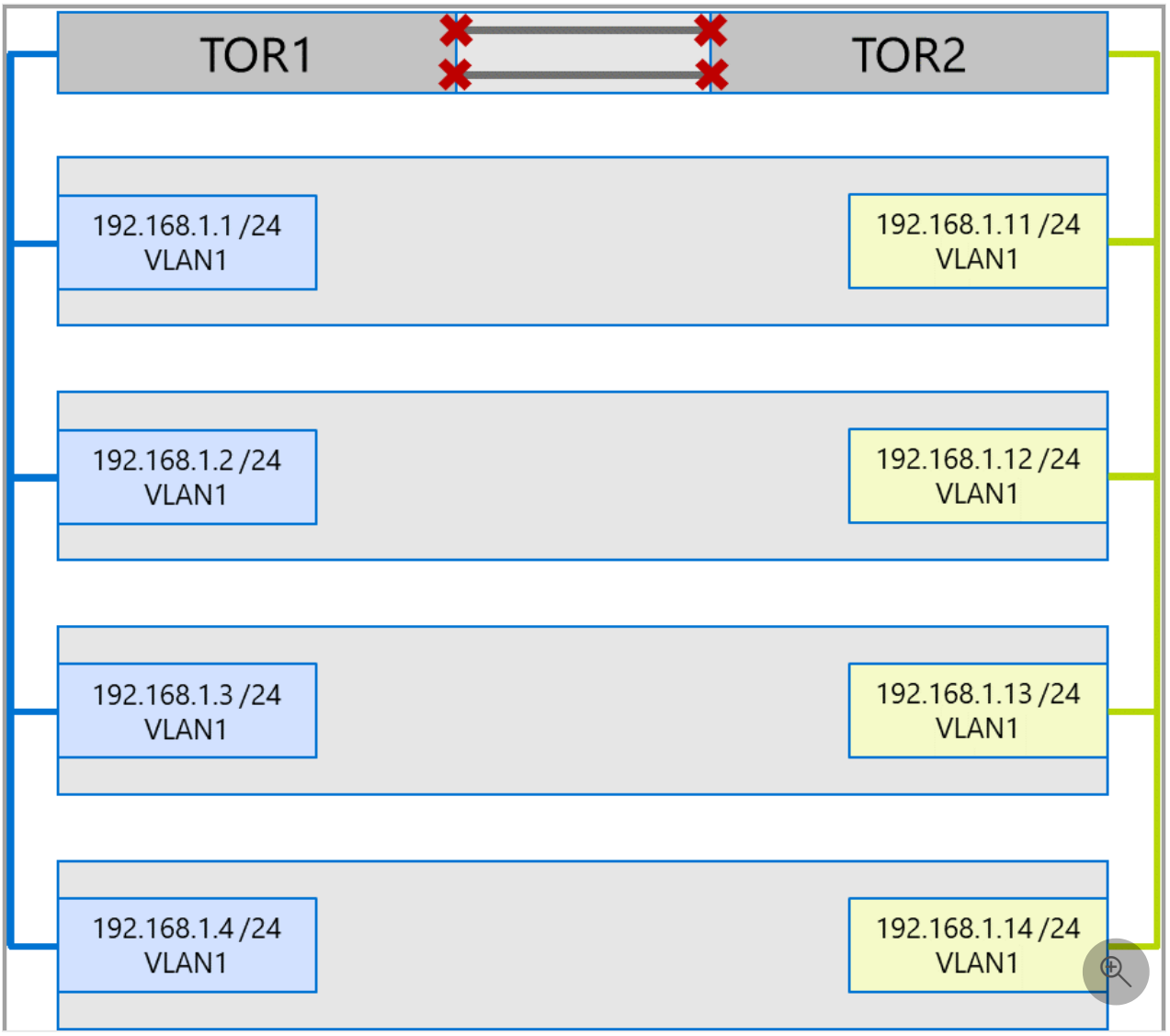
Storage traffic models

SMB provides many benefits as the storage protocol for Azure Stack HCI, including SMB Multichannel. SMB Multichannel isn't covered in this article, but it's important to understand that traffic is multiplexed across every possible link that SMB Multichannel can use.

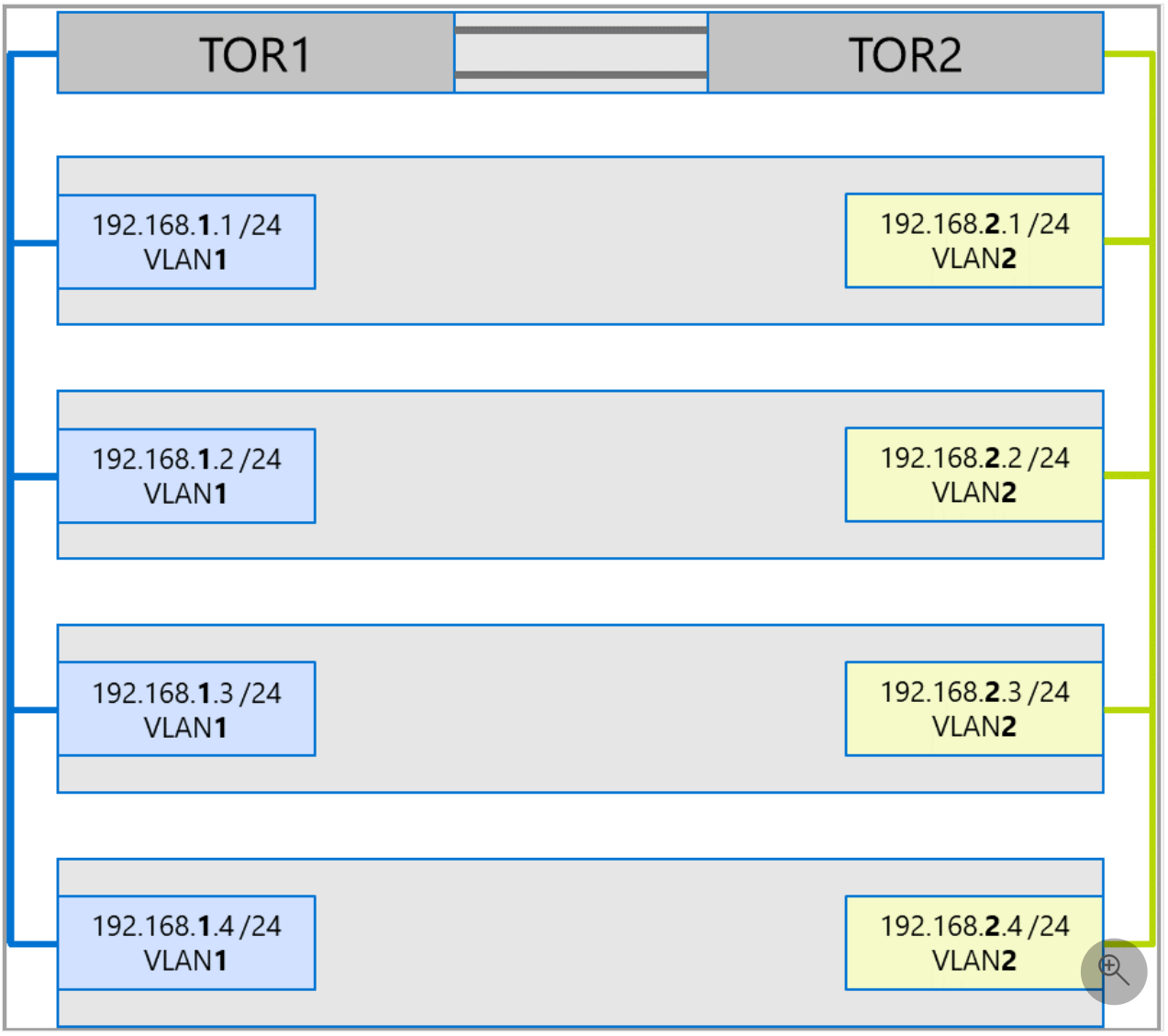
> **Note**
>
> We recommend using multiple subnets and VLANs to separate storage traffic in Azure Stack HCI.

Consider the following example of a four node cluster. Each server has two storage ports (left and right side). Because each adapter is on the same subnet and VLAN, SMB Multichannel will spread connections across all available links. Therefore, the left-side port on the first server (192.168.1.1) will make a connection to the left-side port on the second server (192.168.1.2). The right-side port on the first server (192.168.1.12) will connect to the right-side port on the second server. Similar connections are established for the third and fourth servers.

However, this creates unnecessary connections and causes congestion at the interlink (multi-chassis link aggregation group or MC-LAG) that connects the ToR switches (marked with Xs). See the following diagram:



The recommended approach is to use separate subnets and VLANs for each set of adapters. In the following diagram, the right-hand ports now use subnet 192.168.2.x /24 and VLAN2. This allows traffic on the left-side ports to remain on TOR1 and the traffic on the right-side ports to remain on TOR2.



Traffic bandwidth allocation

The following table shows example bandwidth allocations of various traffic types, using common adapter speeds, in Azure Stack HCI. Note that this is an example of a *converged solution*, where all traffic types (compute, storage, and management) run over the same physical adapters, and are teamed by using SET.

Because this use case poses the most constraints, it represents a good baseline. However, considering the permutations for the number of adapters and speeds, this should be considered an example and not a support requirement.

The following assumptions are made for this example:

- There are two adapters per team.
- Storage Bus Layer (SBL), Cluster Shared Volume (CSV), and Hyper-V (Live Migration) traffic:
  - Use the same physical adapters.
  - Use SMB.
- SMB is given a 50 percent bandwidth allocation by using DCB.

- SBL/CSV is the highest priority traffic, and receives 70 percent of the SMB bandwidth reservation.
- Live Migration (LM) is limited by using the `Set-SMBBandwidthLimit` cmdlet, and receives 29 percent of the remaining bandwidth.
  - If the available bandwidth for Live Migration is >= 5 Gbps, and the network adapters are capable, use RDMA. Use the following cmdlet to do so:

    ```
    Powershell

    Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
    ```

  - If the available bandwidth for Live Migration is < 5 Gbps, use compression to reduce blackout times. Use the following cmdlet to do so:

    ```
    Powershell

    Set-VMHost -VirtualMachineMigrationPerformanceOption Compression
    ```

- If you're using RDMA for Live Migration traffic, ensure that Live Migration traffic can't consume the entire bandwidth allocated to the RDMA traffic class by using an SMB bandwidth limit. Be careful, because this cmdlet takes entry in bytes per second (Bps), whereas network adapters are listed in bits per second (bps). Use the following cmdlet to set a bandwidth limit of 6 Gbps, for example:

  ```
  Powershell

  Set-SMBBandwidthLimit -Category LiveMigration -BytesPerSecond 750MB
  ```

  > **Note**
  >
  > 750 MBps in this example equates to 6 Gbps.

Here is the example bandwidth allocation table:

Expand table

NIC speed	Teamed bandwidth	SMB bandwidth reservation**	SBL/CSV %	SBL/CSV bandwidth	Live Migration %	Max Live Migration bandwidth	Heartbeat %	Heartbeat bandwidth
10 Gbps	20 Gbps	10 Gbps	70%	7 Gbps	*	200 Mbps		
25 Gbps	50 Gbps	25 Gbps	70%	17.5 Gbps	29%	7.25 Gbps	1%	250 Mbps

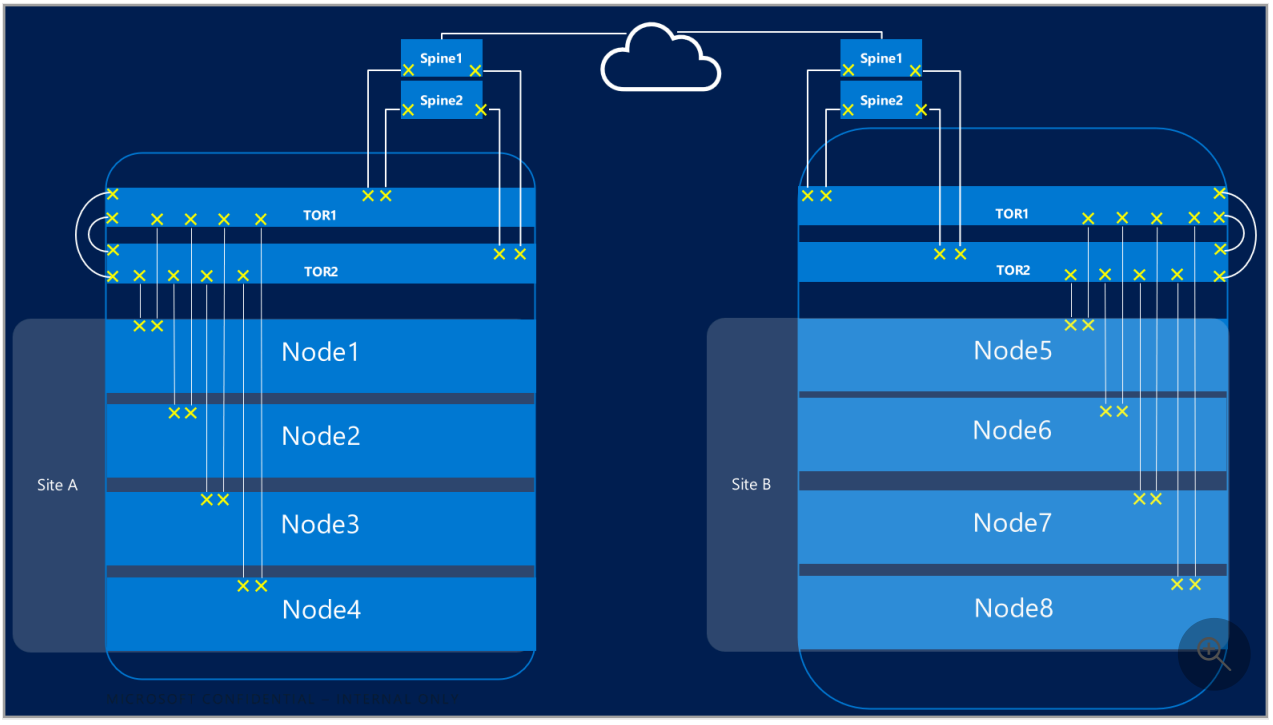
NIC speed	Teamed bandwidth	SMB bandwidth reservation**	SBL/CSV %	SBL/CSV bandwidth	Live Migration %	Max Live Migration bandwidth	Heartbeat %	Heartbeat bandwidth
40 Gbps	80 Gbps	40 Gbps	70%	28 Gbps	29%	11.6 Gbps	1%	400 Mbps
50 Gbps	100 Gbps	50 Gbps	70%	35 Gbps	29%	14.5 Gbps	1%	500 Mbps
100 Gbps	200 Gbps	100 Gbps	70%	70 Gbps	29%	29 Gbps	1%	1 Gbps
200 Gbps	400 Gbps	200 Gbps	70%	140 Gbps	29%	58 Gbps	1%	2 Gbps

* Use compression rather than RDMA, because the bandwidth allocation for Live Migration traffic is <5 Gbps.

** 50 percent is an example bandwidth reservation.

Stretched clusters

Stretched clusters provide disaster recovery that spans multiple datacenters. In its simplest form, a stretched Azure Stack HCI cluster network looks like this:



Stretched cluster requirements

> **Important**

Stretched cluster functionality is only available in Azure Stack HCI, version 22H2.

Stretched clusters have the following requirements and characteristics:

- RDMA is limited to a single site, and isn't supported across different sites or subnets.
- Servers in the same site must reside in the same rack and Layer-2 boundary.
- Host communication between sites must cross a Layer-3 boundary; stretched Layer-2 topologies aren't supported.
- Have enough bandwidth to run the workloads at the other site. In the event of a failover, the alternate site will need to run all traffic. We recommend that you provision sites at 50 percent of their available network capacity. This isn't a requirement, however, if you are able to tolerate lower performance during a failover.
- Adapters used for communication between sites:
  - Can be physical or virtual (host vNIC). If adapters are virtual, you must provision one vNIC in its own subnet and VLAN per physical NIC.
  - Must be on their own subnet and VLAN that can route between sites.
  - RDMA must be disabled by using the `Disable-NetAdapterRDMA` cmdlet. We recommend that you explicitly require Storage Replica to use specific interfaces by using the `Set-SRNetworkConstraint` cmdlet.
  - Must meet any additional requirements for Storage Replica.

Next steps

- Learn about network switch and physical network requirements. See Physical network requirements.
- Learn how to simplify host networking using Network ATC. See Simplify host networking with Network ATC.
- Brush up on failover clustering networking basics.
- See Deploy using Azure portal.
- See Deploy using Azure Resource Manager template.

Feedback

Manage Hyper-V Virtual Switch

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

You can use this topic to access Hyper-V Virtual Switch management content.

This section contains the following topics.

- Configure VLANs on Hyper-V Virtual Switch Ports
- Create Security Policies with Extended Port Access Control Lists

Create Security Policies with Extended Port Access Control Lists

Article • 07/29/2021

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This topic provides information about extended port Access Control Lists (ACLs) in Windows Server 2016. You can configure extended ACLs on the Hyper-V Virtual Switch to allow and block network traffic to and from the virtual machines (VMs) that are connected to the switch via virtual network adapters.

This topic contains the following sections.

Detailed ACL rules

Hyper-V Virtual Switch extended ACLs allow you to create detailed rules that you can apply to individual VM network adapters that are connected to the Hyper-V Virtual Switch. The ability to create detailed rules allows Enterprises and Cloud Service Providers (CSPs) to address network-based security threats in a multitenant shared server environment.

With extended ACLs, rather than having to create broad rules that block or allow all traffic from all protocols to or from a VM, you can now block or allow the network traffic of individual protocols that are running on VMs. You can create extended ACL rules in Windows Server 2016 that include the following 5-tuple set of parameters: source IP address, destination IP address, protocol, source port, and destination port. In addition, each rule can specify network traffic direction (in or out), and the action the rule supports (block or allow traffic).

For example, you can configure port ACLs for a VM to allow all incoming and outgoing HTTP and HTTPS traffic on port 80, while blocking the network traffic of all other protocols on all ports.

This ability to designate the protocol traffic that can or cannot be received by tenant VMs provides flexibility when you configure security policies.

Configuring ACL rules with Windows PowerShell

To configure an extended ACL, you must use the Windows PowerShell command **Add-VMNetworkAdapterExtendedAcl**. This command has four different syntaxes, with a distinct use for each syntax:

1. Add an extended ACL to all of the network adapters of a named VM - which is specified by the first parameter, -VMName. Syntax:

   > ⓘ **Note**
   >
   > If you want to add an extended ACL to one network adapter rather than all, you can specify the network adapter with the parameter -VMNetworkAdapterName.

   ```
   Add-VMNetworkAdapterExtendedAcl [-VMName] <string[]> [-Action]
   <VMNetworkAdapterExtendedAclAction> {Allow | Deny}
       [-Direction] <VMNetworkAdapterExtendedAclDirection> {Inbound |
   Outbound} [[-LocalIPAddress] <string>]
       [[-RemoteIPAddress] <string>] [[-LocalPort] <string>] [[-
   RemotePort] <string>] [[-Protocol] <string>] [-Weight]
       <int> [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID
   <int>] [-Passthru] [-VMNetworkAdapterName
       <string>] [-ComputerName <string[]>] [-WhatIf] [-Confirm]
   [<CommonParameters>]
   ```

2. Add an extended ACL to a specific virtual network adapter on a specific VM. Syntax:

   ```
   Add-VMNetworkAdapterExtendedAcl [-VMNetworkAdapter]
   <VMNetworkAdapterBase[]> [-Action]
       <VMNetworkAdapterExtendedAclAction> {Allow | Deny} [-Direction]
   <VMNetworkAdapterExtendedAclDirection> {Inbound |
       Outbound} [[-LocalIPAddress] <string>] [[-RemoteIPAddress]
   <string>] [[-LocalPort] <string>] [[-RemotePort]
       <string>] [[-Protocol] <string>] [-Weight] <int> [-Stateful <bool>]
   [-IdleSessionTimeout <int>] [-IsolationID
       <int>] [-Passthru] [-WhatIf] [-Confirm]  [<CommonParameters>]
   ```

3. Add an extended ACL to all virtual network adapters that are reserved for use by the Hyper-V host management operating system.

> **Note**
>
> If you want to add an extended ACL to one network adapter rather than all, you can specify the network adapter with the parameter -VMNetworkAdapterName.

```
Add-VMNetworkAdapterExtendedAcl [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow | Deny} [-Direction]
    <VMNetworkAdapterExtendedAclDirection> {Inbound | Outbound} [[-
LocalIPAddress] <string>] [[-RemoteIPAddress]
    <string>] [[-LocalPort] <string>] [[-RemotePort] <string>] [[-
Protocol] <string>] [-Weight] <int> -ManagementOS
    [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID <int>]
[-Passthru] [-VMNetworkAdapterName <string>]
    [-ComputerName <string[]>] [-WhatIf] [-Confirm]
[<CommonParameters>]
```

4. Add an extended ACL to a VM object that you have created in Windows PowerShell, such as **$vm = get-vm "my_vm"**. In the next line of code you can run this command to create an extended ACL with the following syntax:

```
Add-VMNetworkAdapterExtendedAcl [-VM] <VirtualMachine[]> [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow |
    Deny} [-Direction] <VMNetworkAdapterExtendedAclDirection> {Inbound
| Outbound} [[-LocalIPAddress] <string>]
    [[-RemoteIPAddress] <string>] [[-LocalPort] <string>] [[-
RemotePort] <string>] [[-Protocol] <string>] [-Weight]
    <int> [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID
<int>] [-Passthru] [-VMNetworkAdapterName
    <string>] [-WhatIf] [-Confirm]  [<CommonParameters>]
```

Detailed ACL rule examples

Following are several examples of how you can use the **Add-VMNetworkAdapterExtendedAcl** command to configure extended port ACLs and to create security policies for VMs.

- Enforce application-level security
- Enforce both user-level and application-level security
- Provide security support to a non-TCP/UDP application

> **Note**
>
> The values for the rule parameter **Direction** in the tables below are based on traffic flow to or from the VM for which you are creating the rule. If the VM is receiving traffic, the traffic is inbound; if the VM is sending traffic, the traffic is outbound. For example, if you apply a rule to a VM that blocks inbound traffic, the direction of inbound traffic is from external resources to the VM. If you apply a rule that blocks outbound traffic, the direction of outbound traffic is from the local VM to external resources.

Enforce application-level security

Because many application servers use standardized TCP/UDP ports to communicate with client computers, it is easy to create rules that block or allow access to an application server by filtering traffic going to and coming from the port designated to the application.

For example, you might want to allow a user to login to an application server in your datacenter by using Remote Desktop Connection (RDP). Because RDP uses TCP port 3389, you can quickly set up the following rule:

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	*	TCP	*	3389	In	Allow

Following are two examples of how you can create rules with Windows PowerShell commands. The first example rule blocks all traffic to the VM named "ApplicationServer." The second example rule, which is applied to the network adapter of the VM named "ApplicationServer," allows only inbound RDP traffic to the VM.

> **Note**
>
> When you create rules, you can use the **-Weight** parameter to determine the order in which the Hyper-V Virtual Switch processes the rules. Values for **-Weight** are expressed as integers; rules with a higher integer are processed before rules with lower integers. For example, if you have applied two rules to a VM network adapter, one with a weight of 1 and one with a weight of 10, the rule with the weight of 10 is applied first.

```
```

```
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -
Direction "Inbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Allow"
-Direction "Inbound" -LocalPort 3389 -Protocol "TCP" -Weight 10
```

Enforce both user-level and application-level security

Because a rule can match a 5-tuple IP packet (Source IP, Destination IP, Protocol, Source Port, and Destination Port), the rule can enforce a more detailed security policy than a Port ACL.

For example, if you want to provide DHCP service to a limited number of client computers using a specific set of DHCP servers, you can configure the following rules on the Windows Server 2016 computer that is running Hyper-V, where the user VMs are hosted:

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	255.255.255.255	UDP	*	67	Out	Allow
*	10.175.124.0/25	UDP	*	67	Out	Allow
10.175.124.0/25	*	UDP	*	68	In	Allow

Following are examples of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Deny" -
Direction "Outbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -
Direction "Outbound" -RemoteIPAddress 255.255.255.255 -RemotePort 67 -
Protocol "UDP"-Weight 10
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -
Direction "Outbound" -RemoteIPAddress 10.175.124.0/25 -RemotePort 67 -
Protocol "UDP"-Weight 20
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -
Direction "Inbound" -RemoteIPAddress 10.175.124.0/25 -RemotePort 68 -
Protocol "UDP"-Weight 20
```

Provide security support to a non-TCP/UDP application

While most network traffic in a datacenter is TCP and UDP, there is still some traffic that utilizes other protocols. For example, if you want to permit a group of servers to run an

IP-multicast application that relies on Internet Group Management Protocol (IGMP), you can create the following rule.

> ⓘ **Note**
>
> IGMP has a designated IP protocol number of 0x02.

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	*	0x02	*	*	In	Allow
*	*	0x02	*	*	Out	Allow

Following is an example of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -
Direction "Inbound" -Protocol 2 -Weight 20
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -
Direction "Outbound" -Protocol 2 -Weight 20
```

Stateful ACL rules

Another new capability of extended ACLs allows you to configure stateful rules. A stateful rule filters packets based on five attributes in a packet - Source IP, Destination IP, Protocol, Source Port, and Destination Port.

Stateful rules have the following capabilities:

- They always allow traffic and are not used to block traffic.
- If you specify that the value for the parameter **Direction** is inbound and traffic matches the rule, Hyper-V Virtual Switch dynamically creates a matching rule that allows the VM to send outbound traffic in response to the external resource.
- If you specify that the value for the parameter **Direction** is outbound and traffic matches the rule, Hyper-V Virtual Switch dynamically creates a matching rule that allows the external resource inbound traffic to be received by the VM.
- They include a timeout attribute that is measured in seconds. When a network packet arrives at the switch and the packet matches a stateful rule, Hyper-V Virtual

Switch creates a state so that all subsequent packets in both directions of the same flow are allowed. The state expires if there is no traffic in either direction in the period of time that is specified by the timeout value.

Following is an example of how you can use stateful rules.

Allow inbound remote server traffic only after it is contacted by the local server

In some cases, a stateful rule must be employed because only a stateful rule can keep track of a known, established connection, and distinguish the connection from other connections.

For example, if you want to allow a VM application server to initiate connections on port 80 to web services on the Internet, and you want the remote Web servers to be able to respond to the VM traffic, you can configure a stateful rule that allows initial outbound traffic from the VM to the Web services; because the rule is stateful, return traffic to the VM from the Web servers is also allowed. For security reasons, you can block all other inbound network traffic to the VM.

To achieve this rule configuration, you can use the settings in the table below.

> **Note**
>
> Due to formatting restrictions and the amount of information in the table below, the information is displayed differently than in previous tables in this document.

Parameter	Rule 1	Rule 2	Rule 3
Source IP	*	*	*
Destination IP	*	*	*
Protocol	*	*	TCP
Source Port	*	*	*
Destination Port	*	*	80
Direction	In	Out	Out
Action	Deny	Deny	Allow
Stateful	No	No	Yes

Parameter	Rule 1	Rule 2	Rule 3
Timeout (in seconds)	N/A	N/A	3600

The stateful rule allows the VM application server to connect to a remote Web server. When the first packet is sent out, Hyper-V Virtual switch dynamically creates two flow states to allow all packets sent to and all returning packets from the remote Web server. When the flow of packets between the servers stops, the flow states time out in the designated timeout value of 3600 seconds, or one hour.

Following is an example of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -
Direction "Inbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -
Direction "Outbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Allow"
-Direction "Outbound" 80 "TCP" -Weight 100 -Stateful -Timeout 3600
```