Remote Desktop Services

Remote Desktop Services let you deliver virtualized applications and provide secure remote and mobile desktop access through remote desktop sessions.

About Remote Desktop Services

OVERVIEW

What is Remote Desktop Services?

Get started

GET STARTED

Create virtual machines for Remote Desktop

Supported configurations for Remote Desktop Services

Supported security configurations for Windows 10 Virtual Desktop Infrastructure

Planning poster for Remote Desktop Services

Plan and Design

TUTORIAL

Plan and Design your Remote Desktop Services environment

ARCHITECTURE

Remote Desktop Services architecture

Build anywhere

Network guidelines

Windows Server 2025 Capacity Planning whitepaper (PDF)

Build and deploy

TUTORIAL

Build and deploy your Remote Desktop Services deployment

HOW-TO GUIDE

Deploy your Remote Desktop environment

Create a Remote Desktop Services collection

License your RDS deployment with client access licenses

Run and tune

TUTORIAL

Run and tune your Remote Desktop Services environment

HOW-TO GUIDE

Manage your personal desktop session collections

Manage users in your RDS collection

Remote Desktop IP Virtualization

Optimize Windows configuration for VDI desktops

Access your Remote Desktop resources

HOW-TO GUIDE

Get started with the Remote Desktop Connection app

Enable Remote Desktop on your PC

Allow access to your PC from outside your network

Uninstall and reinstall the Remote Desktop Connection app

REFERENCE

Available Remote Desktop clients

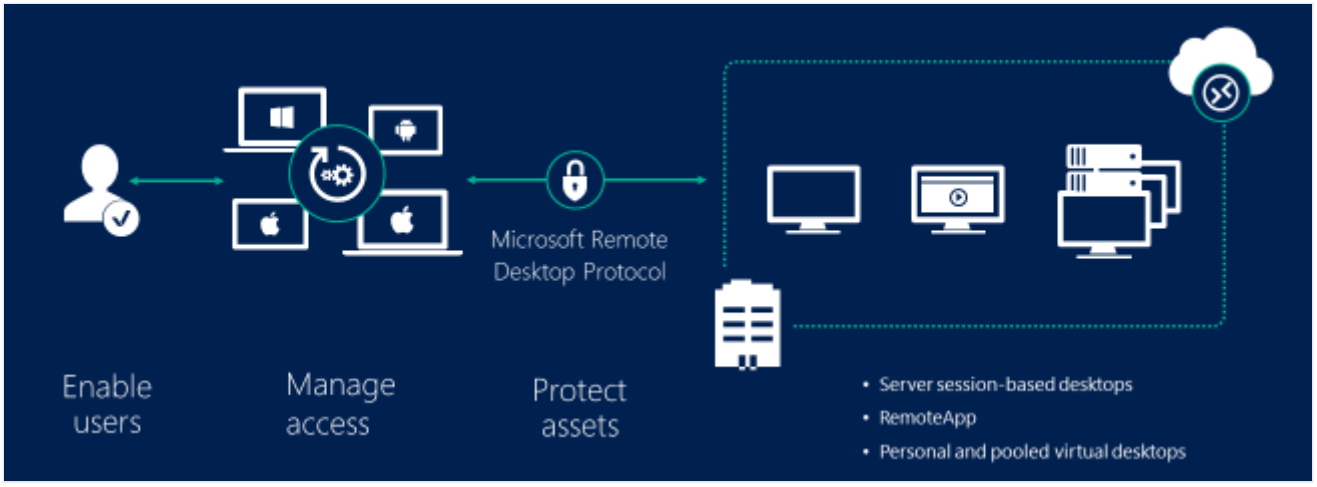
Remote Desktop Services overview in Windows Server

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Summarize this article for me

Remote Desktop Services (RDS) in Windows Server is a built-in platform for securely delivering managed desktops and applications to users, whether they're in the office, working from home, or connecting from branch and partner locations. By centralizing processing in the datacenter and remoting only the UI, Remote Desktop Services helps you reduce management overhead, improve security, and give users consistent, performant access to the resources they need.

Remote Desktop Services supports both multi-session server-based desktops and single-session (or pooled/personal) virtual desktops, plus publishing of individual applications (RemoteApp). You choose the right mix of models to optimize cost, performance, and user experience.



> **Tip**
>
> If you want to evaluate a broader cloud-based desktop solution, see Azure Virtual Desktop. You can even extend Azure Virtual Desktop to your on-premises datacenter with Azure Local.

What is Remote Desktop Services?

Remote Desktop Services is a role-based infrastructure in Windows Server that lets authorized users connect over the network to:

- A full desktop (session-based or virtual machine based).
- Specific applications (RemoteApp programs) that appear and behave like locally installed apps.

Instead of installing and patching applications on many individual endpoints, you manage them once on centralized hosts. Endpoints simply present the remote UI by using the Remote Desktop Protocol (RDP).

Key benefits

Remote Desktop Services centralizes application and desktop management so you patch and secure resources once rather than across many endpoints. Multi-session density lowers per-user cost while a mix of session hosts and VDI provides flexibility for performance or compatibility needs. Data remains in the datacenter; encrypted remote access, MFA, and auditing can strengthen your compliance posture. Users stay productive because RemoteApp windows behave like local apps, supporting taskbar pinning, multi-monitor workflows, and standard window controls. The platform extends easily with profile management, printing, monitoring, GPU acceleration, and broad automation via PowerShell.

Core Remote Desktop Services roles and components

Remote Desktop Services includes several roles that you can distribute and scale:

Expand table

Role	Purpose
RD Session Host (RDSH)	Runs session-based user desktops and RemoteApp programs on Windows Server for multi-session efficiency.
RD Virtualization Host	Hosts Virtual Desktop Infrastructure collections (pooled or personal Windows client VMs). Integrates with Hyper-V for provisioning.
RD Connection Broker	Maintains user sessions, load balances connections, reconnects users to existing sessions, manages collections (session and VDI). Supports high availability.
RD Web Access	Provides a web portal and feeds (Web Access / RemoteApp and Desktop Connections) listing desktops and apps users are authorized to use.
RD Gateway	Enables secure, encrypted RDP access over HTTPS (TCP 443) from external networks without opening internal RDP ports. Supports MFA and conditional policies.
RD Licensing	Issues and tracks Remote Desktop Services Client Access Licenses (RDS CALs)

Role	Purpose
	required for legal use (User or Device).

Supporting components often include file services for user profiles, certificate services for TLS, and monitoring solutions.

Deployment models

You can mix and match models based on user persona and technical requirements:

Expand table

Model	Description	Typical use cases	Cost & density
Session-based (RDSH)	Multiple users share a Windows Server instance; each gets an isolated session.	Task workers, line-of-business apps, seasonal users.	Highest user density, lowest per-user cost.
VDI pooled	Users connect to a dynamically assigned Windows client VM from a pool. Non-persistent or resettable state.	Knowledge workers needing Windows client compatibility; app isolation.	Medium density/cost.
VDI personal	Each user is assigned a dedicated Windows client VM that retains changes.	Developers, power users, customization-heavy apps.	Lowest density, highest flexibility.
Hybrid	Combine RDSH for baseline apps + VDI for specialized needs.	Mixed persona environments.	Optimized balance.

Hosting locations

You can deploy Remote Desktop Services in different environments:

- **On-premises**: Full control of hardware, networking, and data locality.
- **Azure infrastructure (IaaS)**: Deploy Remote Desktop Services roles on Azure VMs for elastic capacity and simplified global access.

Common scenarios

Organizations use Remote Desktop Services to standardize application delivery to branch offices, enable secure access for contractors or partners without exposing internal networks broadly, rapidly onboard seasonal or temporary staff, and host legacy or specialized Windows

applications that are impractical to deploy on many devices. It's also common in regulated sectors (finance, healthcare) where data residency and auditability matter, and for IT administrators who need remote desktop access with constrained privileges rather than full VPN tunnels.

Publishing options

You can publish full desktops (session-based or VM-based) that present a curated application set, or provide individual RemoteApp programs that integrate into the user's Start menu and taskbar, can span multiple monitors, and appear and behave like locally installed software.

Security and compliance

Remote Desktop Services builds on Windows security: TLS protects RDP traffic (deploy valid certificates for Gateway, Broker, and Web Access), and RD Gateway encapsulates RDP in HTTPS to minimize exposed ports while enabling conditional access and MFA integration. Collections and security groups help enforce least-privilege segmentation; centralized execution keeps data resident in the datacenter so only the UI stream leaves. Event logs and Windows auditing feed monitoring or SIEM solutions for compliance and forensic review.

Planning considerations

Effective planning starts with user personas: estimate CPU, memory, storage I/O profiles for task, knowledge, and power users. From there, model capacity for sessions per RD Session Host and sizing for pooled or personal VDI, while designing Connection Broker high availability. Profile strategy (roaming profiles, folder redirection, or third-party profile management) impacts logon performance and disk growth; avoid uncontrolled profile sprawl on hosts. If users need 3D or rich multimedia, plan GPU resources (Hyper-V DDA or supported virtualization technologies). Network topology and latency drive user experience, so place RD Gateway to minimize round-trip time and ensure sufficient bandwidth for peak concurrency. Validate application multi-session behavior early to catch assumptions about per-machine paths or registry keys, and script deployment, scaling, and maintenance with PowerShell to reduce manual effort.

Management and monitoring

Daily operations blend GUI and automation: Server Manager and PowerShell handle role installation, collection creation, and application publishing; the Connection Broker manages load balancing and user assignment across hosts. Monitor sessions and resource consumption

(CPU, memory, disk, GPU) and review authentication or disconnection events for trends. Maintain custom images for pooled VDI and stagger patching of session hosts to keep capacity available. Scheduled scripts streamline certificate renewal, image refresh, and scale-out/scale-in adjustments.

Next steps

To get started with Remote Desktop Services, review the following articles:

- Supported configurations
- Planning and design for capacity, high availability, MFA, and certificates
- Architecture models

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

---

Last updated on 10/30/2025

Create virtual machines for Remote Desktop

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Use the following steps to create the virtual machines in the tenant's environment that will be used to run the Windows Server 2016 roles, services, and features required for a desktop hosting deployment.

For this example of a basic deployment, the minimum of 3 virtual machines will be created. One virtual machine will host the Remote Desktop (RD) Connection Broker and License Server role services and a file share for the deployment. A second virtual machine will host the RD Gateway and Web Access role services. A third virtual machine host the RD Session Host role service. For very small deployments, you can reduce VM costs by using Microsoft Entra App Proxy to eliminate all public endpoints from the deployment and combining all the role services onto a single VM. For larger deployments, you can install the various role services on individual virtual machines to allow better scaling.

This section outlines the steps necessary to deploy virtual machines for each role based on Windows Server images in the Microsoft Azure Marketplace. If you need to create virtual machines from a custom image, which requires PowerShell, check out Create a Windows VM with Resource Manager and PowerShell. Then return here to attach Azure data disks for the file share and enter an external URL for your deployment.

1. Create Windows virtual machines to host the RD Connection Broker, RD License Server, and File server.

   For our purpose, we used the following naming conventions:

   - RD Connection Broker, License Server, and File Server:
     - VM: Contoso-Cb1
     - Availability set: CbAvSet
   - RD Web Access and RD Gateway Server:
     - VM: Contoso-WebGw1
     - Availability set: WebGwAvSet
   - RD Session Host:
     - VM: Contoso-Sh1

- Availability set: ShAvSet

Each VM uses the same resource group.

2. Create and attach an Azure data disk for the user profile disk (UPD) share:
   a. In the Azure portal click **Browse** > **Resource groups**, click the resource group for the deployment, and then click the VM created for the RD Connection Broker (for example, Contoso-Cb1).
   b. Click **Settings** > **Disks** > **Attach new**.
   c. Accept the defaults for name and type.
   d. Enter a size (in GB) that is large enough to hold network shares for the tenant's environment, including user profile disks and certificates. You can approximate 5 GB per user you plan to have
   e. Accept the defaults for location and host caching, and then click **OK**.

3. Create an external load balancer to access the deployment externally:
   a. In the Azure portal click **Browse** > **Load balancers**, and then click **Add**.
   b. Enter a **Name**, select **Public** as the **Type** of load balancer, and select the appropriate **Subscription**, **Resource Group**, and **Location**.
   c. Select **Choose a public IP address**, **Create new**, enter a name, and select **Ok**.
   d. Select **Create** to create the load balancer.

4. Configure the external load balancer for your deployment
   a. In the Azure portal click **Browse** > **Resource groups**, click the resource group for the deployment, and then click the load balancer you created for the deployment.
   b. Add a backend pool for the load balancer to send traffic to:
      i. Select **Backend pool** and **Add**.
      ii. Enter a **Name** and select **+ Add a virtual machine**.
      iii. Select **Availability set** and **WebGwAvSet**.
      iv. Select **Virtual machines**, **Contoso-WebGw1**, **Select**, **OK**, and **OK**.
   c. Add a probe so the load balancer knows what machines are active:
      i. Select **Probes** and **Add**.
      ii. Enter a **Name** (like HTTPS), select **TCP**, enter **Port** 443, and select **OK**.
   d. Enter load balancing rules to balance the incoming traffic:
      i. Select **Load balancing rules** and **Add**
      ii. Enter a **Name** (like HTTPS), select **TCP**, and 443 for both the **Port** and the **Backend port**.

          - For a Windows 10 and Windows Server 2016 Deployment, leave **Session persistence** as **None**, otherwise select **Client IP**.

      iii. Select **OK** to accept the HTTPS rule.

iv. Create a new rule by selecting **Add**.
v. Enter a **Name** (like UDP), select **UDP**, and 3391 for both the **port and the **Backend port**.

- For a Windows 10 and Windows Server 2016 deployment, leave **Session persistence** as **None**, otherwise select **Client IP**.

vi. Select **OK** to accept the UDP rule.

e. Enter an inbound NAT rule to directly connect to Contoso-WebGw1
i. Select **Inbound NAT rules** and **Add**.
ii. Enter a **Name** (like RDP-Contoso-WebGw1), select **Customm** for the service, **TCP** for the protocol, and enter 14000 for the **Port**.
iii. Select **Choose a virtual machine** and Contoso-WebGw1.
iv. Select **Custom** for the port mapping, enter 3389 for the **Target port**, and select **OK**.

5. Enter an external URL/DNS name for your deployment to access it externally:
a. In the Azure portal, click **Browse** > **Resource groups**, click the resource group for the deployment, and then click the public IP address you created for RD Web Access and RD Gateway.
b. Click **Configuration**, enter a DNS name label (like contoso), and then click **Save**. This DNS name label (contoso.westus.cloudapp.azure.com) is the DNS name that you'll use to connect to your RD Web Access and RD Gateway server.

Feedback

Supported configurations for Remote Desktop Services

07/07/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When it comes to the supported configurations for Remote Desktop Services (RDS) environments, the largest concern tends to be version interoperability. Most environments include multiple versions of Windows Server. For example, you might have an existing RDS deployment running an earlier version of Windows Server but want to upgrade to a later version of Windows Server to take advantage of the new features. The question then becomes: which RDS components can work with different versions and which need to use a consistent version?

This article provides basic guidelines for supported configurations of RDS in Windows Server.

> **Note**
>
> Be sure to review the system requirements for Windows Server.

Best practices

- Use the most recent version of Windows Server for your Remote Desktop infrastructure (the Web Access, Gateway, Connection Broker, and license server). Windows Server is backward-compatible with these components. So a Windows Server 2022 RD Session Host can connect to a Windows Server 2025 RD Connection Broker, but not the other way around.
- For RD Session Hosts, all Session Hosts in a collection need to be at the same level, but you can have multiple collections. For example, you can have a collection with Windows Server 2019 Session Hosts and one with Windows Server 2025 Session Hosts.
- An RDS license server can only process client access licenses (CALs) from the same or previous versions of Windows Server. So, if you upgrade your RD Session Host to Windows Server 2025, you also need to upgrade the license server.
- Follow the upgrade order recommended in Upgrading your Remote Desktop Services environment.

- If you're creating a highly available environment, all of your Connection Brokers need to be at the same OS level.

RD Connection Brokers

Starting in Windows Server 2016, there's no restriction for the number of Connection Brokers you can have in a deployment when using Remote Desktop Session Hosts (RDSH) and Remote Desktop Virtualization Hosts (RDVH). The following table shows which versions of RDS components work in a highly available deployment with three or more Connection Brokers.

Expand table

3 or more Connection Brokers in HA	RDSH or RDVH 2025	RDSH or RDVH 2022	RDSH or RDVH 2019	RDSH or RDVH 2016
Windows Server 2025 Connection Broker	Supported	Supported	Supported	Supported
Windows Server 2022 Connection Broker	N/A	Supported	Supported	Supported
Windows Server 2019 Connection Broker	N/A	N/A	Supported	Supported
Windows Server 2016 Connection Broker	N/A	N/A	N/A	Supported

Support for graphics processing unit (GPU) acceleration

RDS supports systems equipped with GPUs. Applications that require a GPU can be used over the remote connection. Additionally, GPU-accelerated rendering and encoding can be enabled for improved app performance and scalability.

Remote Desktop Session Hosts and single-session client operating systems can take advantage of the physical or virtual GPUs presented to the operating system in many ways, including the Azure GPU optimized virtual machine sizes, GPUs available to the physical RDSH server, and GPUs presented to the VMs by supported hypervisors.

See Which graphics virtualization technology is right for you? for help with figuring out what you need. For specific information about Discrete Device Assignment, see Plan for deploying Discrete Device Assignment.

GPU vendors might have a separate licensing scheme for RDSH scenarios or restrict GPU use on the server OS. Verify the requirements with your vendor.

GPUs presented by a non-Microsoft hypervisor or cloud platform must have drivers digitally signed by WHQL and supplied by the GPU vendor.

Remote Desktop Session Host support for GPUs

The following table shows the scenarios supported by different versions of RDSH hosts.

Expand table

Feature	Windows Server 2016	Windows Server 2019	Windows Server 2022	Windows Server 2025
Use of hardware GPU for all RDP sessions	Yes	Yes	Yes	Yes
H.264/AVC hardware encoding (if supported by the GPU)	Yes	Yes	Yes	Yes
Load balancing between multiple GPUs presented to the OS	No	Yes	Yes	Yes
H.264/AVC encoding optimizations for minimizing bandwidth usage	No	Yes	Yes	Yes
H.264/AVC support for 4K resolution	No	Yes	Yes	Yes

VDI support for GPUs

The following table shows support for GPU scenarios in the client OS.

Expand table

Feature	Windows 7 SP1	Windows 8.1	Windows 10
Use of hardware GPU for all RDP sessions	No	Yes	Yes
H.264/AVC hardware encoding (if supported by the GPU)	No	No	Windows 10 1703 or later
Load balancing between multiple GPUs presented to the OS	No	No	Windows 10 1803 or later

Feature	Windows 7 SP1	Windows 8.1	Windows 10
H.264/AVC encoding optimizations for minimizing bandwidth usage	No	No	Windows 10 1803 or later
H.264/AVC support for 4K resolution	No	No	Windows 10 1803 or later

RemoteFX 3D Video Adapter (vGPU) support

> **Note**
>
> Because of security concerns, RemoteFX vGPU is disabled by default on all versions of Windows starting with the July 14, 2020 Security Update and removed starting with the April 13, 2021 Security Update. To learn more, see KB 4570006.

RDS supports RemoteFX vGPUs when the VM is running as a Hyper-V guest on Windows Server. The following guest operating systems have RemoteFX vGPU support:

- Windows 11
- Windows 10
- Windows Server, in a single-session deployment only

Discrete Device Assignment support

RDS supports physical GPUs presented with Discrete Device Assignment from Hyper-V hosts running Windows Server 2016 or later. See Plan for deploying Discrete Device Assignment for more details.

VDI deployment – supported guest operating systems

Windows Server RD Virtualization Host servers support the following guest operating systems:

- Windows 11 Enterprise
- Windows 10 Enterprise

> **Note**

- RDS doesn't support heterogeneous session collections. The operating systems of all VMs in a collection must be the same version.
- You can have separate homogeneous collections with different guest OS versions on the same host.
- The Hyper-V host used to run VMs must be the same version as the Hyper-V host used to create the original VM templates.

Single sign-on

RDS in Windows Server supports two main single sign-on (SSO) experiences:

- In-app (Remote Desktop application on Windows, iOS, Android, and Mac)
- Web SSO

Using the Remote Desktop application, you can store credentials either as part of the connection info (Mac) or as part of managed accounts (iOS, Android, Windows) securely through the mechanisms unique to each OS.

To connect to desktops and RemoteApps with SSO through the inbox Remote Desktop Connection client on Windows, you must connect to the RD Web page through Internet Explorer. The following configuration options are required on the server side. No other configurations are supported for Web SSO.

- RD Web set to form-based authentication (Default)
- RD Gateway set to password authentication (Default)
- RDS Deployment set to "Use RD Gateway credentials for remote computers" (Default) in the RD Gateway properties

> **Note**
>
> Due to the required configuration options, Web SSO isn't supported with smart cards. Users who sign in by using smart cards might face multiple prompts to sign in.

For more information about creating a VDI deployment of RDS, see Supported Windows 10 security configurations for Remote Desktop Services VDI.

Using RDS with application proxy services

You can use RDS with Microsoft Entra application proxy. RDS doesn't support Web Application Proxy.

Supported Windows security configurations for Remote Desktop Services VDI

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Windows and Windows Server have new layers of protection built into the operating system to:

- Safeguard against security breaches
- Help block malicious attacks
- Enhance the security of virtual machines, applications, and data.

> **Note**
>
> - Features like Credential Guard may have performance implication on user density. Ensure to test your scenarios. Learn more about other considerations for credential guard configuration.
> - Make sure to review the Remote Desktop Services supported configuration information.

The following table outlines which of these new features are supported in a VDI deployment using RDS.

Expand table

VDI collection type	Managed pooled	Managed personal	Unmanaged pooled	Unmanaged personal
Credential Guard	Yes	Yes	Yes	Yes
Device Guard	Yes	Yes	Yes	Yes
Remote Credential Guard	No	No	No	No
Shielded & Encryption Supported VMs	No	No	Encryption supported VMs with extra configuration	Encryption supported VMs with extra configuration

Remote Credential Guard

Remote Credential Guard is only supported for direct connections to the target machines and not for the ones via Remote Desktop Connection Broker and Remote Desktop Gateway.

> **Note**
>
> If you have a Connection Broker in a single-instance environment, and the DNS name matches the computer name, you may be able to use Remote Credential Guard, although this isn't supported.

Shielded VMs and Encryption Supported VMs

Shielded VMs aren't supported in Remote Desktop Services VDI.

For leveraging Encryption Supported VMs:

- Use an unmanaged collection and a provisioning technology outside of the Remote Desktop Services collection creation process to provision the virtual machines.
- User Profile Disks aren't supported as they rely on differential disks

---

Feedback

Remote Desktop Services - planning poster

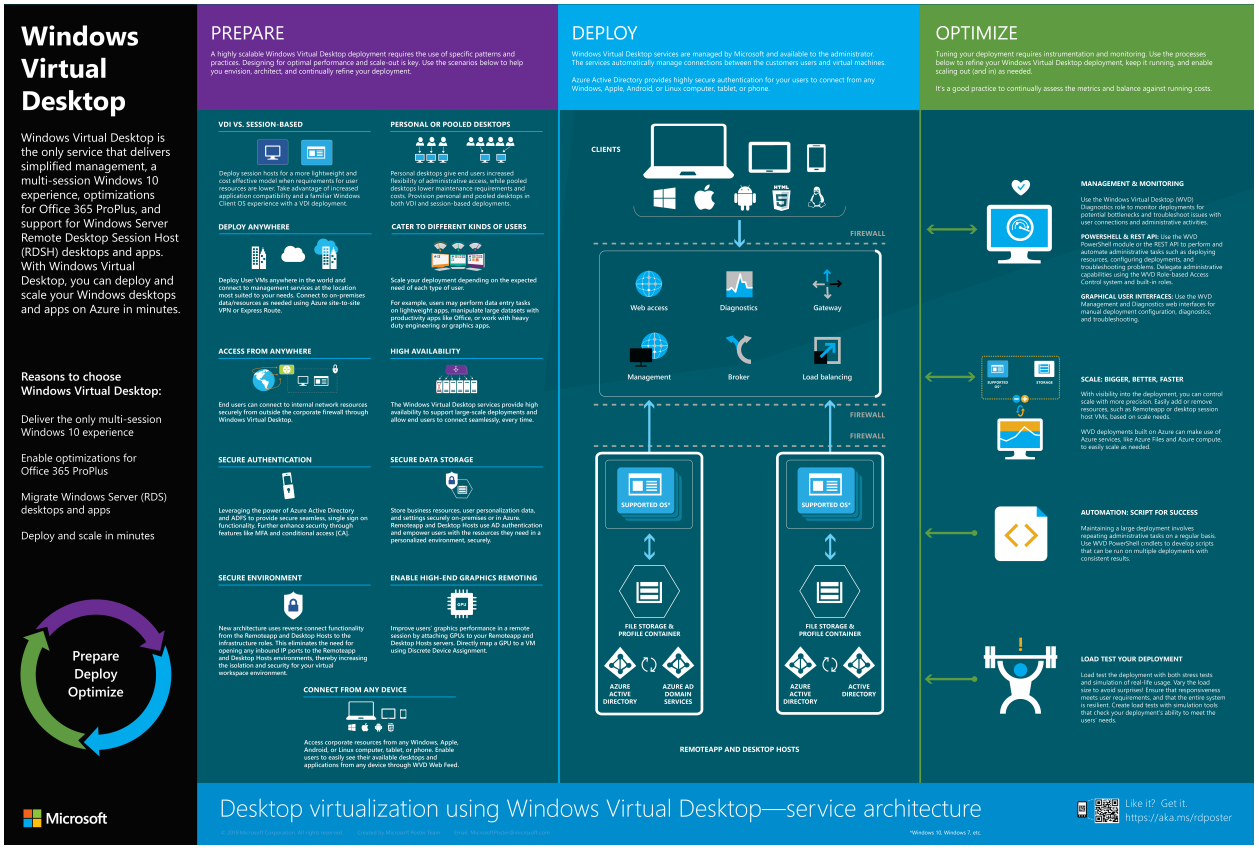
Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Azure Virtual Desktop

You may have heard us talk about a new "modern infrastructure" for Remote Desktop. Maybe you've heard us use the phrase "RDmi." The phrase you need to know is "Azure Virtual Desktop." Learn more at our Azure Virtual Desktop documentation page.

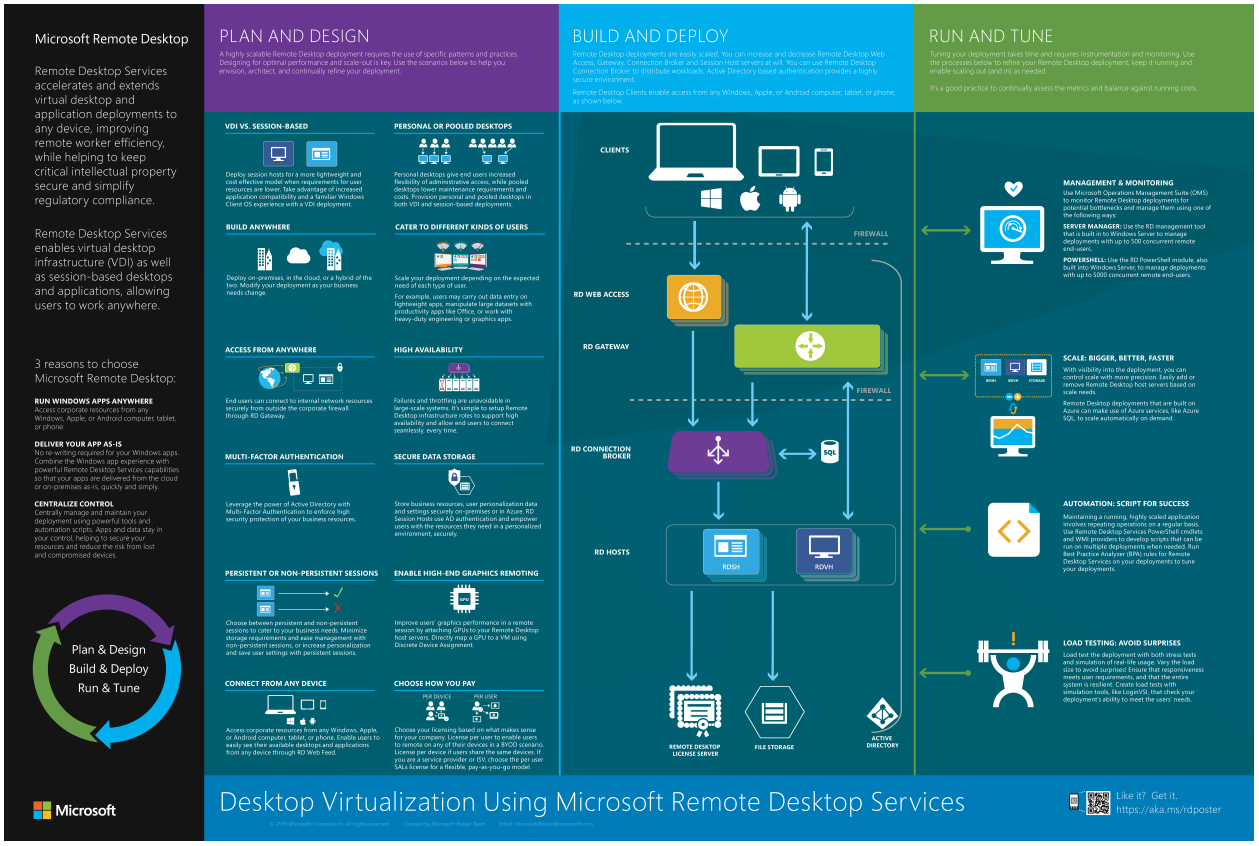
The Remote Desktop Services team have created a poster to help you plan, build, and run your Azure Virtual Desktop environment.



You can get a copy of the poster by right-clicking the image and saving it to your local system.

Remote Desktop Services in Windows Server

The Remote Desktop Services team have created a poster to help you plan, build, and run your RDS environment.



You can get a copy of the poster by right-clicking the image and saving it to your local system.

Check out the following topics to learn more about planning:

- Plan and design your RDS deployment
- Build and deploy RDS
- Run and tune your RDS environment

Feedback

Was this page helpful? Yes No

Plan and design your Remote Desktop Services environment

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Summarize this article for me

A highly scalable Remote Desktop deployment requires the use of specific patterns and practices. Designing for optimal performance and scale-out is key. Use the scenarios below to help you envision, architect, and continually refine your deployment.

Use the following information to plan and design your deployment:

- Build anywhere
- Network guidance
- Access from anywhere
- High availability
- MultiFactor Authentication
- Secure data storage
- GPU acceleration
- Connect from any device
- Choose how you pay

Be sure to also review the following architecture and capacity planning resources:

- Desktop Hosting Reference Architecture, which provides an overview of the Remote Desktop architecture and helps you plan a hybrid RDS environment that includes Azure infrastructure.
- Windows Server 2025 Capacity Planning whitepaper (PDF), which provides authoritative guidance for planning Windows Server 2025 Remote Desktop Services capacity.

Last updated on 02/24/2026

Remote Desktop Services - Build anywhere

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Deploy on-premises, in the cloud, or a hybrid of the two. Modify your deployment as your business needs change.

Regardless of where you are, the underlying architecture of the Remote Desktop Services environment remains the same:

- You still must have an internet-facing server to utilize RD Web Access and RD Gateway for external users
- You still must have an Active Directory and--for highly available environments--a SQL database to house user and Remote Desktop properties
- You still must have communication access between the RD infrastructure roles (RD Connection Broker, RD Gateway, RD Licensing, and RD Web Access) and the end RDSH or RDVH hosts to be able to connect end-users to their desktops or applications.

This flexibility allows you to get the best of both worlds:

- The simplicity and pay-as-you-go methods associated with the cloud and the online world.
- The familiarity and hassle-free way of leveraging heavy resources that already exist on-premises.

For additional information, look at how to build and deploy your Remote Desktop Services deployment.

Feedback

Network guidelines

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When using a remote Windows session, your network's available bandwidth greatly impacts the quality of your experience. Different applications and display resolutions require different network configurations, so it's important to make sure your network is configured to meet your needs.

> **Note**
>
> The following recommendations apply to networks with less than 0.1% loss. These recommendations apply regardless of how many sessions you're hosting on your virtual machines (VMs).

Applications

The following table lists the minimum recommended bandwidths for a smooth user experience. These recommendations are based on the guidelines in Remote Desktop workloads.

Expand table

Workload type	Recommended bandwidth
Light	1.5 Mbps
Medium	3 Mbps
Heavy	5 Mbps
Power	15 Mbps

Keep in mind that the stress put on your network depends on both your app workload's output frame rate and your display resolution. If either the frame rate or display resolution increases, the bandwidth requirement will also rise. For example, a light workload with a high-resolution display requires more available bandwidth than a light workload with regular or low resolution.

Other scenarios can have their bandwidth requirements change depending on how you use them, such as:

- Voice or video conferencing
- Real-time communication
- Streaming 4K video

Make sure to load test these scenarios in your deployment using simulation tools like Login VSI. Vary the load size, run stress tests, and test common user scenarios in remote sessions to better understand your network's requirements.

Display resolutions

Different display resolutions require different available bandwidths. The following table lists the bandwidths we recommend for a smooth user experience at typical display resolutions with a frame rate of 30 frames per second (fps). These recommendations apply to single and multiple user scenarios. Keep in mind that scenarios involving a frame rate under 30 fps, such as reading static text, require less available bandwidth.

Expand table

Typical display resolutions at 30 fps	Recommended bandwidth
About 1024 × 768 px	1.5 Mbps
About 1280 × 720 px	3 Mbps
About 1920 × 1080 px	5 Mbps
About 3840 × 2160 px (4K)	15 Mbps

Azure Virtual Desktop experience estimator

The Azure region you're in can affect user experience as much as network conditions. Check out the Azure Virtual Desktop experience estimator to learn more.

Assistive technologies

Assistive technology workloads, like using Narrator in the remote session, require connections with a connection round trip time (RTT) of 20 milliseconds (ms) or better for the best user experience.

Feedback

Was this page helpful? Yes No

Session host virtual machine sizing guidelines for Azure Virtual Desktop and Remote Desktop Services

09/30/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Session host sizing for Azure Virtual Desktop and Remote Desktop Services requires careful consideration of workload types and hardware configurations. Different types of workloads require different hardware configurations to ensure optimal performance.

There are two types of session hosts to consider when sizing them appropriately for users:

- **Single-session**: dedicated to a single user at a time.
- **Multi-session**: shared between multiple users concurrently.

In an environment where desktops and apps are accessed remotely, execution and data processing occurs on the session host, unless apps support local offload. Correctly sizing both each session host and the number of session hosts is important so you don't run out of resources during peak loads, which would otherwise lead to disruption to users.

Session hosts can run on virtual machines or on physical hardware for Remote Desktop Services. Virtual machines do have some overhead, so you should account for that when sizing your session hosts, which is covered in this article.

The examples in this article are generic guidelines, and you should only use them for initial performance estimates. For the best possible experience, scale your deployment depending on your users' needs.

Capacity planning

The capacity and resources you need to provide is different for everyone as it depends on many contributing factors. Capacity planning is the process of determining the session hosts and their resources required to meet the expected workload demands. It involves analyzing current and future resource needs, estimating the number of users per session host, and determining the appropriate sizes to ensure optimal performance.

When planning capacity for session hosts, consider the following areas:

Expand table

Area	Description
User workload	Understand the types of apps and tasks users perform. Different workloads have varying resource requirements, such as CPU, memory, and storage.
User count	Estimate the number of concurrent users who access the session hosts. This helps determine the required resources to support the expected user load.
Resource requirements	Analyze the resource requirements of the apps and tasks users perform. This includes CPU, memory, storage, and network bandwidth.
Performance expectations	Define the performance expectations for the session hosts, such as response time, logon time, application launch time, and overall user experience. Consider logon performance for key times, such as the start of a work day or shift as these can impact performance compared to a steady state.
Scalability	Consider the ability to scale the session hosts as user demands increase. This might involve adding more session hosts or resizing existing ones to accommodate extra users or workloads.
Resilience and redundancy	Consider implementing redundancy and failover mechanisms to ensure high availability and minimize downtime if there are hardware or software failures.
Monitoring and optimization	Implement monitoring tools to track resource utilization and performance metrics. Use this data to optimize the session hosts and make ongoing adjustments as needed.

The following two approaches are commonly used to help you determine the capacity of session hosts:

- **Pilot approach**: deploy a single test server and gradually increase the load while monitoring user feedback and system performance indicators such as CPU, paging, disk, and network. This approach is reliable for smaller deployments but might require initial hardware investments that might not meet final deployment goals.
- **Simulation approach**: use automation tools to generate simulated user workloads that mimic real user behavior. Typically simulation involves gradually increasing the number of simulated users over time and performance metrics are collected throughout the test. Analysis helps identify the point where performance degrades beyond acceptable thresholds. This approach is more suitable for larger deployments where accurate capacity determination significantly influences purchasing decisions.

Piloting tends to be more time and cost effective for smaller deployments, while the simulation approach might be more suitable for larger deployments where accurately determining session host capacity can significantly influence purchasing decisions.

Whichever approach you use, you also need to consider key times for user logon, such as the start of a work day or shift, which can impact performance compared to a steady state and cause long logon times. A session host might be able to support enough users for a certain scenario, but it might not have the capacity to service those users all logging on concurrently. Also plan for some headroom to accommodate unexpected spikes in user activity or resource demands.

We recommend you document the capacity planning process, including assumptions, calculations, and decisions made. Communicate the plan to stakeholders to ensure alignment and understanding.

Key factors affecting capacity and performance

There are several key factors that affect the capacity of session hosts. Understanding these factors can help you make informed decisions about sizing and scaling your session hosts.

- **CPU scaling**:
  - The number of CPU cores directly impacts the number of users that can be supported on a session host VM.
  - Doubling the number of CPU cores doesn't necessarily double the user capacity due to diminishing returns and synchronization overhead. The scaling factor is higher when the initial number of CPUs is small, and it decreases as the number of cores get higher. For example, the scaling factor going from 4 cores to 8 cores is larger than the one going from 8 cores to 16 cores.
  - The scaling factor typically ranges between 1.5 and 1.9, meaning that for every extra core, you can expect a proportional increase in user capacity, but not a linear one.

- **Memory impact**:
  - The amount of memory allocated to a session host VM directly affects the number of users it can support.
  - When memory is the limiting factor, adding more memory at lower capacities can significantly improve performance. For example, increasing memory from 8 GB to 16 GB can more than double the number of users you can support.

- **User logon impact**:
  - User logon is a CPU-intensive operation, and high concurrent logon rates can significantly impact system performance.
  - Plan for expected logon patterns, such as the start of a work day at 9 AM, where many users log on simultaneously. Otherwise users might experience extended logon times.

- **Virtualization overhead**

    - Running on virtual machines can incur a 15-20% capacity cost compared to bare metal, based on internal testing.
    - A hypervisor introduces more latency and CPU overhead that can result in user response times being 10% to 20% higher than on bare metal.

- **Hyperthreading benefits**
    - Hyperthreading can improve user capacity by allowing more threads to run concurrently on each core, making more efficient use of the processor's resources.
    - The benefits of hyperthreading vary depending on the workload and the number of cores. Workloads that are less CPU-intensive can benefit from extra parallel processing capabilities and achieve better performance through hyperthreading.

- **Network performance**
    - Network latency, packet loss and jitter can impact user experience, especially for applications that require frequent communication with remote servers or databases. Any combination of high latency, packet loss, and jitter can lead to slower response times and degraded performance.
    - Lower network RTT, packet loss, and jitter lead to faster response times and better overall performance. Consider using low-latency network connections to minimize the impact of network performance on user experience.

- **Storage performance**
    - Storage performance can impact user experience, especially for applications that require frequent disk access.
    - Use high-performance storage solutions, such as SSDs or NVMe drives, to ensure fast data access and minimize latency.

- **Graphics processing unit (GPU) requirements**
    - Some workloads, such as graphics-intensive applications for video rendering, 3D design, and simulations or virtual desktops with high-resolution displays, might require dedicated GPUs to ensure optimal performance.
    - Consider using session hosts with GPU capabilities if your users run graphics-intensive applications or require high-resolution displays.

All these factors can impact the overall performance and capacity of session hosts. The measurement of user input delay, or end-to-end session response time, is a key metric to consider when evaluating the performance for users. This metric measures the time it takes for a user's input to be processed and reflected in the session, providing a more accurate representation of user experience. Users generally expect a response time of less than 200 milliseconds for their actions, and any delay beyond that can lead to a degraded user experience. For more information about measuring user experience, see Use performance counters to diagnose app performance problems on Remote Desktop Session Hosts.

Workloads

When sizing session hosts, it's important to consider the type of workload that users run as they can be significantly different. For example, light data entry workers have low resource utilization that would lead to a high user density. However, expert workers using heavy 3D apps consume higher resources that would lead to low user density with the same hardware.

Here's an example that categorizes workloads into four types: *light*, *medium*, *heavy*, and *power*. Each workload type has different resource requirements and user expectations.

The following table describes each workload. *Example users* are the types of users that might find each workload most helpful. *Example apps* are the kinds of apps that work best for each workload.

Expand table

Workload type	Example users	Example apps
Light	Users doing basic data entry tasks	Database entry apps, command-line interfaces
Medium	Consultants and market researchers	Database entry apps, command-line interfaces, Microsoft Word, static web pages
Heavy	Software engineers, content creators	Database entry apps, command-line interfaces, Microsoft Word, static web pages, Microsoft Outlook, Microsoft PowerPoint, dynamic web pages, software development
Power	Graphic designers, 3D model makers, machine learning researchers	Database entry apps, command-line interfaces, Microsoft Word, static web pages, Microsoft Outlook, Microsoft PowerPoint, dynamic web pages, photo and video editing, computer-aided design (CAD), computer-aided manufacturing (CAM)

Single-session session host sizing recommendations

In a *single-session* scenario, only one user is signed in to a session host at any one time. For example, if you use personal host pools in Azure Virtual Desktop, you're using a single-session scenario.

These sizing recommendations for single-session scenarios are based on Azure VMs. You can also use these figures as a baseline for physical session hosts, consider your capacity planning approach to refine these recommendations for your workloads.

We recommend you use at least two physical CPU cores per VM, typically four vCPUs with hyper-threading. If you need more specific VM sizing recommendations for single-session scenarios, ask the software vendors specific to your workload. VM sizing for single-session session hosts usually align with physical device guidelines.

The following table shows examples of typical workloads:

Expand table

Workload type	vCPU/RAM/OS storage minimum	Example Azure instances	Profile container storage minimum
Light	2 vCPUs, 8 GB RAM, 32 GB storage	D2s_v5, D2s_v4	30 GB
Medium	4 vCPUs, 16 GB RAM, 32 GB storage	D4s_v5, D4s_v4	30 GB
Heavy	8 vCPUs, 32 GB RAM, 32 GB storage	D8s_v5, D8s_v4	30 GB

Multi-session session host sizing recommendations

In a *multi-session* scenario, more than one user is signed in to a session host virtual machine at any given time. For example, when you use pooled host pools in Azure Virtual Desktop with the Windows 11 Enterprise multi-session operating system (OS), that's a multi-session deployment.

A multi-session computing environment experiences significantly higher peak loads compared to single-session environments. A session host with a specific hardware capacity has a maximum workload limit that it can support before its resources are exhausted.

These sizing recommendations for multi-session scenarios are based on Azure VMs. You can also use these figures as a baseline for physical session hosts, consider your capacity planning approach to refine these recommendations for your workloads.

The following table lists the maximum suggested number of users per virtual central processing unit (vCPU) and the minimum VM configuration for a standard or larger user workload. If you need more specific VM sizing recommendations for single-session scenarios, ask the software vendors specific to your workload.

Expand table

Workload type	Maximum users per vCPU	Minimum vCPU/RAM/OS storage	Example Azure instances	Minimum profile storage
Light	6	8 vCPUs, 16 GB RAM, 32 GB storage	D8s_v5, D8s_v4, F8s_v2, D8as_v4, D16s_v5, D16s_v4, F16s_v2, D16as_v4	30 GB
Medium	4	8 vCPUs, 16 GB RAM, 32 GB storage	D8s_v5, D8s_v4, F8s_v2, D8as_v4, D16s_v5, D16s_v4, F16s_v2, D16as_v4	30 GB
Heavy	2	8 vCPUs, 16 GB RAM, 32 GB storage	D8s_v5, D8s_v4, F8s_v2, D8as_v4, D16s_v5, D16s_v4, F16s_v2, D16as_v4	30 GB
Power	1	6 vCPUs, 56 GB RAM, 340 GB storage	D16ds_v5, D16s_v4, D16as_v4, NV6, NV16as_v4	30 GB

For multi-session workloads, you should limit VM size to between 4 vCPUs and 24 vCPUs for the following reasons:

- All VMs should have more than two cores. The UI components in Windows rely on the use of at least two parallel threads for some of the heavier rendering operations. For multi-session scenarios, having multiple users on a two-core VM leads to the UI and apps becoming unstable, which lowers the quality of user experience. Four cores are the lowest recommended number of cores that a stable multi-session VM should have.
- VMs shouldn't have more than 32 cores. As the number of cores increases, the system's synchronization overhead also increases. For most workloads, at around 16 cores, the return on investment gets lower, with most of the extra capacity offset by synchronization overhead. User experience is better with two 16-core VMs instead of one 32-core VM.

The recommended range between 4 and 24 cores generally provides better capacity returns for your users as you increase the number of cores. For example, if you have 12 users sign in at the same time to a VM that has four cores, the ratio is three users per core. On a VM with 8 cores and 14 users, the ratio is 1.75 users per core. In this scenario, the latter configuration with a ratio of 1.75 offers greater burst capacity for applications that have short-term CPU demand.

This recommendation is true at a larger scale. For scenarios with 20 or more users connected to a single VM, several smaller VMs would perform better than one or two large VMs. For example, if you're expecting 30 or more users to sign in within 10 minutes of each other on the same session host with 16 cores, two 8-core VMs would handle the workload better. You can also use breadth-first load balancing to evenly distribute users across different VMs instead of depth-first load balancing, where you can only use a new session host after the existing one is full of users.

It's also better to use a large number of smaller VMs instead of a few large VMs. It's easier to shut down VMs that need to be updated or aren't currently in use. With larger VMs, you're more likely to have at least one user signed in at any given time, which prevents you from shutting down the VM. When you have many smaller VMs, it's more likely you have some VMs without active users. You can safely shut down these unused VMs to conserve resources, either manually or automatically by using autoscale in Azure Virtual Desktop. Conserving resources makes your deployment more resilient, easier to maintain, and less expensive.

Related content

- Use performance counters to diagnose app performance problems on Remote Desktop Session Hosts.

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

Remote Desktop Services - Access from anywhere

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

End users can connect to internal network resources securely from outside the corporate firewall through RD Gateway.

Regardless of how you configure the desktops for your end-users, you can easily plug the RD Gateway into the connection flow for a fast, secure connection. For end-users connecting through published feeds, you can configure the RD Gateway property as you configure the overall deployment properties. For end-users connecting through to their desktops without a feed, they can easily add the name of the organization's RD Gateway as a connection property no matter which Remote Desktop client application they use.

The three primary purposes of the RD Gateway, in the order of the connection sequence, are:

1. **Establish an encrypted SSL tunnel between the end-user's device and the RD Gateway Server**: In order to connect through any RD Gateway server, the RD Gateway server must have a certificate installed that the end-user's device recognizes. In testing and proofs of concepts, self-signed certificates can be used, but only publicly trusted certificates from a certificate authority should be used in any production environment.
2. **Authenticate the user into the environment**: The RD Gateway uses the inbox IIS service to perform authentication, and can even utilize the RADIUS protocol to leverage multi-factor authentication solutions such as Azure MFA. Aside from the default policies created, you can create additional RD Resource Authorization Policies (RD RAPs) and RD Connection Authorization Policies (RD CAPs) to more specifically define which users should have access to which resources within the secure environment.
3. **Pass traffic back and forth between the end-user's device and the specified resource**: The RD Gateway continues to perform this task for as long as the connection is established. You can specify different timeout properties on the RD Gateway servers to maintain the security of the environment in case the user walks away from the device.

You can find additional details on the overall architecture of a Remote Desktop Services deployment in the desktop hosting reference architecture.

Feedback

Was this page helpful? Yes No

Remote Desktop Services - High availability

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Failures and throttling are unavoidable in large-scale systems. It's simple to set up Remote Desktop infrastructure roles to support high availability and allow end users to connect seamlessly, every time.

In Remote Desktop Services, the following items represent the Remote Desktop infrastructure roles, with their respective guidance to establish high availability:

- Remote Desktop Connection Broker
- Remote Desktop Gateway
- Remote Desktop Licensing
- Remote Desktop Web Access

High availability is established by duplicating each of the roles services on a second machines. In Azure, you can receive a guaranteed uptime by placing the set of the two virtual machines (hosting the same role) in an availability sets.

Along with availability sets, you can now leverage the power of Azure SQL Database and its Azure-backed SLA to ensure that you always have connection information and can redirect users to their desktops and applications.

For best practices on creating your RDS environment, please see the desktop hosting architecture.

Feedback

Plan multifactor authentication for Remote Desktop Services

06/30/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Multifactor Authentication (MFA) enhances the security of Remote Desktop Services (RDS) by requiring users to verify their identity through multiple authentication methods. This added layer of protection helps safeguard sensitive resources and reduces the risk of unauthorized access. This article provides an overview of the user experience, architecture components, and security benefits of integrating MFA with RDS, along with planning considerations for deployment.

User Experience

For your end-users connecting to their desktops and applications, the experience is similar to any other second authentication measure to connect to the desired resource. The process typically involves:

1. Launch a desktop or RemoteApp.
2. Upon connecting to the RD Gateway for secure, remote access, receive a mobile application MFA challenge.
3. Correctly authenticate and get connected to their resources.

Architecture components

RDS can integrate with Network Policy Server in Windows Server and its extension to Microsoft Entra ID. A typical RDS MFA implementation includes:

- **RD Gateway**: Acts as the entry point for remote connections.
- **Network Policy Server (NPS)**: Processes authentication requests and enforces policies.
- **Microsoft Entra ID**: Provides cloud-based MFA services.
- **NPS Extension**: Bridges on-premises NPS with cloud-based MFA.

For more information, see Integrate your Remote Desktop Gateway infrastructure using the Network Policy Server (NPS) extension and Microsoft Entra ID.

When planning your MFA deployment, you should consider the following components:

- **User enrollment**: Plan how users will register their MFA methods.
- **Backup authentication**: Configure alternative methods for users who lose primary devices.
- **Conditional access**: Consider implementing location-based or device-based policies.
- **Network requirements**: Ensure firewall rules allow communication with MFA service endpoints.

Security Benefits

Implementing MFA with Remote Desktop Services provides:

- **Reduced breach risk**: Even if passwords are compromised, extra verification prevents unauthorized access.
- **Compliance support**: Helps meet regulatory requirements for secure remote access.
- **Centralized management**: Unified MFA policies across cloud and on-premises resources.
- **Audit capabilities**: Detailed logging of authentication attempts and methods used.

Remote Desktop Services - Secure data storage with UPDs

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Store business resources, user personalization data, and settings securely on-premises or in Azure. RD Session Hosts use AD authentication and empower users with the resources they need in a personalized environment, securely.

Ensuring users have a consistent experience, regardless of the endpoint from which they access their remote resources, is an important aspect of managing an RDS deployment. User Profile Disks (UPDs) allow user data, customizations, and application settings to follow a user within a single collection. A UPD is a per-user, per-collection VHD file saved in a central share that is mounted to a user's session when they sign in - the UPD is treated as a local drive for the duration of that session.

From the user's perspective, the UPD provides a familiar experience - they save their documents to their Documents folder (on what appears to be a local drive), change their app settings as usual, and make any customizations to their Windows environment. All this data, including the registry hive, is stored on the UPD and persists in a central network share. UPDs are only available to the user when the user is actively connected to a desktop or RemoteApp. UPDs can only roam within a collection because the user's entire `C:\Users\<username\>` directory (including AppData\Local) is stored on the UPD.

You can use PowerShell cmdlets to designate the path to the central share, the size of each UPD, and which folders should be included or excluded from the user profile saved to the UPD. Alternatively, you can enable UPDs through Server Manager by going to **Remote Desktop Services** > **Collections** > **Desktop Collection** > **Desktop Collection Properties** > **User Profile Disks**. Note that you enable or disable UPDs for all users of an entire collection, not for specific users in that collection. UPDs must be stored on a central file share where the servers in the collection have full control permissions.

You can achieve high availability for your UPDs by storing them in Azure with Storage Spaces Direct.

Feedback

Was this page helpful? Yes No

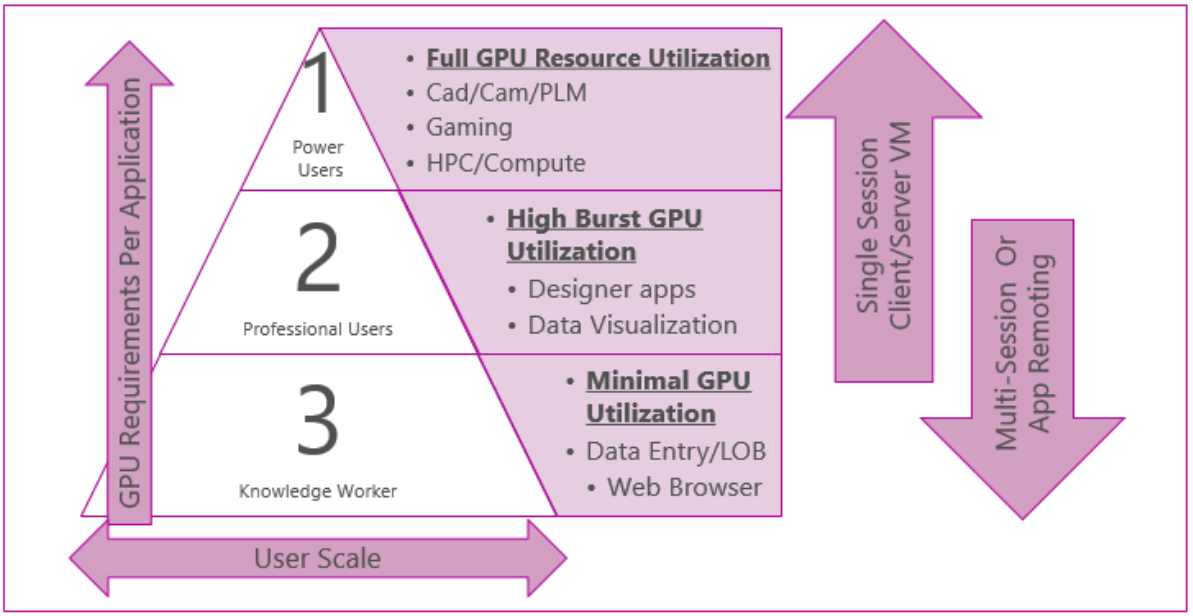
Remote Desktop Services - GPU acceleration

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) works with native graphics acceleration and the graphics virtualization technologies supported by Windows Server. For information on those technologies, their differences, and how to deploy them, see Plan for GPU acceleration in Windows Server.

When you plan for graphics acceleration in your RDS environment, your choice of user scale and user workloads drives which graphics rendering technology you use:



Feedback

Was this page helpful? Yes No

Remote Desktop Services - Connect from any device

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Access corporate resources from any Windows, Apple, or Android computer, tablet, or phone. Enable users to easily see their available desktops and applications from any device through RD Web Feed.

Learn more about Microsoft Remote Desktop clients.

Feedback

Remote Desktop Services - Choose how you pay

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Choose your licensing based on what makes sense for your company. License per user to enable users to remote on any of their devices in a BYOD scenario. License per device if users share the same devices. If you are a service provider (HSP or MSP) or ISV, choose the per user SALs license for a flexible, pay-as-you-go model.

For more information, check out License your RDS deployment with client access licenses (CALs).

Feedback

Office 2016 in RDSH and VDI Deployments

Article • 01/03/2024

Use the following information to plan how best to integrate Office 2016 into your Remote Desktop (RDSH) and VDI deployments.

Outlook 2016

In pooled VDI and RDSH deployments, using search within Outlook has limitations. Search indexing depends on the machine ID, which is different for different VMs. It's possible that every time a user logs into a pooled VDI infrastructure, they're directed to a new VM. That would mean, if we enable local search, the indexer will run every time the machine ID changes (when the user is on a different VM). Depending on the size of the .OST file, the indexer could take a long time to complete and use up resources needed for other apps. Search wouldn't only be slow but might not produce results. Using an Online Mode account profile would work around this, but overall performance would suffer due to the lack of a local cache.

Learn more about the difference between cached and online mode

Outlook 2016 has a solution to tackle this in cached mode by providing a new service search experience for mailboxes hosted on Exchange 2016 (or hosted in Office 365). This uses service search results against the local cache (OST). Outlook might fall back to using the local search indexer in some scenarios, but most searches would use this new service search feature. The recommendation for pooled VDI and RDSH deployments would be to use Outlook 2016 in cache mode with network connectivity to allow service search.

Learn how to configure cached exchange mode in Outlook 2016

OneDrive

The OneDrive Desktop App isn’t supported for client sessions that are hosted on Windows 2008 Terminal Services or Windows 2012 Remote Desktop Services (RDS) in non-persistent environments. Persistent Virtual Desktop Infrastructure (VDI) environments are supported. For more information, see Use the sync app on virtual desktops.

Skype for Business

Skype for Business isn't supported for RDSH deployments. For VDI deployments, check out the documentation on planning for Skype for Business in VDI environments.

Feedback

Was this page helpful? Yes No

Provide product feedback

Dealing with Outlook search in non-persistent RDS environments

Article • 12/01/2023

Applies To: Windows Server (Semi-Annual Channel), Windows Server 2016

A common issue customers face with their non-persistent (pooled) Remote Desktop Services environments is handling users' Outlook data. When Outlook is running in cached exchange mode, the .OST storing a user's Outlook data must follow the user as they roam from host to host. Windows Search Service indexes the .OST and creates an index catalog to enable search functionality in Outlook. In non-persistent RDS environments, the index catalog doesn't roam with user data and must be rebuilt every time the user signs into a new PC, which could potentially be every sign-on. Until the Windows Search Service finishes indexing the .OST, users get limited or incomplete search functionality.

According to a published report from RDS Gurus, FSLogix (a third party solution provider) has a solution that aims to solve this issue: FSLogix's Office 365 Container roams a user's Outlook data and their search index catalog, giving users access to their emails and enabling users to search in Outlook, even when they roam between sessions on different hosts within a collection.

RDS Gurus performed testing on FSLogix's Office 365 Container, comparing it with RDS's native User Profile Disk roaming solution. The test scenarios covered both on-premises and Azure RDS environments for non-persistent sessions on an RD session host (RDSH). Tests also included pooled VMs on RD virtualization host (RDVH), only for on-premises (RDVH isn't available in Azure). RDS Gurus primarily focused on the user experience when there are "noisy neighbors," or other users logged on to the same session host running similar workloads on the system.

The performance counters collected in these tests revealed similar resource usage (CPU, RAM, network activity) with both UPD and FSLogix. The similarity in resource usage is because Windows Search Service throttles its CPU usage when indexing. When it comes to user experience, RDS Gurus found that FSLogix's Office 365 Container exceeds UPD in Outlook search functionality. In the UPD case, search doesn't return results or returns incomplete results as Windows Search Service indexes the .OST. Because FSLogix roams the index catalog, users see search results immediately. RDS Gurus observed a significant improvement in user experience when searching in Outlook in non-persistent RDS environments using FSLogix.

Read more about the results and conclusions on the RDS Gurus blog.

Use the sync app on virtual desktops

10/27/2025

For all supported operating systems, the OneDrive sync app supports:

- Virtual desktops that persist between sessions.
- Non-persistent virtual desktops that use Azure Virtual Desktop.
- Non-persistent virtual desktops that have FSLogix Apps or FSLogix Office Container, and a Microsoft 365 subscription for all of the following operating systems:
  - Windows 10 and 11, 32-bit or 64-bit (supports VMDK/Virtual Machine Disk files)
  - Windows Server 2022 (supports VHDX/Virtual Hard Disk)
  - Windows Server 2019 (supports VHDX/Virtual Hard Disk)
  - Windows Server 2016 (supports VHDX/Virtual Hard Disk)
  - Windows Server 2012 R2 (supports VHDX/Virtual Hard Disk)

> **Note**
>
> Check the latest versions at these links: OneDrive, FSLogix App Using the OneDrive sync app with non-persistent environments requires that you install the sync app per machine. For Windows Server, the SMB network file sharing protocol is also required. The OneDrive sync app is supported in a remote app scenario hosted as a Citrix Virtual App. The OneDrive sync app with FSLogix doesn't support running multiple instances of the same container simultaneously.

Set up OneDrive in Citrix Virtual Apps

This section describes how to enable and use OneDrive in Citrix Virtual Apps.

Prerequisites

To enable OneDrive in Citrix Virtual Apps, you must have the following versions of Windows and Citrix Virtual Apps and Desktops (CVAD):

Windows:

- Windows 11: KB5014019
- Windows Server 2022: KB5014021
- Windows 10: KB5014023
- Windows Server 2019: KB5014022

Citrix:

- CVAD 7 2203 LTSR/Long Term Service Release CU1 or later.
- VDA/Virtual Delivery Agent 2212 enables Shellbridge by default. All earlier versions require Shellbridge to be enabled manually.
- To enable this feature, On 2203 LTSR TS VDA (2019 Server, 2022 Server, Windows 10 RDSH/Remote Desktop Session Host, or Windows 11 RDSH/Remote Desktop Service Host) add the following registry details:
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Citrix Virtual Desktop Agent`
  - `Name: Shellbridge`
  - `Type: REG_DWORD`
  - `Value: 1`

To ensure that the feature is correctly enabled, open a command window (cmd.exe) and run `start ms-settings:printers`. If the feature is enabled, the printer setting window is displayed.

We recommend adding OneDrive.exe to `LogoffCheckSysModules`.

- `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI`
- `Value Name:LogoffCheckSysModules`
- `Type:REG_SZ`
- `String:OneDrive.exe, Microsoft.Sharepoint.exe`

> ⓘ **Important**
>
> FSLogix must be used in conjunction with Citrix Virtual Apps for OneDrive to be supported.

How to set up OneDrive

1. Install OneDrive Sync app per machine. See Install the sync app per-machine.
2. Install the latest version of FSLogix. See Install FSLogix Applications.

   > ⓘ **Note**
   >
   > All non-persistent VDI environments require the latest version of FSLogix. Ensure you install the latest version. See OneDrive sync error FSLogix_unsupported_environment on VMs.

3. Add OneDrive to `HKLM\Software\Microsoft\Windows\CurrentVersion\` by using the following command:

   ```
   REG ADD HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v OneDrive /t REG_SZ /d "\"C:\Program Files\Microsoft OneDrive\OneDrive.exe\" /background"
   ```

4. Silently configure user accounts. See Silently configure user accounts.

   > **Note**
   >
   > Silent sign-in should work if your machine is connected to Microsoft Entra ID. Make sure to turn off this setting if your computer isn't Microsoft Entra joined.

Set up OneDrive in Omnissa Horizon Virtual Apps

This section describes how to enable and use OneDrive in Omnissa Horizon Virtual Apps.

Prerequisites

- Omnissa Horizon
- Windows
  - Windows 10 and Windows 11 Guest Operating Systems for Horizon Agent and Remote Experience, for Omnissa Horizon 8.x (2006 and later) (78714). For more information, see this article.
  - Non-Windows 10 and 11 Guest Operating Systems for Horizon 8 Agent (78715). For more information, see this article.
- FSLogix
- Omnissa Dynamic Environment Manager (DEM) or a product which enables user environment personalization. The system on which you plan to install DEM must meet certain software requirements. For more information, see this article.

Registry Keys

The following registry keys help to roam the user environment on multiple nodes in the virtual application farm. Omnissa Dynamic Environment Manager or a similar user environment management tool can be used to deploy the registry keys to all farm servers.

- [IncludeRegistryTrees]
  - `HKCU\Software\Microsoft\Office`
  - `HKCU\Software\Microsoft\Internet Explorer`
  - `HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings`
  - `HKCU\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached`
  - `HKCU\Software\Microsoft\OneDrive`
- [IncludeFolderTrees]
  - `<Appdata>\Microsoft\Windows\Recent`
  - `<Appdata>\Microsoft\crypto`
  - `<Appdata>\SystemCertificates`
  - `<LocalAppdata>\Microsoft\IdentityCache`
  - `<LocalAppdata>\Microsoft\Internet Explorer`
  - `<LocalAppdata>\Microsoft\Windows\INetCache`

Configure Omnissa Dynamic Environment Manager with Horizon Apps

1. Launch the Omnissa Dynamic Environment Manager management console, select **Create Config File**, and select **Use an Application Template**.
2. Select the application template (Microsoft Office 2016 and 2019, or Microsoft 365), select **OneDrive for Business**, and click **Next**.
3. Provide the file name and description and select **Finish**.
4. Add the previously listed **required registry keys** to **Import / Export** settings.

Configure FSLogix with Omnissa Dynamic Environment Manager

Configuring FSLogix in combination with Dynamic Environment Manager will help with store OneDrive cache and the save location for Microsoft and non-Microsoft applications.

For more info on how to configure FSLogix Office Container (ODFC) on all Horizon Virtual App farm servers, refer to this article.

- Install the per-machine version of the OneDrive sync app on all the Horizon Virtual App farm hosts.
- Create the following entries in each Horizon farm servers. You can use DEM or similar user environment management tool to deploy the registry to all virtual app farm servers.
  - Key: `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`
  - Type: `REG_SZ`
  - Name: `OneDrive`
  - Data: `"C:\Program Files\Microsoft OneDrive\OneDrive.exe"/background`
  - Key: `HKLM\Software\Policies\Microsoft\OneDrive`
  - Type: `REG_DWORD`
  - Name: `SilentAccountconfig`
  - Data: `1`

> **Note**
>
> Sometimes the silent login will take several seconds. If the first attempt fails, a second attempt might be required.

See also

Learn more about VHDX/Virtual Hard Disk and VHD/Virtual Hard Disk.

For info about creating virtual hard disks, see Manage virtual hard disks.

Desktop Hosting Reference Architecture

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article defines a set of architectural blocks for using Remote Desktop Services (RDS) and Microsoft Azure virtual machines to create multitenant, hosted Windows desktop and application services, which we call "desktop hosting." You can use this architecture reference to create highly secure, scalable, and reliable desktop hosting solutions for small- and medium-sized organizations with 5 to 5000 users.

The primary audience for this reference architecture is hosting providers who want to leverage Microsoft Azure Infrastructure Services to deliver desktop hosting services and Subscriber Access Licenses (SALs) to multiple tenants via the Microsoft Service Provider Licensing Agreement (SPLA) program. A second audience for this reference architecture are end customers who want to create and manage desktop hosting solutions in Microsoft Azure Infrastructure Services for their own employees using RDS User CALs extended rights through Software Assurance (SA).

To deliver a desktop hosting solutions, hosting partners and SA customers leverage Windows Server to deliver Windows users an application experience that is familiar to business users and consumers. Built on the foundations of Windows 10, Windows Server 2016 provides familiar application support and user experience.

The scope of this document is limited to:

- Architectural design guidance for a desktop hosting service. Detailed information, such as deployment procedures, performance, and capacity planning is explained in separate documents. For more general information about Azure Infrastructure Services, see Virtual Machines in Azure.
- Session-based desktops, RemoteApp applications, and server-based personal desktops that use Windows Server 2016 Remote Desktop Session Host (RD Session Host). Windows client-based virtual desktop infrastructures aren't covered because there's no Service Provider License Agreement (SPLA) for Windows client operating systems. Windows Server-based virtual desktop infrastructures are allowed under the SPLA, and Windows client-based virtual desktop infrastructures are allowed on dedicated hardware with end-customer licenses in certain scenarios. However, client-based virtual desktop infrastructures are out-of-scope for this document.
- Microsoft products and features, primarily Windows Server 2016 and Microsoft Azure Infrastructure Services.

- Desktop hosting services for tenants ranging in size from 5 to 5000 users. For larger tenants, you may need to modify this architecture to provide adequate performance. The Server Manager RDS graphical user interface (GUI) isn't recommended for deployments over 500 users. PowerShell is recommended for managing RDS deployments between 500 and 5000 users.

- The minimum set of components and services required for a desktop hosting service. There are many optional components and services that can be added to enhance a desktop hosting service, but these are out-of-scope for this document.

After reading this document, the reader should understand:

- The building blocks for providing a secure, reliable, multitenant desktop hosting solution based in Microsoft Azure Services.
- The purpose of each building block and how they fit together.

There are multiple ways to build a desktop hosting solution based on this architecture. This architecture outlines integration and improvements in Azure with Windows Server 2016. Other deployment options are available with the Desktop Hosting Reference Architecture Guide.

The following topics are covered:

- Desktop hosting logical architecture
- Understand the RDS Roles
- Understand the desktop hosting environment
- Azure services and considerations for desktop hosting

Feedback

Remote Desktop Services architecture

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) provides a flexible platform for hosting Windows applications and desktops in the cloud or on-premises. This article describes common RDS deployment architectures and shows how to integrate RDS with Azure services to meet your organization's needs.

Use these architecture diagrams to understand:

- How RDS roles work together in different deployment scenarios.
- Options for basic and highly available RDS configurations.
- Integration patterns with Azure Platform as a Service (PaaS) offerings.

Whether you're planning a new RDS deployment or modernizing an existing one, these architectures provide proven patterns to help you design a solution that meets your requirements for performance, availability, and security.

The architecture diagrams in this article show using RDS in Azure. However, as Remote Desktop Services is a role in Windows Server, you can deploy it on-premises and on other clouds. These diagrams are primarily intended to illustrate how the RDS roles are colocated and use other services.

Standard RDS deployment architectures

Remote Desktop Services has two standard architectures:

- **Basic deployment**: contains the minimum number of servers to create a fully effective RDS environment, but with no redundancy.
- **Highly available deployment**: contains all necessary components to have the highest guaranteed uptime for your RDS environment.

The following sections show the components of each architecture and how they work together. The diagrams also show how the RDS roles are colocated on the servers, which is a common practice to reduce costs and complexity.

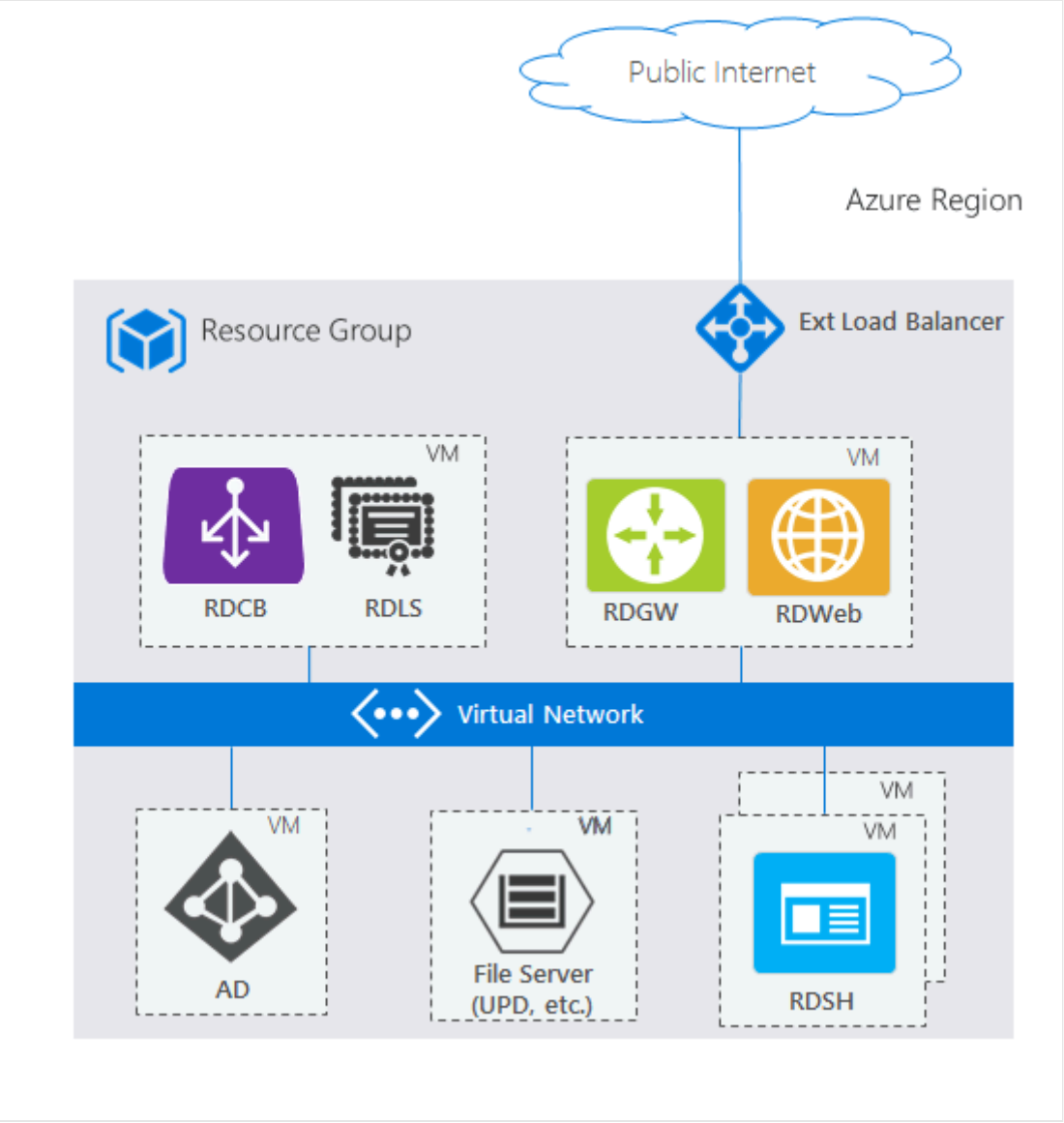
Basic deployment

This architecture illustrates a foundational Remote Desktop Services deployment in Azure that provides remote access to desktops and applications through a single Azure region. The

deployment uses an external load balancer to distribute incoming connections from the public internet across the RDS infrastructure.

The core RDS roles are distributed across multiple virtual machines within a single resource group. The RD Connection Broker (RDCB) and RD Licensing Server (RDLS) share one virtual machine, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate VM. Supporting infrastructure includes an Active Directory domain controller and a file server for user profile disks and shared storage. The RD Session Host (RDSH) server, deployed on its own virtual machine, provides the actual desktop sessions and hosted applications to end users.

All virtual machines communicate through an Azure Virtual Network, which provides secure network connectivity between the RDS components while isolating the deployment from other Azure resources. This architecture provides a cost-effective starting point for organizations looking to migrate their desktop hosting to Azure, with the flexibility to scale individual components as usage grows.

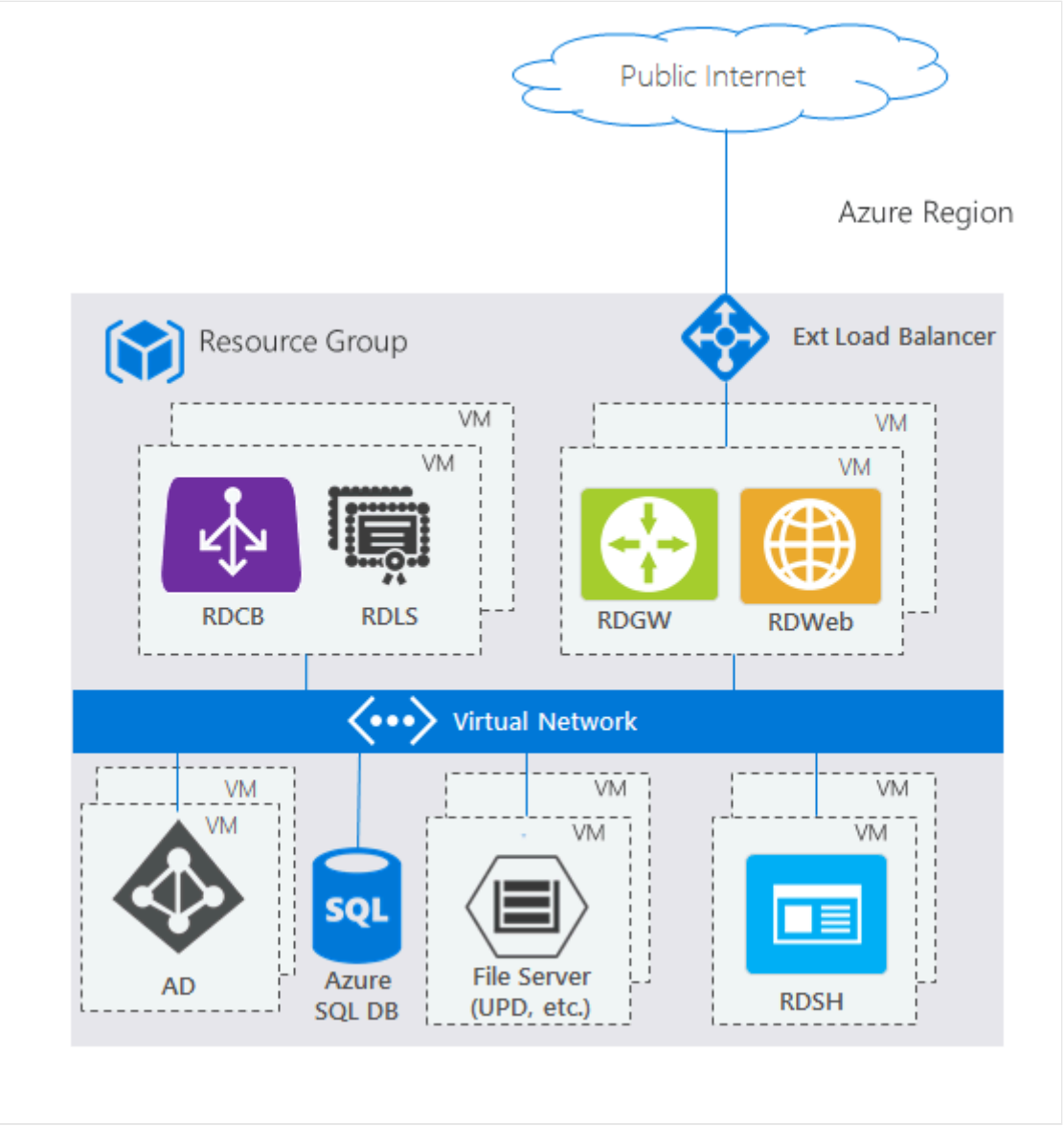


Highly available deployment

This architecture demonstrates a Remote Desktop Services deployment that integrates Azure Platform as a Service (PaaS) offerings to improve scalability and reduce management overhead. The key difference from a basic RDS deployment is the replacement of a traditional SQL Server virtual machine with Azure SQL Database for storing RDS configuration and user session data.

The RDS roles maintain the same distribution pattern, but with multiple instances of each; the RD Connection Broker (RDCB) and RD Licensing Server (RDLS) sharing one set of virtual machines, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate set of VMs. The RD Session Hosts (RDSH) continue to provide desktop sessions and applications from their dedicated virtual machines. Supporting infrastructure includes an Active Directory domain controller and a file server for user profiles and shared storage.

By using Azure SQL Database instead of a self-managed SQL Server instance, this architecture provides built-in high availability, automatic backups, and simplified database management. The Azure SQL Database handles the RDS Connection Broker database requirements while eliminating the need to maintain, patch, and monitor a separate database server. This hybrid approach combines the flexibility of Infrastructure as a Service (IaaS) for the RDS roles with the managed benefits of PaaS for the database tier, resulting in reduced operational complexity and improved reliability.

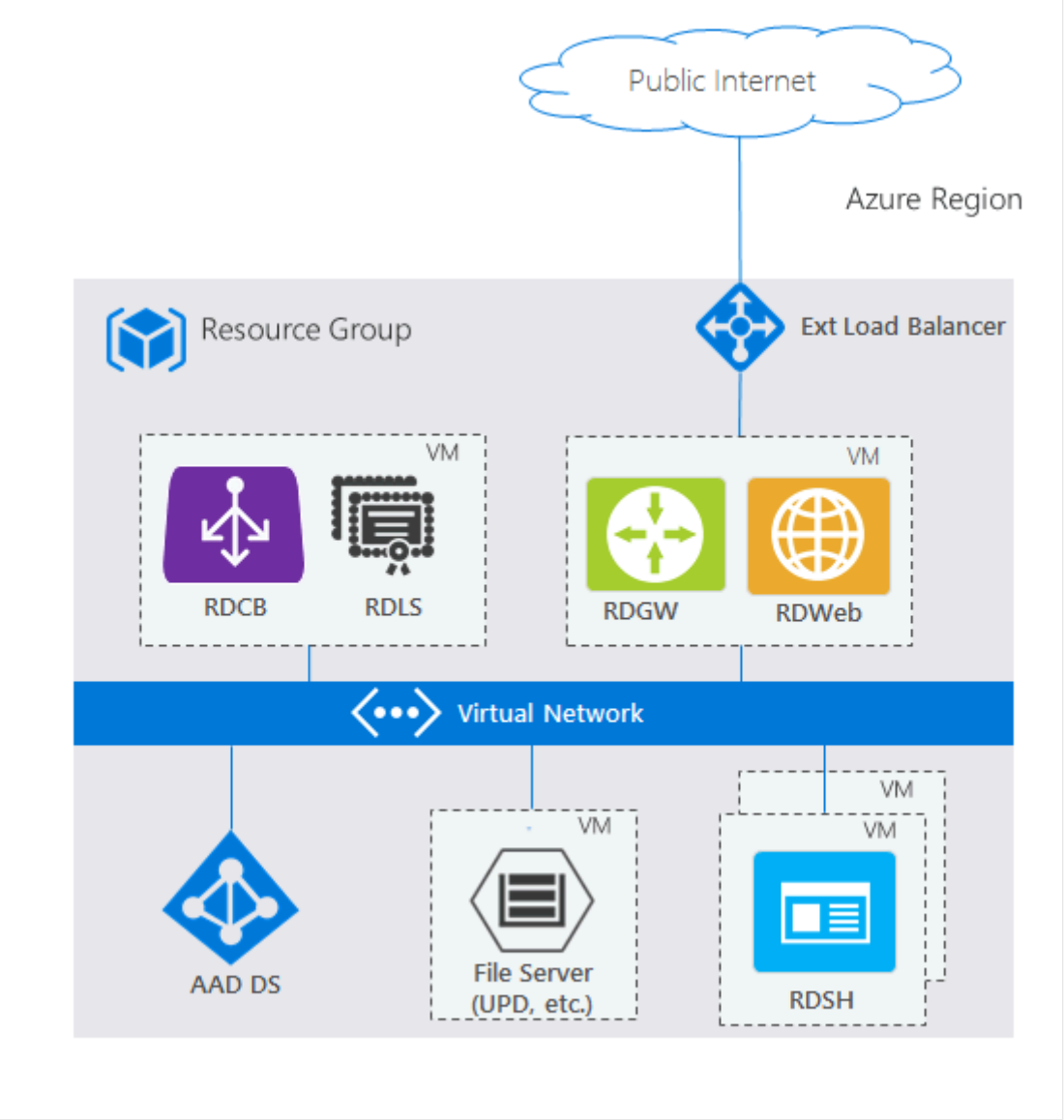


RDS architectures with unique Azure PaaS roles

Though the standard RDS deployment architectures fit most scenarios, Azure continues to invest in first-party PaaS solutions that drive customer value. The following architectures show how they incorporate with RDS.

RDS deployment with Microsoft Entra Domain Services

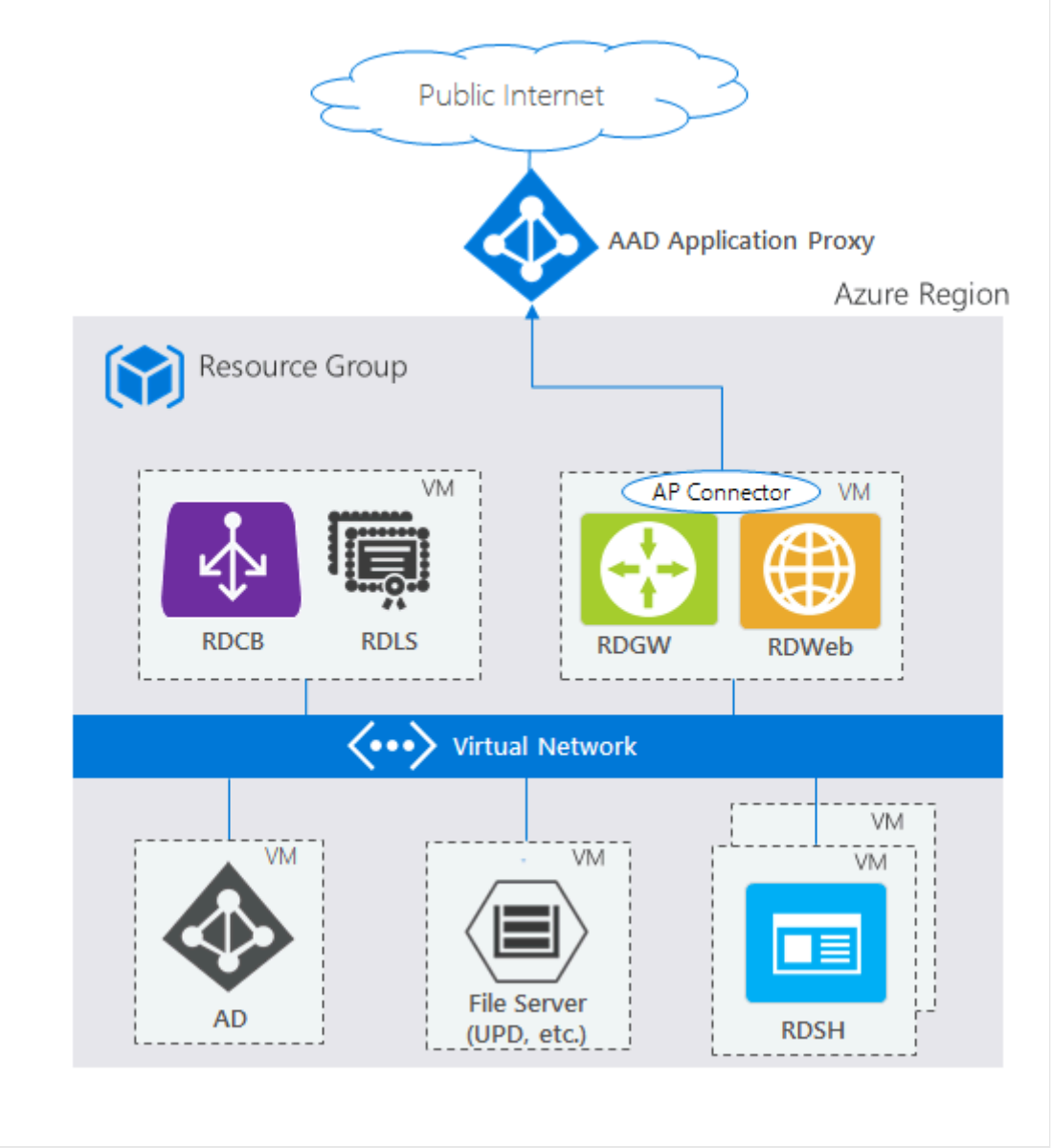
The two standard architecture diagrams are based on a traditional Active Directory (AD) deployed on a Windows Server VM. However, if you don't have a traditional AD and only have a Microsoft Entra tenant, for example through services like Microsoft 365, but still want to use RDS, you can use Microsoft Entra Domain Services to create a fully managed domain in your Azure IaaS environment that uses the same users that exist in your Microsoft Entra tenant. This option removes the complexity of manually syncing users and managing more virtual machines. Microsoft Entra Domain Services can work in either deployment: basic or highly available.



RDS deployment with Microsoft Entra application proxy

The two standard architecture diagrams use the RD Web/Gateway servers as the Internet-facing entry point into the RDS system. For some environments, administrators would prefer to remove their own servers from the perimeter and instead use technologies that also provide extra security through reverse proxy technologies. The Microsoft Entra application proxy PaaS role fits nicely with this scenario.

For supported configurations and how to create this setup, see Publish Remote Desktop with Microsoft Entra application proxy.



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

---

Last updated on 07/07/2025

Desktop hosting service

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article will tell you more about the desktop hosting service's components.

Tenant environment

As described in Remote Desktop service roles, each role plays a distinct part in the tenant environment.

The provider's desktop hosting service is implemented as a set of isolated tenant environments. Each tenant's environment consists of a storage container, a set of virtual machines, and a combination of Azure services, all communicating over an isolated virtual network. Each virtual machine contains one or more of the components that make up the tenant's hosted desktop environment. The following subsections describe the components that make up each tenant's hosted desktop environment.

Active Directory Domain Services

Active Directory Domain Services (AD DS) provides the domain and forest information, such that the tenant's users can sign in to the desktops and applications to carry out their workloads. This also enables you to set up or connect to required file shares and databases that may be required for Windows applications.

The tenant's forest does not require any trust relationship with the provider's management forest. A domain administrator account may be set up in the tenant's domain to allow the provider's technical personnel to perform administrative tasks in the tenant's environment (such as monitoring system status and applying software updates) and to assist with troubleshooting and configuration.

There are multiple ways to deploy AD DS:

1. Enable Microsoft Entra Domain Services in the tenant's virtual networking environment. This will create a managed AD DS instance for the tenant based on the users and groups that exist in Microsoft Entra ID.
2. Set up a stand-alone AD DS server in the tenant's virtual networking environment. This gives you all of the full control of the AD DS instance running on virtual machines.

3. Create a site-to-site VPN connection to an AD DS server located on the tenant's premises. This allows the tenant to connect to their existing AD DS instance and reduce duplication of users, groups, organizational units, and so on.

For more information, see the following articles:

- Microsoft Entra Domain Services documentation
- Desktop Hosting Reference Architecture Guide
- Create a site-to-site connection in the Azure portal

SQL database

A highly-available SQL database is used by the Remote Desktop Connection Broker to store deployment information, such as the mapping of current users' connections to the host servers.

There are multiple ways to deploy an SQL database:

1. Create an Azure SQL Database in the tenant's environment. This provides you with the functionality of a redundant SQL database without you having to manage the servers themselves. This also allows you to pay for what you consume instead of investing in infrastructure.
2. Create an SQL Server AlwaysOn cluster. This allows you to leverage existing SQL Server infrastructure and gives you complete control over the SQL Server instances.

For more information about how to set up a highly-available SQL database infrastructure, see the following articles:

- What is the Azure SQL Database service?
- Creation and configuration of availability groups (SQL Server).
- Add the RD Connection Broker server to the deployment and configure high availability.

File server

The file server uses the Server Message Block (SMB) 3.0 protocol to provide shared folders. These shared folders are used to create and store user profile disk files (.vhdx) to back up data and let users share data with each other within the tenant's cloud service.

The virtual machine that deploys the file server must have an Azure data disk attached and configured with shared folders. Azure data disks use write-through caching,

guaranteeing that writes to the disk will not be erased whenever the virtual machine is restarted.

Small tenants can reduce costs by combining the file server and RD Licensing role on a single virtual machine in the tenant's environment.

For more information, see the following articles:

- Storage in Windows Server
- How to attach a managed data disk to a Windows VM in the Azure portal

User profile disks

User profile disks allow users to save personal settings and files when they are signed in to a session on an RD Session Host server in one collection, then access the same settings and files when signing in to a different RD Session Host server in the collection. When the user first signs in, the tenant's file server creates a user profile disk that gets mounted to the RD Session Host server that the user is currently connected to. For each subsequent sign-in, the user profile disk is mounted to the appropriate RD Session host server, and it is unmounted with each sign-out. Only the user can access the profile disk's contents.

Feedback

Remote Desktop Services roles

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article describes the roles within a Remote Desktop Services environment.

Remote Desktop Session Host

The Remote Desktop Session Host (RD Session Host) holds the session-based apps and desktops you share with users. Users get to these desktops and apps through one of the Remote Desktop clients that run on Windows, MacOS, iOS, and Android. Users can also connect through a supported browser by using the web client.

You can organize desktops and apps into one or more RD Session Host servers, called "collections." You can customize these collections for specific groups of users within each tenant. For example, you can create a collection where a specific user group can access specific apps, but anyone outside of the group you designated won't be able to access those apps.

For small deployments, you can install applications directly onto the RD Session Host servers. For larger deployments, we recommend building a base image and provisioning virtual machines from that image.

You can expand collections by adding RD Session Host server virtual machines to a collection farm with each RDSH virtual machine within a collection assigned to same availability set. This provides higher collection availability and increases scale to support more users or resource-heavy applications.

In most cases, multiple users share the same RD Session Host server, which most efficiently utilizes Azure resources for a desktop hosting solution. In this configuration, users must sign in to collections with non-administrative accounts. You can also give some users full administrative access to their remote desktop by creating personal session desktop collections.

You can customize desktops even more by creating and uploading a virtual hard disk with the Windows Server OS that you can use as a template for creating new RD Session Host virtual machines.

For more information, see the following articles:

- Remote Desktop Services - Secure data storage

- Upload a generalized VHD and use it to create new VMs in Azure
- Update RDSH collection (ARM template)

Remote Desktop Connection Broker

Remote Desktop Connection Broker (RD Connection Broker) manages incoming remote desktop connections to RD Session Host server farms. RD Connection Broker handles connections to both collections of full desktops and collections of remote apps. RD Connection Broker can balance the load across the collection's servers when making new connections. If RD Connection Broker is enabled, using DNS round robin to RD Session Hosts for balancing servers is not supported. If a session disconnects, RD Connection Broker will reconnect the user to the correct RD Session Host server and their interrupted session, which still exists in the RD Session Host farm.

You'll need to install matching digital certificates on both the RD Connection Broker server and the client to support single sign-on and application publishing. When developing or testing a network, you can use a self-generated and self-signed certificate. However, released services require a digital certificate from a trusted certification authority. The name you give the certificate must be the same as the internal Fully Qualified Domain Name (FQDN) of the RD Connection Broker virtual machine.

You can install the Windows Server 2016 RD Connection Broker on the same virtual machine as AD DS to reduce cost. If you need to scale out to more users, you can also add additional RD Connection Broker virtual machines in the same availability set to create an RD Connection Broker cluster.

Before you can create an RD Connection Broker cluster, you must either deploy an Azure SQL Database in the tenant's environment or create an SQL Server AlwaysOn Availability Group.

For more information, see the following articles:

- Add the RD Connection Broker server to the deployment and configure high availability
- SQL database in Desktop hosting service.

Remote Desktop Gateway

Remote Desktop Gateway (RD Gateway) grants users on public networks access to Windows desktops and applications hosted in Microsoft Azure's cloud services.

The RD Gateway component uses Secure Sockets Layer (SSL) to encrypt the communications channel between clients and the server. The RD Gateway virtual machine must be accessible through a public IP address that allows inbound TCP connections to port 443 and inbound UDP connections to port 3391. This lets users connect through the internet using the HTTPS communications transport protocol and the UDP protocol, respectively.

The digital certificates installed on the server and client have to match for this to work. When you're developing or testing a network, you can use a self-generated and self-signed certificate. However, a released service requires a certificate from a trusted certification authority. The name of the certificate must match the FQDN used to access RD Gateway, whether the FQDN is the public IP address' externally facing DNS name or the CNAME DNS record pointing to the public IP address.

For tenants with fewer users, the RD Web Access and RD Gateway roles can be combined on a single virtual machine to reduce cost. You can also add more RD Gateway virtual machines to an RD Gateway farm to increase service availability and scale out to more users. Virtual machines in larger RD Gateway farms should be configured in a load-balanced set. IP affinity isn't required.

For more information, see the following articles:

- Add high availability to the RD Web and Gateway web front
- Remote Desktop Services - Access from anywhere
- Remote Desktop Services - Multifactor authentication
- Set up the RD Gateway role

Remote Desktop Web Access

Remote Desktop Web Access (RD Web Access) lets users access desktops and applications through a web portal and launches them through the device's native Microsoft Remote Desktop client application. You can use the web portal to publish Windows desktops and applications to Windows and non-Windows client devices, and you can also selectively publish desktops or apps to specific users or groups.

RD Web Access needs Internet Information Services (IIS) to work properly. A Hypertext Transfer Protocol Secure (HTTPS) connection provides an encrypted communications channel between the clients and the RD Web server. The RD Web Access virtual machine must be accessible through a public IP address that allows inbound TCP connections to port 443 to allow the tenant's users to connect from the internet using the HTTPS communications transport protocol.

Matching digital certificates must be installed on the server and clients. For development and testing purposes, this can be a self-generated and self-signed certificate. For a released service, the digital certificate must be obtained from a trusted certification authority. The name of the certificate must match the Fully Qualified Domain Name (FQDN) used to access RD Web Access. Possible FQDNs include the externally facing DNS name for the public IP address and the CNAME DNS record pointing to the public IP address.

For tenants with fewer users, you can reduce costs by combining the RD Web Access and Remote Desktop Gateway workloads into a single virtual machine. You can also add additional RD Web virtual machines to an RD Web Access farm to increase service availability and scale out to more users. In an RD Web Access farm with multiple virtual machines, you'll have to configure the virtual machines in a load-balanced set.

For more information about how to configure RD Web Access, see the following articles:

- Set up the Remote Desktop web client for your users
- Create and deploy a Remote Desktop Services collection
- Create a Remote Desktop Services collection for desktops and apps to run

Remote Desktop Licensing

Activated Remote Desktop Licensing (RD Licensing) servers let users connect to the RD Session Host servers hosting the tenant's desktops and apps. Tenant environments usually come with the RD Licensing server already installed, but for hosted environments you'll have to configure the server in per-user mode.

The service provider needs enough RDS Subscriber Access Licenses (SALs) to cover all authorized unique (not concurrent) users that sign in to the service each month. Service providers can purchase Microsoft Azure Infrastructure Services directly, and can purchase SALs through the Microsoft Service Provider Licensing Agreement (SPLA) program. Customers looking for a hosted desktop solution must purchase the complete hosted solution (Azure and RDS) from the service provider.

Small tenants can reduce costs by combining the file server and RD Licensing components onto a single virtual machine. To provide higher service availability, tenants can deploy two RD License server virtual machines in the same availability set. All RD servers in the tenant's environment are associated with both RD License servers to keep users able to connect to new sessions even if one of the servers goes down.

For more information, see the following articles:

- License your RDS deployment with client access licenses (CALs)

- Activate the Remote Desktop Services license server
- Track your Remote Desktop Services client access licenses (RDS CALs)
- Microsoft Volume Licensing: licensing options for service providers

Feedback

Was this page helpful? Yes No

Azure services and considerations for desktop hosting

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The following sections describe Azure Infrastructure Services.

Azure portal

After the provider creates an Azure subscription, the Azure portal can be used to manually create each tenant's environment. This process can also be automated using PowerShell scripts.

For more information, visit the Microsoft Azure website.

Azure Load Balancer

The tenant's components run on virtual machines that communicate with each other on an isolated network. During the deployment process, you can externally access these virtual machines through the Azure Load Balancer using Remote Desktop Protocol endpoints or a Remote PowerShell endpoint. Once a deployment is complete, these endpoints will typically be deleted to reduce the attack surface area. The only endpoints will be the HTTPS and UDP endpoints created for the virtual machine running the RD Web and RD Gateway components. This allows clients on the internet to connect to sessions running in the tenant's desktop hosting service. If a user opens an application that connects to the internet, such as a web browser, the connections will be passed through the Azure Load Balancer.

For more information, see What is Azure Load Balancer?

Security considerations

This Azure Desktop Hosting Reference Architecture Guide is designed to provide a highly secure and isolated environment for each tenant. System security also depends on safeguards taken by the provider during deployment and operation of the hosted service. The following list describes some considerations the provider should take to keep their desktop hosting solution based on this reference architecture secure.

- All administrative passwords must be strong, ideally randomly generated, changed frequently, and saved in a secure central location only accessible to a select few provider administrators.
- When replicating the tenant environment for new tenants, avoid using the same or weak administrative passwords.
- The RD Web Access site URL, name, and certificates must be unique and recognizable to each tenant to prevent spoofing attacks.
- During the normal operation of the desktop hosting service, all public IP addresses should be deleted for all virtual machines except the RD Web and RD Gateway virtual machine that lets users securely connect to the tenant's desktop hosting cloud service. Public IP addresses may be temporarily added when necessary for management tasks, but they should always be deleted afterwards.

For more information, see the following articles:

- Security and protection
- Security best practices for IIS 8

Design considerations

It's important to consider the constraints of Microsoft Azure Infrastructure Services when designing a multitenant desktop hosting service. The following list describes considerations the provider must take to achieve a functional and cost-effective desktop hosting solution based on this reference architecture.

- An Azure subscription has a maximum number of virtual networks, VM cores, and Cloud Services that can be used. If a provider needs more resources than this, they may need to create multiple subscriptions.
- An Azure Cloud Service has a maximum number of virtual machines that can be used. The provider may need to create multiple Cloud Services for larger tenants that exceed the maximum.
- Azure deployment costs are based partially on the number and size of virtual machines. The provider should optimize the number and size of the virtual machines for each tenant to provide a functional and highly secure Desktop Hosting environment at the lowest cost.
- The physical computer resources in the Azure data center are virtualized by using Hyper-V. Hyper-V hosts are not configured in host clusters, so the availability of the virtual machines is dependent on the availability of the individual servers used in the Azure infrastructure. To provide higher availability, multiple instances of each role service virtual machine can be created in an availability set, then guest clustering can be implemented within the virtual machines.

- In a typical storage configuration, a service provider will have a single storage account with multiple containers (for example, one for each tenant), and multiple disks within each container. However, there is a limit to the total storage and performance that can be achieved for a single storage account. For service providers that support large numbers of tenants or tenants with high storage capacity or performance requirements, the service provider may need to create multiple storage accounts.

For more information, see the following articles:

- Sizes for Cloud Services
- Microsoft Azure virtual machine pricing details
- Hyper-V overview
- Azure Storage scalability and performance targets

Microsoft Entra application proxy

Microsoft Entra application proxy is a service provided in paid SKUs of Microsoft Entra ID that allow users to connect to internal applications through Azure's own reverse-proxy service. This allows the RD Web and RD Gateway endpoints to be hidden inside of the virtual network, eliminating the need to be exposed to the internet by a public IP address. Hosters can use Microsoft Entra application proxy to condense the number of virtual machines in the tenant's environment while still maintaining a full deployment. Microsoft Entra application proxy also enables many of the benefits that Microsoft Entra ID provides, such as conditional access and multi-factor authentication.

For more information, see Get started with Application Proxy and install the connector.

Feedback

Understanding the desktop hosting environment

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The following information describes the components of the desktop hosting service.

Tenant environment

The provider's desktop hosting service is implemented as a set of isolated tenant environments. Each tenant's environment consists of a storage container, a set of virtual machines, and a combination of Azure services, all communicating over an isolated virtual network. Each virtual machine contains one or more of the components that make up the tenant's hosted desktop environment. The following subsections describe the components that make up each tenant's hosted desktop environment.

Remote Desktop Services

In a desktop hosting environment, the following Remote Desktop Services roles are installed amongst various virtual machines:

- Remote Desktop Connection Broker
- Remote Desktop Gateway
- Remote Desktop Licensing
- Remote Desktop Session Host
- Remote Desktop Web Access

For a full description of each of these roles and how they interact with each other, please review the Understanding RDS roles document.

(Azure) Active Directory Domain Services

There are multiple ways to connect to and manage Active Directory Domain Services (AD DS) for a desktop hosting environment in Azure:

1. Create a virtual machine in the tenant's environment running the AD DS role
2. Create a site-to-site VPN connection with the tenant's on-premises environment to use an existing AD DS

3. Use the Microsoft Entra Domain Services PaaS role, which creates a domain on the tenant's virtual network based on the tenant's Microsoft Entra ID

With Remote Desktop Services, the tenant must have an Active Directory to manage access into the environment, user profile storage, and monitoring within the deployment. When using the standard (non-Azure) AD DS, the tenant's forest does not require any trust relationship with the provider's management forest. A domain administrator account may be set up in the tenant's domain to allow the provider's technical personnel to perform administrative tasks in the tenant's environment (such as monitoring system status and applying software updates) and to assist with troubleshooting and configuration.

Additional information: Microsoft Entra Domain Services Documentation Install a new Active Directory forest on an Azure virtual network Create a resource manager VNet with a Site-to-Site VPN connection using the Azure Portal

Azure SQL Database

Azure SQL Database allows for hosters to extend their Remote Desktop Services deployment without needing to deploy and maintain a full SQL Server Always-On cluster. The Azure SQL Database is used by the Remote Desktop Connection Broker to store deployment information, such as the mapping of current users' connections to end-host servers. Like other Azure services, Azure SQL DB follows a consumption model, ideal for any size deployment.

Additional information: What is SQL Database?

Microsoft Entra application proxy

Microsoft Entra application proxy is a service provided in paid-SKUs of Microsoft Entra ID that allow users to connect to internal applications through Azure's own reverse-proxy service. This allows the RD Web and RD Gateway endpoints to be hidden inside of the virtual network, eliminating the need to be exposed to the internet via a public IP address. This further allows hosters to condense the number of virtual machines in the tenant's environment while still maintaining a full deployment.

Additional information: Enabling Microsoft Entra application proxy

File server

The file server provides shared folders by using the Server Message Block (SMB) 3.0 protocol. The shared folders are used to create and store user profile disk files (.vhdx), to backup data, and to allow users a place to share data with other users in the tenant's virtual network.

The VM used to deploy the file server must have an Azure data disk attached and configured with shared folders. Azure data disks use write-through caching which guarantees that writes to the disk persist across restarts of the VM.

For small tenants, the cost can be reduced by combining the file server with the virtual machine running the RD Connection Broker and RD Licensing roles on a single virtual machine in the tenant's environment.

Additional information File and Storage Services Overview How to Attach a Data Disk to a Virtual Machine

User Profile Disks

User profile disks allow users to save personal settings and files when they are signed in to a session on an RD Session Host server in a collection, and then have access to the same settings and files when signing in to a different RD Session Host server in the collection. When the user first signs in, a user profile disk is created on the tenant's file server, and that disk is mounted to the RD Session Host server to which the user is connected. For each subsequent sign-in, the user profile disk is mounted to the appropriate RD Session host server, and with each sign-out, it is un-mounted. The contents of the profile disk can only be accessed by that user.

Feedback

Build and deploy your Remote Desktop Services deployment

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

A Remote Desktop Services deployment is the infrastructure used to share apps and resources with your users. Depending on the experience you want to provide, you can make it as small or complex as you need. Remote Desktop deployments are easily scaled. You can increase and decrease Remote Desktop Web Access, Gateway, Connection Broker and Session Host servers at will. You can use Remote Desktop Connection Broker to distribute workloads. Active Directory based authentication provides a highly secure environment.

Remote Desktop clients enable access from any Windows, Apple, or Android computer, tablet, or phone.

See Remote Desktop Services architecture for a detailed discussion of the different pieces that work together to make up your Remote Desktop Services deployment.

Have an existing Remote Desktop deployment built on a previous version of Windows Server? Check out your options for moving to the latest version of Windows Server, where you can take advantage of new and better functionality around performance and scale:

- Migrate your RDS deployment
- Upgrade your RDS deployment

Want to create a new Remote Desktop deployment? Use the following information to deploy Remote Desktop in Windows Server:

- Deploy the Remote Desktop Services infrastructure
- Create a session collection to hold the apps and resources you want to share
- License your RDS deployment
- Have your users install a Remote Desktop client so they can access the apps and resources.
- Enable high availability by adding additional Connection Brokers and Session Hosts:
  - Scale out an existing RDS collection with an RD Session Host farm
  - Add high availability to the RD Connection Broker infrastructure
  - Add high availability to the RD Web and RD Gateway web front

- Deploy a two-node Storage Spaces Direct file system for UPD storage

Feedback

Was this page helpful? Yes No

Remote Desktop Services architecture

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) provides a flexible platform for hosting Windows applications and desktops in the cloud or on-premises. This article describes common RDS deployment architectures and shows how to integrate RDS with Azure services to meet your organization's needs.

Use these architecture diagrams to understand:

- How RDS roles work together in different deployment scenarios.
- Options for basic and highly available RDS configurations.
- Integration patterns with Azure Platform as a Service (PaaS) offerings.

Whether you're planning a new RDS deployment or modernizing an existing one, these architectures provide proven patterns to help you design a solution that meets your requirements for performance, availability, and security.

The architecture diagrams in this article show using RDS in Azure. However, as Remote Desktop Services is a role in Windows Server, you can deploy it on-premises and on other clouds. These diagrams are primarily intended to illustrate how the RDS roles are colocated and use other services.

Standard RDS deployment architectures

Remote Desktop Services has two standard architectures:

- **Basic deployment**: contains the minimum number of servers to create a fully effective RDS environment, but with no redundancy.
- **Highly available deployment**: contains all necessary components to have the highest guaranteed uptime for your RDS environment.

The following sections show the components of each architecture and how they work together. The diagrams also show how the RDS roles are colocated on the servers, which is a common practice to reduce costs and complexity.

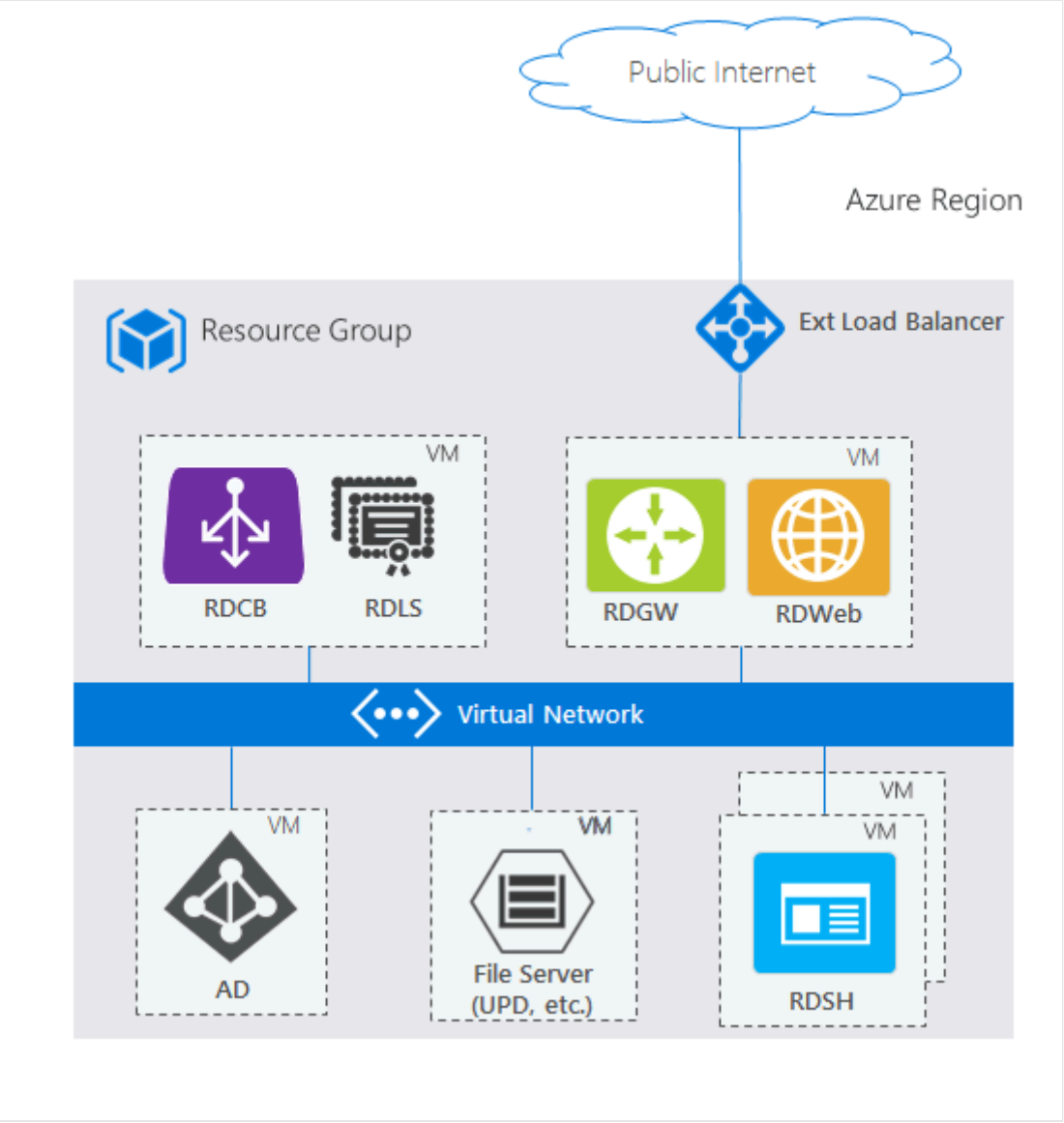
Basic deployment

This architecture illustrates a foundational Remote Desktop Services deployment in Azure that provides remote access to desktops and applications through a single Azure region. The

deployment uses an external load balancer to distribute incoming connections from the public internet across the RDS infrastructure.

The core RDS roles are distributed across multiple virtual machines within a single resource group. The RD Connection Broker (RDCB) and RD Licensing Server (RDLS) share one virtual machine, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate VM. Supporting infrastructure includes an Active Directory domain controller and a file server for user profile disks and shared storage. The RD Session Host (RDSH) server, deployed on its own virtual machine, provides the actual desktop sessions and hosted applications to end users.

All virtual machines communicate through an Azure Virtual Network, which provides secure network connectivity between the RDS components while isolating the deployment from other Azure resources. This architecture provides a cost-effective starting point for organizations looking to migrate their desktop hosting to Azure, with the flexibility to scale individual components as usage grows.

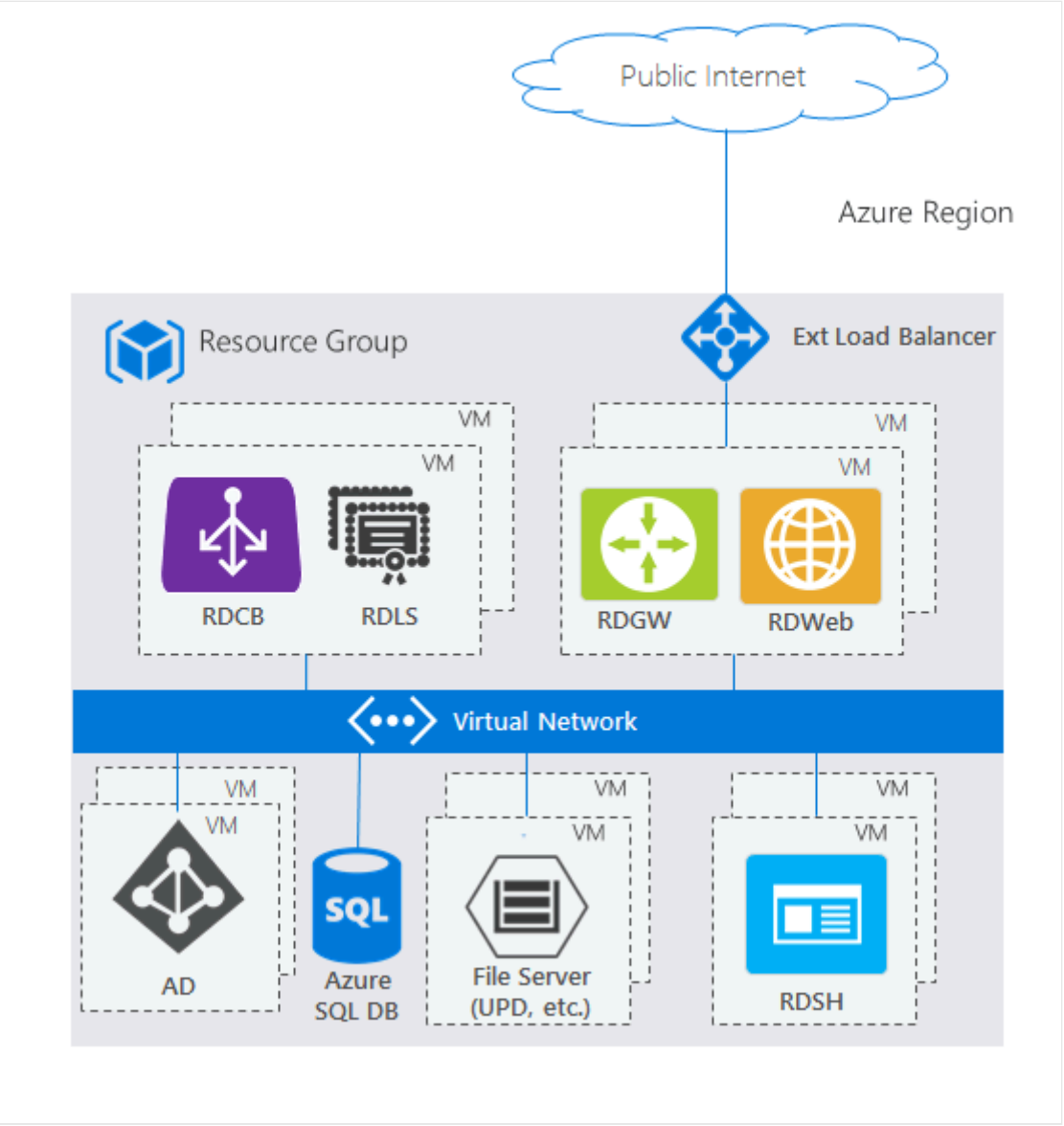


Highly available deployment

This architecture demonstrates a Remote Desktop Services deployment that integrates Azure Platform as a Service (PaaS) offerings to improve scalability and reduce management overhead. The key difference from a basic RDS deployment is the replacement of a traditional SQL Server virtual machine with Azure SQL Database for storing RDS configuration and user session data.

The RDS roles maintain the same distribution pattern, but with multiple instances of each; the RD Connection Broker (RDCB) and RD Licensing Server (RDLS) sharing one set of virtual machines, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate set of VMs. The RD Session Hosts (RDSH) continue to provide desktop sessions and applications from their dedicated virtual machines. Supporting infrastructure includes an Active Directory domain controller and a file server for user profiles and shared storage.

By using Azure SQL Database instead of a self-managed SQL Server instance, this architecture provides built-in high availability, automatic backups, and simplified database management. The Azure SQL Database handles the RDS Connection Broker database requirements while eliminating the need to maintain, patch, and monitor a separate database server. This hybrid approach combines the flexibility of Infrastructure as a Service (IaaS) for the RDS roles with the managed benefits of PaaS for the database tier, resulting in reduced operational complexity and improved reliability.

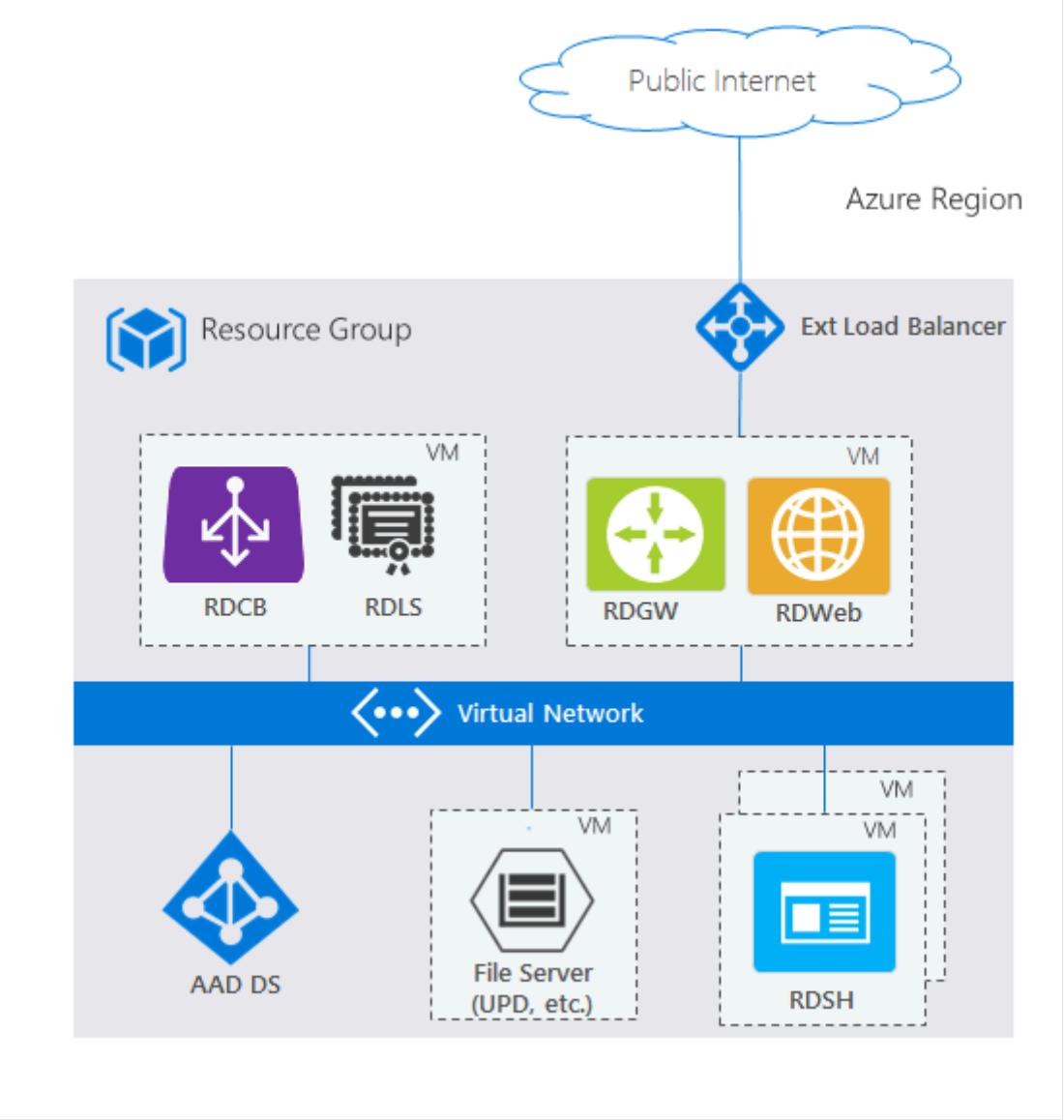


RDS architectures with unique Azure PaaS roles

Though the standard RDS deployment architectures fit most scenarios, Azure continues to invest in first-party PaaS solutions that drive customer value. The following architectures show how they incorporate with RDS.

RDS deployment with Microsoft Entra Domain Services

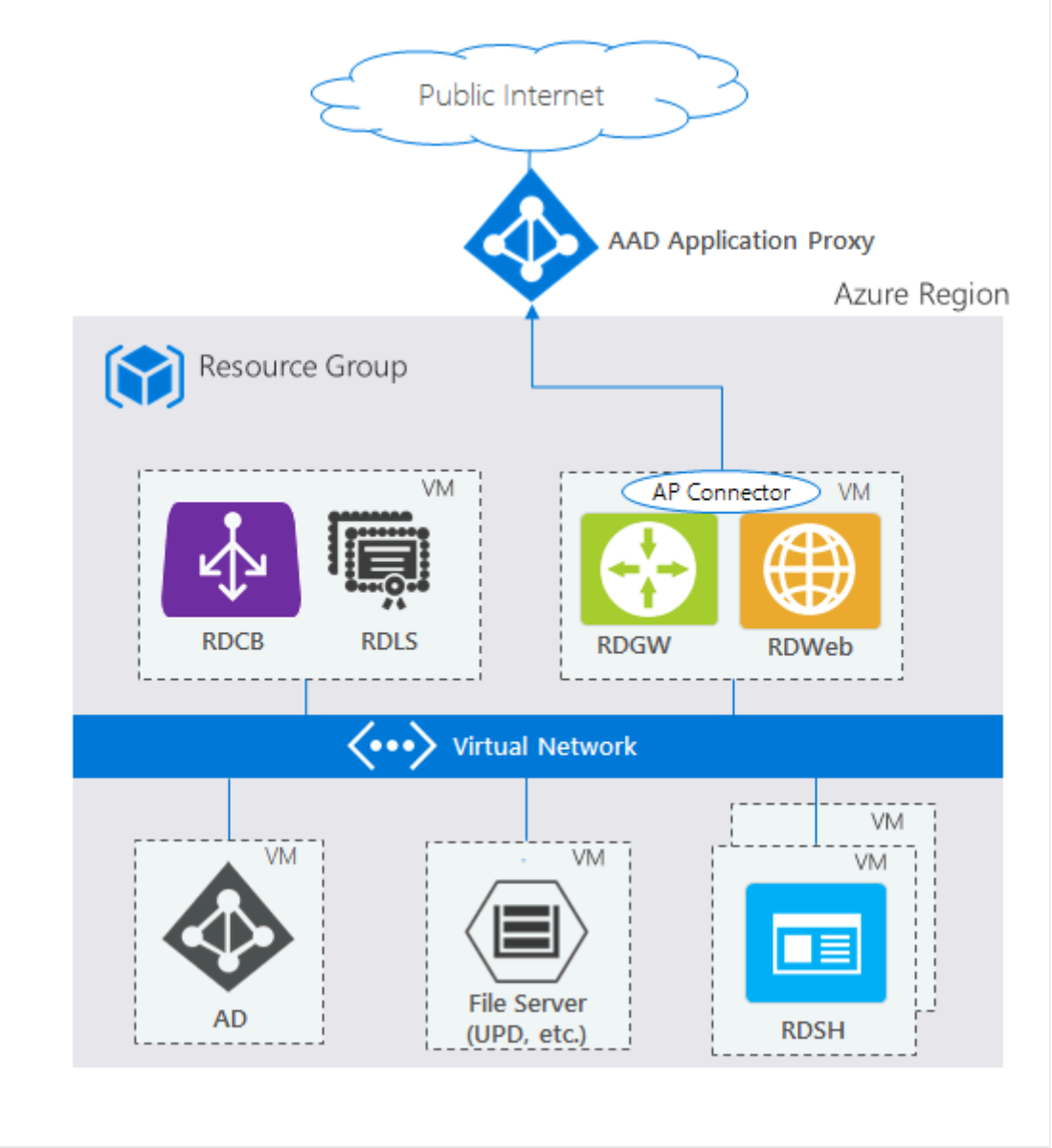
The two standard architecture diagrams are based on a traditional Active Directory (AD) deployed on a Windows Server VM. However, if you don't have a traditional AD and only have a Microsoft Entra tenant, for example through services like Microsoft 365, but still want to use RDS, you can use Microsoft Entra Domain Services to create a fully managed domain in your Azure IaaS environment that uses the same users that exist in your Microsoft Entra tenant. This option removes the complexity of manually syncing users and managing more virtual machines. Microsoft Entra Domain Services can work in either deployment: basic or highly available.



RDS deployment with Microsoft Entra application proxy

The two standard architecture diagrams use the RD Web/Gateway servers as the Internet-facing entry point into the RDS system. For some environments, administrators would prefer to remove their own servers from the perimeter and instead use technologies that also provide extra security through reverse proxy technologies. The Microsoft Entra application proxy PaaS role fits nicely with this scenario.

For supported configurations and how to create this setup, see Publish Remote Desktop with Microsoft Entra application proxy.



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

---

Last updated on 07/07/2025

Migrate your Remote Desktop Services deployment to a newer Windows Server version

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Migration of a Remote Desktop Services deployment is supported from source servers running a Windows Server version to destination servers running the same Windows Server version. For example, from Windows Server 2025 to Windows Server 2025. Meaning, there is no direct in-place migration from RDS in an earlier version of Windows Server to a later Windows Server version. Instead, you need to upgrade most RDS components first to a later Windows Server version, then migrate data and licenses. The only components that support a direct migration are RD Web, RD Gateway, and the licensing server.

For more information on the upgrade process and requirements, see upgrading your Remote Desktop Services deployments.

Use the following steps to migrate your Remote Desktop Services deployment:

- Migrate RD Connection Broker servers
- Migrate session collections
- Migrate virtual desktop collections
- Migrate RD Web Access servers
- Migrate RD Gateway servers
- Migrate RD Licensing servers
- Migrate certificates

Migrate RD Connection Broker servers

This is the first and most important step for migrating: migrating your RD Connection Brokers to destination servers running the latest version of Windows Server.

> ⓘ **Important**

The Remote Desktop Connection Broker (RD Connection Broker) source servers must be configured for high availability to support migration. For more information, see Deploy a Remote Desktop Connection Broker cluster.

1. If you have more than one RD Connection Broker server in the high availability setup, remove all the RD Connection Broker servers except the one that is currently active.
2. Upgrade the remaining RD Connection Broker server in the deployment to a later Windows Server version.
3. Add the new Windows Server version RD Connection Broker servers into the high availability deployment.

> **Note**
>
> - A mixed high availability configuration with different versions of Windows Server is not supported for RD Connection Broker servers.
> - An RD Connection Broker running a newer version of Windows Server can serve session collections with RD Session Host servers running a previous version of Windows Server, and it can serve virtual desktop collections with RD Virtualization Host servers running a previous version of Windows Server.

Migrate session collections

Follow these steps to migrate a session collection in an earlier version of Windows Server to a session collection in a later version of Windows Server.

> **Important**
>
> Migrate session collections only after successfully completing the previous step, Migrate RD Connection Broker servers.

1. Upgrade the session collection to a later version of Windows Server.
2. Add the new RD Session Host server running the new Windows Server version to the session collection.
3. Sign out of all sessions in the RD Session Host servers, and remove the servers that require migration from the session collection.

> **Note**
>
> If the UVHD template (UVHD-template.vhdx) is enabled in the session collection and the file server has been migrated to a new server, update the User Profile Disks: Location collection property with the new path. The User Profile Disks must be available at the same relative path in the new location as they were on the source server.
>
> A session collection of RD Session Host servers with a mix of Windows Server versions isn't supported.

Migrate virtual desktop collections

Follow these steps to migrate a virtual desktop collection from a source server running the earlier version of Windows Server to a destination server running a later version of Windows Server.

> **Important**
>
> Migrate virtual desktop collections only after successfully completing the previous step, Migrate RD Connection Broker servers.

1. Upgrade the virtual desktop collection from the server running the earlier version of Windows Server to a later version of Windows Server.
2. Add the new Windows Server version RD Virtualization Host servers to the virtual desktop collection.
3. Migrate all virtual machines in the current virtual desktop collection that are running on RD Virtualization Host servers to the new servers.
4. Remove all RD Virtualization Host servers that required migration from the virtual desktop collection in the source server.

> **Note**
>
> If the UVHD template (UVHD-template.vhdx) is enabled in the session collection and the file server has been migrated to a new server, update the User Profile Disks: Location collection property with the new path. The User Profile Disks must be available at the same relative path in the new location as they were on the source server.

A virtual desktop collection of RD Virtualization Host servers with a mix of servers running earlier versions of Windows Server and later versions of Windows Server is not supported.

Migrate RD Web Access servers

Follow these steps to migrate RD Web Access servers:

1. Join the destination servers running the new version of Windows Server to the Remote Desktop Services deployment and install the RD Web role
2. Use IIS Web Deploy tool to migrate the RD Web website settings from the current RD Web Access servers to the destination servers running the new version of Windows Server.
3. Migrate certificates to the destination servers running the new version of Windows Server.
4. Remove the source servers from the Remote Desktop Services deployment.

Migrate RD Gateway servers

Follow these steps to migrate RD Gateway servers:

1. Join the destination servers running the new version of Windows Server to the Remote Desktop Services deployment and install the RD Gateway role
2. Use the IIS Web Deploy tool to migrate the RD Gateway endpoint settings from the current RD Gateway servers to the destination servers running the new version of Windows Server.
3. Migrate certificates to the destination servers running the new version of Windows Server.
4. Remove the source servers from the Remote Desktop Services deployment.

Migrate RD Licensing servers

Follow these steps to migrate an RD Licensing server from a source server running an earlier version of Windows Server to a destination server running a later version Windows Server.

1. Migrate the Remote Desktop Services client access licenses (RDS CALs) from the source server to the destination server.
2. Edit the **Deployment Properties** in **Server Manager** on the Remote Desktop management server (which is typically being run on the first RD Connection Broker server) to include only the new RD Licensing servers running the new version of Windows Server.
3. Deactivate the source RD Licensing server: In **Remote Desktop Licensing Manager**, right-click the appropriate server, hover over **Advanced** to select **Deactivate Server**, and then follow the steps in the wizard.
4. Remove the source RD Licensing servers from the deployment in **Server Manager** on the Remote Desktop management server.

Migrate certificates

Successful certificate migration requires both the actual process of migrating certificates and updating certificate information in the Remote Desktop Services Deployment Properties.

Typical certificate migration includes the following steps:

- Export the certificate to a PFX file with the private key.
- Import the certificate from a PFX file.

After migrating the appropriate certificates, update the following required certificates for the Remote Desktop Services deployment in server manager or PowerShell:

- RD Connection Broker - single sign-on
- RD Connection Broker - RDP file publishing
- RD Gateway - HTTPS connection
- RD Web Access - HTTPS connection and RemoteApp/desktop connection subscription

---

Feedback

Migrate your Remote Desktop Services Client Access Licenses (RDS CALs)

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You have three options to migrate your RDS CALs:

- **Automatic connection method**: This recommended method communicates via internet directly to the Microsoft Clearinghouse outbound over TCP port 443.
- **Using a web browser**: This method allows migration when the server running the Remote Desktop Licensing Manager tool does not have internet connectivity, but the administrator has internet connectivity on a separate device. The URL for the Web migration method is displayed in the Manage RDS CALs Wizard.
- **Using a telephone**: This method allows the administrator to complete the migration process over the phone with a Microsoft representative. The appropriate telephone number is determined by the country/region that you chose in the Activate Server Wizard and is displayed in the Manage RDS CALs Wizard.

In this article, the establish RDS CAL migration method highlights the general steps common across any RDS CAL migration method, while migrate RDS CALs highlights the steps specific to each migration method.

Regardless of migration method, you must, at a minimum, be a member of the local Administrators group to perform the migration steps.

Before migration ensure that the destination license server is activated. Follow these steps to activate the Remote Desktop Services license server.

Establish RDS CAL migration method

1. On the destination license server, open **Remote Desktop Licensing Manager**. (Alternatively, the licensing manager can be launched with the following steps: Select **Start** > **Administrative Tools**. Enter the **Remote Desktop Services** directory, and launch **Remote Desktop Licensing Manager**.)
2. Verify the connection method for the Remote Desktop license server: right-click the license server to which you want to migrate the RDS CALs, and then select

Properties. On the **Connection Method** tab, verify the **Connection method** - you can change it in the dropdown menu. Select **OK**.

3. Right-click the license server to which you want to migrate the RDS CALs, and then select **Manage Licenses**.

4. Follow the steps in the wizard to the **Action Selection** page. Select **Migrate RDS CALs from another license server to this license server**.

5. Choose the reason for migrating the RDS CALs, and then select **Next**. You have the following choices:

   - The source license server is being replaced by this license server.
   - The source license server is no longer functioning.

6. The next page in the wizard depends on the migration reason that you chose.

   - If you chose **The source license server is being replaced by this license server** as the reason for migrating the RDS CALs, the **Source License Server Information** page is displayed.

     On the Source License Server Information page, enter the name or IP address of the source license server.

     If the source license server is available on the network, select **Next**. The wizard contacts the source license server and has an option to **Obtain Client License Key Pack**.

     If the source license server isn't available on the network, select **The specified source license server isn't available on the network**. Specify the operating system that the source license server is running, and then provide the license server ID for the source license server. After you select **Next**, you're reminded that you must remove the RDS CALs manually from the source license server after the wizard has completed. After you confirm that you understand this requirement, the **Obtain Client License Key Pack** page appears.

   - If you chose **The source license server is no longer functioning** as the reason for migrating the RDS CALs, you're reminded that you must remove the RDS CALs manually from the source license server after the wizard has completed. After you confirm that you understand this requirement, the **Obtain Client License Key Pack** page appears.

The next step is to migrate the CALs - use the information in the following to complete the wizard. What you see in the wizard depends on the connection method you identified in Step 2 of this section.

Migrate RDS CALs

There are three ways to migrate licenses to the destination license server; follow the steps corresponding to the **Connection method** verified in Step 2 in the previous section:

- Automatic connection method
- Using a web browser
- Using a telephone

Automatic connection method

1. On the **License Program** page, select the appropriate program through which you purchased your RDS CALs, then select **Next**.
2. Enter the required information (typically a license code or an agreement number, depending on the **License program**), and then select **Next**. Consult the documentation provided when you purchased your RDS CALs.
3. Select the appropriate product version, license type, and quantity of RDS CALs for your environment based on your RDS CAL purchase agreement, and then select **Next**.
4. The Microsoft Clearinghouse is automatically contacted and processes your request. The RDS CALs are then migrated onto the license server.
5. Select **Finish** to complete the RDS CAL migration process.

Using a web browser

1. On the **Obtain Client License Key Pack** page, select the hyperlink to connect to the Remote Desktop Services Licensing Web site. If you're running Remote Desktop Licensing Manager on a computer that doesn't have internet connectivity, note the address for the Remote Desktop Services Licensing Web site, and then connect to the Web site from a computer that has internet connectivity.
2. On the Remote Desktop Services Licensing Web page, under **Select Option**, select **Manage CALs**, and then select **Next**.
3. Provide the following required information, then select **Next**:

    - **Target License Server ID**: A 35-digit number, in groups of 5 numerals, which is displayed on the **Obtain Client License Key Pack** page in the Manage

RDS CALs Wizard.

- **Reason for recovery**: Choose the reason for migrating the RDS CALs.
- **License Program**: Choose the program through which you purchased your RDS CALs.

4. Provide the following required information, then select **Next**:

   - Last name or surname
   - First name or given name
   - Company name
   - Country/region

   You can also provide the optional information requested, such as company address, e-mail address, and phone number. In the organizational unit field, you can describe the unit within your organization that this license server serves.

5. The License Program that you selected on the previous page determines what information you need to provide on the next page. In most cases, you must provide either a license code or an agreement number. Consult the documentation provided when you purchased your RDS CALs. In addition, you need to specify which type of RDS CAL and the quantity that you want to migrate to the license server.

6. After you enter the required information, select **Next**.

7. Verify that all of the information that you entered is correct, then select **Next** to submit your request to the Microsoft Clearinghouse. The web page then displays a license key pack ID generated by the Microsoft Clearinghouse.

   > **ⓘ Important**
   >
   > Keep a copy of the license key pack ID. Having this information with you facilitates communications with the Microsoft Clearinghouse, should you need assistance with recovering RDS CALs.

8. On the same **Obtain Client License Key Pack** page, enter the license key pack ID, and then select **Next** to migrate the RDS CALs to your license server.

9. Select **Finish** to complete the RDS CAL migration process.

Using a telephone

1. On the **Obtain Client License Key Pack** page, use the displayed telephone number to call the Microsoft Clearinghouse. Give the representative your Remote Desktop license server ID and the required information for the licensing program through which you purchased your RDS CALs. The representative then processes your request to migrate the RDS CALs, and gives you a unique ID for the RDS CALs. This unique ID is referred to as the **license key pack ID**.

   > **Important**
   >
   > Keep a copy of the license key pack ID. Having this information with you facilitates communications with the Microsoft Clearinghouse should you need assistance with recovering RDS CALs.

2. On the same **Obtain Client License Key Pack** page, enter the license key pack ID, and then select **Next** to migrate the RDS CALs to your license server.

3. Select **Finish** to complete the RDS CAL migration process.

Feedback

Use certificates in Remote Desktop Services

07/14/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use certificates to secure connections to your Remote Desktop Services (RDS) deployment and between RDS server roles. RDS uses Secure Socket Layer (SSL) or Transport Layer Security (TLS) to encrypt connections to the RDS Web, Connection Broker, and Gateway role services.

Certificates prevent man-in-the-middle attacks, where a bad actor intercepts traffic between the Remote Desktop Protocol (RDP) server and client to steal confidential information or deny access to credentials, by verifying that the server sending information to the client is authentic. When this trust relationship is set up, the client considers the connection secure and can accept data going to and from the server.

Prerequisites

The following are required to use certificates in RDS:

- A computer or computers where the RDS role is configured. To learn more, see Install or uninstall roles, role services, or features.
- An account with administrator rights or equivalent to one or more RDS servers.
- A server certificate that meets the following requirements:
  - Issued for Server Authentication (EKU 1.3.6.1.5.5.7.3.1).
  - Issued for Enhanced Key Usage (OID 2.5.29.37).
  - Issued for Key Usage (OID 2.5.29.15).
  - Issued by a certificate authority trusted by one or more RDS servers and clients.
  - Issued with an exportable private key.
  - An export of the certificate with the corresponding private key in `.pfx` format. To learn more about exporting the private key, see Export a certificate with its private key.

> ⓘ Note

If you're using Active Directory Certificate Services (AD CS) to issue certificates, you can also create a certificate template or duplicate the Web Server certificate template. To learn more about creating certificate templates, see Create a new certificate template.

Configure Remote Desktop to use certificates

Now that you created your certificates and understand their contents, you must configure Remote Desktop to use those certificates.

To configure Remote Desktop to use specific certificates:

GUI

1. In **Server Manager**, on the left pane, select **Remote Desktop Services**.
2. On the **Overview** tab, under **Deployment Overview**, select **TASKS**, then select **Edit Deployment Properties**.
3. In the **Configure the deployment** window, select **Certificates**.
4. Choose **Select existing certificate**, select **Browse**, locate your certificate file in `.pfx` format, then select **Open**.
5. In the **Password** field, enter the password for the certificate you created, then select **OK**.
6. Select the **Allow the certificate to be added to the Trusted Root Certification Authorities certificate store on the destination computers** checkbox, then select **OK**.
7. Select **OK** to finalize your deployment.

> **Note**
>
> Even if you have multiple servers in the deployment, Server Manager imports the certificate to all servers. Server Manager places the certificate in the trusted root for each server, then binds the certificate to its respective roles.

You might want to use certificates for the RDS Session Host along with the certificates you configured in Server Manager. For more information about RDS Session Host certificates, see Remote Desktop listener certificate configurations.

Related content

- Remote Desktop Services - Secure data storage with UPDs
- Remote Desktop Services - Multifactor Authentication

Upgrade Remote Desktop Services deployments

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

In this article, learn about which Remote Desktop Services (RDS) versions can be upgraded and the order to upgrade your Remote Desktop (RD) role services.

Supported OS upgrades with RDS role installed

You can upgrade to a newer version of Windows Server by two versions at a time. For example, you can upgrade to Windows Server 2025 from Windows Server 2019.

Flow for deployment upgrades

In order to keep the downtime to a minimum, use the following guide:

1. **RD Connection Broker servers** should be upgraded first. If you have an active/active configuration, remove all but one server from the deployment and perform an in-place upgrade. Perform upgrades on the remaining RD Connection Broker servers offline and then reapply them to the deployment. The deployment isn't available during the RD Connection Broker servers' upgrade.

   > ⓘ **Note**
   >
   > It's mandatory to upgrade all RD Connection Broker servers. Windows Server RD Connection Broker servers in a mixed deployment are not supported. Once the RD Connection Broker server(s) are running a new Windows Server version, the deployment remains functional, even if the rest of the servers in the deployment are still running the previous version.

2. **RD License servers** should be upgraded before you upgrade your RD Session Host servers.

   > ⓘ **Note**

RD license servers from an older version of Windows Server work with newer versions, but they can only process client access licenses (CALs) from the older Windows Server version. They can't use the newer Windows Server CALs. For more information about RD license servers, see RDS CAL version compatibility.

3. **RD Session Host servers** can be upgraded next. Avoid downtime during upgrade by splitting the servers to be upgraded into steps as detailed. All will be functional after the upgrade. To upgrade, use the steps described in Upgrading your Remote Desktop Session Host to the latest Windows Server version.

4. **RD Virtualization Host servers** can be upgraded next. To upgrade, use the steps described in Upgrading your Remote Desktop Virtualization Host to the latest Windows Server version.

5. **RD Web Access servers** can be upgraded anytime.

   > ⓘ **Note**
   >
   > - Upgrading RD Web might reset Internet Information Services (IIS) properties, such as any configuration files. To not lose your changes, make notes or copies of customizations done to the RD Web site in IIS.
   > - RD Web Access servers from an older version of Windows Server work with newer versions.

6. **RD Gateway servers** can be upgraded anytime.

   > ⓘ **Note**
   >
   > - Windows Server 2016 and later doesn't include Network Access Protection (NAP) policies—they have to be removed. The easiest way to remove the correct policies is by running the upgrade wizard. If there are any NAP policies you must delete, the upgrade blocks and creates a text file on the desktop that includes the specific policies. To manage NAP policies, open the Network Policy Server tool. After deleting them, select **Refresh** in the Setup tool to continue with the upgrade process.
   > - RD Gateway servers from an older version of Windows Server work with newer versions.

VDI deployment – supported guest OS upgrade

Administrators have the following options to upgrade VM collections:

Upgrade managed shared VM collections

Administrators need to create VM templates with the desired OS version and use it to patch all the VMs in the pool.

Windows 10 can be patched to Windows 11.

Upgrade unmanaged shared VM collections

End users can't upgrade their personal desktops. Administrators should perform the upgrade. The exact steps are to be determined.

Known issues

Issue: If the RD deployment has the RD Web Access (RDWA) Role already installed and has been upgraded from a previous windows installation, a new upgrade might fail. For example, if the deployment containing RDWA upgraded from Server 2012 R2 to Server 2019, another upgrade to Server 2022 might encounter a failure.

Workaround: Before migrating for the second time, check if the following registry key is present: `HKLM\SOFTWARE\Microsoft\Terminal Server Web Access\IsInstalled`

If it isn't present, open an elevated PowerShell prompt, then run the following commands:

PowerShell

```
$registryPath = "HKLM:SOFTWARE\Microsoft\Terminal Server Web
Access\IsInstalled"
New-Item -Path $registryPath
New-ItemProperty -Path $registryPath -Name Version -PropertyType String -
Value "6.0"
```

Feedback

Upgrading your Remote Desktop Session Host

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

> **ⓘ Important**
>
> All applications must be uninstalled before the upgrade and reinstalled after the upgrade to avoid any app compatibility issues that may rise because of the upgrade.

Upgrading a RDS session-based collection

In order to keep the down-time to a minimum, it's best to follow these steps while upgrading a RDS session-based collection:

1. Identify the servers to be upgraded, say, half the servers in the collection.
2. Prevent new connections to these servers by setting **Allow New Connections** to false.
3. Log off all sessions on these servers.
4. Remove these servers from the collection.
5. Upgrade the servers to the latest Windows Server version.
6. Set **Allow New Connections** to "false" on the remaining servers in the collection.
7. Add the upgraded servers back to their corresponding collections.
8. Remove the remaining set of servers to be upgraded from the collection.
9. Set **Allow New Connections** to "true" on the upgraded servers in the collection.
10. Upgrade the remaining servers in the deployment by following steps 3 through 9.

Upgrading a standalone RD Session Host server

A standalone RD Session Host server can be upgraded anytime.

Feedback

Was this page helpful? Yes No

Upgrading your Remote Desktop Virtualization Host

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

RD Virtualization Host servers in the deployment where VMs are stored locally

These servers should be upgraded all at once. Complete the following steps to upgrade:

1. Log off all users.
2. Turn off or save all virtual machines on each host.
3. Upgrade the servers to the new Windows Server version.
4. All collections should be available and functional after the upgrades are complete.

RD Virtualization Host servers in the deployment where VMs are stored in Cluster Shared Volumes (CSV)

1. Determine an upgrade strategy where some of the RDVH servers are upgraded and some continue to host VMs on the earlier version of Windows Server.
2. Isolate one or more of the RDVH servers targeted for the initial round of upgrading. By migrating all VMs to other 'not to be upgraded yet' RDVH servers that remain part of the original cluster.
    a. Open Failover Cluster Manager.
    b. Select **Roles**.
    c. Select one or more VMs. Right-click to open the context menu.
    d. Select **Move** and choose either **Live** or **Quick Migration** to move the VMs to one or more of the RD Virtualization Host Servers that aren't part of the initial upgrade. Use **Live** or **Quick** Migration depending on factors such as hardware compatibility or online requirements.

3. Evict the RDVH servers, prepared for upgrading, from the original cluster.
4. Upgrade the isolated RDVH servers.
5. After the targeted RDVH servers have been successfully upgraded, create a new cluster and CSV, which needs to be on an entirely different SAN volume.
6. Join all upgraded RDVH servers to the new cluster.
7. Create a folder structure in the new CSV that mimics the existing folder structure in the existing CSV. This includes the collection folders and each VM's top level subfolders.
8. From the various VM Collection folders on the original CSV, copy over the /IMGS folder and contents to the new collection folders in the same locations on the new CSV.
9. On the source RDVH machine, use Cluster Manager to remove the VM's configuration for high availability:
    a. Launch Cluster Manager.
    b. Select **Roles**.
    c. Right-click the VM objects, and then select **Remove**.
10. On one of the nonupgraded RDVH servers, use Hyper-V Manager to move all VMs to one of the upgraded RDVH servers and new Cluster CSV:
    a. Open Hyper-V Manager.
    b. Select one of the nonupgraded RDVH servers.
    c. Right-click one of the VMs to be moved, and then select **Move**.
    d. Choose **Move the virtual machine**, and then select **Next**.
    e. Provide the targeted upgraded RDVH server's name on the **Specify Destination Computer** page, and then select **Next**.
    f. Choose **Move the virtual machine's data to a single location**, and then select **Next**.
    g. Browse to the destination location.

> **Important**

Ensure this path is to an empty folder for the specific VM.

> **Note**
>
> As mentioned, you need to have already created a new destination subfolder prior to this step. The Select Folder dialog won't allow you to create a subfolder in this step.

Select **Next**, and then select **Finished**.

11. Once the VMs are relocated, add them as cluster **High Availability** objects:

    a. Open Failover Cluster Manager on an upgraded RD Virtualization Host Server.

    b. Right-click the **Roles** node, and then select **Configure Role**. Select **Next** on the **Start** page of the High Availability wizard.

    c. Choose **Virtual Machine** from the list of available roles, and then select **Next**. A list of VMs that aren't configured is shown.

    d. Select all the VMs. Select **Next** and then select **Next** again on the confirmation page to start the configuration task.

12. Once you have relocated all VMs, upgrade the remaining RDVH servers. Use the above steps for balancing VM locations as appropriate.

> **Note**
>
> Heterogeneous Hyper-V servers in a cluster aren't supported.

Feedback

Deploy your Remote Desktop environment

06/17/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Use the following steps to deploy the Remote Desktop servers in your environment. You can install the server roles on physical machines or virtual machines, depending on whether you're creating an on-premises, cloud-based, or hybrid environment.

If you're using virtual machines for any of the Remote Desktop Services servers, make sure you prepared those virtual machines.

1. Add all the servers you're going to use for Remote Desktop Services to Server Manager:
   a. In Server Manager, click **Manage** > **Add Servers**.
   b. Click **Find Now**.
   c. Click each server in the deployment (for example, Contoso-Cb1, Contoso-WebGw1, and Contoso-Sh1) and click **OK**.

2. Create a session-based deployment to deploy the Remote Desktop Services components:
   a. In Server Manager, click **Manage** > **Add Roles and Features**.
   b. Click **Remote Desktop Services installation**, **Standard Deployment**, and **Session-based desktop deployment**.
   c. Select the appropriate servers for the RD Connection Broker server, RD Web Access server, and RD Session Host server (for example, Contoso-Cb1, Contoso-WebGw1, and Contoso-SH1, respectively).
   d. Select **Restart the destination server automatically if required**, and then click **Deploy**.
   e. Wait for the deployment to complete successfully.

3. Add RD License Server:
   a. In Server Manager, click **Remote Desktop Services** > **Overview** > **+RD Licensing**.
   b. Select the virtual machine on which you want to install the RD license server. For example, Contoso-Cb1.
   c. Click **Next**, and then click **Add**.

4. Activate the RD License Server and add it to the License Servers group:
   a. In Server Manager, click **Remote Desktop Services** > **Servers**. Right-click the server with the Remote Desktop Licensing role installed and select **RD Licensing Manager**.
   b. In RD Licensing Manager, select the server, and then click **Action** > **Activate Server**.
   c. Accept the default values in the Activate Server Wizard. Continue accepting default values until you reach the **Company information** page. Then, enter your company information.

d. Accept the defaults for the remaining pages until the final page. Clear **Start Install Licenses Wizard now**, and then click **Finish**.
e. Select **Action** > **Review Configuration** > **Add to Group** > **OK**. Enter credentials for a user in the AAD DC Administrators group, and register as SCP. This step might not work if you're using Microsoft Entra Domain Services, but you can ignore any warnings or errors.

5. Add the RD Gateway server and certificate name:
   a. In Server Manager, click **Remote Desktop Services** > **Overview** > **+ RD Gateway**.
   b. In the Add RD Gateway Servers wizard, select the virtual machine where you want to install the RD Gateway server (for example, Contoso-WebGw1).
   c. Enter the SSL certificate name for the RD Gateway server using the external fully qualified DNS Name (FQDN) of the RD Gateway server. In Azure, this is the **DNS name** label and uses the format servicename.location.cloudapp.azure.com. For example, contoso.westus.cloudapp.azure.com.
   d. Click **Next**, and then click **Add**.

6. Create and install self-signed certificates for the RD Gateway and RD Connection Broker servers.

   > **Note**
   >
   > If you're providing and installing certificates from a trusted certificate authority, perform the procedures from step h to step k for each role. You need to have the .pfx file available for each of these certificates.

   a. In Server Manager, click **Remote Desktop Services** > **Overview** > **Tasks** > **Edit Deployment Properties**.
   b. Expand **Certificates**, and then scroll down to the table. Click **RD Gateway** > **Create new certificate**.
   c. Enter the certificate name, using the external FQDN of the RD Gateway server (for example, contoso.westus.cloudapp.azure.com) and then enter the password.
   d. Select **Store this certificate** and then browse to the shared folder you created for certificates in a previous step. (For example,\Contoso-Cb1\Certificates.)
   e. Enter a file name for the certificate (for example, ContosoRdGwCert), and then click **Save**.
   f. Select **Allow the certificate to be added to the Trusted Root Certificate Authorities certificate store on the destination computers**, and then click **OK**.
   g. Click **Apply**, and then wait for the certificate to be successfully applied to the RD Gateway server.
   h. Click **RD Web Access** > **Select existing certificate**.

i. Browse to the certificate created for the RD Gateway server (for example, ContosoRdGwCert), and then click **Open**.
j. Enter the password for the certificate, select **Allow the certificate to be added to the Trusted Root Certificate store on the destination computers**, and then click **OK**.
k. Click **Apply**, and then wait for the certificate to be successfully applied to the RD Web Access server.
l. Repeat substeps 1-11 for the **RD Connection Broker - Enable Single Sign On** and **RD Connection Broker - Publishing services**, using the internal FQDN of the RD Connection Broker server for the new certificate's name (for example, Contoso-Cb1.Contoso.com).

7. Export self-signed public certificates and copy them to a client computer. If you're using certificates from a trusted certificate authority, you can skip this step.
   a. Launch certlm.msc.
   b. Expand **Personal**, and then click **Certificates**.
   c. In the right-hand pane right-click the RD Connection Broker certificate intended for client authentication, for example **Contoso-Cb1.Contoso.com**.
   d. Click **All Tasks** > **Export**.
   e. Accept the default options in the Certificate Export Wizard accept defaults until you reach the **File to Export** page.
   f. Browse to the shared folder you created for certificates, for example \Contoso-Cb1\Certificates.
   g. Enter a File name, for example ContosoCbClientCert, and then click **Save**.
   h. Click **Next**, and then click **Finish**.
   i. Repeat substeps 1-8 for the RD Gateway and Web certificate, (for example contoso.westus.cloudapp.azure.com), giving the exported certificate an appropriate file name, for example **ContosoWebGwClientCert**.
   j. In File Explorer, navigate to the folder where the certificates are stored, for example \Contoso-Cb1\Certificates.
   k. Select the two exported client certificates, then right-click them, and click **Copy**.
   l. Paste the certificates on the local client computer.

8. Configure the RD Gateway and RD Licensing deployment properties:
   a. In Server Manager, click **Remote Desktop Services** > **Overview** > **Tasks** > **Edit Deployment Properties**.
   b. Expand **RD Gateway** and clear the **Bypass RD Gateway server for local addresses** option.
   c. Expand **RD licensing** and select **Per User**.
   d. Click **OK**.

9. Create a session collection. These steps create a basic collection. Check out Create a Remote Desktop Services collection for desktops and apps to run for more information

about collections.

a. In Server Manager, click **Remote Desktop Services** > **Collections** > **Tasks** > **Create Session Collection**.
b. Enter a collection Name (for example, ContosoDesktop).
c. Select an RD Session Host Server (Contoso-Sh1), accept the default user groups (Contoso\Domain Users), and enter the Universal Naming Convention (UNC) Path to the user profile disks created previously (\Contoso-Cb1\UserDisks).
d. Set a Maximum size, and then click **Create**.

You now created a basic Remote Desktop Services infrastructure. If you need to create a highly available deployment, you can add a connection broker cluster or a second RD Session Host server.

Create a Remote Desktop Services collection for desktops and apps to run

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Use the following steps to create a Remote Desktop Services session collection. A session collection holds the apps and desktops you want to make available to users. After you create the collection, publish it so users can access it.

Before you create a collection, you need to decide what kind of collection you need: pooled desktop sessions or personal desktop sessions.

- **Use pooled desktop sessions for session-based virtualization**: Leverage the compute power of Windows Server to provide a cost-effective multi-session environment to drive your users' everyday workloads
- **Use personal desktop sessions for to create a virtual desktop infrastructure (VDI)**: Leverage Windows client to provide the high performance, app compatibility, and familiarity that your users have come to expect of their Windows desktop experience.

With a pooled session, multiple users access a shared pool of resources, while with a personal desktop session, users are assigned their own desktop from within the pool. The pooled session provides lower overall cost, while personal sessions enable users to customize their desktop experience.

If you need to share graphics-intensive hosted applications, you can combine personal session desktops with the new Discrete Device Assignment (DDA) capability to also provide support for hosted applications that require accelerated graphics. Check out Which graphics virtualization technology is right for you for more information.

Regardless of the type of collection you choose, you'll populate those collections with RemoteApps - programs and resources that users can access from any supported device and work with as though the program was running locally.

Create a pooled desktop session collection

1. In Server Manager, click **Remote Desktop Services** > **Collections** > **Tasks** > **Create Session Collections**.
2. Enter a name for the collection, for example **ContosoAps**.

3. Select the RD Session Host server you created (for example, Contoso-Shr1).
4. Accept the default **User Groups**.
5. Enter the location of the file share you created for the user profile disks for this collection (for example, **\Contoso-Cb1\UserDisksr**).
6. Click **Create**. When the collection is created, click **Close**.

Create a personal desktop session collection

Use the New-RDSessionCollection cmdlet to create a personal session desktop collection. The following three parameters provide the configuration information required for personal session desktops:

- **-PersonalUnmanaged** - Specifies the type of session collection that lets you assign users to a personal session host server. If you don't specify this parameter, then the collection is created as a traditional RD Session Host collection, where users are assigned to the next available session host when they sign in.
- **-GrantAdministrativePrivilege** - If you use **-PersonalUnmanaged**, specifies that the user assigned to the session host be given administrative privileges. If you don't use this parameter, users are granted only standard user privileges.
- **-AutoAssignUser** - If you use **-PersonalUnmanaged**, specifies that new users connecting through the RD Connection Broker are automatically assigned to an unassigned session host. If there are no unassigned session hosts in the collection, the user will see an error message. If you don't use this parameter, you have to manually assign users to a session host before they sign in.

You can use PowerShell cmdlets to manage your personal desktop session collections. See Manage your personal desktop session collections for more information.

Publish RemoteApp programs

Use the following steps to publish the apps and resources in your collection:

1. In Server Manager, select the new collection (**ContosoApps**).
2. Under RemoteApp Programs, click **Publish RemoteApp programs**.
3. Select the programs you want to publish, and then click **Publish**.

Feedback

Deploy the Remote Desktop Gateway role

06/24/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Desktop Gateway (RD Gateway) role enables secure, encrypted connections to Remote Desktop Services (RDS) resources over the internet. With RD Gateway, users can access internal network resources from remote locations without the need for a VPN. You can deploy RD Gateway servers on either physical or virtual machines, supporting on-premises, cloud, or hybrid environments. This article guides you through installing and configuring the RD Gateway role to enhance the security and accessibility of your Remote Desktop environment.

Prerequisites

Your device must have the **Remote Desktop Services** role installed. To learn more, see Add or remove roles and features in Windows Server.

- During the installation process, select the **Remote Desktop Gateway** Role service, and proceed with installation.

> ⓘ **Note**
>
> For this article, we're using default settings for installing the RD Gateway, progressing through without additional changes. Based on your own organizational needs, during the Role service selection for the Web Server Role (IIS), select the additional services necessary for your environment.

You must be a part of the **Administrators** group or have equivalent permissions.

Configure the RD Gateway role

Once the RD Gateway role is installed, it needs to be configured. To configure the RD Gateway role, follow these steps:

1. In **Server Manager**, select **Remote Desktop Services**, then select **Servers**
2. Right-click the name of your server, then select **RD Gateway Manager**.
3. In the **RD Gateway Manager** on the left pane, right-click the name of your gateway, then select **Properties**.

4. Select the **SSL Certificate** tab, select the **Import a certificate into the RD Gateway** radio button, then select **Browse and Import Certificate**.
5. Locate and select your PFX file, then select **Open**.
6. Enter the password for the PFX file when prompted.

After you've imported the certificate and its private key, the display should show the certificate's key attributes.

> **Note**
>
> Because the RD Gateway role is supposed to be public, we recommend you use a publicly issued certificate. If you use a privately issued certificate, you'll need to make sure to configure all clients with the certificate's trust chain beforehand.

Next steps

If you want to add high availability to your RD Gateway role, see Add high availability to the RD Web and Gateway web front.

Set up the Remote Desktop web client for your users

Article • 04/15/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Desktop web client lets users access your organization's Remote Desktop infrastructure through a compatible web browser. They'll be able to interact with remote apps or desktops like they would with a local PC no matter where they are. Once you set up your Remote Desktop web client, all your users need to get started is the URL where they can access the client, their credentials, and a supported web browser.

> **Important**
>
> The web client does support using Microsoft Entra application proxy but doesn't support Web Application Proxy at all. See Using RDS with application proxy services for details.

What you'll need to set up the web client

Before getting started, keep the following things in mind:

- Make sure your Remote Desktop deployment has an RD Gateway, an RD Connection Broker, and RD Web Access running on Windows Server 2016 or 2019.
- Make sure your deployment is configured for per-user client access licenses (CALs) instead of per-device, otherwise all licenses are consumed.
- Install the Windows 10 KB4025334 update on the RD Gateway. Later cumulative updates might already contain this KB.
- Make sure public trusted certificates are configured for the RD Gateway and RD Web Access roles.
- Make sure that any computers your users connect to are running one of the following OS versions:
  - Windows 10 or later
  - Windows Server 2016 or later

You'll see better performance connecting to Windows Server 2016 (or later) and Windows 10 (version 1611 or later).

If you used the web client during the preview period and installed version 1.0.0 or earlier, you must first uninstall the old client before moving to the new version. If you receive an error that says "*The web client was installed using an older version of RDWebClientManagement and must first be removed before deploying the new version*", follow these steps:

1. Open an elevated PowerShell prompt.
2. Run **Uninstall-Module RDWebClientManagement** to uninstall the new module.
3. Close and reopen the elevated PowerShell prompt.
4. Run **Install-Module RDWebClientManagement -RequiredVersion <old version>** to install the old module.
5. Run **Uninstall-RDWebClient** to uninstall the old web client.
6. Run **Uninstall-Module RDWebClientManagement** to uninstall the old module.
7. Close and reopen the elevated PowerShell prompt.
8. Proceed with the normal installation steps as follows.

How to publish the Remote Desktop web client

To install the web client for the first time, follow these steps:

1. On the RD Connection Broker server, obtain the certificate used for Remote Desktop connections and export it as a `.cer` file. Copy the `.cer` file from the RD Connection Broker to the server running the RD Web role.

2. On the RD Web Access server, open an elevated PowerShell prompt.

3. On Windows Server 2016, update the PowerShellGet module since the inbox version doesn't support installing the web client management module. To update PowerShellGet, run the following cmdlet:

   PowerShell

   ```
   Install-Module -Name PowerShellGet -Force
   ```

   > **Note**
   >
   > To access the PowerShell Gallery, Transport Layer Security (TLS) 1.2 or higher is required. Use the following command to enable TLS 1.2 in your PowerShell session:
   >
   > PowerShell
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   > [Net.ServicePointManager]::SecurityProtocol -bor
   > ```

```
[Net.SecurityProtocolType]::Tls12
```

> **Important**
>
> You'll need to restart PowerShell before the update can take effect, otherwise the module might not work.

4. Install the Remote Desktop web client management PowerShell module from the PowerShell gallery with this cmdlet:

```PowerShell
Install-Module -Name RDWebClientManagement
```

5. After that, run the following cmdlet to download the latest version of the Remote Desktop web client:

```PowerShell
Install-RDWebClientPackage
```

6. Next, run this cmdlet with the bracketed value replaced with the path of the `.cer` file that you copied from the RD Broker:

```PowerShell
Import-RDWebClientBrokerCert <.cer file path>
```

7. Finally, run this cmdlet to publish the Remote Desktop web client:

```PowerShell
Publish-RDWebClientPackage -Type Production -Latest
```

   Make sure you can access the web client at the web client URL with your server name, formatted as `https://server_FQDN/RDWeb/webclient/index.html`. It's important to use the server name that matches the RD Web Access public certificate in the URL (typically the server FQDN).

> **Note**

When running the **Publish-RDWebClientPackage** cmdlet, you might see a warning that says per-device CALs aren't supported, even if your deployment is configured for per-user CALs. If your deployment uses per-user CALs, you can ignore this warning. We display it to make sure you're aware of the configuration limitation.

8. When you're ready for users to access the web client, just send them the web client URL you created.

> **Note**
>
> To see a list of all supported cmdlets for the RDWebClientManagement module, run the following cmdlet in PowerShell:
>
> ```PowerShell
> Get-Command -Module RDWebClientManagement
> ```

How to update the Remote Desktop web client

When a new version of the Remote Desktop web client is available, follow these steps to update the deployment with the new client:

1. Open an elevated PowerShell prompt on the RD Web Access server and run the following cmdlet to download the latest available version of the web client:

   ```PowerShell
   Install-RDWebClientPackage
   ```

2. Optionally, you can publish the client for testing before official release by running this cmdlet:

   ```PowerShell
   Publish-RDWebClientPackage -Type Test -Latest
   ```

   The client should appear on the test URL that corresponds to your web client URL (for example, `<https://server_FQDN/RDWeb/webclient-test/index.html>`).

3. Publish the client for users by running the following cmdlet:

```PowerShell
Publish-RDWebClientPackage -Type Production -Latest
```

This replaces the client for all users when they relaunch the web page.

How to uninstall the Remote Desktop web client

To remove all traces of the web client, follow these steps:

1. On the RD Web Access server, open an elevated PowerShell prompt.
2. Unpublish the Test and Production clients, uninstall all local packages and remove the web client settings:

   ```PowerShell
   Uninstall-RDWebClient
   ```

3. Uninstall the Remote Desktop web client management PowerShell module:

   ```PowerShell
   Uninstall-Module -Name RDWebClientManagement
   ```

How to install the Remote Desktop web client without an internet connection

Follow these steps to deploy the web client to an RD Web Access server that doesn't have an internet connection.

> **Note**
>
> Installing without an internet connection is available in version 1.0.1 and above of the RDWebClientManagement PowerShell module. You still need an admin PC with internet access to download the necessary files before transferring them to the offline server. The end-user PC needs an internet connection for now. This will be addressed in a future release of the client to provide a complete offline scenario.

From a device with internet access

1. Open a PowerShell window.
2. Import the Remote Desktop web client management PowerShell module from the PowerShell gallery:

```
PowerShell

Import-Module -Name RDWebClientManagement
```

3. Download the latest version of the Remote Desktop web client for installation on a different device:

```
PowerShell

Save-RDWebClientPackage "C:\WebClient\"
```

4. Download the latest version of the RDWebClientManagement PowerShell module:

```
PowerShell

Find-Module -Name "RDWebClientManagement" -Repository "PSGallery" | Save-
Module -Path "C:\WebClient\"
```

5. Copy the content of "C:\WebClient" to the RD Web Access server.

From the RD Web Access server

Follow the instructions under How to publish the Remote Desktop web client, replacing steps 4 and 5 with the following.

1. You have two options to retrieve the latest web client management PowerShell module:

   a. Import the Remote Desktop web client management PowerShell module:

   ```
   PowerShell

   Import-Module -Name RDWebClientManagement
   ```

   b. Copy the downloaded RDWebClientManagement folder to one of the local PowerShell module folders listed under **$env:psmodulePath**, or add the path to the folder with the downloaded files to the **$env:psmodulePath**.

2. Deploy the latest version of the Remote Desktop web client from the local folder (replace with the appropriate zip file):

PowerShell

```
Install-RDWebClientPackage -Source "C:\WebClient\rdwebclient-1.0.1.zip"
```

Connecting to RD Broker without RD Gateway in Windows Server 2019

This section describes how to enable a web client connection to an RD Broker without an RD Gateway in Windows Server 2019.

Setting up the RD Broker server

Follow these steps if there's no certificate bound to the RD Broker server

1. Open **Server Manager** > **Remote Desktop Services**.
2. In **Deployment Overview** section, select the **Tasks** dropdown menu.
3. Select **Edit Deployment Properties**, a new window titled **Deployment Properties** opens.
4. In the **Deployment Properties** window, select **Certificates** in the left menu.
5. In the list of Certificate Levels, select **RD Connection Broker - Enable Single Sign On**. You have two options: (1) create a new certificate or (2) an existing certificate.

Follow these steps if there's a certificate previously bound to the RD Broker server

1. Open the certificate bound to the Broker and copy the **Thumbprint** value.
2. To bind this certificate to the secure port 3392, open an elevated PowerShell window and run the following command, replacing "< **thumbprint** >" with the value copied from the previous step:

PowerShell

```
netsh http add sslcert ipport=0.0.0.0:3392 certhash="<thumbprint>"
certstorename="Remote Desktop" appid="{00000000-0000-0000-0000-000000000000}"
```

> ⓘ Note

To check if the certificate is bound correctly, run the following command:

```
PowerShell

netsh http show sslcert
```

In the list of SSL Certificate bindings, ensure that the correct certificate is bound to port 3392.

3. Open the Windows Registry (regedit), go to `HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp` and locate the key **WebSocketURI**. Next, set the value to `https://+:3392/rdp/`.

Setting up the RD Session Host

Follow these steps if the RD Session Host server is different from the RD Broker server:

1. Create a certificate for the RD Session Host machine, open it, and copy the **Thumbprint** value.

2. To bind this certificate to the secure port 3392, open an elevated PowerShell window and run the following command, replacing "< **thumbprint** >" with the value copied from the previous step:

```
PowerShell

netsh http add sslcert ipport=0.0.0.0:3392 certhash="<thumbprint>" appid="
{00000000-0000-0000-0000-000000000000}"
```

> ⓘ **Note**
>
> To check if the certificate is bound correctly, run the following command:
>
> ```
> PowerShell
>
> netsh http show sslcert
> ```
>
> In the list of SSL Certificate bindings, ensure that the correct certificate is bound to port 3392.

3. Open the Windows Registry (regedit) and navigate to `HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp` and locate

the key **WebSocketURI**. The value must be set to `https://+:3392/rdp/`.

General Observations

Ensure that both the RD Session Host and RD Broker server are running Windows Server 2019.

Ensure that public trusted certificates are configured for both the RD Session Host and RD Broker server.

> **Note**
>
> If both the RD Session Host and the RD Broker server share the same machine, set the RD Broker server certificate only. If the RD Session Host and RD Broker server use different machines, both must be configured with unique certificates.

The **Subject Alternative Name (SAN)** for each certificate must be set to the machine's **Fully Qualified Domain Name (FQDN)**. The **Common Name (CN)** must match the SAN for each certificate.

How to pre-configure settings for Remote Desktop web client users

This section tells you how to use PowerShell to configure settings for your Remote Desktop web client deployment. These PowerShell cmdlets control a user's ability to change settings based on your organization's security concerns or intended workflow. The following settings are all located in the **Settings** side panel of the web client.

Suppress telemetry

By default, users might choose to enable or disable collection of telemetry data that is sent to Microsoft. For information about the telemetry data Microsoft collects, refer to our Privacy Statement via the link in the **About** side panel.

As an administrator, you can choose to suppress telemetry collection for your deployment using the following PowerShell cmdlet:

PowerShell

```
Set-RDWebClientDeploymentSetting -Name "SuppressTelemetry" $true
```

By default, the user might select to enable or disable telemetry. A boolean value **$false** will match the default client behavior. A boolean value **$true** disables telemetry and restricts the user from enabling telemetry.

Remote resource launch method

> **Note**
>
> This setting currently only works with the RDS web client, not the Azure Virtual Desktop web client.

By default, users might choose to launch remote resources (1) in the browser or (2) by downloading an `.rdp` file to handle with another client installed on their machine. As an administrator, you can choose to restrict the remote resource launch method for your deployment with the following PowerShell command:

```
PowerShell

Set-RDWebClientDeploymentSetting -Name "LaunchResourceInBrowser" ($true|$false)
```

By default, the user might select either launch method. A boolean value **$true** will force the user to launch resources in the browser. A boolean value **$false** forces the user to launch resources by downloading an `.rdp` file to handle with a locally installed RDP client.

Reset RDWebClientDeploymentSetting configurations to default

To reset a deployment-level web client setting to the default configuration, run the following PowerShell cmdlet and use the -name parameter to specify the setting you want to reset:

```
PowerShell

Reset-RDWebClientDeploymentSetting -Name "LaunchResourceInBrowser"
Reset-RDWebClientDeploymentSetting -Name "SuppressTelemetry"
```

Troubleshooting

If a user reports any of the following issues when opening the web client for the first time, the following sections explain what to do to fix them.

What to do if the user's browser shows a security warning when they try to access the web client

The RD Web Access role might not be using a trusted certificate. Make sure the RD Web Access role is configured with a publicly trusted certificate.

If that doesn't work, your server name in the web client URL might not match the name provided by the RD Web certificate. Make sure your URL uses the FQDN of the server hosting the RD Web role.

What to do if the user can't connect to a resource with the web client even though they can see the items under All Resources

If the user reports that they can't connect with the web client even though they can see the resources listed, check the following things:

- Is the RD Gateway role properly configured to use a trusted public certificate?
- Does the RD Gateway server have the required updates installed? Make sure that your server has the KB4025334 update installed.

If the user gets an "unexpected server authentication certificate was received" error message when they try to connect, then the message shows the certificate's thumbprint. Search the RD Broker server's certificate manager using that thumbprint to find the right certificate. Verify that the certificate is configured to be used for the RD Broker role in the Remote Desktop deployment properties page. After making sure the certificate hasn't expired, copy the certificate in `.cer` file format to the RD Web Access server and run the following command on the RD Web Access server with the bracketed value replaced by the certificate's file path:

```
PowerShell

Import-RDWebClientBrokerCert <certificate file path>
```

Diagnose issues with the console log

If you can't solve the issue based on the troubleshooting instructions in this article, you can try to diagnose the source of the problem yourself by watching the console sign in the browser. The web client provides a method for recording the browser console log activity while using the web client to help diagnose issues.

- Select the ellipsis in the upper-right corner and navigate to the **About** page in the dropdown menu.
- Under **Capture support information** select the **Start recording** button.
- Perform one or more operations in the web client that produced the issue you're trying to diagnose.
- Navigate to the **About** page and select **Stop recording**.
- Your browser will automatically download a .txt file titled **RD Console Logs.txt**. This file contains the full console log activity generated while reproducing the target issue.

The console might also be accessed directly through your browser. The console is located under the developer tools. For example, you can access the sign-in in Microsoft Edge by pressing the **F12** key, or by selecting the ellipsis, then navigating to **More tools** > **Developer Tools**.

Get help with the web client

If you've encountered an issue that can't be solved by the information in this article, you can report it on the Azure Virtual Desktop forum of Microsoft Tech Community.

Disable Automatic Reconnection

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Learn about Automatic Reconnection in Remote Desktop Service (RDS), lock screen security, and how to disable Automatic Reconnection for RDS session hosts and clients using Server Manager, Group Policy, and Remote Desktop Protocol (RDP) properties.

Automatic Reconnection

Microsoft Remote Desktop offers a wide range of features designed to enhance your remote working experience, such as Automatic Reconnection. Automatic Reconnection allows the client to seamlessly reconnect to their existing sessions, giving a smooth, uninterrupted user experience when temporary network disruptions occur. To learn more about the automatic reconnection behavior, see the Automatic Reconnection open specification. Automation Reconnection is available to Remote Desktop when connecting to a local PC or Remote Desktop Services (RDS).

> **ⓘ Important**
>
> Automatic Reconnection is enabled by default, and therefore, requires explicit action from the administrators to disable it.

Lock Screen Security

When a policy or the user locks the remote session and the network connection is lost or disrupted, RDS retains the session state and connection information. If the automatic reconnection of locked sessions raises concerns for your specific use case, we recommend implementing extra security measures. Because RDS retains the session state and connection information, the client reconnects without needing to reauthenticate. The lock screen of the Remote Desktop session isn't designed to function as a security boundary. Security measures can include disabling Automatic Reconnection on either the RDS session host or the client. This article describes how to disabled Automatic Reconnection.

Prerequisites

Before you can configure Automatic Reconnection for Remote Desktop, you need to complete the following prerequisites:

- A Windows client or Windows Server machine to connect from and to.
- An account that is a member of RDS session host administrators group, or equivalent.
- If your machine is a domain member, you also need a domain account that is a member of the Group Policy Creator Owners group, or equivalent.

If you're using RDS, you also need:

- A Windows Server with the RDS installed and configured. To learn more about deploying RDS, see Deploy your Remote Desktop environment.
- A Remote Desktop Session Collection. To learn more about creating a Remote Desktop Session Collection, see Create a Remote Desktop Services collection for desktops and apps to run.

Methods to disable Automatic Reconnection

To disable Automatic Reconnection, you can configure your server, client, or both.

> **Tip**
>
> - If you disable Automatic Reconnection from your server, clients will be unable to perform Automatic Reconnection regardless of the client configuration.
> - Changes to the Automatic Reconnection setting only apply to new sessions. Existing sessions will continue to use the Automatic Reconnection setting from the time of connection.

Client RDP Properties

You can configure the following Remote Desktop Protocol (RDP) property to disable Automatic Reconnection using the Remote Desktop Connection app or by editing the `.rdp` file. More information can be found here: Supported RDP properties with Remote Desktop Services. To disabled Automatic Reconnection, select the relevant method and follow the steps.

RDP file

Here's how to disable Automatic Reconnection by editing the `.rdp` file.

1. Locate your `.rdp` file, right-click the file, expand the **Open with** menu, then select **Choose another app**.
2. Select **Notepad**, then select **Just once**
3. Scroll to the last line of the file, then enter the following text.

```
RDP

autoreconnection enabled:i:0
```

Remote Desktop Services server configuration

To disable Automatic Reconnection for your RDS session host, select the relevant method and follow the steps.

> **Tip**
>
> If you have an RDS deployment and want to configure Automatic Reconnection using the Session Collection properties, Group Policy must be in the **Not Configured** state for each session host. The Group Policy setting applied to each session host takes priority over the Automatic Reconnection setting for the Remote Desktop Session Collection.

Group Policy

Here's how to disable Automatic Reconnection for RDS session hosts using Group Policy.

1. Open the **Group Policy Management Console**, create or edit a policy applied to your server.
2. In the console tree, select **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Service** > **Remote Desktop Session Host** > **Connections**.
3. For the setting, right-click **Automatic reconnection** and select **Edit**.
4. Select **Disable**, from the radio buttons.

5. Select **OK** to complete the configuration.

Next steps

- Remote Desktop clients for Remote Desktop Services and remote PCs.

Feedback

Was this page helpful? Yes No

Set up email discovery to subscribe to your RDS feed

Article • 05/01/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Have you ever had trouble getting your end users connected to their published RDS feed, either because of a single missing character in the feed URL or because they lost the email with the URL? Nearly all Remote Desktop client applications support finding your subscription by entering your email address, making it easier than ever to get your users connected to their RemoteApps and desktops.

Before you set up email discovery, do the following:

- Make sure you have permission to add a TXT record to the domain associated with your email (for example, if your users have @contoso.com email addresses, you would need permissions for the contoso.com domain).
- Create an RD Web feed URL (`https://<rdweb-dns-name>.<domain>/RDWeb/Feed/webfeed.aspx`), such as `https://rdweb.contoso.com/RDWeb/Feed/webfeed.aspx`.

> **Note**
>
> If you're using Azure Virtual Desktop instead of Remote Desktop, you'll want to use these URLs instead:
>
> - If you're using Azure Virtual Desktop (classic): `https://rdweb.wvd.microsoft.com/api/feeddiscovery/webfeeddiscovery.aspx`
> - If you're using Azure Virtual Desktop: `https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery`

Now, follow these steps to set up email discovery:

1. In your browser, connect to the website of the domain name registrar where your domain is registered.
2. Navigate to the appropriate page for your registered domain where you can view, add, and edit DNS records.
3. Enter a new DNS record with the following properties:

- **Host:** _msradc
- **Text:** <RD Web Feed URL>
- **TTL:** 300 seconds

The names of the DNS records fields vary by domain name registrar, but this process will result in a TXT record named _msradc.<domain_name> (such as _msradc.contoso.com) that has a value of the full RD Web feed.

That's it! Now, launch the Remote Desktop application on your device and subscribe yourself!

Fair Share technologies are enabled by default in Remote Desktop Services

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes how a Remote Desktop Session Host (RDSH) server, Windows 10 Enterprise multi-session, Windows 11 Enterprise multi-session, and Windows Server use Fair Share technologies to balance CPU, disk, and network bandwidth resources among multiple Remote Desktop sessions.

_ *Original KB number:* 4494631

Introduction

Remote Desktop Services (RDS) server, Windows 10 Enterprise multi-session and Windows 11 Enterprise multi-session use Fair Share technologies for CPU resources to manage resources. RDS builds on the Fair Share technologies to add features for allocating network bandwidth and disk resources. Fair Share CPU Scheduling is enabled by default, while Dynamic Disk Fair Share and Dynamic Network Fair Share are disabled. You can change the defaults by using PowerShell and WMI.

For more information about the related properties in WMI, see Win32_TerminalServiceSetting class: Properties.

> ⓘ **Note**
>
> Before turning on Dynamic Disk Fair Share or Dynamic Network Fair Share, it's recommended to review performance on applications that require exchanging larger amounts of data.

Fair Share CPU Scheduling

Fair Share CPU Scheduling dynamically distributes processor time across all RDS and Azure Virtual Desktop (AVD) multi-session sessions on the same Session Host server, based on the number of sessions and the demand for processor time within each session. This process creates a consistent user experience across all of the active sessions, while sessions are being created and deleted dynamically. This feature builds

on the Dynamic Fair Share Scheduling technology (DFSS) that was part of Windows Server.

Dynamic Disk Fair Share

When disk-intensive processes run in one or more sessions, they can starve non-disk intensive processes and prevent them from ever accessing disk resources. To fix this issue, the Dynamic Disk Fair Share feature balances disk access among the different sessions by balancing disk IO and throttling excess disk usage.

Dynamic Network Fair Share

When bandwidth-intensive applications run in one or more sessions, they can starve applications in other sessions of bandwidth. To equalize network consumption among the sessions, the Network Fair Share feature uses a round-robin approach to allocate bandwidth for each session.

In a centralized computing scenario, the Dynamic Network Fair Share feature tries to fairly distribute network interface bandwidth load among the sessions.

Feedback

License Remote Desktop Services with Client Access Licenses (CALs)

06/16/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Each user and device that connects to a Remote Desktop Services session host or Azure Virtual Desktop session host running Windows Server needs a Remote Desktop Services (RDS) client access license (CAL).

This article explains RDS CAL licensing to help you deploy and manage licenses effectively.

Understand the RDS CAL model

You use a Remote Desktop Licensing server to install, issue, and track RDS CALs. When a user or a device connects to a session host, the session host determines if an RDS CAL is needed. If an RDS CAL is needed, the session host then requests an RDS CAL from a Remote Desktop Licensing server. If an appropriate RDS CAL is available from a license server, the RDS CAL is issued to the client, and the user is able to connect to a remote session for the desktop or apps they're trying to use.

There are two types of RDS CALs: **per device** and **per user**. The type of RDS CAL you need depends on how your users or devices access the session host.

The following table provides a summary comparison of the two types of RDS CAL:

Expand table

Per device RDS CAL	Per user RDS CAL
Physically assigned to each device.	Assigned to a user in Active Directory.
Tracked by the license server.	Tracked by the license server.
Can be tracked regardless of Active Directory membership.	Can't be tracked within a workgroup.
You can revoke up to 20% of RDS CALs.	You can't revoke any RDS CALs.
Temporary RDS CALs are assigned on first sign-in per device and are valid for 90 days.	Temporary RDS CALs aren't available.
Permanent RDS CALs are valid for a random period of 52–89 days before renewal.	Permanent RDS CALs are valid for 60 days before renewal or 90 days before reassignment.

Per device RDS CAL	Per user RDS CAL
Can't be over-allocated.	Can be over-allocated, in breach of the Remote Desktop licensing agreement.

Here's an example of when you could use each of the RDS CAL types:

- The per device model would be appropriate in an environment where there are two or more shifts of workers using the same computers to access the session hosts.
- The per user model would be best for environments where each user has their own dedicated Windows devices to access the session hosts.

There's a licensing grace period of 120 days during which no license server is required. Once the grace period ends, clients must have a valid RDS CAL issued by a license server before they can sign in a remote session.

A license server can issue RDS CALs across different Active Directory domains or forests, or in a workgroup environment. However, there are some limitations to consider. For more information, see Best practices for setting up Remote Desktop licensing across Active Directory domains/forests or workgroups.

Per device RDS CALs

When you use the per device model, a temporary license is issued the first time a device connects to a session host. After a user signs in to the session, the session host instructs the license server to mark the issued temporary RDS CAL token as validated.

The next time that device connects, as long as the license server is activated and there are available RDS CALs, the license server upgrades the temporary RDS CAL token to a full RDS CAL token and issues a permanent per device RDS CAL. If no RDS CAL tokens are available, the temporary RDS CAL token continues to function for 90 days.

Every time the client device connects to the session host, it presents its RDS CAL certificate to the server. The server checks not only whether the client device has a valid certificate, but also the expiration date of that certificate. If the expiration date of the certificate is within seven days of the current date, the session host connects to the license server to renew the license for another random period of 52 to 89 days.

Per user RDS CALs

When you use the per user model, licensing isn't enforced, and each user is granted a license to connect to a session host from any number of devices. The license server issues an RDS CAL from the available pool or the overused RDS CAL pool. It's the administrator's responsibility to

ensure that all users have valid licenses and not using overused RDS CALs, to avoid violating the Remote Desktop Services license terms.

Per user RDS CALs show as expiring 60 days after they're issued. Shortly before their expiration date, when the user signs in, the date is extended another 60 days. If a user doesn't sign in before the expiration date, they drop off the list, but the next time they sign in they show up again with a new expiration date.

For most license agreements, 90 days is the more relevant time period, because it's the minimum time required before a license can be reassigned to a different user, except under special circumstances.

You can use the Remote Desktop Licensing Manager to track and generate reports on per user RDS CALs. To ensure you're in compliance with the Remote Desktop Services license terms, track the number of per user RDS CALs used in your organization. Be sure to have enough per user RDS CALs installed on the license server for all of your users.

RDS CAL version compatibility

The RDS CAL for your users or devices must be compatible with the version of Windows Server that the user or device is connecting to. You can't use RDS CALs for earlier versions to access later versions of Windows Server, but you can use later versions of RDS CALs to access earlier versions of Windows Server. For example, if you have RDS CALs for Windows Server 2022, you can connect to a session host running Windows Server 2022 or earlier, but you can't use it to connect to a session host running Windows Server 2025.

The following table shows which version of Windows Server for RDS CALs and session hosts are compatible with each other.

Expand table

Session host Window Server version	Windows Server 2025 RDS CAL	Windows Server 2022 RDS CAL	Windows Server 2019 RDS CAL	Windows Server 2016 RDS CAL
Windows Server 2025	Yes	No	No	No
Windows Server 2022	Yes	Yes	No	No
Windows Server 2019	Yes	Yes	Yes	No

Session host Window Server version	Windows Server 2025 RDS CAL	Windows Server 2022 RDS CAL	Windows Server 2019 RDS CAL	Windows Server 2016 RDS CAL
Windows Server 2016	Yes	Yes	Yes	Yes

You must also install your RDS CALs on a Remote Desktop Licensing server running a compatible version of Windows Server. You can install RDS CALs on a license server running the same version of Windows Server as the RDS CALs or earlier. For example, if you have RDS CALs for Windows Server 2022, you can install them on a license server running Windows Server 2022 or earlier, but you can't use it to install RDS CALs for Windows Server 2025.

The following table shows which RDS CAL and license server versions are compatible with each other.

Expand table

License server Window Server version	Windows Server 2025 RDS CAL	Windows Server 2022 RDS CAL	Windows Server 2019 RDS CAL	Windows Server 2016 RDS CAL
Windows Server 2025	Yes	Yes	Yes	Yes
Windows Server 2022	No	Yes	Yes	Yes
Windows Server 2019	No	No	Yes	Yes
Windows Server 2016	No	No	No	Yes

Next step

Install Remote Desktop Services client access licenses.

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

Activate the Remote Desktop Services license server

Article • 02/14/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Desktop Services license server issues client access licenses (CALs) to users and devices when they access the RD Session Host. You can activate the license server by using the Remote Desktop Licensing Manager.

Prerequisites

Before you can activate your Remote Desktop Services license server, ensure that you have:

- A Windows Server with the Remote Desktop Services licensing role service installed, including the Remote Desktop Licensing Tools feature.
- Administrative privileges on the server.
- A basic understanding of Remote Desktop licensing.
- You need to have you company information ready to enter during the activation process.

Install the Remote Desktop Licensing role

1. Sign into the server you want to use as the license server using an administrator account.
2. In Server Manager, select **Manage** > **Add Roles and Features**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**.
4. Specify the server on which you install the licensing role.
5. On the **Server Roles** page, check the box for **Remote Desktop Services**, then select **Next** until you see the **Remote Desktop Services** page.
6. Select the roles you want to install. Make sure you include the **Remote Desktop Licensing** role.

7. In the **Add Roles and Features Wizard** dialog box, select **Add Features**.

8. Select **Next** until you see the **Confirmation** page, then select **Install**.

For detailed information and other installation options, see Install or uninstall roles, role services, or features

Activate the license server

1. In Server Manager, select **Tools** > **Remote Desktop Services** > **Remote Desktop Licensing Manager**.

2. In the **RD Licensing Manager**, select the server, and then select **Action** > **Activate Server**.

3. Confirm your preferred Connection method for license server activation and select **Next**. The three options available are:

   - **Automatic connection (recommended)**: This recommended method communicates via internet directly to the Microsoft Clearinghouse outbound over TCP port 443.

   - **Web Browser**: This method requires the administrator to connect to the Microsoft Clearinghouse web site. Use this method if the license server doesn't have internet access, but you have internet access through another computer.

   - **Telephone**: This method allows the administrator to complete the migration process over the phone with a Microsoft Clearinghouse operator. Use this method if you don't have internet access.

Activate with automatic connection

1. Enter your required **Company information** including First name, Family name, Company, and Country or Region. Select **Next**.

2. Then enter your optional company information. Select **Next** until you complete the Activate Server Wizard.

3. Accept the defaults for the remaining pages until the final page. Clear **Start Install Licenses Wizard now**, and then select **Finish**.

4. Select **Action** > **Install Licenses**. Enter your license code ready to enter when prompted.

Activate using a web browser

1. On the **License Server Activation** page, copy the URL for the Remote Desktop Licensing Web Site. Then open a web browser and navigate to the site.
2. Complete the steps on the Remote Desktop Licensing Web Site.
3. Return to the **License Server Activation** page and enter in the **license server ID**. Select **Next**.

Activate by telephone

1. Select your Country or Region. Then select **Next**.
2. On the **License Server Activation** page, call Microsoft at the number displayed. The representative provides you with a license server ID to enter. Select **Next**.

Feedback

Reactivate or deactivate a Remote Desktop Services license server

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

In this article learn how-to reactivate or deactivate a Remote Desktop Services license server automatically over the internet, using a web browser, or by telephone.

Prerequisites

Consider the following prerequisites before either reactivating or deactivating the license server:

- You'll need to know which server is the license server.
- You'll need membership in the **Administrators** group, or equivalent.

Reactive a license server

You must reactivate a Remote Desktop license server when any of the following occur:

- The license server certificate expired.
- The license server certificate was corrupted.
- The license server was upgraded.
- The license server was redeployed.
- The license server private key was compromised.

Reactivate a license server by using one of the following methods:

- Reactivate a license server automatically
- Reactivate a license server using a web browser
- Reactivate a license server by telephone

Reactivate a license server automatically

The automatic reactivation method requires internet connectivity from the computer running the Remote Desktop Licensing Manager tool. Complete the following steps to reactivate a license server automatically:

1. In Server Manager, select **Tools** > **Remote Desktop Services** > **Remote Desktop Licensing Manager**.
2. In RD Licensing Manager, verify that the connection method for the license server is set to **Automatic connection (recommended)**. To do this, right-click the license server that you want to reactivate, and then select **Properties**. On the **Connection Method** tab, change the connection method if necessary.
3. Right-click the license server that you want to reactivate, point to **Advanced**, and then select **Reactivate Server**. The Reactivate Server Wizard starts.
4. On the **Welcome** page, select **Next**.
5. On the **Information Needed** page, provide the requested information, and then select **Next**.
6. Your request to reactivate the license server is sent to the Microsoft Clearinghouse for processing, and the license server is reactivated.
7. On the Completing the Reactivate Server Wizard page, select **Finish**.

Reactivate a license server using a web browser

The web reactivation method can be used when the computer running the Remote Desktop Licensing Manager tool doesn't have internet connectivity, but you have access to the web from another computer. The URL for the web method is displayed in the Reactivate Server Wizard.

To reactivate a Remote Desktop Licensing Manager server by using a web browser, complete the following steps:

1. In Server Manager, select **Tools** > **Remote Desktop Services** > **Remote Desktop Licensing Manager**.
2. Verify that the connection method for the Remote Desktop license server is set to **Web Browser** by right-clicking the license server that you want to reactivate, and then select **Properties**.
3. Using a computer that has internet connectivity, connect to the Remote Desktop Licensing website.
4. On the Remote Desktop Licensing website, select the option to **Reactivate a license server** and then select **Next**. Follow the steps to reactivate the license server.

Reactivate a license server by telephone

The telephone reactivation method allows you to talk to a Microsoft customer service representative to complete the reactivation process. The appropriate telephone number depends on the country/region that is configured for the Remote Desktop Licensing Manager, and that telephone number is displayed in the Reactivate Server Wizard.

To reactivate a Remote Desktop Licensing Manager server by telephone, complete the following steps:

1. In Server Manager, select **Tools** > **Remote Desktop Services** > **Remote Desktop Licensing Manager**.
2. Verify that the connection method for the Remote Desktop license server is set to **Telephone** by right-clicking the license server that you want to reactivate, and then select **Properties**. On the **Connection Method** tab, change the connection method, if necessary, ensure that the correct country or region is selected in the Select Country or Region list, and then select **OK**.
3. Right-click the license server that you want to reactivate, point to **Advanced**, and then select **Reactivate Server**. The Reactivate Server Wizard starts.
4. On the **Welcome** page, select **Next**.
5. Call Microsoft by using the telephone number that is displayed on the **License Server Reactivation** page, and then provide the Microsoft customer support representative with the product ID and license server ID that is displayed on your screen.
6. The representative processes your request to reactivate the license server, and provides you with a new license server ID. On the **License Server Reactivation** page in the Reactivate Server Wizard, type the new license server ID that the representative provides, and then select **Next**. The license server is reactivated.
7. On the **Completing the Reactivate Server Wizard** page, select **Finish**.

Deactive a license server

You may have to deactivate a license server if the certificate of the server expires, gets damaged, or if you redeploy the server.

The following steps describe how to deactivate a license server:

1. In Server Manager, select **Tools** > **Remote Desktop Services** > **Remote Desktop Licensing Manager**.
2. In the console tree, right-click the license server that you want to deactivate, select **Advanced**, and then select **Deactivate Server**.
3. In the Deactivate Server Wizard, confirm that your name, your phone number (optional), and your e-mail address that are listed under Information Needed are correct. Then select **Next**. Your request to deactivate the license server is sent to Microsoft Clearinghouse for processing.

   > **Note**
   >
   > Your e-mail address is required if you are using the Internet method.

4. Select Finish.

When you deactivate a license server, you can't license other client computers from this server until the license server is activated again.

Feedback

Install Remote Desktop Services client access licenses

05/29/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) client access licenses (CALs) are required for users to connect to your Remote Desktop environment and in compliance with licensing requirements. This article provides instructions for installing RDS CALs on your license server. Once the CALs are installed and you configure your RDS deployment to use the license server, licenses are issued to users as appropriate.

Prerequisites

Before you begin, ensure you meet the following prerequisites:

- A Remote Desktop license server installed and activated. For activation instructions, see Activate the Remote Desktop Services license server.
- Internet connectivity on a computer running Remote Desktop Licensing Manager to activate the licenses. The license server itself doesn't require internet connectivity.
- A user account that's an administrator on the Remote Desktop license server to install licenses.
- The license code or agreement number provided when you purchased your Remote Desktop Services CALs.

Install the client access licenses

1. Open the Remote Desktop Licensing Manager.
2. Right-click the name of the license server, then select **Install licenses**.
3. On the welcome page, select **Next** .
4. Select the program you purchased your RDS CALs from, then select **Next**. If you're a service provider, select **Service Provider License Agreement**.
5. Enter the information for your license program. In most cases, you need to enter the license code or an agreement number, but it can vary depending on the license program

you're using.

6. Select **Next**.
7. Select the product version, license type, and number of licenses for your environment, then select **Next**. The license manager contacts the Microsoft Clearinghouse to validate and retrieve your licenses.
8. Select **Finish** to complete the process.

Next step

After you install the RDS CALs, you need to configure your RDS deployment to use the license server. For more information, see License Remote Desktop session hosts.

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

License Remote Desktop session hosts

Article • 05/06/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use the information in this article to configure licensing for session hosts on your Remote Desktop Services (RDS) deployments. The process is slightly different depending on which roles you assigned to the session host you're licensing.

Prerequisites

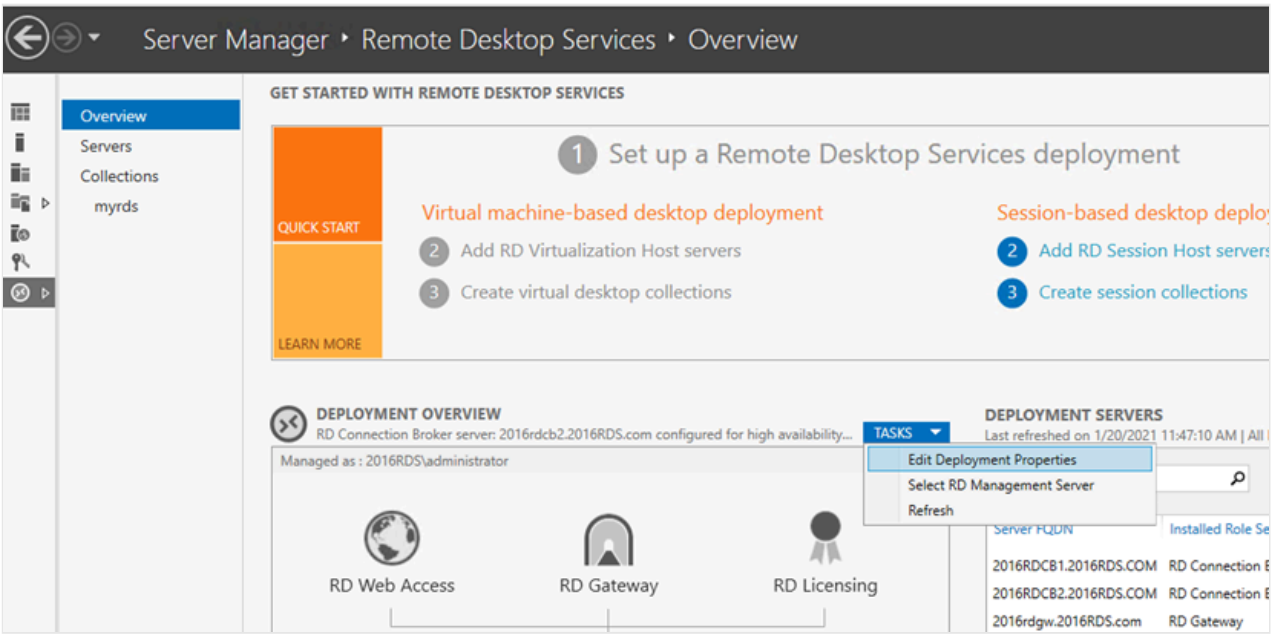
In order to install licenses for your session hosts, you need a Remote Desktop license server with per-user or per-device client access licenses (CALs) activated.

Configure licensing for an RDS deployment that includes the RD Connection Broker role

If you need to license session hosts where your RDS deployment doesn't include the connection broker role, you must specify a license server by using group policy either centrally from your Active Directory domain, or locally on each session host. You also need to do specify a license server when using Windows Server with Azure Virtual Desktop.

To specify a license server:

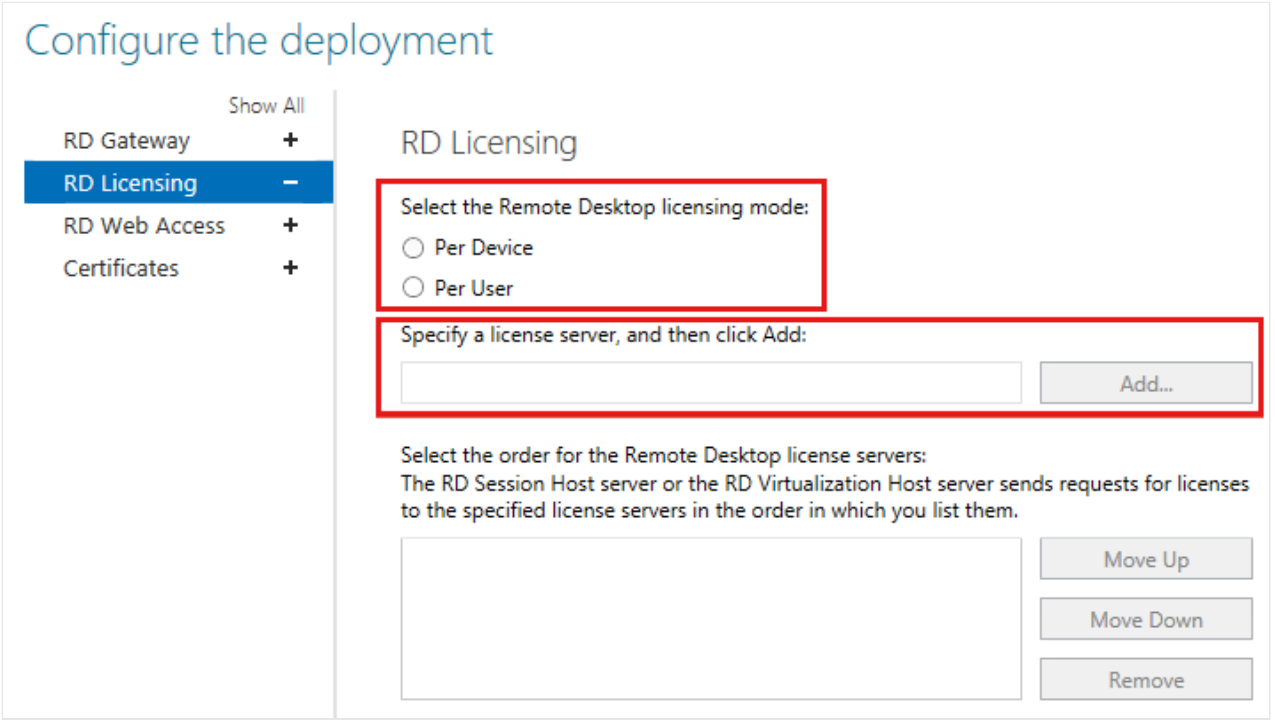
1. On the RD Connection Broker computer, open **Server Manager**.
2. In **Server Manager**, select **Remote Desktop Services** > **Overview** > **Edit Deployment Properties** > **RD Licensing**.



3. Select the Remote Desktop licensing mode (either **Per User** or **Per Device**, as appropriate for your deployment).

> **Note**
>
> If you use domain-joined servers for your RDS deployment, you can use both Per User and Per Device CALs. If you use workgroup servers for your RDS deployment, you have to use Per Device CALs In that case, Per User CALs aren't permitted.

4. Specify a license server, and then select **Add**.



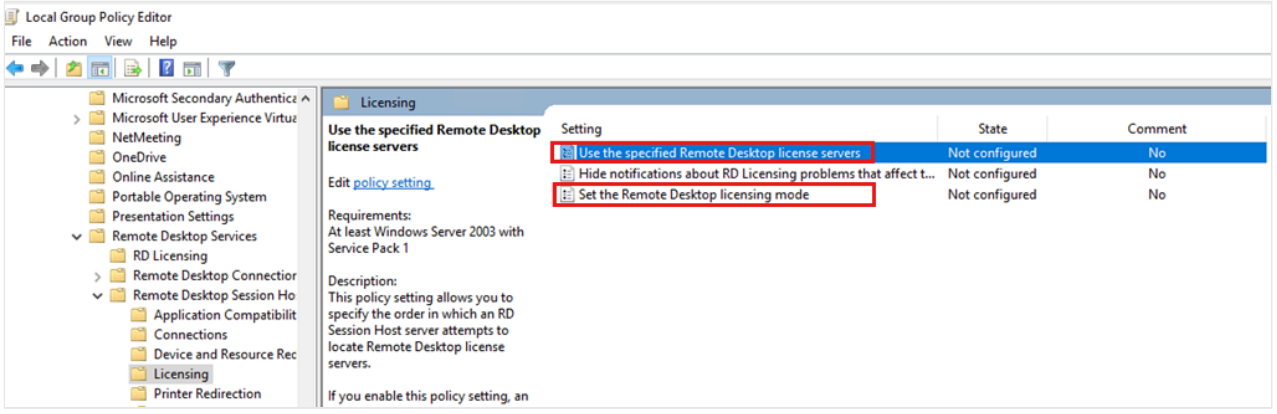
Configure licensing for an RDS deployment that includes only the RD Session Host role and the RD Licensing role

1. Depending on whether you want to configure Group Policy centrally from your domain or locally on each session host:

    - Open the **Group Policy Management Console** (GPMC) and create or edit a policy that targets your session hosts.

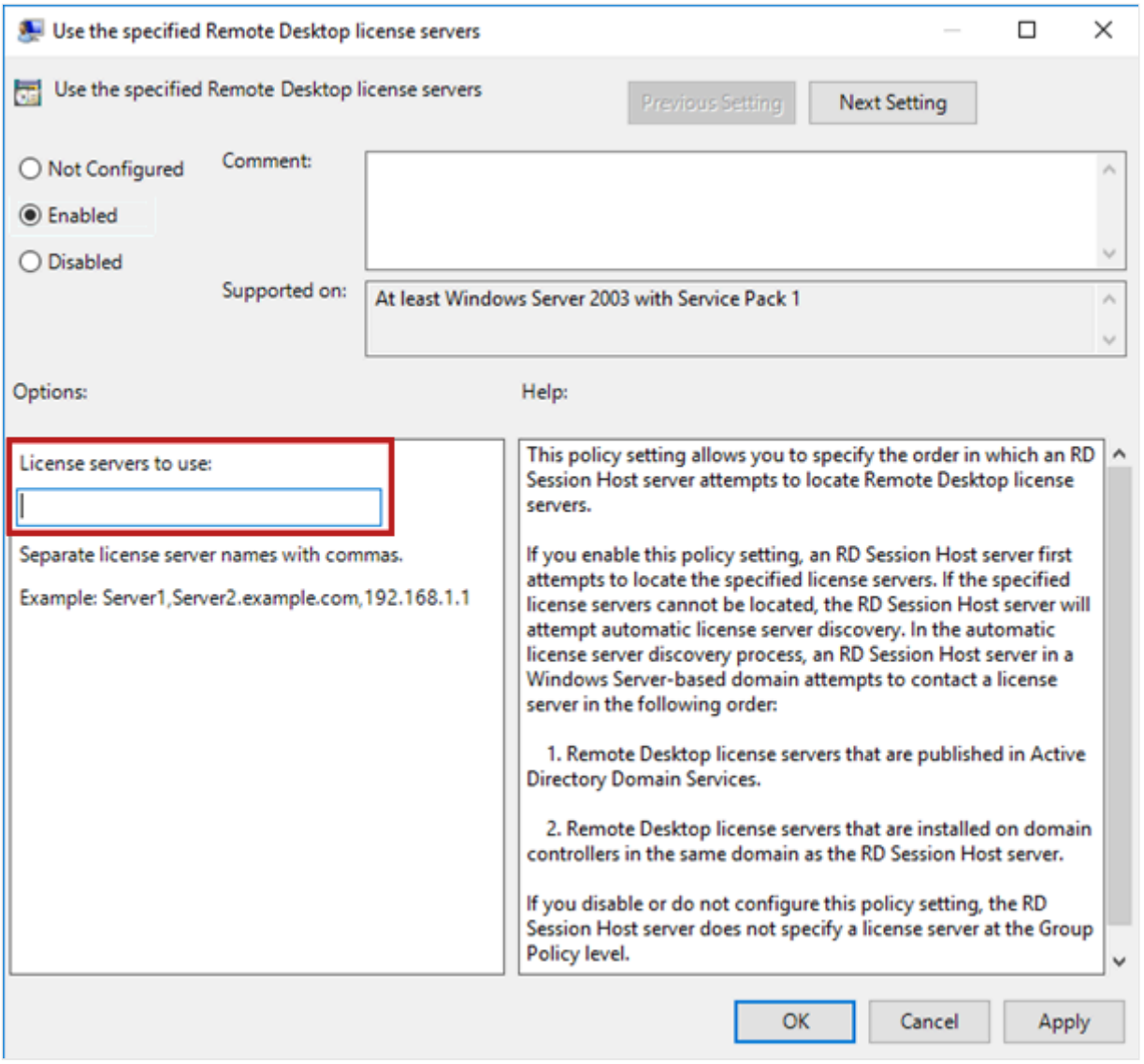
    - Open the **Local Group Policy Editor** on the session host.

2. Go to **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Licensing**.



3. In the policy list, right-click **Use the specified Remote Desktop license servers**, and then select **Properties**.

4. Select **Enabled**, and then enter the name of the license server under **License servers to use**. If you have more than one license server, use commas to separate their names.

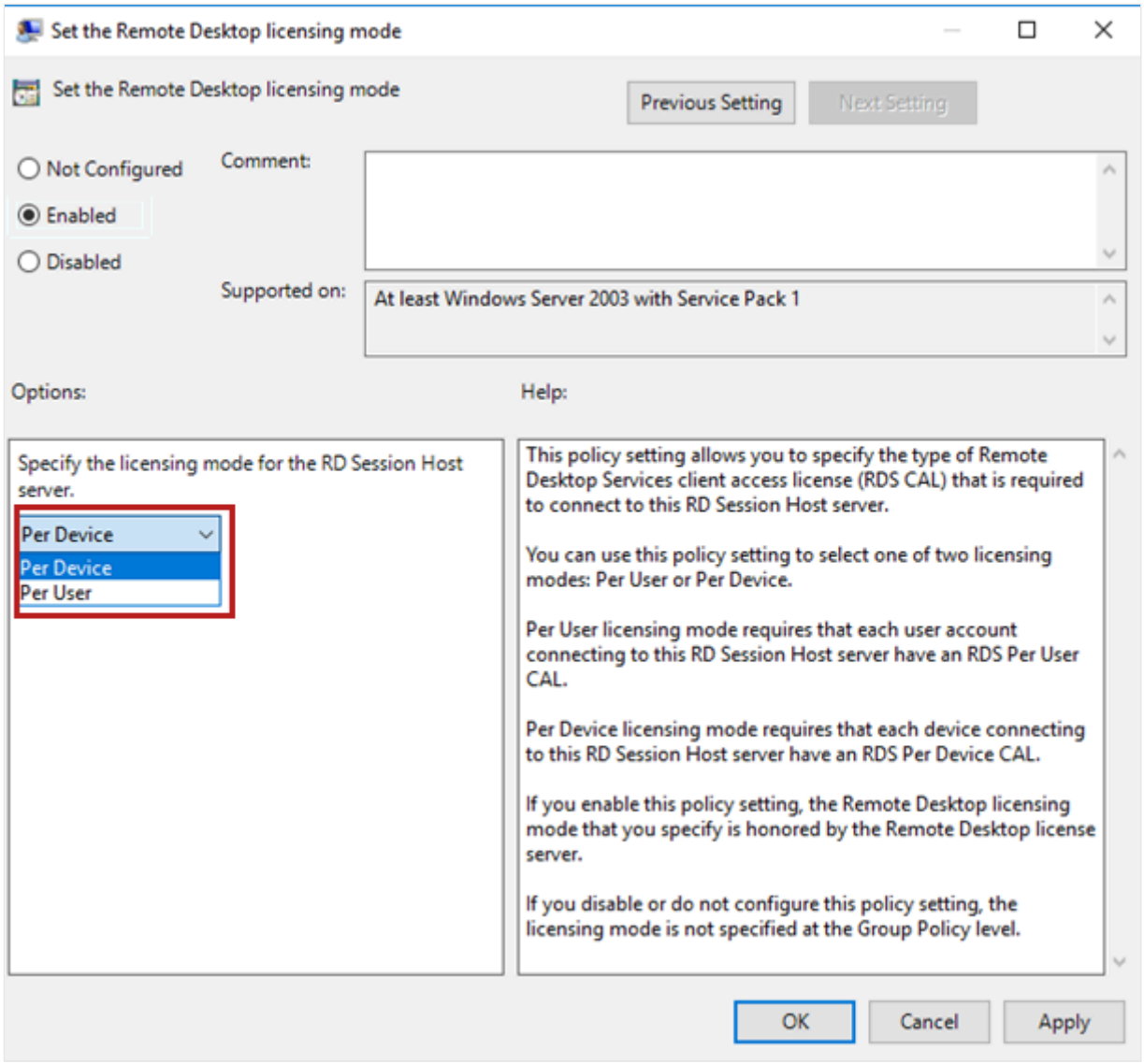


5. Select **OK**.

6. In the policy list, right-click **Set the Remote Desktop licensing mode**, and then select **Properties**.

7. Select **Enabled**.

8. Under **Specify the licensing mode for the Remote Desktop Session Host server**, select **Per Device** or **Per User**, as appropriate for your deployment.



Ensure an RD Session Host can access an RD licensing server in the same work group

This section only applies to work groups. Skip this section if your RD Session Host and RD licensing server are joined to a domain in Active Directory. You can also skip this section if the RD licensing server and RD Session Host server are the same machine.

After you apply the security update for CVE-2024-38099, RD licensing servers enforce that RD Session Host servers present nonanonymous credentials when requesting or querying licenses. To enforce nonanymous credentials exist, confirm that the *NT AUTHORITY\NETWORK SERVICE* account under which the Remote Desktop Service runs on the RD Session Host has access to credentials. Configure the machines in a work group using the following steps.

First, we recommend creating a dedicated user on the RD licensing server:

1. Connect to the RD licensing server. If doing so remotely, you might need to start the **Remote Desktop Connection** application using the `mstsc.exe /admin` command if the

target machine can't contact an RD licensing server.

2. Once connected, right-click **Start**, then select **Run**, and enter `lusrmgr.msc`. Then press ENTER.
3. Select **Users** in the left pane.
4. Open the **Action** menu and select **New User**.
5. Choose a username and a unique strong password for the user. Then confirm the password.
6. Uncheck the **User must change password at next logon** checkbox.
7. Select **Create**.

Then, on each RD Session Host server that needs to connect to the RD licensing server, add the user:

1. Connect to the RD Session Host machine. If doing so remotely, you might need to start the **Remote Desktop Connection** application if the target machine can't contact any RD licensing server. Open **Remote Desktop Connection** as an administrator, or use the command: `mstsc.exe /admin`.
2. Start a Command Prompt as *NT AUTHORITY\NETWORK SERVICE*. You can do this with PsExec from the Sysinternals Utilities, by running the following command as an administrator:

   Windows Command Prompt

   ```
   psexec.exe -I -u "NT AUTHORITY\NETWORK SERVICE" cmd.exe
   ```

3. Then, add the hostname or IP address of your licensing server, and a username and password to the licensing server with the following command:

   Windows Command Prompt

   ```
   cmdkey /add:<NAME-OF-THE-LICENSING-SERVER> /user:<NAME-OF-THE-LICENSING-SERVER>\<USERNAME> /pass
   ```

4. When prompted for the password, enter the password previously selected and press ENTER.

The RD Session Host should now be able to connect to the RD licensing server.

Alternatively, the requirement for proper authentication can be disabled on the licensing server. If you would like to disable the enforcement of authentication on your RD licensing server despite the risk, you can modify the registry.

> ⚠ **Warning**
>
> Disabling the enforcement of authentication on the RD licensing server isn't recommended and can result in increased security risks. Use it at your own risk.
>
> If you use Registry Editor incorrectly, you might cause serious problems that might require you to reinstall your operating system. Microsoft can't guarantee that you can solve problems that result from using Registry Editor incorrectly. Use Registry Editor at your own risk.

To update the registry key and value on the RD licensing server:

1. Select **Start**, type **Registry Editor**, then open it.
2. Navigate to and modify the key: `HKLM\SYSTEM\CurrentControlSet\Services\TermServLicensing\Parameters` with the following values:
   - **Name**: DisableWorkgroupAuthEnforcement
   - **Type**: REG_DWORD
   - **Data**: 1

> ⚠ **Warning**
>
> Future versions of Windows might stop honoring this setting.

Next steps

Learn how to create reports to track RDS per-user CALs issued by a Remote Desktop license server at Track your Remote Desktop Services client access licenses (RDS CALs).

Track your Remote Desktop Services client access licenses (RDS CALs)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use the Remote Desktop Licensing Manager tool to create reports to track the RDS Per User CALs that have been issued by a Remote Desktop license server.

> ⓘ **Note**
>
> If you're using Microsoft Entra Domain Services in your environment, the Remote Desktop Licensing Manager tool won't work to obtain Per User CALs. Instead, you need to track licensing manually, either through logon events, polling active Remote Desktop connections through the Connection Broker, or another mechanism that works for you.

Use the following steps to generate a per User CALs report:

1. In Remote Desktop Licensing Manager right-click the license server, click **Create Report**, and then click **CAL Usage**.

2. The report is created and a message appears to confirm that the report was successfully created. Click **OK** to close the message.

The report that you created appears in the Reports section under the node for the license server. The report provides the following information:

- Date and time the report was created
- The scope of the report (e.g., Domain, OU=Sales, or All trusted domains)
- The number of RDS Per User CALs that are installed on the license server
- The number of RDS Per User CALs that have been issued by the license server specific to the scope of the report

You can also save the report as a CSV file to a folder location on the computer. To save the report, right-click the report that you want to save, click Save As, and then specify the file name and location to save the report.

Reports that you create are listed in the Reports node under the node for the license server in Remote Desktop Licensing Manager. If you no longer need a report, you can delete it.

Feedback

Was this page helpful? Yes No

Use multiple Remote Desktop license servers

Article • 09/10/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When using multiple Remote Desktop (RD) license servers, after applying the security update for CVE-2024-38231, ensure that the servers can properly communicate with one another. It's important that RD license servers can communicate with one another in either of the following scenarios:

- A license is returned to an RD license server that didn't issue it
- Automatic license server discovery, a mechanism that was abandoned starting with Windows Server 2008 R2, is still in use in a Remote Desktop deployment

Workgroup-joined deployment

Workgroup-joined Remote Desktop deployments are meant for small deployments. We don't recommend using multiple RD license servers in workgroup-joined Remote Desktop deployments.

> **Important**
>
> Support for multiple license servers in workgroups may be removed in a future version of Windows.

To use multiple RD license servers in the same workgroup, ensure that each license server can authenticate to one another, and that they recognize each other as license servers.

Ensure license servers are authenticated

As an example, let's consider two license servers called LICSVR1 and LICSVR2.

To ensure that LICSVR1 can authenticate to LICSVR2, you need to decide which account LICSVR1 uses to connect to LICSVR2. We recommend creating a dedicated user account on LICSVR2 with the following steps:

1. Connect to LICSVR2 using an administrator account. If doing so remotely, you may need to start the **Remote Desktop Connection** application using the *mstsc.exe /admin* command if the target machine can't contact an RD license server.
2. Once connected, right-click **Start**, then select **Run**, and enter **lusrmgr.msc**. Then press **ENTER**.
3. Select **Users** in the left pane.
4. Open the **Action** menu and select **New User...**.
5. Choose a username and a unique strong password for the user. Then confirm the password.
6. Uncheck the "User must change password at next logon" checkbox.
7. Select **Create**.

Then, on LICSVR1, add the user and its credentials so that the *NT AUTHORITY\NETWORK SERVICE* account can authenticate to LICSVR2 with the following steps:

1. Connect to LICSVR1. If doing so remotely, you may need to start the Remote Desktop Connection application using the *mstsc.exe /admin* command if the target machine can't contact an RD license server.
2. Start a Command Prompt as *NT AUTHORITY\NETWORK SERVICE*. You can do this with PsExec from the Sysinternals Utilities, by running the following command as an administrator:

   ```
   Bash

   psexec.exe -i -u "NT AUTHORITY\NETWORK SERVICE" cmd.exe
   ```

3. Then, add a username and password to the host computer with the following command:

   ```
   Bash

   cmdkey /add:LICSVR2 /user:LICSVR2\<USERNAME> /pass
   ```

   where <USERNAME> is the name of the user you decided that LICSVR1 uses to authenticate to LICSVR2.
4. When prompted for the password, enter the password of that user.

LICSVR1 should now be able to authenticate to LICSVR2. For LICSVR2 to recognize LICSVR1 as another license server, you need to add the user to a local group on LICSV2 and register that local group with the RD licensing service. In PowerShell running as administrator on LICSVR2, use the following command:

```
PowerShell

New-LocalGroup -Name <GROUP-NAME>
Add-LocalGroupMember -Group <GROUP-NAME> -Member "LICSVR2\<USERNAME>"
```

Where <GROUP-NAME> is the desired name for the group and <USERNAME> is the name of the user whose credentials are registered in LICSVR1.

To register that local group with the RD licensing service in the registry, run the following PowerShell command:

```
PowerShell

Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\TermServLicensing\Parameters" -Name
" WorkgroupLicenseServerAccountsGroup" -Value "LICSVR2\<GROUP-NAME>" -Type
String
```

Domain-joined deployment

For domain-joined RD license servers to properly communicate with one another, they need to know that communication is coming from another RD license server. This can be achieved using one of the three manners described in this section.

- A domain administrator can publish each RD license server to Active Directory Domain Services (AD DS) using the PublishLS WMI method of the Win32_TSLicenseServer class. This creates a site-level record in AD DS that can be used to authorize communication between RD license servers. In PowerShell as a domain administrator on a license server, run the command:

  ```
  PowerShell

  Invoke-WmiMethod -Class Win32_TSLicenseServer -Name PublishLS
  ```

- Alternatively, each RD license server can be configured to authorize communication from a particular set of RD license servers by configuring the **Use the specified Remote Desktop license servers** group policy. That group policy is described in more detail in License Remote Desktop session hosts. Or the following

registry value can be set to specify license servers. In PowerShell as an administrator on an RD license server, run the command:

```
PowerShell

Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\TermServLicensing\Parameters"
-Name " SpecifiedLicenseServers" -Value "<LicSrv1DnsHostName>","
<LicSrv2DnsHostName>" -Type MultiString
```

Where <LicSrv1DnsHostName> and <LicSrv1DnsHostName> are the DNS host names of the other RD license servers.

- For historical reasons, RD licensing services that run on Active Directory domain controllers don't require extra configuration.

> **Important**
>
> We strongly advise against installing the RD licensing server on domain controllers. Use this approach at your own risk. Per the Active Directory security best practices, domain controllers should be treated as critical infrastructure components and should minimize the amount of unrelated software they run. For more information, se Protecting Domain Controllers.

Feedback

Was this page helpful? Yes No

Remote Desktop Services architecture

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) provides a flexible platform for hosting Windows applications and desktops in the cloud or on-premises. This article describes common RDS deployment architectures and shows how to integrate RDS with Azure services to meet your organization's needs.

Use these architecture diagrams to understand:

- How RDS roles work together in different deployment scenarios.
- Options for basic and highly available RDS configurations.
- Integration patterns with Azure Platform as a Service (PaaS) offerings.

Whether you're planning a new RDS deployment or modernizing an existing one, these architectures provide proven patterns to help you design a solution that meets your requirements for performance, availability, and security.

The architecture diagrams in this article show using RDS in Azure. However, as Remote Desktop Services is a role in Windows Server, you can deploy it on-premises and on other clouds. These diagrams are primarily intended to illustrate how the RDS roles are colocated and use other services.

Standard RDS deployment architectures

Remote Desktop Services has two standard architectures:

- **Basic deployment**: contains the minimum number of servers to create a fully effective RDS environment, but with no redundancy.
- **Highly available deployment**: contains all necessary components to have the highest guaranteed uptime for your RDS environment.

The following sections show the components of each architecture and how they work together. The diagrams also show how the RDS roles are colocated on the servers, which is a common practice to reduce costs and complexity.

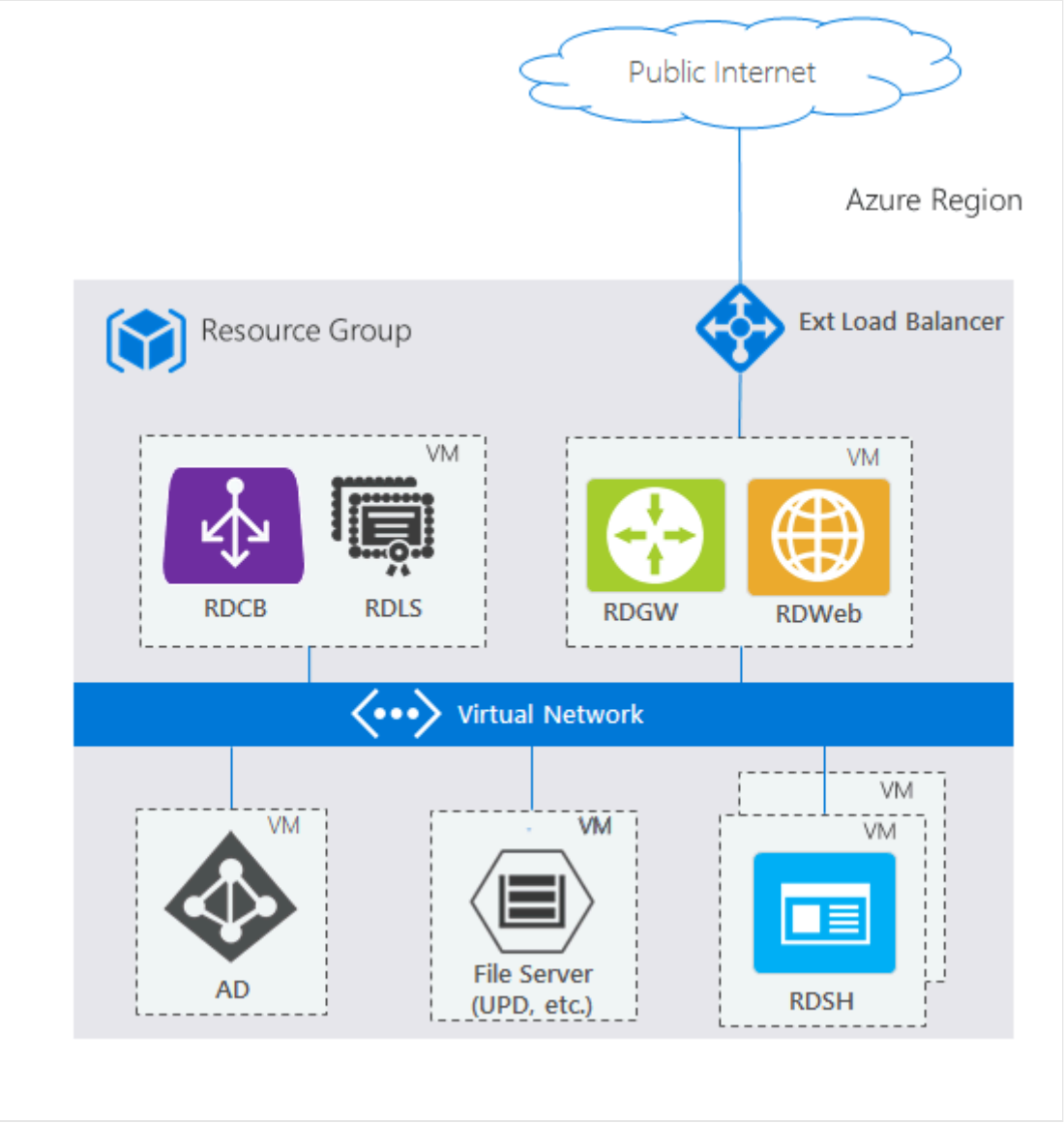
Basic deployment

This architecture illustrates a foundational Remote Desktop Services deployment in Azure that provides remote access to desktops and applications through a single Azure region. The

deployment uses an external load balancer to distribute incoming connections from the public internet across the RDS infrastructure.

The core RDS roles are distributed across multiple virtual machines within a single resource group. The RD Connection Broker (RDCB) and RD Licensing Server (RDLS) share one virtual machine, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate VM. Supporting infrastructure includes an Active Directory domain controller and a file server for user profile disks and shared storage. The RD Session Host (RDSH) server, deployed on its own virtual machine, provides the actual desktop sessions and hosted applications to end users.

All virtual machines communicate through an Azure Virtual Network, which provides secure network connectivity between the RDS components while isolating the deployment from other Azure resources. This architecture provides a cost-effective starting point for organizations looking to migrate their desktop hosting to Azure, with the flexibility to scale individual components as usage grows.

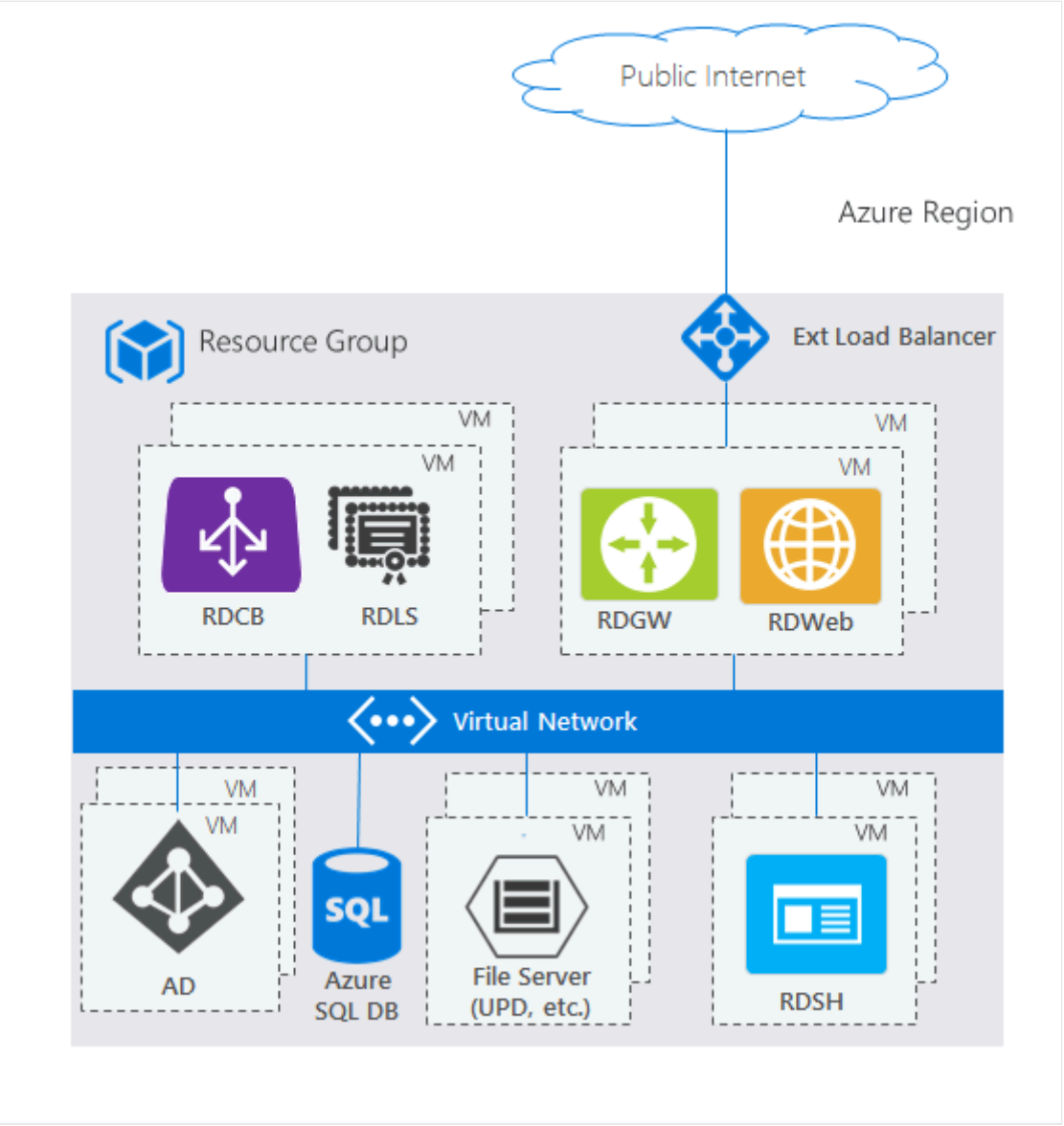


Highly available deployment

This architecture demonstrates a Remote Desktop Services deployment that integrates Azure Platform as a Service (PaaS) offerings to improve scalability and reduce management overhead. The key difference from a basic RDS deployment is the replacement of a traditional SQL Server virtual machine with Azure SQL Database for storing RDS configuration and user session data.

The RDS roles maintain the same distribution pattern, but with multiple instances of each; the RD Connection Broker (RDCB) and RD Licensing Server (RDLS) sharing one set of virtual machines, while the RD Gateway (RDGW) and RD Web Access (RDWeb) components are deployed on a separate set of VMs. The RD Session Hosts (RDSH) continue to provide desktop sessions and applications from their dedicated virtual machines. Supporting infrastructure includes an Active Directory domain controller and a file server for user profiles and shared storage.

By using Azure SQL Database instead of a self-managed SQL Server instance, this architecture provides built-in high availability, automatic backups, and simplified database management. The Azure SQL Database handles the RDS Connection Broker database requirements while eliminating the need to maintain, patch, and monitor a separate database server. This hybrid approach combines the flexibility of Infrastructure as a Service (IaaS) for the RDS roles with the managed benefits of PaaS for the database tier, resulting in reduced operational complexity and improved reliability.

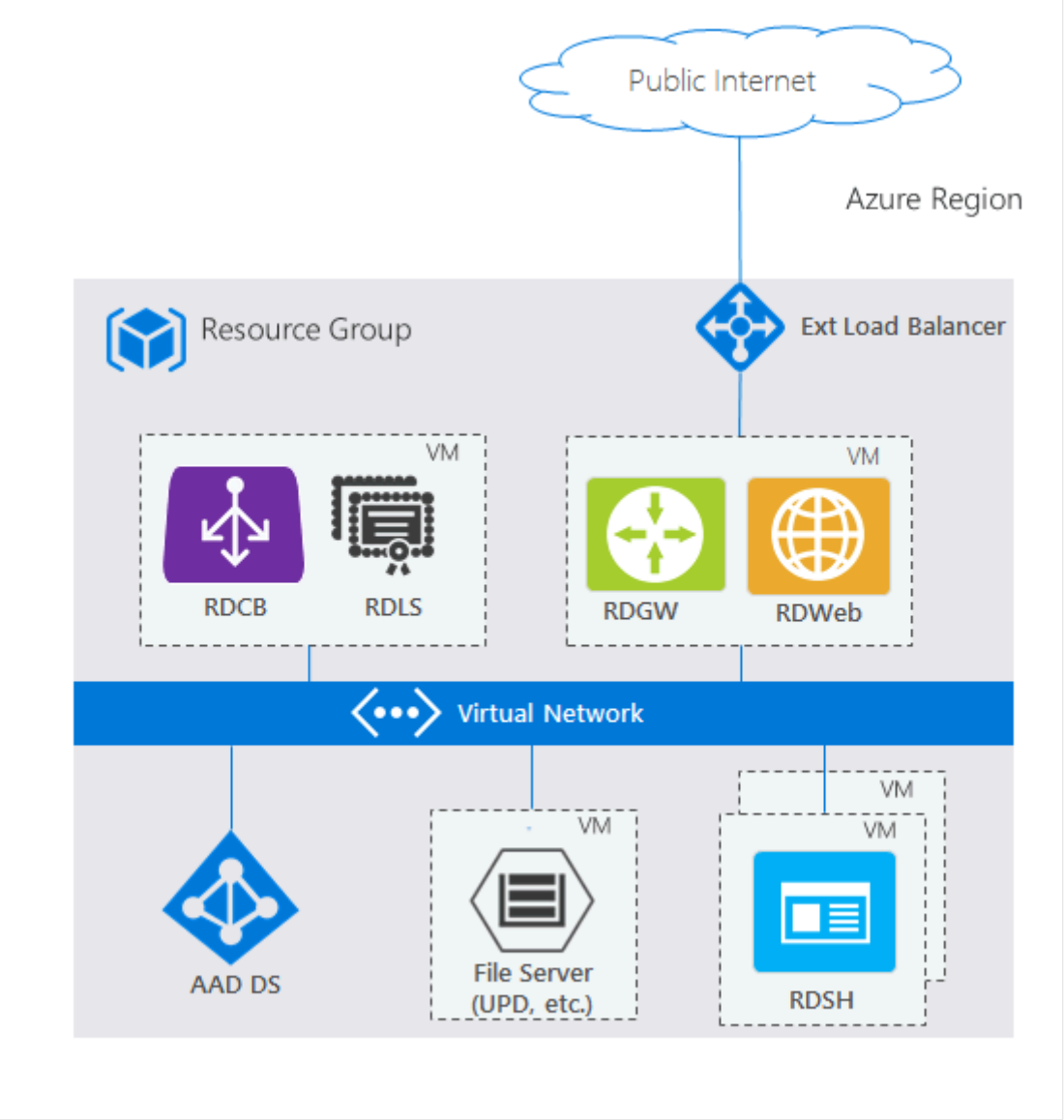


RDS architectures with unique Azure PaaS roles

Though the standard RDS deployment architectures fit most scenarios, Azure continues to invest in first-party PaaS solutions that drive customer value. The following architectures show how they incorporate with RDS.

RDS deployment with Microsoft Entra Domain Services

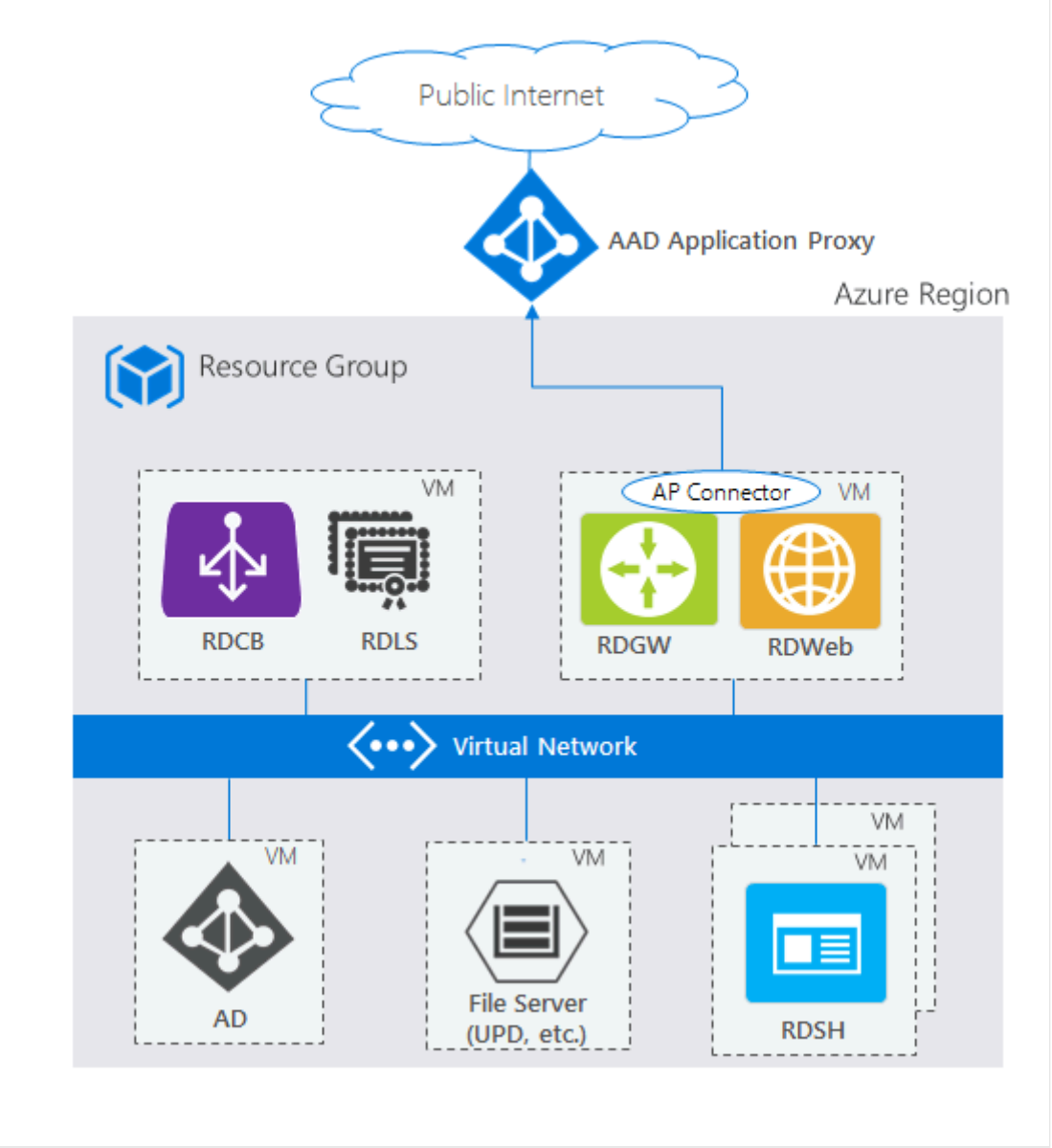
The two standard architecture diagrams are based on a traditional Active Directory (AD) deployed on a Windows Server VM. However, if you don't have a traditional AD and only have a Microsoft Entra tenant, for example through services like Microsoft 365, but still want to use RDS, you can use Microsoft Entra Domain Services to create a fully managed domain in your Azure IaaS environment that uses the same users that exist in your Microsoft Entra tenant. This option removes the complexity of manually syncing users and managing more virtual machines. Microsoft Entra Domain Services can work in either deployment: basic or highly available.



RDS deployment with Microsoft Entra application proxy

The two standard architecture diagrams use the RD Web/Gateway servers as the Internet-facing entry point into the RDS system. For some environments, administrators would prefer to remove their own servers from the perimeter and instead use technologies that also provide extra security through reverse proxy technologies. The Microsoft Entra application proxy PaaS role fits nicely with this scenario.

For supported configurations and how to create this setup, see Publish Remote Desktop with Microsoft Entra application proxy.



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

---

Last updated on 07/07/2025

Integrate your Remote Desktop Gateway infrastructure using the Network Policy Server (NPS) extension and Microsoft Entra ID

Summarize this article for me

This article provides details for integrating your Remote Desktop Gateway infrastructure with Microsoft Entra multifactor authentication using the Network Policy Server (NPS) extension for Microsoft Azure.

The Network Policy Server (NPS) extension for Azure allows customers to safeguard Remote Authentication Dial-In User Service (RADIUS) client authentication using Azure's cloud-based multifactor authentication. This solution provides two-step verification for adding a second layer of security to user sign-ins and transactions.

This article provides step-by-step instructions for integrating the NPS infrastructure with Microsoft Entra multifactor authentication using the NPS extension for Azure. This enables secure verification for users attempting to sign in to a Remote Desktop Gateway.

> **Note**
>
> This article shouldn't be used with MFA Server deployments and should only be used with Microsoft Entra multifactor authentication (Cloud-based) deployments.

The Network Policy and Access Services (NPS) gives organizations the ability to do the following:

- Define central locations for the management and control of network requests by specifying who can connect, what times of day connections are allowed, the duration of connections, and the level of security that clients must use to connect, and so on. Rather than specifying these policies on each VPN or Remote Desktop (RD) Gateway server, these policies can be specified once in a central location. The RADIUS protocol provides the centralized Authentication, Authorization, and Accounting (AAA).
- Establish and enforce Network Access Protection (NAP) client health policies that determine whether devices are granted unrestricted or restricted access to network resources.
- Provide a means to enforce authentication and authorization for access to 802.1x-capable wireless access points and Ethernet switches.

Typically, organizations use NPS (RADIUS) to simplify and centralize the management of VPN policies. However, many organizations also use NPS to simplify and centralize the management of RD Desktop Connection Authorization Policies (RD CAPs).

Organizations can also integrate NPS with Microsoft Entra multifactor authentication to enhance security and provide a high level of compliance. This helps ensure that users establish two-step verification to sign in to the Remote Desktop Gateway. For users to be granted access, they must provide their username/password combination along with information that the user has in their control. This information must be trusted and not easily duplicated, such as a cell phone number, landline number, application on a mobile device, and so on. RDG currently supports phone call and **Approve/Deny** push notifications from Microsoft authenticator app methods for 2FA. For more information about supported authentication methods, see the section Determine which authentication methods your users can use.

If your organization uses Remote Desktop Gateway and the user is registered for a TOTP code along with Authenticator push notifications, the user can't meet the MFA challenge and the Remote Desktop Gateway sign-in fails. In that case, you can override this behaviour by creating a new registry key (**OVERRIDE_NUMBER_MATCHING_WITH_OTP**) to fallback to push notifications to Approve/Deny with Authenticator. To perform it, follow NPS extension override number matching procedure, assuming final value will be *OVERRIDE_NUMBER_MATCHING_WITH_OTP = FALSE*.

Prior to the availability of the NPS extension for Azure, customers who wished to implement two-step verification for integrated NPS and Microsoft Entra multifactor authentication environments had to configure and maintain a separate MFA Server in the on-premises environment as documented in Remote Desktop Gateway and Azure Multi-Factor Authentication Server using RADIUS.

The availability of the NPS extension for Azure now gives organizations the choice to deploy either an on-premises based MFA solution or a cloud-based MFA solution to secure RADIUS client authentication.

Authentication Flow

For users to be granted access to network resources through a Remote Desktop Gateway, they must meet the conditions specified in one RD Connection Authorization Policy (RD CAP) and one RD Resource Authorization Policy (RD RAP). RD CAPs specify who is authorized to connect to RD Gateways. RD RAPs specify the network resources, such as remote desktops or remote apps, that the user is allowed to connect to through the RD Gateway.

An RD Gateway can be configured to use a central policy store for RD CAPs. RD RAPs can't use a central policy, as they're processed on the RD Gateway. An example of an RD Gateway

configured to use a central policy store for RD CAPs is a RADIUS client to another NPS server that serves as the central policy store.

When the NPS extension for Azure is integrated with the NPS and Remote Desktop Gateway, the successful authentication flow is as follows:

1. The Remote Desktop Gateway server receives an authentication request from a remote desktop user to connect to a resource, such as a Remote Desktop session. Acting as a RADIUS client, the Remote Desktop Gateway server converts the request to a RADIUS Access-Request message and sends the message to the RADIUS (NPS) server where the NPS extension is installed.
2. The username and password combination is verified in Active Directory and the user is authenticated.
3. If all the conditions as specified in the NPS Connection Request and the Network Policies are met (for example, time of day or group membership restrictions), the NPS extension triggers a request for secondary authentication with Microsoft Entra multifactor authentication.
4. Microsoft Entra multifactor authentication communicates with Microsoft Entra ID, retrieves the user's details, and performs the secondary authentication using supported methods.
5. Upon success of the MFA challenge, Microsoft Entra multifactor authentication communicates the result to the NPS extension.
6. The NPS server, where the extension is installed, sends a RADIUS Access-Accept message for the RD CAP policy to the Remote Desktop Gateway server.
7. The user is granted access to the requested network resource through the RD Gateway.

Prerequisites

This section details the prerequisites necessary before integrating Microsoft Entra multifactor authentication with the Remote Desktop Gateway. Before you begin, you must have the following prerequisites in place.

- Remote Desktop Services (RDS) infrastructure
- Microsoft Entra multifactor authentication License
- Windows Server software
- Network Policy and Access Services (NPS) role
- Microsoft Entra synced with on-premises Active Directory
- Microsoft Entra GUID ID

Remote Desktop Services (RDS) infrastructure

You must have a working Remote Desktop Services (RDS) infrastructure in place. If you don't, then you can quickly create this infrastructure in Azure using the following quickstart template: Create Remote Desktop Session Collection deployment.

If you wish to manually create an on-premises RDS infrastructure quickly for testing purposes, follow the steps to deploy one. **Learn more**: Deploy RDS with Azure quickstart and Basic RDS infrastructure deployment.

Windows Server software

The NPS extension requires Windows Server 2008 R2 SP1 or above with the NPS role service installed. All the steps in this section were performed using Windows Server 2016.

Network Policy and Access Services (NPS) role

The NPS role service provides the RADIUS server and client functionality and Network Access Policy health service. This role must be installed on at least two computers in your infrastructure: The Remote Desktop Gateway and another member server or domain controller. By default, the role is already present on the computer configured as the Remote Desktop Gateway. You must also install the NPS role on at least on another computer, such as a domain controller or member server.

For information on installing the NPS role service Windows Server 2012 or older, see Install a NAP Health Policy Server. For a description of best practices for NPS, including the recommendation to install NPS on a domain controller, see Best Practices for NPS.

Microsoft Entra synced with on-premises Active Directory

To use the NPS extension, on-premises users must be synced with Microsoft Entra ID and enabled for MFA. This section assumes that on-premises users are synced with Microsoft Entra ID using AD Connect. For information on Microsoft Entra Connect, see Integrate your on-premises directories with Microsoft Entra ID.

Microsoft Entra GUID ID

To install NPS extension, you need to know the GUID of the Microsoft Entra ID. The following provides instructions for finding the GUID of the Microsoft Entra ID.

Configure multifactor authentication

This section provides instructions for integrating Microsoft Entra multifactor authentication with the Remote Desktop Gateway. As an administrator, you must configure the Microsoft Entra multifactor authentication service before users can self-register their multifactor devices or applications.

Follow the steps in Getting started with Microsoft Entra multifactor authentication in the cloud to enable MFA for your Microsoft Entra users.

Configure accounts for two-step verification

Once an account has been enabled for MFA, you can't sign in to resources governed by the MFA policy until you have successfully configured a trusted device to use for the second authentication factor and have authenticated using two-step verification.

Follow the steps in What does Microsoft Entra multifactor authentication mean for me? to understand and properly configure your devices for MFA with your user account.

> **Important**
>
> The sign-in behavior for Remote Desktop Gateway doesn't provide the option to enter a verification code with Microsoft Entra multifactor authentication. A user account must be configured for phone verification or the Microsoft Authenticator App with **Approve/Deny** push notifications.
>
> If neither phone verification or the Microsoft Authenticator App with **Approve/Deny** push notifications is configured for a user, the user won't be able to complete the Microsoft Entra multifactor authentication challenge and sign in to Remote Desktop Gateway.
>
> The SMS text method doesn't work with Remote Desktop Gateway because it doesn't provide the option to enter a verification code.

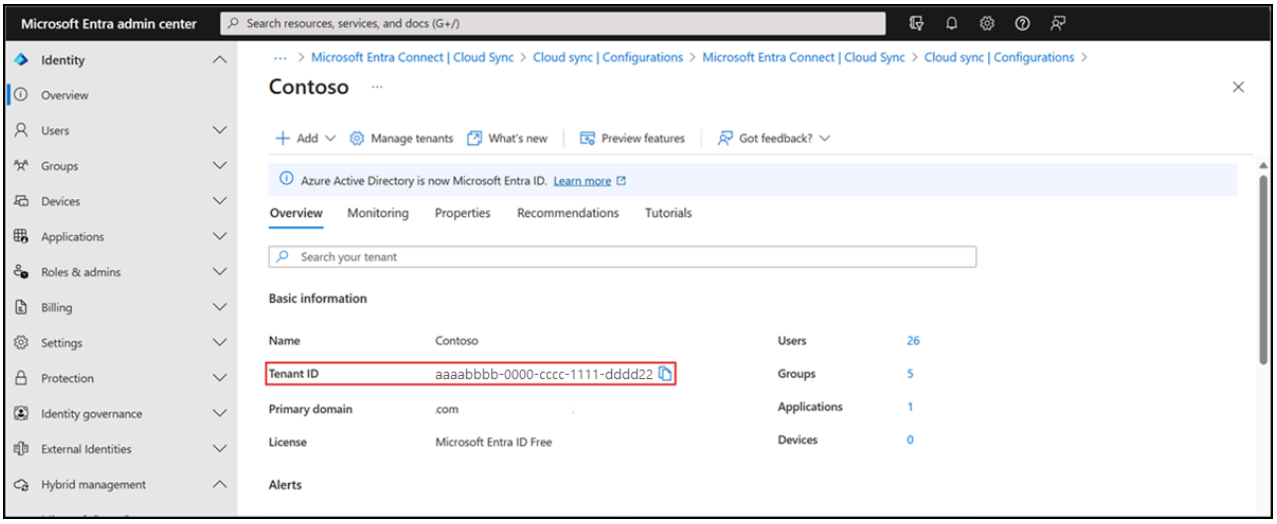
Install and configure NPS extension

This section provides instructions for configuring RDS infrastructure to use Microsoft Entra multifactor authentication for client authentication with the Remote Desktop Gateway.

Obtain the directory tenant ID

As part of the configuration of the NPS extension, you must supply administrator credentials and the ID of your Microsoft Entra tenant. To get the tenant ID, complete the following steps:

1. Sign in to the Microsoft Entra admin center.
2. Browse to **Entra ID** > **Overview** > **Properties**.



Install the NPS extension

Install the NPS extension on a server that has the Network Policy and Access Services (NPS) role installed. This functions as the RADIUS server for your design.

> **Important**
>
> Don't install the NPS extension on your Remote Desktop Gateway (RDG) server. The RDG server doesn't use the RADIUS protocol with its client, so the extension can't interpret and perform the MFA.
>
> When the RDG server and NPS server with NPS extension are different servers, RDG uses NPS internally to talk to other NPS servers and uses RADIUS as the protocol to correctly communicate.

1. Download the NPS extension.
2. Copy the setup executable file (NpsExtnForAzureMfaInstaller.exe) to the NPS server.
3. On the NPS server, double-select **NpsExtnForAzureMfaInstaller.exe**. If prompted, select **Run**.
4. In the NPS Extension For Microsoft Entra multifactor authentication Setup dialog box, review the software license terms, check **I agree to the license terms and conditions**, and select **Install**.
5. In the NPS Extension For Microsoft Entra multifactor authentication Setup dialog box, select **Close**.

Configure certificates for use with the NPS extension using a PowerShell script

Next, you need to configure certificates for use by the NPS extension to ensure secure communications and assurance. The NPS components include a PowerShell script that configures a self-signed certificate for use with NPS.

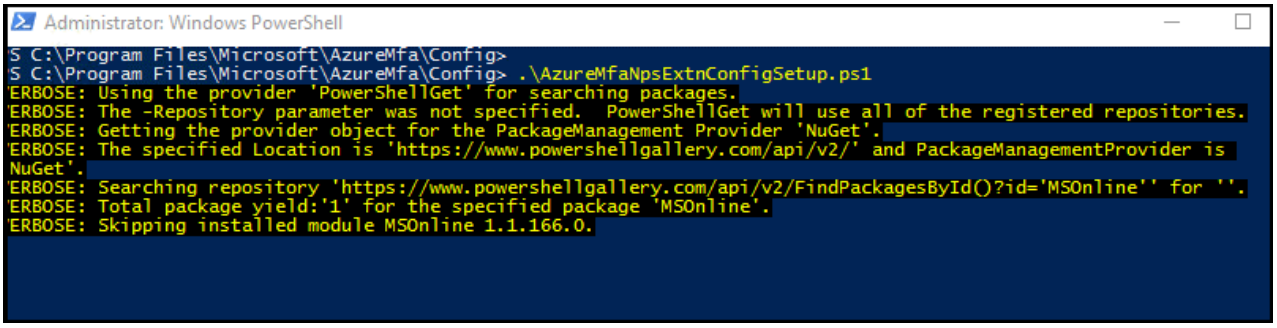
The script performs the following actions:

- Creates a self-signed certificate
- Associates public key of certificate to service principal on Microsoft Entra ID
- Stores the cert in the local machine store
- Grants access to the certificate's private key to the network user
- Restarts Network Policy Server service

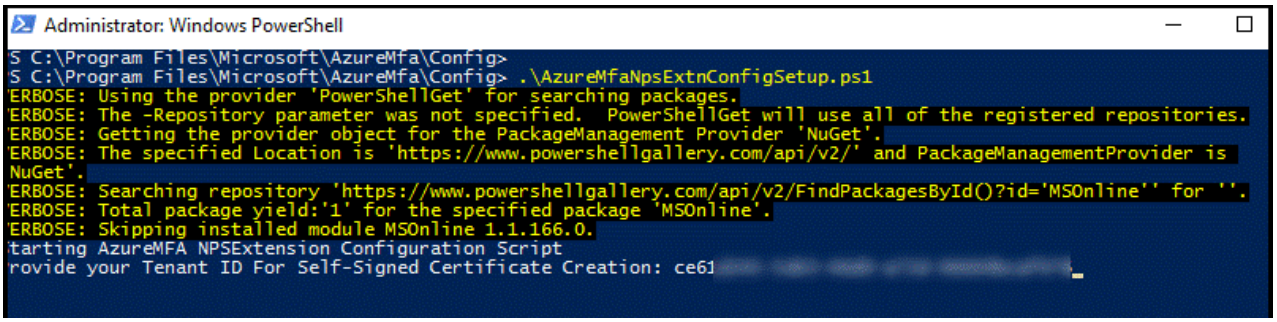
If you want to use your own certificates, you need to associate the public key of your certificate to the service principal on Microsoft Entra ID, and so on.

To use the script, provide the extension with your Microsoft Entra Admin credentials and the Microsoft Entra tenant ID that you copied earlier. Run the script on each NPS server where you installed the NPS extension. Then do the following:

1. Open an administrative Windows PowerShell prompt.
2. At the PowerShell prompt, type `cd 'c:\Program Files\Microsoft\AzureMfa\Config'`, and press **ENTER**.
3. Type `.\AzureMfaNpsExtnConfigSetup.ps1`, and press **ENTER**. The script checks to see if the PowerShell module is installed. If not installed, the script installs the module for you.



4. After the script verifies the installation of the PowerShell module, it displays the PowerShell module dialog box. In the dialog box, enter your Microsoft Entra admin credentials and password, and select **Sign In**.
5. When prompted, paste the *Tenant ID* you copied to the clipboard earlier, and press **ENTER**.



6. The script creates a self-signed certificate and performs other configuration changes.

Configure NPS components on Remote Desktop Gateway

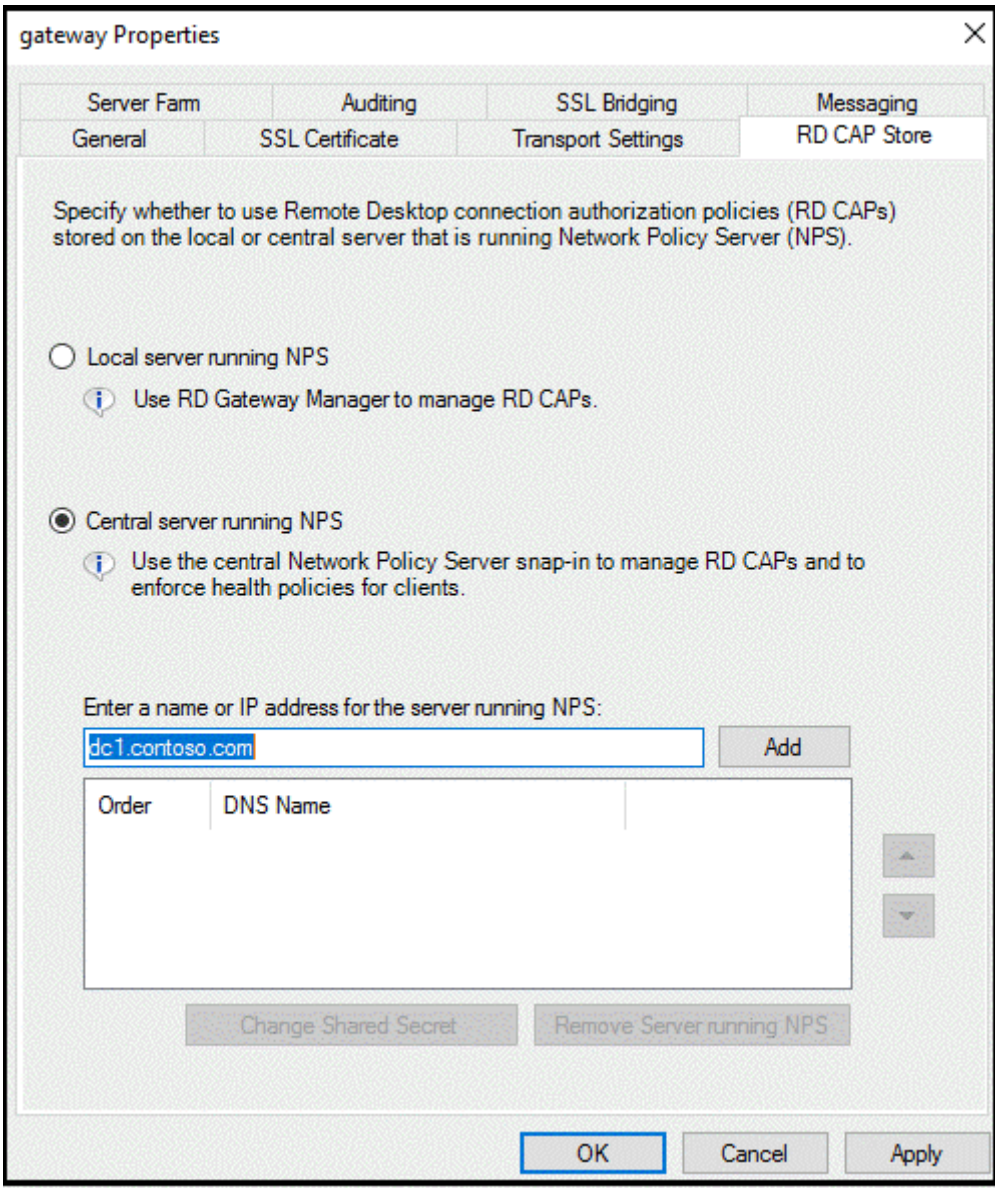
In this section, you configure the Remote Desktop Gateway connection authorization policies and other RADIUS settings.

The authentication flow requires that RADIUS messages be exchanged between the Remote Desktop Gateway and the NPS server where the NPS extension is installed. This means that you must configure RADIUS client settings on both Remote Desktop Gateway and the NPS server where the NPS extension is installed.

Configure Remote Desktop Gateway connection authorization policies to use central store

Remote Desktop connection authorization policies (RD CAPs) specify the requirements for connecting to a Remote Desktop Gateway server. RD CAPs can be stored locally (default) or they can be stored in a central RD CAP store that is running NPS. To configure integration of Microsoft Entra multifactor authentication with RDS, you need to specify the use of a central store.

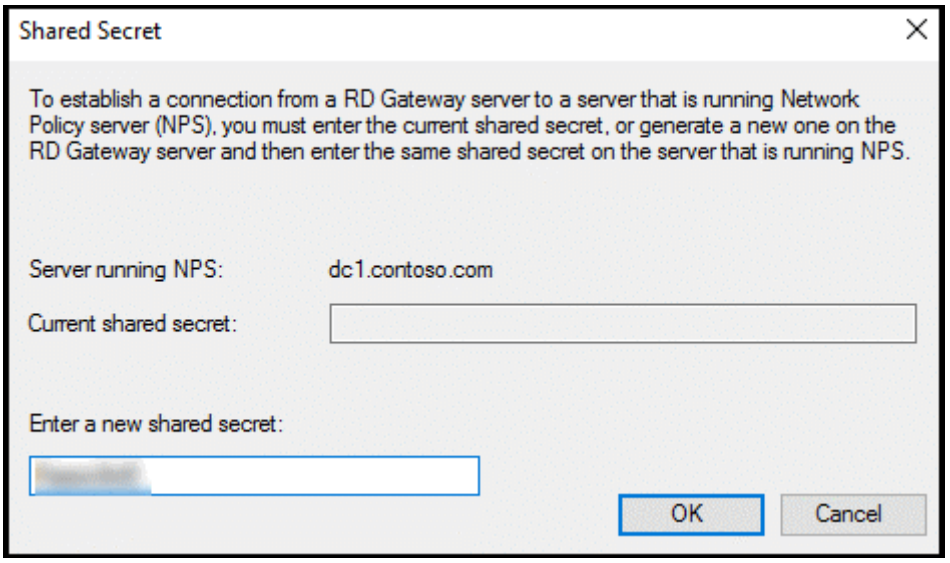
1. On the RD Gateway server, open **Server Manager**.
2. On the menu, select **Tools**, point to **Remote Desktop Services**, and then select **Remote Desktop Gateway Manager**.
3. In the RD Gateway Manager, right-select **[Server Name] (Local)**, and select **Properties**.
4. In the Properties dialog box, select the **RD CAP Store** tab.
5. On the RD CAP Store tab, select **Central server running NPS**.
6. In the **Enter a name or IP address for the server running NPS** field, type the IP address or server name of the server where you installed the NPS extension.



7. Select **Add**.

8. In the **Shared Secret** dialog box, enter a shared secret, and then select **OK**. Ensure you record this shared secret and store the record securely.

> **Note**
>
> Shared secret is used to establish trust between the RADIUS servers and clients. Create a long and complex secret.



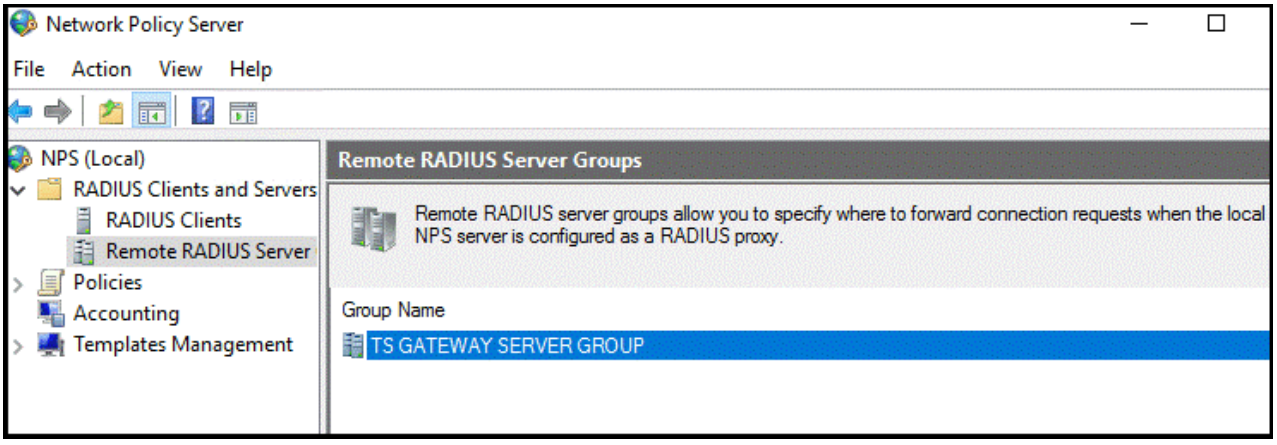
9. Select **OK** to close the dialog box.

Configure RADIUS timeout value on Remote Desktop Gateway NPS

To ensure there is time to validate users' credentials, perform two-step verification, receive responses, and respond to RADIUS messages, it's necessary to adjust the RADIUS timeout value.

1. On the RD Gateway server, open Server Manager. On the menu, select **Tools**, and then select **Network Policy Server**.

2. In the **NPS (Local)** console, expand **RADIUS Clients and Servers**, and select **Remote RADIUS Server**.

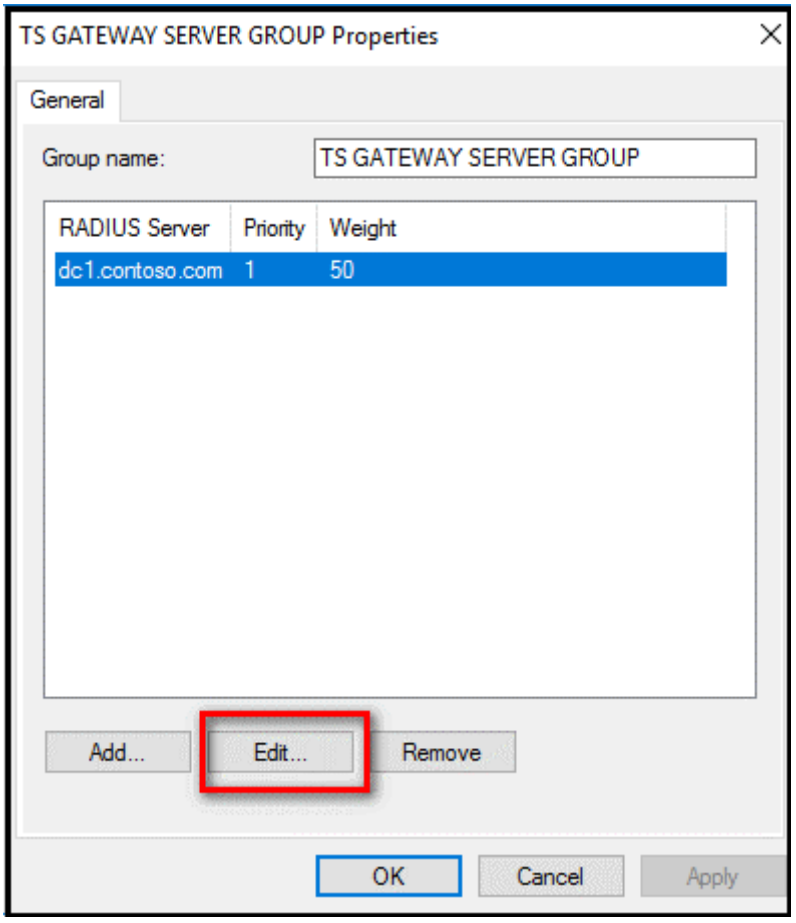


3. In the details pane, double-select **TS GATEWAY SERVER GROUP**.

> ⓘ Note

This RADIUS Server Group was created when you configured the central server for NPS policies. The RD Gateway forwards RADIUS messages to this server or group of servers, if more than one in the group.

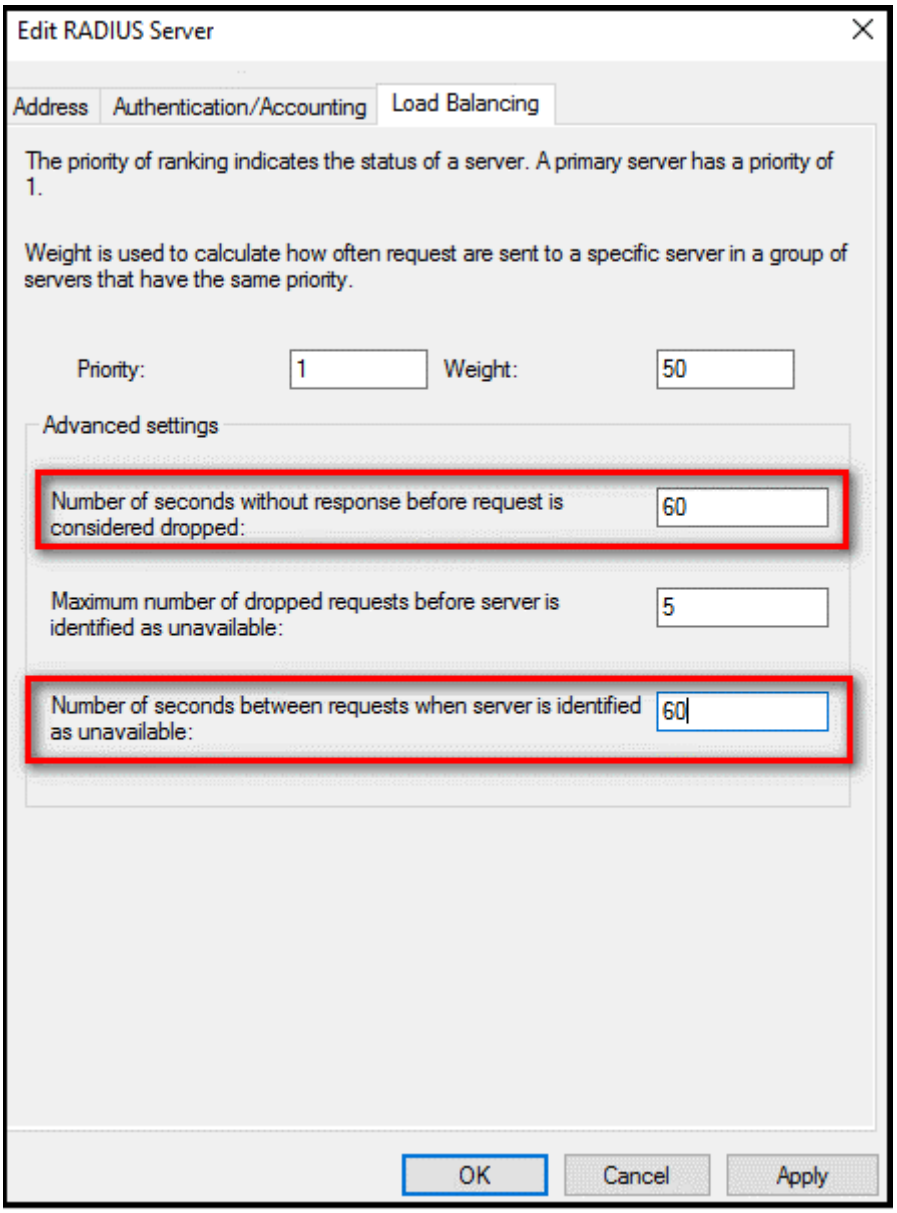
4. In the **TS GATEWAY SERVER GROUP Properties** dialog box, select the IP address or name of the NPS server you configured to store RD CAPs, and then select **Edit**.



5. In the **Edit RADIUS Server** dialog box, select the **Load Balancing** tab.

6. In the **Load Balancing** tab, in the **Number of seconds without response before request is considered dropped** field, change the default value from 3 to a value between 30 and 60 seconds.

7. In the **Number of seconds between requests when server is identified as unavailable** field, change the default value of 30 seconds to a value that is equal to or greater than the value you specified in the previous step.



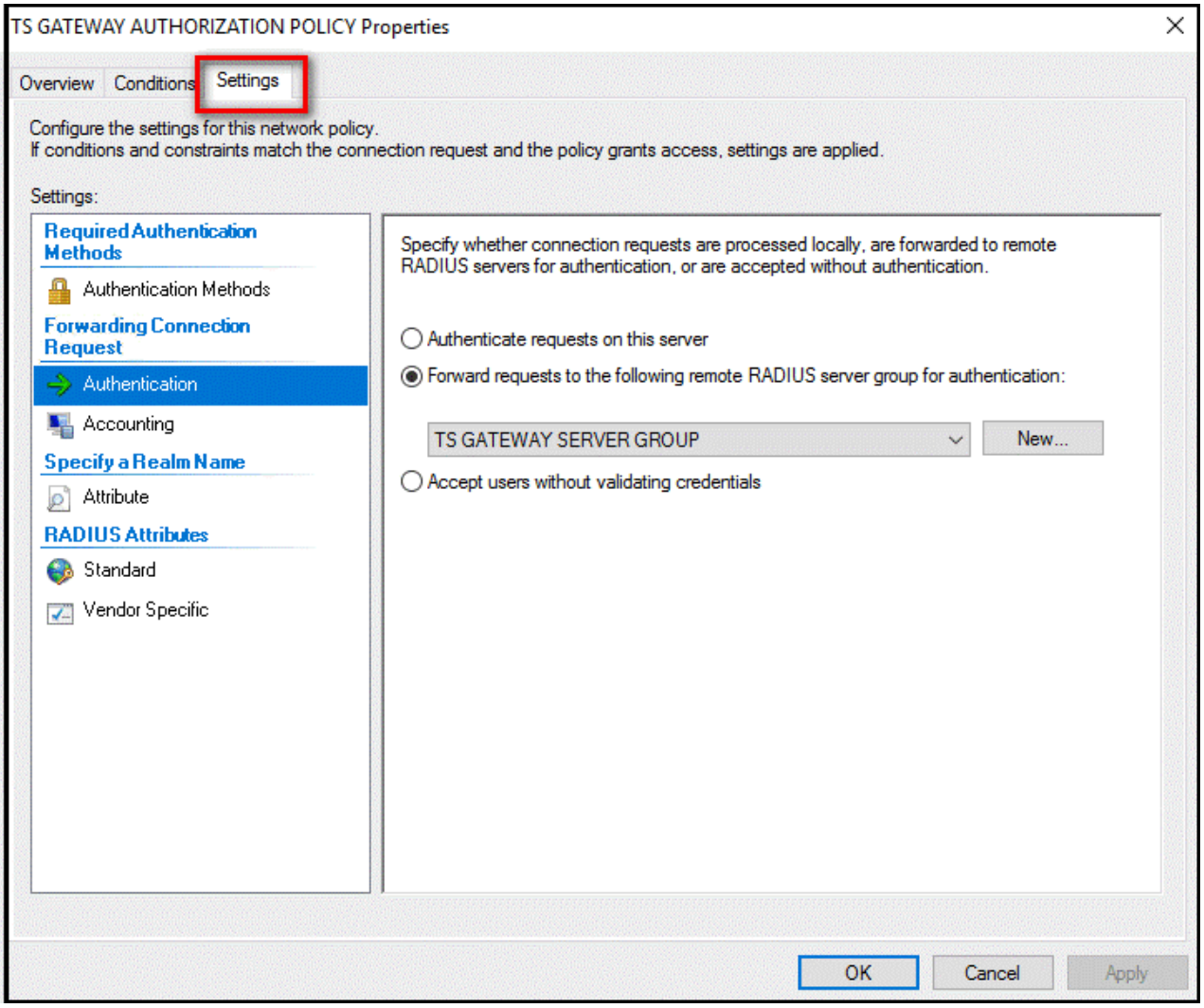
8. Select **OK** two times to close the dialog boxes.

Verify Connection Request Policies

By default, when you configure the RD Gateway to use a central policy store for connection authorization policies, the RD Gateway is configured to forward CAP requests to the NPS server. The NPS server with the Microsoft Entra multifactor authentication extension installed, processes the RADIUS access request. The following steps show you how to verify the default connection request policy.

1. On the RD Gateway, in the NPS (Local) console, expand **Policies**, and select **Connection Request Policies**.
2. Double-select **TS GATEWAY AUTHORIZATION POLICY**.
3. In the **TS GATEWAY AUTHORIZATION POLICY properties** dialog box, select the **Settings** tab.

4. On **Settings** tab, under Forwarding Connection Request, select **Authentication**. RADIUS client is configured to forward requests for authentication.



5. Select **Cancel**.

> ⓘ **Note**
>
> For more information about creating a connection request policy, see the article Configure connection request policies documentation for the same.

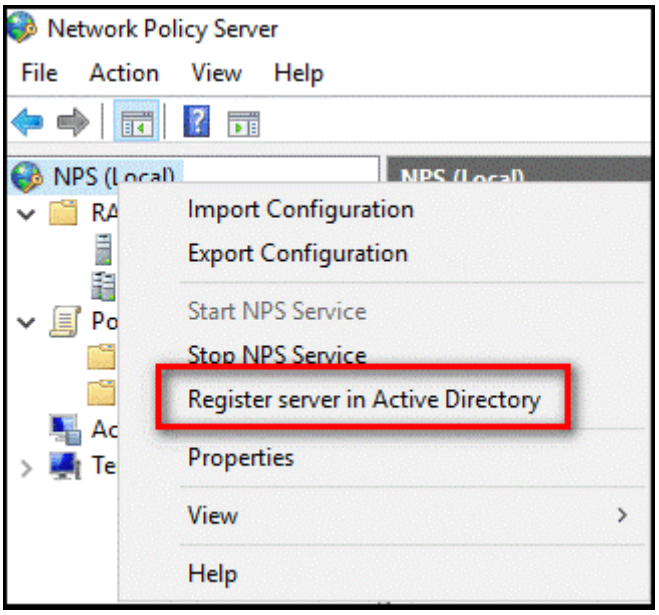
Configure NPS on the server where the NPS extension is installed

The NPS server where the NPS extension is installed needs to be able to exchange RADIUS messages with the NPS server on the Remote Desktop Gateway. To enable this message exchange, you need to configure the NPS components on the server where the NPS extension service is installed.

Register Server in Active Directory

To function properly in this scenario, the NPS server needs to be registered in Active Directory.

1. On the NPS server, open **Server Manager**.
2. In Server Manager, select **Tools**, and then select **Network Policy Server**.
3. In the Network Policy Server console, right-select **NPS (Local)**, and then select **Register server in Active Directory**.
4. Select **OK** two times.

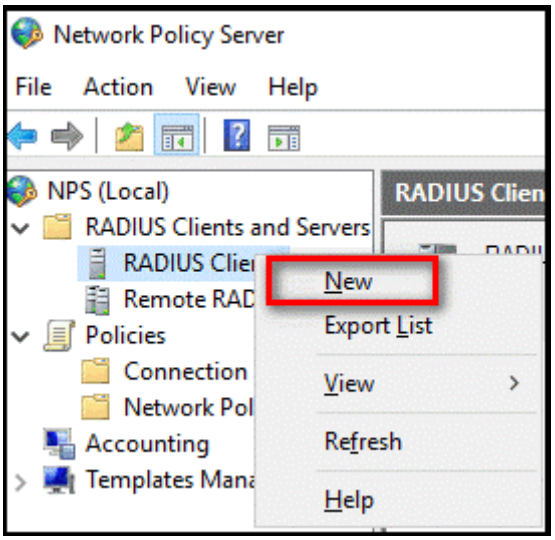


5. Leave the console open for the next procedure.

Create and configure RADIUS client

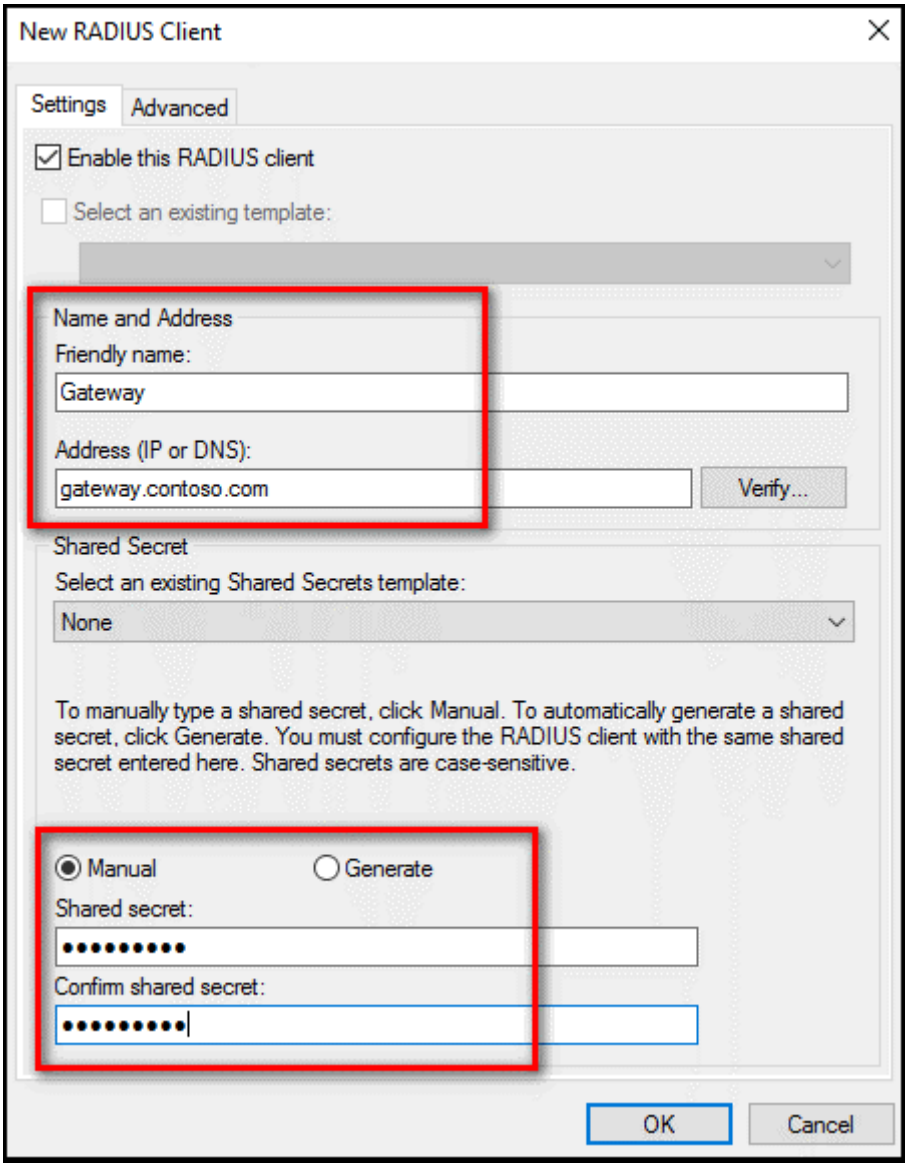
The Remote Desktop Gateway needs to be configured as a RADIUS client to the NPS server.

1. On the NPS server where the NPS extension is installed, in the **NPS (Local)** console, right-select **RADIUS Clients** and select **New**.



2. In the **New RADIUS Client** dialog box, provide a friendly name, such as *Gateway*, and the IP address or DNS name of the Remote Desktop Gateway server.

3. In the **Shared secret** and the **Confirm shared secret** fields, enter the same secret that you used before.



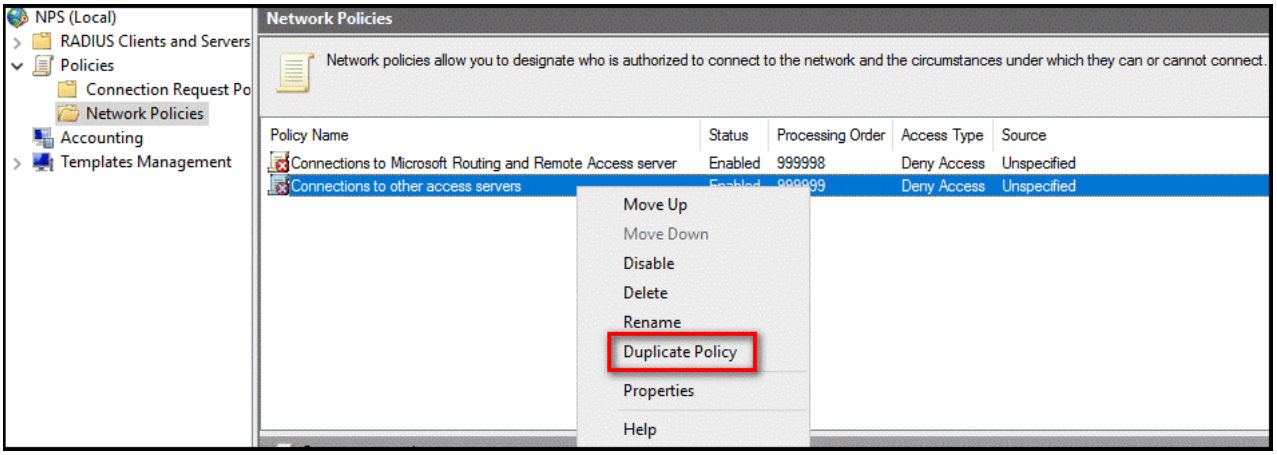
4. Select **OK** to close the New RADIUS Client dialog box.

Configure Network Policy

Recall that the NPS server with the Microsoft Entra multifactor authentication extension is the designated central policy store for the Connection Authorization Policy (CAP). Therefore, you need to implement a CAP on the NPS server to authorize valid connections requests.

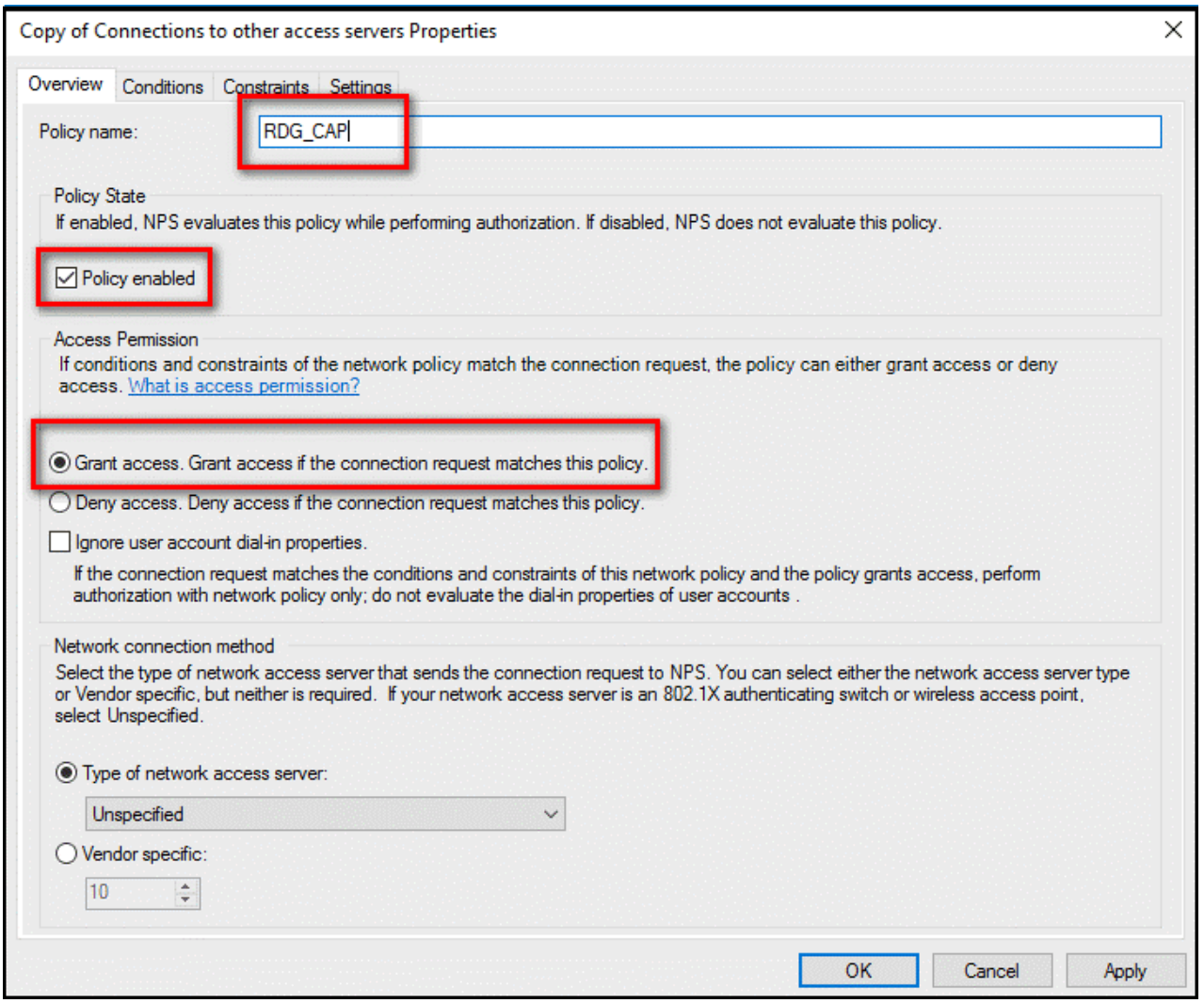
1. On the NPS Server, open the NPS (Local) console, expand **Policies**, and select **Network Policies**.

2. Right-select **Connections to other access servers**, and select **Duplicate Policy**.

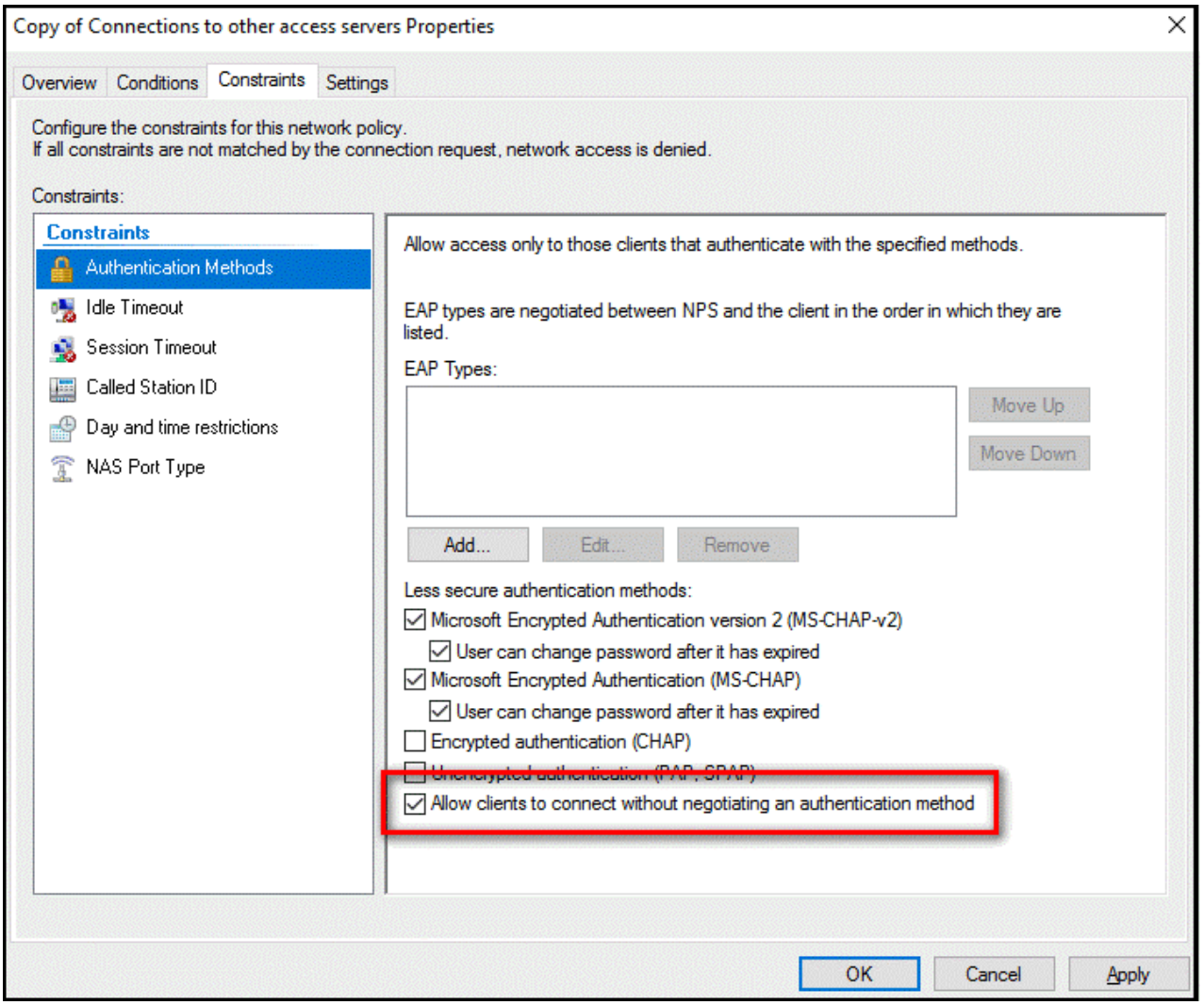


3. Right-select **Copy of Connections to other access servers**, and select **Properties**.

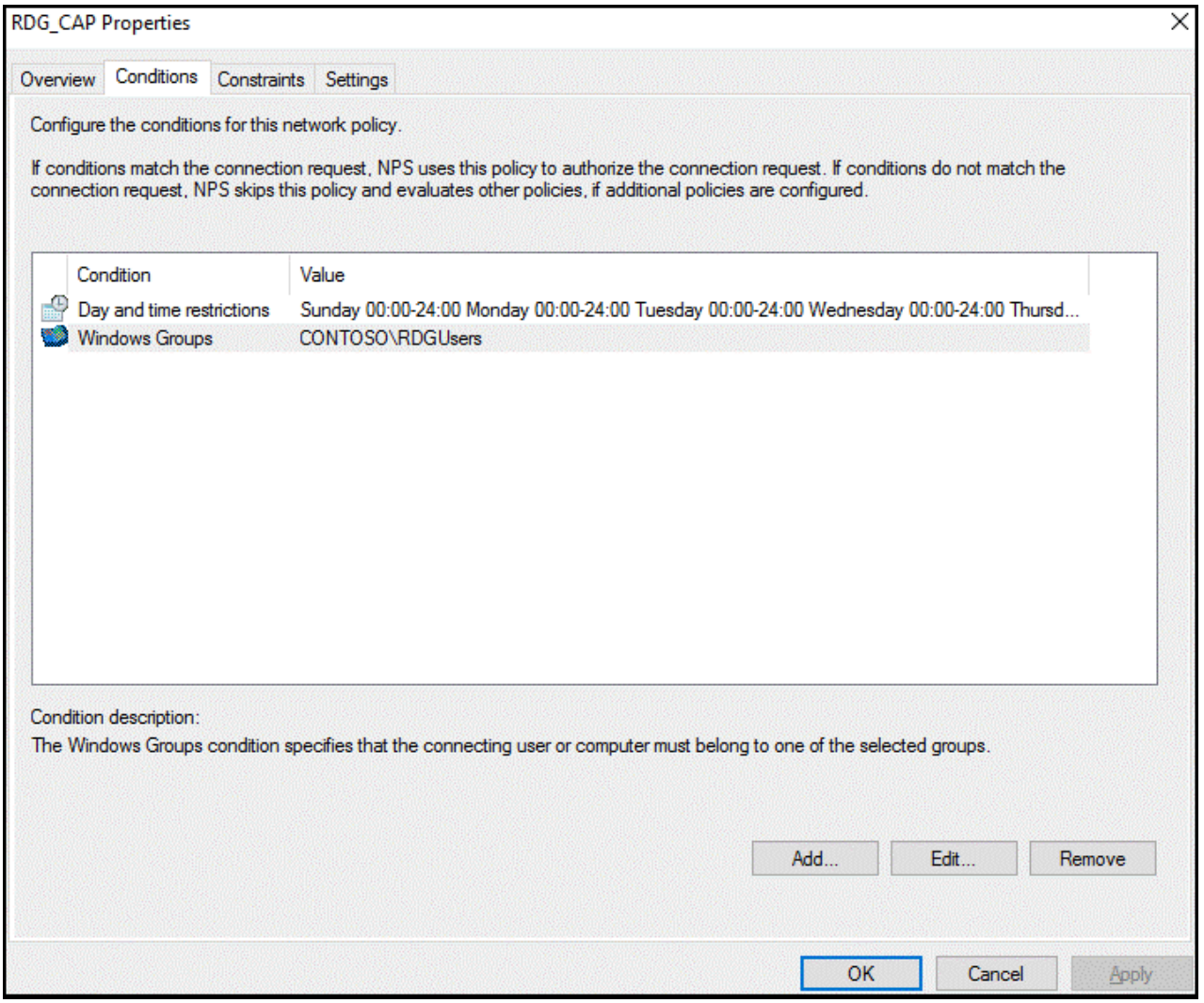
4. In the **Copy of Connections to other access servers** dialog box, in **Policy name**, enter a suitable name, such as *RDG_CAP*. Check **Policy enabled**, and select **Grant access**. Optionally, in **Type of network access server**, select **Remote Desktop Gateway**, or you can leave it as **Unspecified**.



5. Select the **Constraints** tab, and check **Allow clients to connect without negotiating an authentication method**.

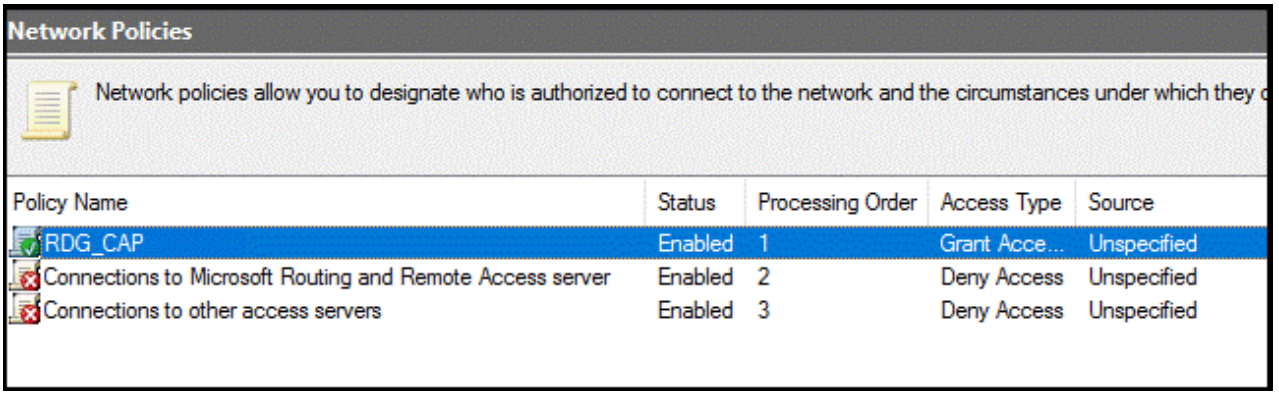


6. Optionally, select the **Conditions** tab and add conditions that must be met for the connection to be authorized, for example, membership in a specific Windows group.



7. Select **OK**. When prompted to view the corresponding Help topic, select **No**.

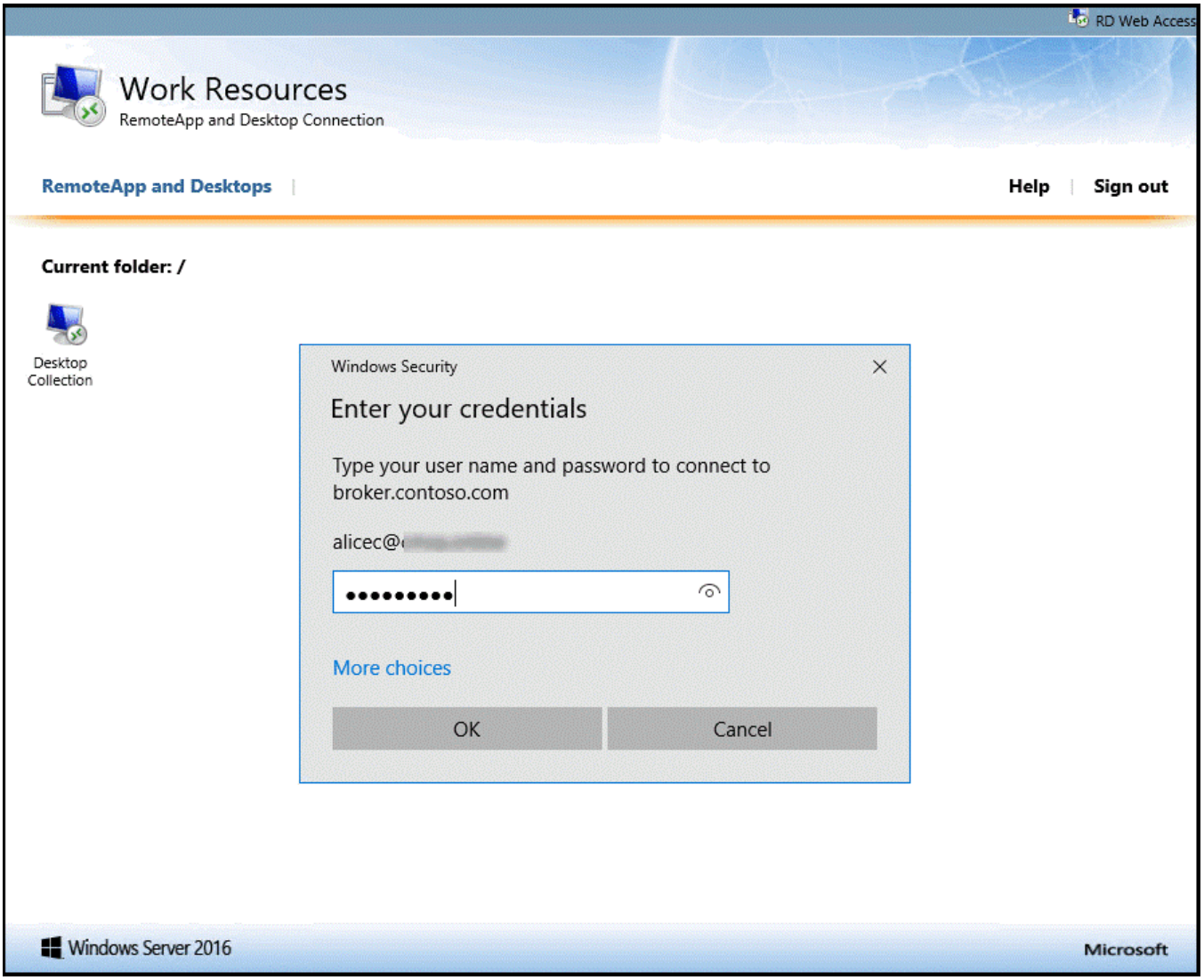
8. Ensure that your new policy is at the top of the list, that the policy is enabled, and that it grants access.



Verify configuration

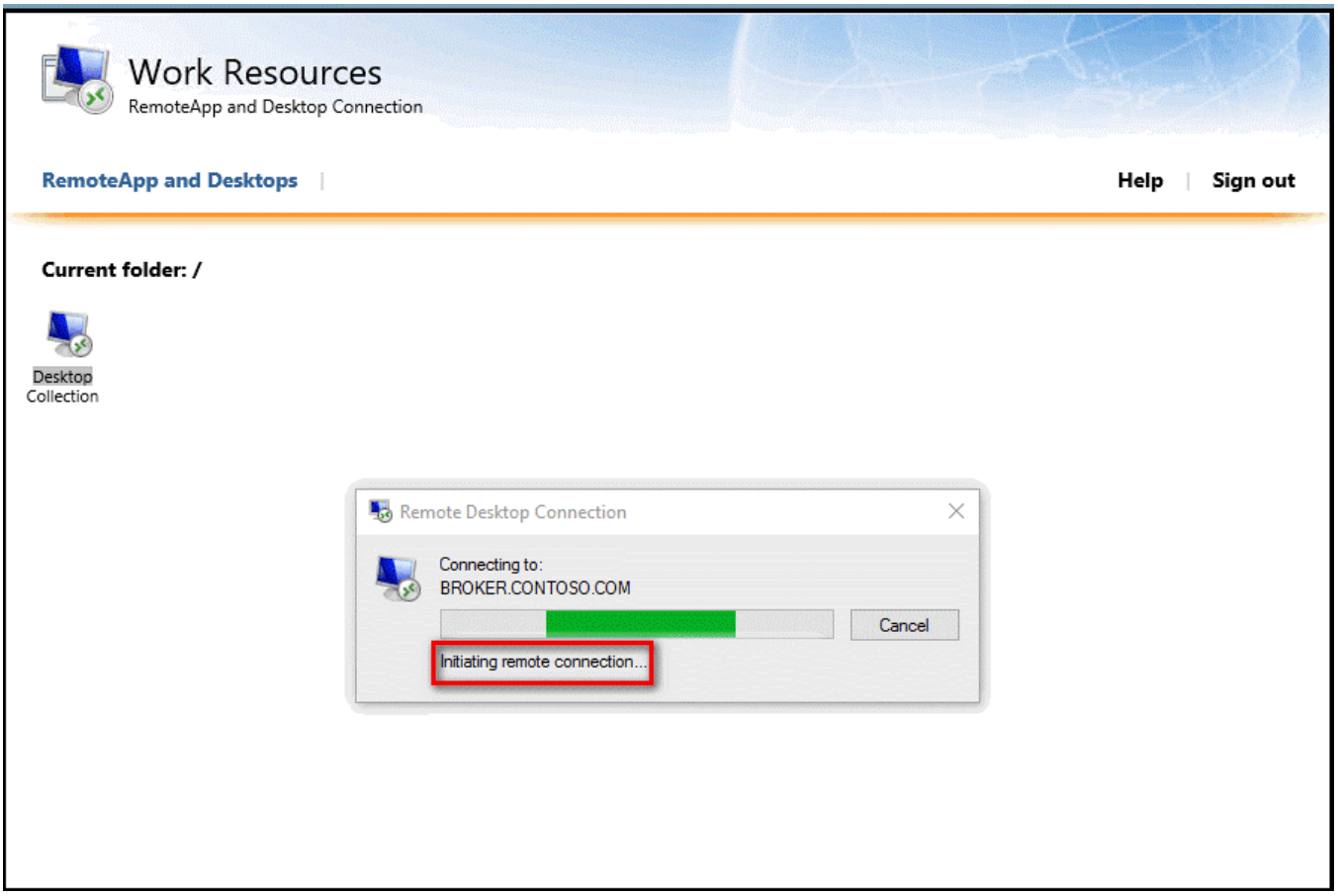
To verify the configuration, you need to sign in to the Remote Desktop Gateway with a suitable RDP client. Be sure to use an account that is allowed by your Connection Authorization Policies and is enabled for Microsoft Entra multifactor authentication.

As show in the following image, you can use the **Remote Desktop Web Access** page.

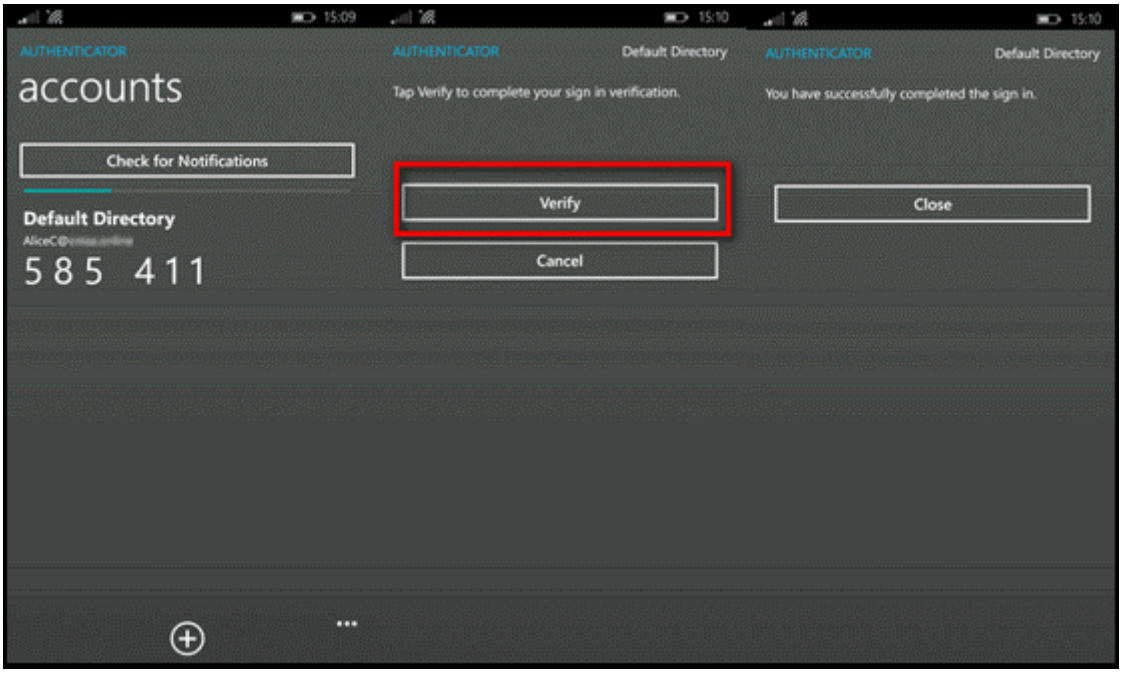


When you successfully entering your credentials for primary authentication, the Remote Desktop Connect dialog box shows a status of Initiating remote connection, as shown in the following section.

If you successfully authenticate with the secondary authentication method you previously configured in Microsoft Entra multifactor authentication, you're connected to the resource. However, if the secondary authentication isn't successful, you're denied access to the resource.



In the following example, the Authenticator app on a Windows phone is used to provide the secondary authentication.



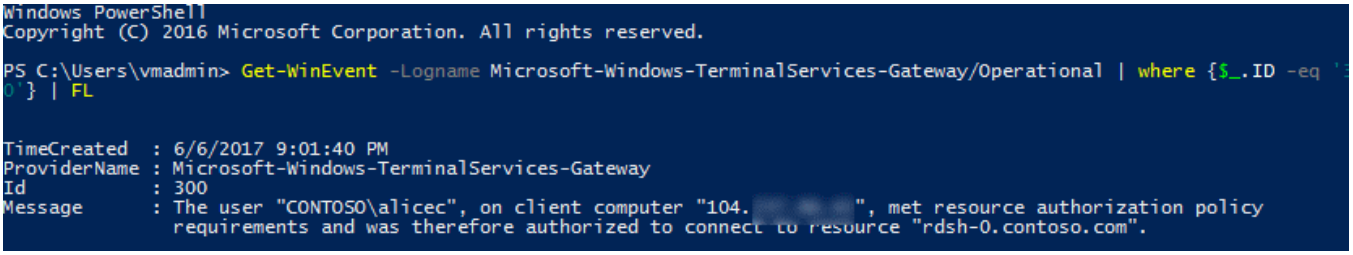
Once you have successfully authenticated using the secondary authentication method, you're logged into the Remote Desktop Gateway as normal. However, because you're required to use a secondary authentication method using a mobile app on a trusted device, the sign in process is more secure than it would be otherwise.

View Event Viewer logs for successful logon events

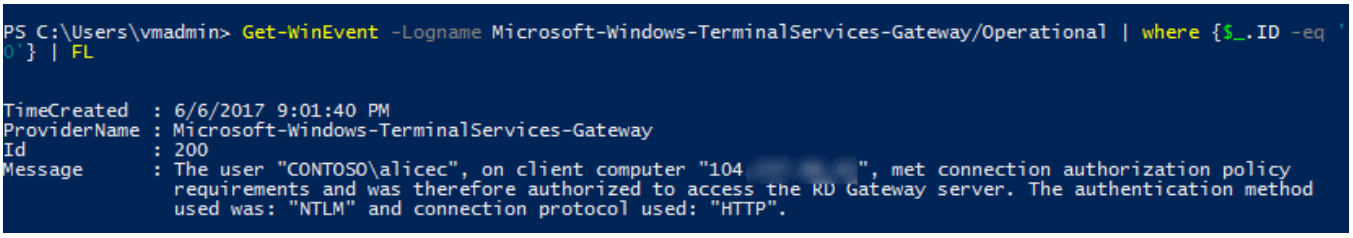
To view the successful sign-in events in the Windows Event Viewer logs, you can issue the following PowerShell command to query the Windows Terminal Services and Windows Security logs.

To query successful sign-in events in the Gateway operational logs *(Event Viewer\Applications and Services Logs\Microsoft\Windows\TerminalServices-Gateway\Operational)*, use the following PowerShell commands:

- `Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '300'} | FL`
- This command displays Windows events that show the user met resource authorization policy requirements (RD RAP) and was granted access.

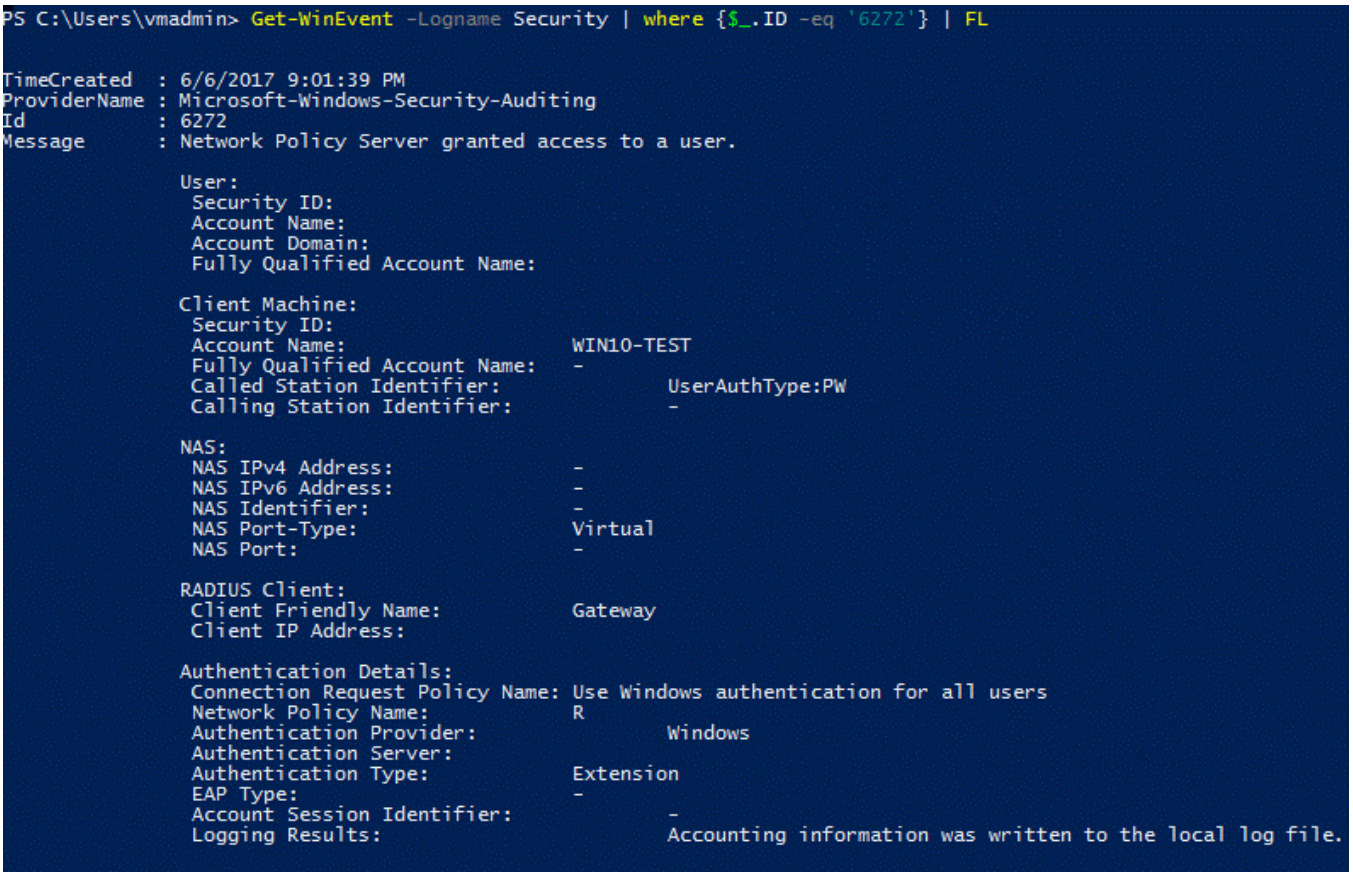


- `Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '200'} | FL`
- This command displays the events that show when user met connection authorization policy requirements.

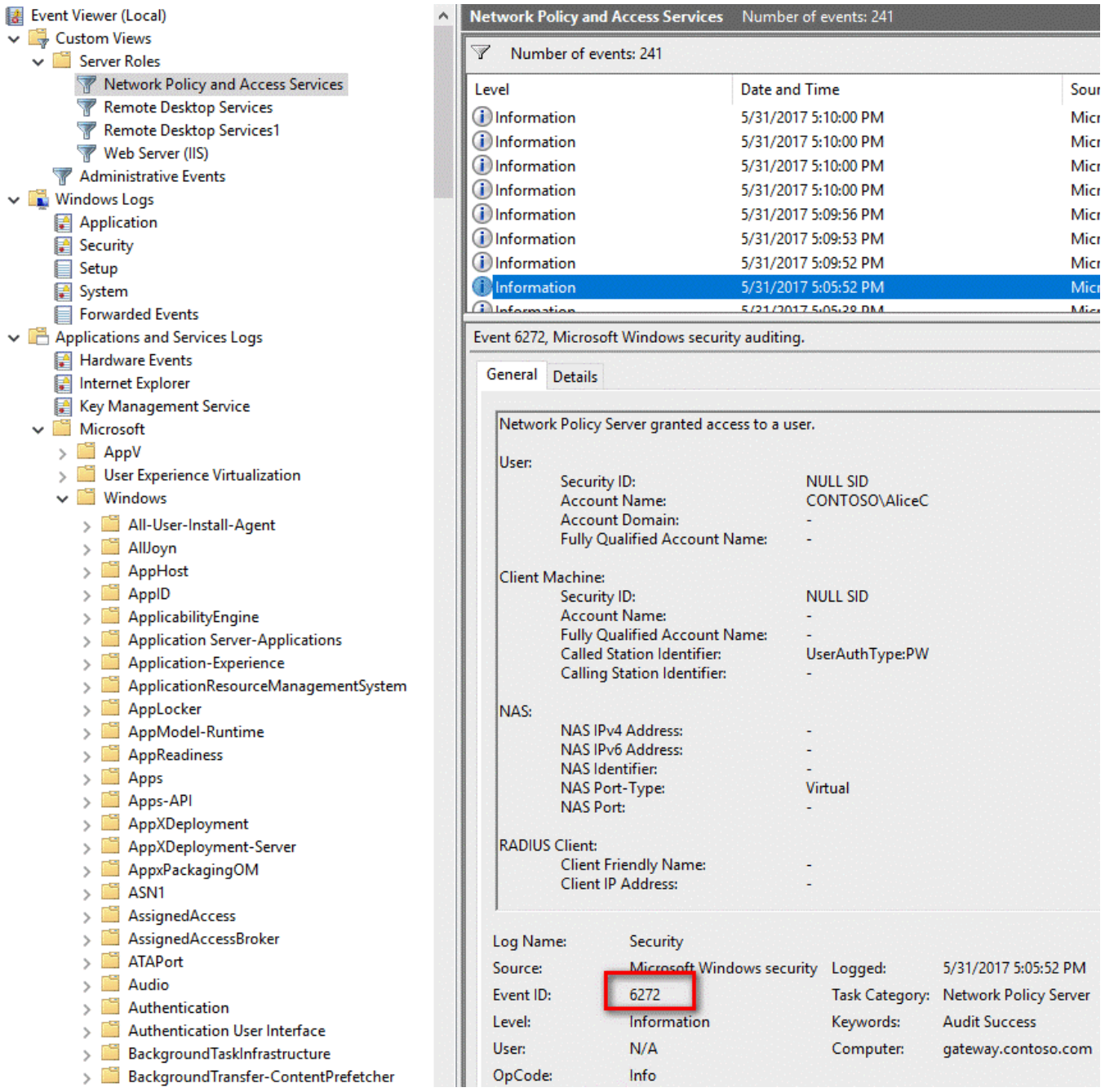


You can also view this log and filter on event IDs, 300 and 200. To query successful logon events in the Security event viewer logs, use the following command:

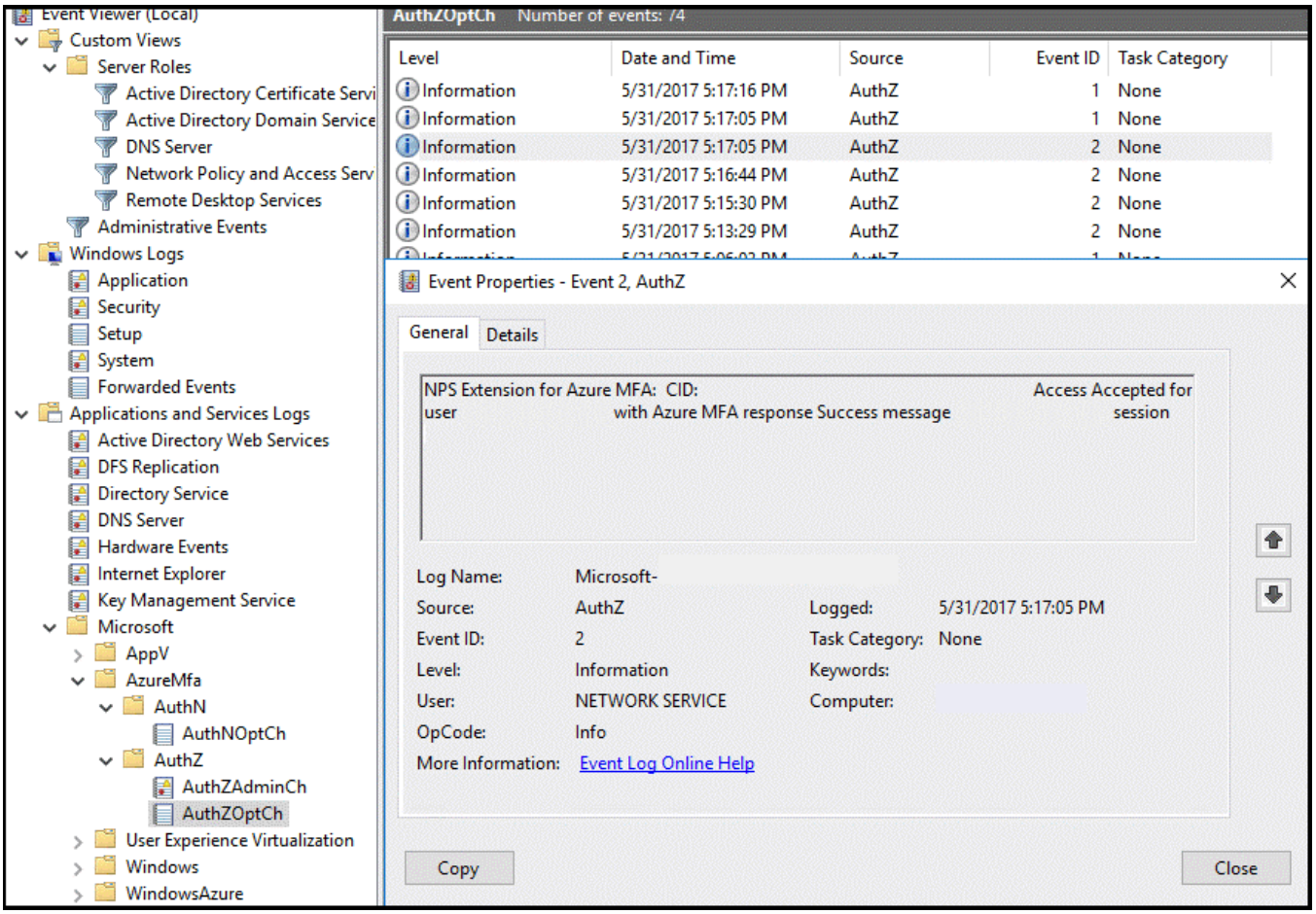
- `Get-WinEvent -Logname Security | where {$_.ID -eq '6272'} | FL`
- This command can be run on either the central NPS or the RD Gateway Server.



You can also view the Security log or the Network Policy and Access Services custom view:



On the server where you installed the NPS extension for Microsoft Entra multifactor authentication, you can find Event Viewer application logs specific to the extension at *Application and Services Logs\Microsoft\AzureMfa*.

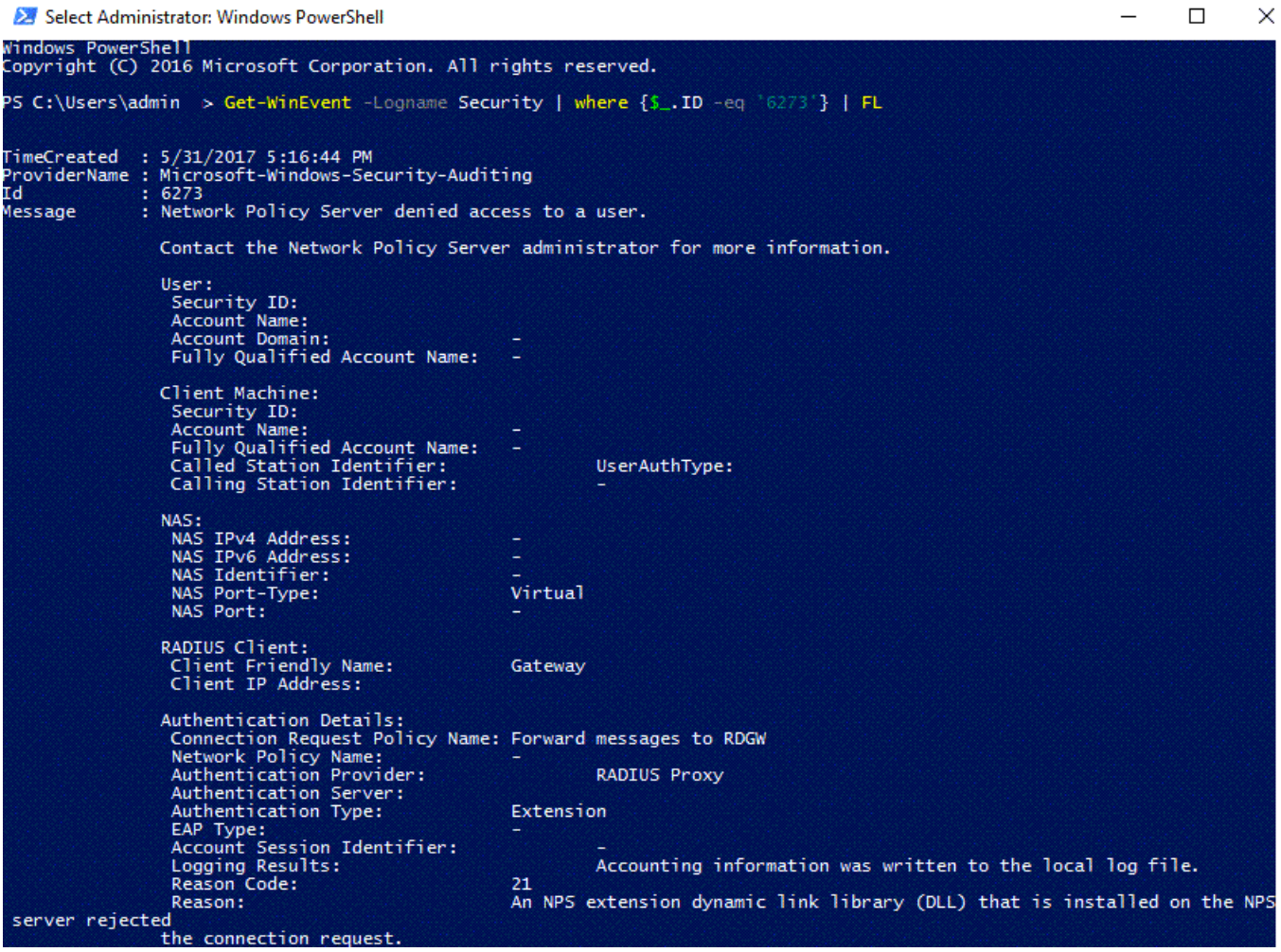


Troubleshoot Guide

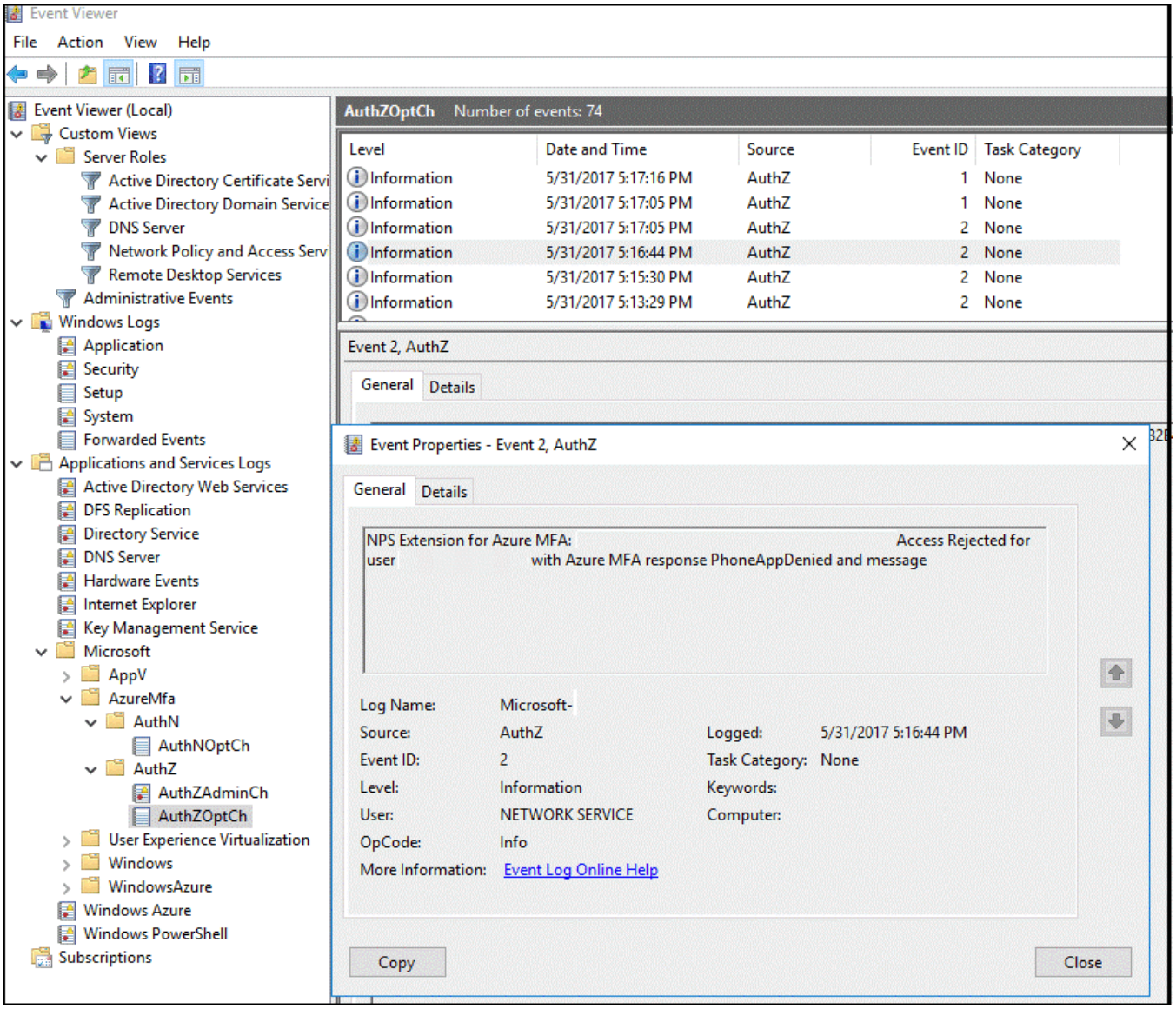
If the configuration isn't working as expected, the first place to start to troubleshoot is to verify that the user is configured to use Microsoft Entra multifactor authentication. Have the user sign in to the Microsoft Entra admin center. If users are prompted for secondary verification and can successfully authenticate, you can eliminate an incorrect configuration of Microsoft Entra multifactor authentication.

If Microsoft Entra multifactor authentication is working for the user(s), you should review the relevant Event logs. These include the Security Event, Gateway operational, and Microsoft Entra multifactor authentication logs that are discussed in the previous section.

See the following example output of Security log showing a failed logon event (Event ID 6273).



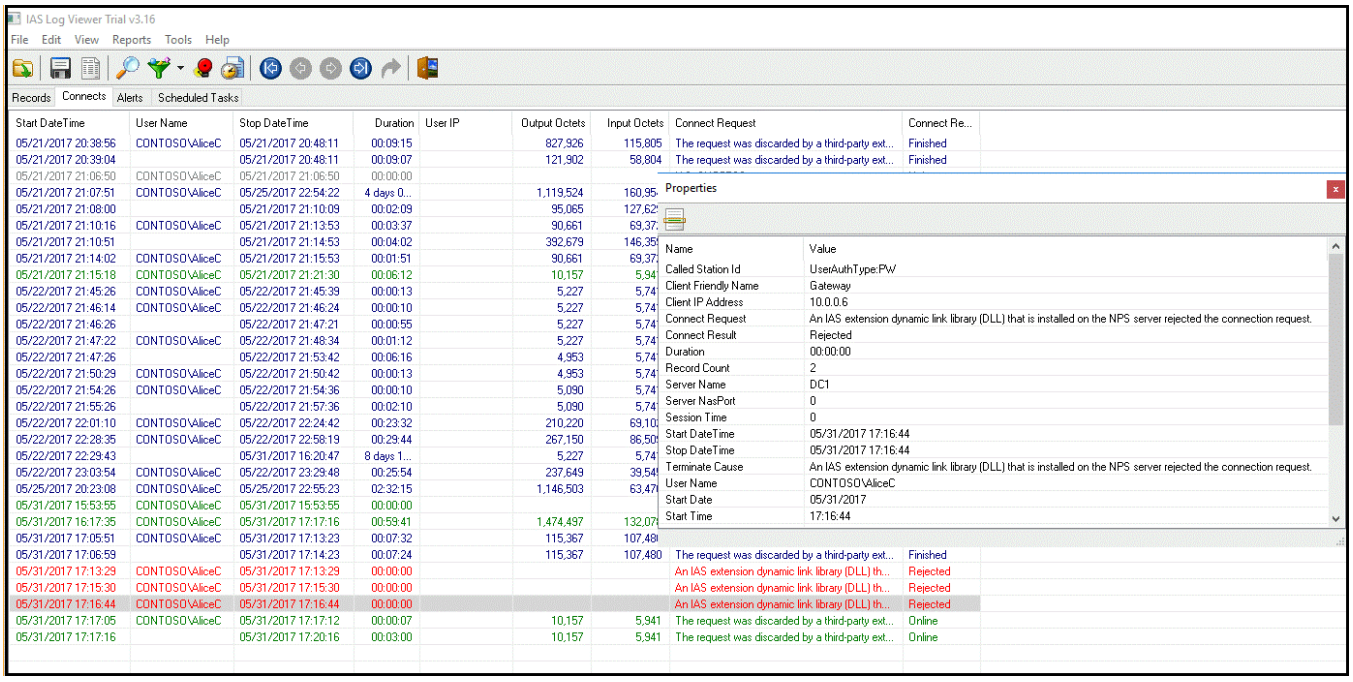
What follows is a related event from the AzureMFA logs:



To perform advanced troubleshoot options, consult the NPS database format log files where the NPS service is installed. These log files are created in *%SystemRoot%\System32\Logs* folder as comma-delimited text files.

For a description of these log files, see Interpret NPS Database Format Log Files. The entries in these log files can be difficult to interpret without importing them into a spreadsheet or a database. You can find several IAS parsers online to assist you in interpreting the log files.

The following image shows the output of one such downloadable shareware application.



Next steps

How to get Microsoft Entra multifactor authentication

Remote Desktop Gateway and Azure Multi-Factor Authentication Server using RADIUS

Integrate your on-premises directories with Microsoft Entra ID

Last updated on 03/04/2025

Integrate Microsoft Entra Domain Services with your RDS deployment

Article • 07/03/2024 •

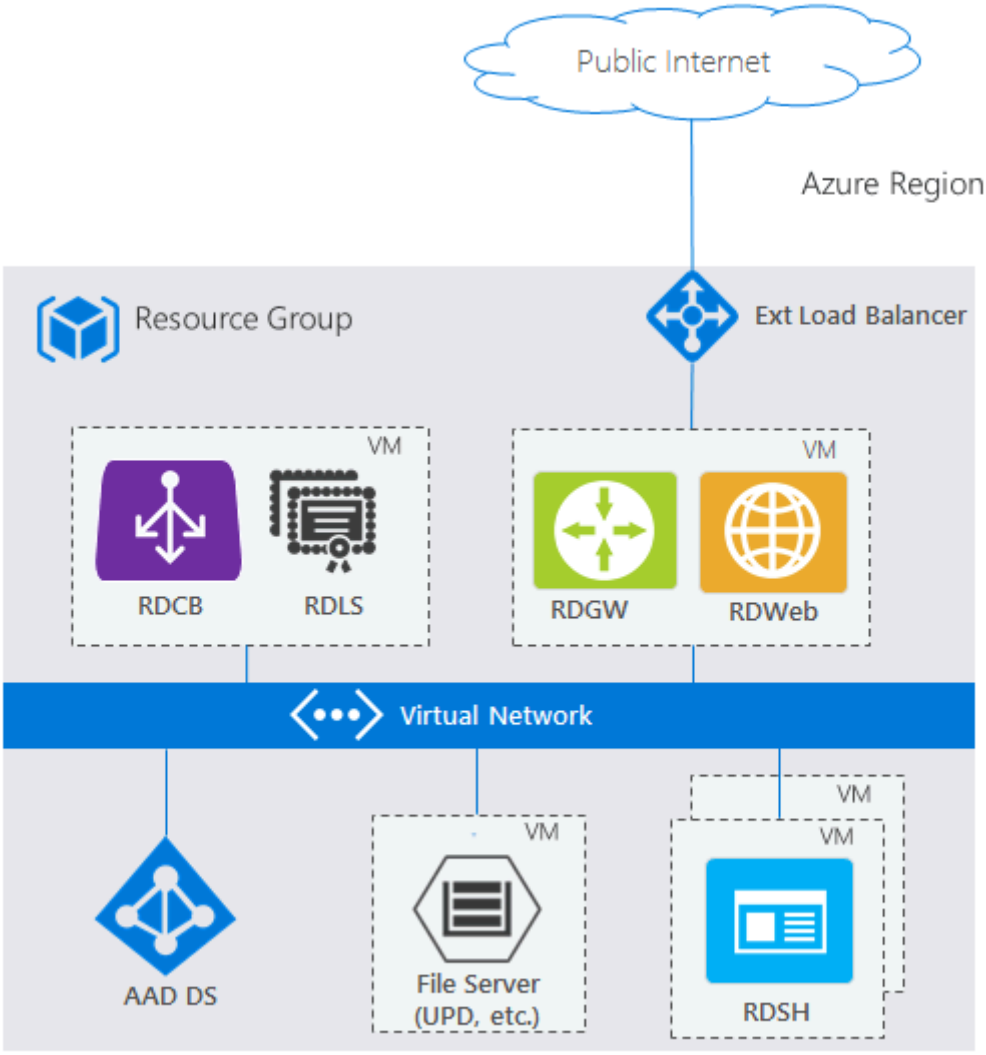
Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use Microsoft Entra Domain Services in your Remote Desktop Services deployment in the place of Windows Server Active Directory. Microsoft Entra Domain Services lets you use your existing Microsoft Entra identities in with classic Windows workloads.

With Microsoft Entra Domain Services you can:

- Create an Azure environment with a local domain for born-in-the-cloud organizations.
- Create an isolated Azure environment with the same identities used for your on-premises and online environment, without needing to create a site-to-site VPN or ExpressRoute.

When you finish integrating Microsoft Entra Domain Services into your Remote Desktop deployment, your architecture will look something like this:



To see how this architecture compares with other RDS deployment scenarios, check out Remote Desktop Services architectures.

To get a better understanding of Microsoft Entra Domain Services, check out the Microsoft Entra Domain Services overview and How to decide if Microsoft Entra Domain Services is right for your use-case.

Use the following information to deploy Microsoft Entra Domain Services with RDS.

Prerequisites

Before you can bring your identities from Microsoft Entra ID to use in an RDS deployment, configure Microsoft Entra ID to save the hashed passwords for your users' identities. Born-in-the-cloud organizations don't need to make any additional changes in their directory; however, on-premises organizations need to allow password hashes to be synchronized and stored in Microsoft Entra ID, which may not be permissible to

some organizations. Users will have to reset their passwords after making this configuration change.

Deploy Microsoft Entra Domain Services and RDS

Use the following steps to deploy Microsoft Entra Domain Services and RDS.

1. Enable Microsoft Entra Domain Services. Note that the linked article does the following:

    - Walk through creating the appropriate Microsoft Entra groups for domain administration.
    - Highlight when you might have to force users to change their password so their accounts can work with Microsoft Entra Domain Services.

2. Set up RDS. You can either use an Azure template or deploy RDS manually.

    - Use the Existing AD template. Make sure to customize the following:

        - **Settings**

            - **Resource group**: Use the resource group where you want to create the RDS resources.

                > **Note**
                >
                > Right now this has to be the same resource group where the Azure resource manager virtual network exists.

            - **Dns Label Prefix**: Enter the URL that you want users to use to access RD Web.

            - **Ad Domain Name**: Enter the full name of your Microsoft Entra instance, for example, "contoso.onmicrosoft.com" or "contoso.com".

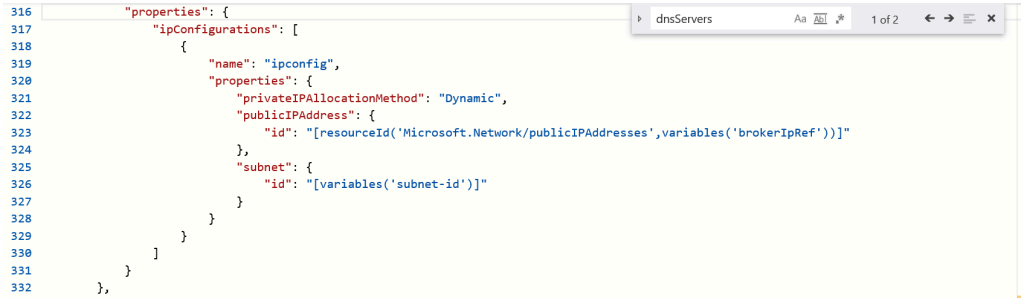
            - **Ad Vnet Name** and **Ad Subnet Name**: Enter the same values that you used when you created the Azure resource manager virtual network. This is the subnet to which the RDS resources will connect.

            - **Admin Username** and **Admin Password**: Enter the credentials for an admin user that's a member of the **AAD DC Administrators** group in Microsoft Entra ID.

- **Template**
  - Remove all properties of **dnsServers**: after selecting **Edit template** from the Azure quickstart template page, search for "dnsServers" and remove the property.

    For example, before removing the **dnsServers** property:

    And here's the same file after removing the property:



- Deploy RDS manually.

Feedback

Was this page helpful? Yes No

Publish Remote Desktop with Microsoft Entra application proxy

Overview

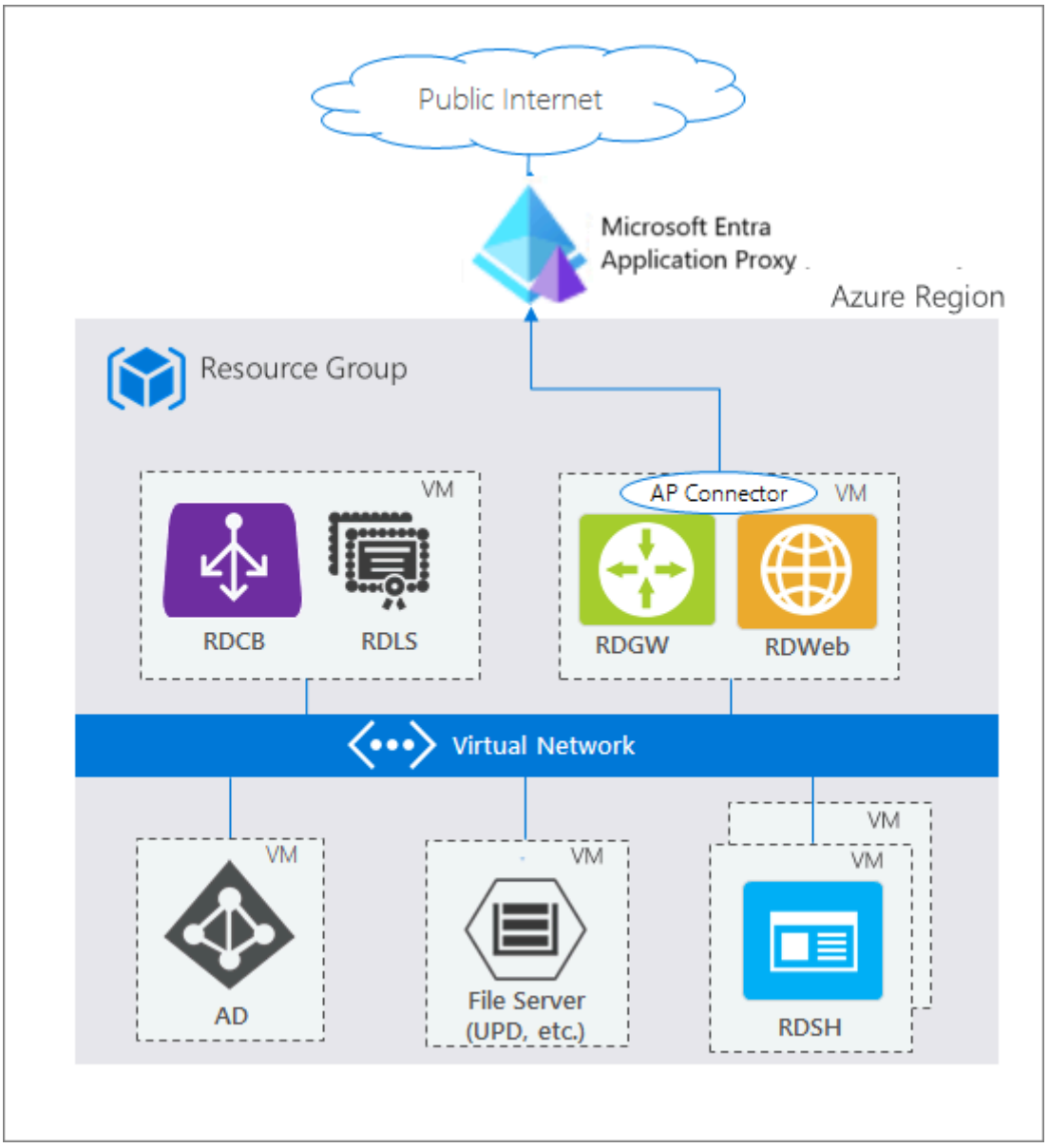
Remote Desktop Service and Microsoft Entra application proxy work together to improve the productivity of workers who are away from the corporate network.

The intended audience for this article is:

- Current application proxy customers who want to offer more applications to their end users by publishing on-premises applications through Remote Desktop Services.
- Current Remote Desktop Services customers who want to reduce the attack surface of their deployment by using Microsoft Entra application proxy. This scenario gives a set of two-step verification and Conditional Access controls to RDS.

How application proxy fits in the standard RDS deployment

A standard RDS deployment includes various Remote Desktop role services running on Windows Server. Multiple deployment options exist in the Remote Desktop Services architecture. Unlike other RDS deployment options, the RDS deployment with Microsoft Entra application proxy (shown in the following diagram) has a permanent outbound connection from the server running the connector service. Other deployments leave open inbound connections through a load balancer.



In an RDS deployment, the Remote Desktop (RD) Web role and the RD Gateway role run on Internet-facing machines. These endpoints are exposed for the following reasons:

- RD Web provides the user a public endpoint to sign in and view the various on-premises applications and desktops they can access. When you select a resource, a Remote Desktop Protocol (RDP) connection is created using the native app on the OS.
- RD Gateway comes into the picture once a user launches the RDP connection. The RD Gateway handles encrypted RDP traffic coming over the internet and translates it to the on-premises server that the user is connecting to. In this scenario, the traffic the RD Gateway is receiving comes from the Microsoft Entra application proxy.

> **Tip**
>
> For more information, see how to seamlessly deploy RDS with Azure Resource Manager and Azure Marketplace.

Requirements

- Both the RD Web and RD Gateway endpoints must be located on the same machine, and with a common root. RD Web and RD Gateway are published as a single application with application proxy so that you can have a single sign-on experience between the two applications.
- Deploy RDS, and enabled application proxy. Enable application proxy and open required ports and URLs, and enable Transport Layer Security (TLS) 1.2 on the server. To learn which ports need to be opened, and other details, see Tutorial: Add an on-premises application for remote access through application proxy in Microsoft Entra ID.
- Your end users must use a compatible browser to connect to RD Web or the RD Web client. For more information, see Support for client configurations.
- When publishing RD Web, use the same internal and external Fully Qualified Domain Name (FQDN) when possible. If the internal and external Fully Qualified Domain Names (FQDNs) are different, disable Request Header Translation to avoid the client receiving invalid links.
- If you're using the RD Web client, you *must* use the same internal and external FQDN. If the internal and external FQDNs are different, you encounter websocket errors when making a RemoteApp connection through the RD Web client.
- If you're using RD Web on Internet Explorer, you need to enable the RDS ActiveX add-on.
- If you're using the RD Web client, you need to use the application proxy connector version 1.5.1975 or later.
- For the Microsoft Entra pre authentication flow, users can only connect to resources published to them in the **RemoteApp and Desktops** pane. Users can't connect to a desktop using the **Connect to a remote PC** pane.
- If you're using Windows Server 2019, you need to disable HTTP2 protocol. For more information, see Tutorial: Add an on-premises application for remote access through application proxy in Microsoft Entra ID.

Deploy the joint RDS and application proxy scenario

After setting up RDS and Microsoft Entra application proxy for your environment, follow the steps to combine the two solutions. These steps walk through publishing the two web-facing RDS endpoints (RD Web and RD Gateway) as applications, and then directing traffic on your RDS to go through application proxy.

Publish the RD host endpoint

1. Publish a new application proxy application with the values.

- Internal URL: `https://<rdhost>.com/`, where `<rdhost>` is the common root that RD Web and RD Gateway share.
- External URL: This field is automatically populated based on the name of the application, but you can modify it. Your users go to this URL when they access RDS.
- Pre authentication method: Microsoft Entra ID.
- Translate URL headers: No.
- Use HTTP-Only Cookie: No.

2. Assign users to the published RD application. Make sure they all have access to RDS, too.

3. Leave the single sign-on method for the application as **Microsoft Entra single sign-on disabled**.

   > ⓘ **Note**
   >
   > Your users are asked to authenticate once to Microsoft Entra ID and once to RD Web, but they have single sign-on to RD Gateway.

4. Browse to **Entra ID** > **App registrations**. Choose your app from the list.

5. Under **Manage**, select **Branding**.

6. Update the **Home page URL** field to point to your RD Web endpoint (like `https://<rdhost>.com/RDWeb`).

Direct RDS traffic to application proxy

Connect to the RDS deployment as an administrator and change the RD Gateway server name for the deployment. This configuration ensures that connections go through the Microsoft Entra application proxy service.

1. Connect to the RDS server running the RD Connection Broker role.

2. Launch **Server Manager**.

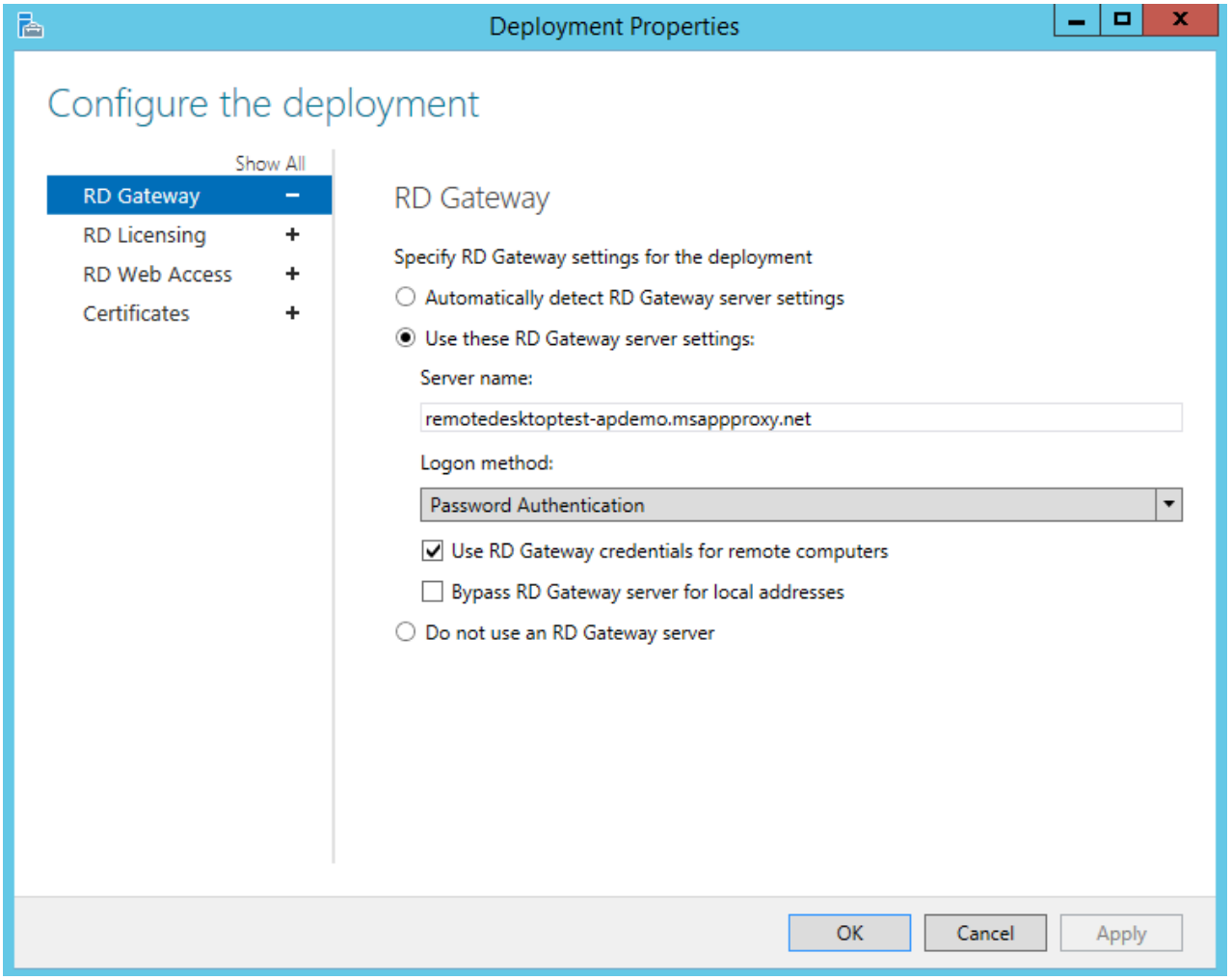
3. Select **Remote Desktop Services** from the pane on the left.

4. Select **Overview**.

5. In the Deployment Overview section, select the drop-down menu and choose **Edit deployment properties**.

6. In the RD Gateway tab, change the **Server name** field to the External URL that you set for the RD host endpoint in application proxy.

7. Change the **Logon method** field to **Password Authentication**.



8. Run this command for each collection. Replace <*yourcollectionname*> and <*proxyfrontendurl*> with your own information. This command enables single sign-on between RD Web and RD Gateway, and optimizes performance.

PowerShell

```
Set-RDSessionCollectionConfiguration -CollectionName "<yourcollectionname>" -
CustomRdpProperty "pre-authentication server address:s:
<proxyfrontendurl>`nrequire pre-authentication:i:1"
```

For example:

PowerShell

```
Set-RDSessionCollectionConfiguration -CollectionName "QuickSessionCollection"
-CustomRdpProperty "pre-authentication server
address:s:https://remotedesktoptest-aadapdemo.msappproxy.net/`nrequire pre-
authentication:i:1"
```

> ⓘ Note

The command uses a backtick in ``nrequire`.

9. To verify the modification of the custom RDP properties and view the RDP file contents that are downloaded from RDWeb for this collection, run the following command.

```
PowerShell

(get-wmiobject -Namespace root\cimv2\terminalservices -Class
Win32_RDCentralPublishedRemoteDesktop).RDPFileContents
```

Now that Remote Desktop is configured, Microsoft Entra application proxy takes over as the internet-facing component of RDS. Remove the other public internet-facing endpoints on your RD Web and RD Gateway machines.

Enable the RD Web Client

If you want users to use the RD Web Client follow the steps at Set up the Remote Desktop web client for your users.

The Remote Desktop web client provides access for your organization's Remote Desktop infrastructure. An HTML5-compatible web browser such as Microsoft Edge, Google Chrome, Safari, or Mozilla Firefox (v55.0 and later) is required.

Test the scenario

Test the scenario with Internet Explorer on a Windows 7 or 10 computer.

1. Go to the external URL you set up, or find your application in the MyApps panel.
2. Authenticate to Microsoft Entra ID. Use an account that you assigned to the application.
3. Authenticate to RD Web.
4. Once your RDS authentication succeeds, you can select the desktop or application you want, and start working.

Support for other client configurations

The configuration outlined in this article is for access to RDS via RD Web or the RD Web Client. If you need to, however, you can support other operating systems or browsers. The difference is in the authentication method that you use.

Expand table

Authentication method	Supported client configuration
Pre authentication	RD Web- Windows 7/10/11 using Microsoft Edge Chromium IE mode + RDS ActiveX add-on
Pre authentication	RD Web Client- HTML5-compatible web browser such as Microsoft Edge, Internet Explorer 11, Google Chrome, Safari, or Mozilla Firefox (v55.0 and later)
Passthrough	Any other operating system that supports the Microsoft Remote Desktop application

> **Note**
>
> `Microsoft Edge Chromium IE` mode is required when the My Apps portal is used for accessing the Remote Desktop app.

The pre authentication flow offers more security benefits than the passthrough flow. With pre authentication you can use Microsoft Entra authentication features like single sign-on, Conditional Access, and two-step verification for your on-premises resources. You also ensure that only authenticated traffic reaches your network.

To use passthrough authentication, there are just two modifications to the steps listed in this article:

1. In Publish the RD host endpoint step 1, set the Preauthentication method to **Passthrough**.
2. In Direct RDS traffic to application proxy, skip step 8 entirely.

Next steps

- Enable remote access to SharePoint with Microsoft Entra application proxy
- Security considerations for accessing apps remotely by using Microsoft Entra application proxy
- Best practices for load balancing multiple app servers

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

Last updated on 03/25/2026

Scale out your Remote Desktop Services deployment by adding an RD Session Host farm

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can improve the availability and scale of your RDS deployment by adding a Remote Desktop Session Host (RDSH) farm.

Use the following steps to add another RD Session Host to your deployment:

1. Create a server to host the second RD Session Host. If you are using Azure virtual machines, make sure to include the new VM in the same availability set that holds your first RD Session Host.

2. Enable remote management on the new server or virtual machine:
   a. In Server Manager, click **Local Server** > **Remote management current setting (disabled)**.
   b. Select **Enable remote management for this server**, and then click **OK**.
   c. Optional: You can temporarily set Windows Update to not automatically download and install updates. This helps prevent changes and system restarts while you deploy the RDSH server. In Server Manager, click **Local Server** > **Windows Update current setting**. Click **Advanced options** > **Defer upgrades**.

3. Add the server or VM to the domain:
   a. In Server Manager, click **Local Server** > **Workgroup current setting**.
   b. Click **Change** > **Domain**, and then enter the domain name (for example, Contoso.com).
   c. Enter the domain administrator credentials.
   d. Restart the server or VM.

4. Add the new RD Session Host to the farm:

> ⓘ **Note**
>
> Step 1, creating a public IP address for the RDMS virtual machine, is only necessary if you are using a VM for the RDMS and if it does not already have an IP address assigned.

a. Create a public IP address for the virtual machine running Remote Desktop Management Services (RDMS). The RDMS virtual machine will typically be the virtual machine running the first instance of the RD Connection Broker role.
   i. In the Azure portal, click **Browse** > **Resource groups**, click the resource group for the deployment and then click the RDMS virtual machine (for example, Contoso-Cb1).
   ii. Click **Settings** > **Network interfaces**, and then click the corresponding network interface.
   iii. Click **Settings** > **IP address**.
   iv. For **Public IP address**, select **Enabled**, and then click **IP address**.
   v. If you have an existing public IP address you want to use, select it from the list. Otherwise, click **Create new**, enter a name, and then click **OK** and then **Save**.

b. Sign into the RDMS.

c. Add the new RDSH server to Server Manager:
   i. Launch Server Manager, click **Manage** > **Add Servers**.
   ii. In the Add Servers dialog, click **Find Now**.
   iii. Select the server you want to use for the RD Session Host or the newly created virtual machine (for example, Contoso-Sh2) and click **OK**.

d. Add the RDSH server to the deployment
   i. Launch Server Manager.
   ii. Click **Remote Desktop Services** > **Overview** > **Deployment Servers** > **Tasks** > **Add RD Session Host Servers**.
   iii. Select the new server (for example, Contoso-Sh2), and then click **Next**.
   iv. On the Confirmation page, select **Restart remote computers as needed**, and then click **Add**.

e. Add RDSH server to the collection farm:
   i. Launch Server Manager.
   ii. Click **Remote Desktop Services** and then click the collection to which you want to add the newly created RDSH server (for example, ContosoDesktop).
   iii. Under **Host Servers**, click **Tasks** > **Add RD Session Host Servers**.
   iv. Select the newly created server (for example, Contoso-Sh2), and then click **Next**.
   v. On the Confirmation page, click **Add**.

Feedback

Configure the Remote Desktop Connection Broker for high availability

07/02/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

To ensure the reliability and scalability of your Remote Desktop Services infrastructure, you can configure the Remote Desktop Connection Broker for high availability. This article shows you how to set up a highly available Connection Broker cluster, including prerequisites, database configuration, load balancing, and final deployment steps. By following these instructions, you can minimize downtime and optimize performance for your remote desktop environment.

Prerequisites

Before you begin, you need to meet the following prerequisites:

- Set up a server to act as a second RD Connection Broker. This server can be either a physical server or a VM.
- Set up a database for the Connection Broker. You can use Azure SQL Database instance or SQL Server in your local environment. We give an example using Azure SQL, but the steps still apply to SQL Server. You need to find the connection string for the database and make sure you have the correct ODBC driver.

Configure the database for the Connection Broker

1. Find the connection string for the database you created - you need it both to identify the version of ODBC driver you need and later, when you're configuring the Connection Broker itself (step 3), so save the string somewhere you can reference it easily. Here's how you find the connection string for Azure SQL:

   a. In the Azure portal, select **Browse** > **Resource groups** and select the resource group for the deployment.

   b. Select the SQL database you created (for example, CB-DB1).

   c. Select **Settings** > **Properties** > **Show database connection strings**.

   d. Copy the connection string for **ODBC (includes Node.js)**, which should look like this. Replace the `<values>` with your values. You use this entire string, with your included password, when connecting to the database.

```
connectionstring
```

```
Driver={ODBC Driver 13 for SQL Server};Server=tcp:<YourHost>,
<HostPort>;Database=<DatabaseName>;Uid=<UserID>;Pwd=
<Password>;Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;
```

2. Install the ODBC driver on the new Connection Broker:

   a. If you're using a VM for the Connection Broker, create a public IP address for the first RD Connection Broker. (You only have to do this step if the RDMS virtual machine doesn't already have a public IP address to allow RDP connections.)
      i. In the Azure portal, select **Browse** > **Resource groups**, select the resource group for the deployment, and then select the first RD Connection Broker virtual machine (for example, Contoso-Cb1).
      ii. Select **Settings** > **Network interfaces**, and then select the corresponding network interface.
      iii. Select **Settings** > **IP address**.
      iv. For **Public IP address**, select **Enabled**, and then select **IP address**.
      v. If you have an existing public IP address you want to use, select it from the list. Otherwise, select **Create new**, enter a name, and then select **OK** and then **Save**.

   b. Connect to the first RD Connection Broker:
      i. In the Azure portal, select **Browse** > **Resource groups**, select the resource group for the deployment, and then select the first RD Connection Broker virtual machine (for example, Contoso-Cb1).
      ii. Select **Connect** > **Open** to open the Remote Desktop client.
      iii. In the client, select **Connect**, and then select **Use another user account**. Enter the user name and password for a domain administrator account.
      iv. Select **Yes** when warned about the certificate.

   c. Download the ODBC driver for SQL Server that matches the version in the ODBC connection string. For the example string, we need to install the version 13 ODBC driver.

   d. Copy the `sqlincli.msi` file to the first RD Connection Broker server.

   e. Open the `sqlincli.msi` file and install the native client.

   f. Repeat steps 1-5 for each RD Connection Brokers (for example, Contoso-Cb2).

   g. Install the ODBC driver on each server that runs the connection broker.

Configure load balancing on the RD Connection Brokers

You can use a load balancer, such as Azure load balancer; if not, you can set up DNS round-robin.

Create a load balancer

1. Create an Azure Load Balancer
   a. In the Azure portal, select **Browse** > **Load balancers** > **Add**.
   b. Enter a name for the new load balancer (for example, `hacb`).
   c. Select **Internal** for the **Scheme**, **Virtual Network** for your deployment (for example, Contoso-VNet), and the **Subnet** with all of your resources (for example, default).
   d. Select **Static** for the **IP address assignment** and enter a **Private IP address** that isn't currently in use (for example, 10.0.0.32).
   e. Select the appropriate **Subscription**, the **Resource group** with all of your resources, and the appropriate **Location**.
   f. Select **Create**.

2. Create a probe to monitor which servers are active:
   a. In Azure portal, select **Browse** > **Load Balancers**, and then select the load balancer you created, for example, `CBLB`. Select **Settings**.
   b. Select **Probes** > **Add**.
   c. Enter a name for the probe (for example, **RDP**), select **TCP** as the **Protocol**, enter **3389** for the **Port**, and then select **OK**.

3. Create the backend pool of the Connection Brokers:
   a. In **Settings**, select **Backend address pools** > **Add**.
   b. Enter a name (for example, CBBackendPool), then select **Add a virtual machine**.
   c. Choose an availability set (for example, CbAvSet), and then select **OK**.
   d. Select **Choose the virtual machines**, select each virtual machine, and then select **Select** > **OK** > **OK**.

4. Create the RDP load balancing rule:
   a. In **Settings**, select **Load balancing rules**, and then select **Add**.
   b. Enter a name (for example, RDP), select **TCP** for the **Protocol**, enter **3389** for both **Port** and **Backend port**, and select **OK**.

5. Add a DNS record for the Load Balancer:
   a. Connect to the RDMS server virtual machine (for example, Contoso-CB1). Check out the Prepare the RD Connection Broker VM article for steps on how you connect to the

VM.
   b. In Server Manager, select **Tools** > **DNS**.
   c. In the left-hand pane, expand **DNS**, select the DNS machine, select **Forward Lookup Zones**, and then select your domain name (for example, Contoso.com). (It might take a few seconds to process the query to the DNS server for the information.)
   d. Select **Action** > **New Host (A or AAAA)**.
   e. Enter the name (for example, `hacb`) and the IP address specified earlier (for example, 10.0.0.32).

Configure DNS round-robin

The following steps are an alternative to creating an Azure Internal Load Balancer.

1. Connect to the RDMS server in the Azure portal. using Remote Desktop Connection client

2. Create DNS records:
   a. In Server Manager, select **Tools** > **DNS**.
   b. In the left-hand pane, expand **DNS**, select the DNS machine, select **Forward Lookup Zones**, and then select your domain name (for example, Contoso.com). (It might take a few seconds to process the query to the DNS server for the information.)
   c. Select **Action** and **New Host (A or AAAA)**.
   d. Enter the **DNS Name** for the RD Connection Broker cluster (for example, `hacb`), and then enter the **IP address** of the first RD Connection Broker.
   e. Repeat steps 3-4 for each RD Connection Broker, providing each unique IP address for each record.

For example, if the IP addresses for the two RD Connection Broker virtual machines are 10.0.0.8 and 10.0.0.9, you would create two DNS host records:

- Host name: `hacb.contoso.com`, IP address: `10.0.0.8`
- Host name: `hacb.contoso.com`, IP address: `10.0.0.9`

Configure the Connection Brokers for high availability

1. Add the new RD Connection Broker server to Server Manager:
   a. In Server Manager, select **Manage** > **Add Servers**.
   b. Select **Find Now**.
   c. Select the newly created RD Connection Broker server (for example, Contoso-Cb2) and select **OK**.

2. Configure high availability for the RD Connection Broker:
   a. In Server Manager, select **Remote Desktop Services** > **Overview**.
   b. Right-click **RD Connection Broker**, and then select **Configure High Availability**.
   c. Page through the wizard until you get to the Configuration type section. Select **Shared database server**, and then select **Next**.
   d. Enter the DNS name for the RD Connection Broker cluster.
   e. Enter the connection string for the SQL DB, and then page through the wizard to establish high availability.

3. Add the new RD Connection Broker to the deployment
   a. In Server Manager, select **Remote Desktop Services** > **Overview**.
   b. Right-click the RD Connection Broker, and then select **Add RD Connection Broker Server**.
   c. Page through wizard until you get to Server Selection, then select the newly created RD Connection Broker server (for example, Contoso-CB2).
   d. Complete the wizard, accepting the default values.

4. Configure trusted certificates on RD Connection Broker servers and clients.

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

Add high availability to the RD Web and Gateway web front

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can deploy a Remote Desktop Web Access (RD Web Access) and Remote Desktop Gateway (RD Gateway) farm to improve the availability and scale of a Windows Server Remote Desktop Services (RDS) deployment

Use the following steps to add an RD Web and Gateway server to an existing Remote Desktop Services basic deployment.

Pre-requisites

Set up a server to act as an additional RD Web and RD Gateway - this can be either a physical server or VM. This includes joining the server to the domain and enabling remote management.

Step 1: Configure the new server to be part of the RDS environment

1. Connect to the RDMS server in the Azure portal, using Remote Desktop Connection client.
2. Add the new RD Web and Gateway server to Server Manager:
   a. Launch Server Manager, click **Manage** > **Add Servers**.
   b. In the Add Servers dialog, click **Find Now**.
   c. Select the newly created RD Web and Gateway server (for example, Contoso-WebGw2) and click **OK**.
3. Add RD Web and Gateway servers to the deployment
   a. Launch Server Manager .
   b. Click **Remote Desktop Services** > **Overview** > **Deployment Servers** > **Tasks** > **Add RD Web Access Servers**.
   c. Select the newly created server (for example, Contoso-WebGw2), and then click **Next**.
   d. On the Confirmation page, select **Restart remote computers as needed**, and then click **Add**.

e. Repeat these steps to add the RD Gateway server, but choose **RD Gateway Servers** in step b.

4. Re-install certificates for the RD Gateway servers:
   a. In Server Manager on the RDMS server, click **Remote Desktop Services** > **Overview** > **Tasks** > **Edit Deployment Properties**.
   b. Expand **Certificates**.
   c. Scroll down to the table. Click RD **Gateway Role Service** > **Select existing certificate.**
   d. Click **Choose a different certificate** and then browse to the certificate location. For example, \Contoso-CB1\Certificates). Select the certificate file for the RD Web and Gateway server created during the prerequisites (e.g. ContosoRdGwCert), and then click **Open**.
   e. Enter the password for the certificate, select **Allow the certificate to be added to the Trusted Root Certificate Authorities certificate store on the destination computers**, and then click **OK**.
   f. Click **Apply**.

> ⓘ **Note**
>
> You may need to manually restart the TSGateway service running on each RD Gateway server, either through Server Manager or Task Manager.

   g. Repeat steps a through f for the RD Web Access Role Service.

Step 2: Configure RD Web and RD Gateway properties on the new server

1. Configure the server to be part of an RD Gateway farm:
   a. In Server Manager on the RDMS server, click **All Servers**. Right-click one of the RD Gateway servers, and then click **Remote Desktop Connection**.
   b. Sign into to the RD Gateway server using a domain admin account.
   c. In Server Manager on the RD Gateway server, click **Tools** > **Remote Desktop Services** > **RD Gateway Manager**.
   d. In the navigation pane, click the local computer (e.g. Contoso-WebGw1).
   e. Click **Add RD Gateway Server Farm members**.
   f. On the **Server Farm** tab, enter the name of each RD Gateway server, then click **Add** and **Apply**.
   g. Repeat steps a through f on each RD Gateway server so that they recognize each other as RD Gateway servers in a farm. Do not be alarmed if there are warnings, as it might take time for DNS settings to propagate.

2. Configure the server to be part of an RD Web Access farm. The steps below configure the Validation and Decryption Machine Keys to be the same on both RDWeb sites.
    a. In Server Manager on the RDMS server, click **All Servers**. Right-click the first RD Web Access server (e.g. Contoso-WebGw1) and then click **Remote Desktop Connection**.
    b. Sign into the RD Web Access server using a domain admin account.
    c. In Server Manager on the RD Web Access server, click **Tools** > **Internet Information Services (IIS) Manager**.
    d. In the left pane of IIS Manager, expand the **Server (e.g. Contoso-WebGw1)** > **Sites** > **Default Web Site**, and then click **RDWeb**.
    e. Right-click **Machine Key**, and then click **Open Feature**.
    f. On the Machine Key page, in the **Actions** pane, select **Generate Keys**, and then click **Apply**.
    g. Copy the validation key (you can right-click the key and then click **Copy**.)
    h. In IIS Manager, under **Default Web Site**, select **Feed**, **FeedLogon** and **Pages** in turn.
    i. For each:
        i. Right-click **Machine Key**, and then click **Open Feature**.
        ii. For the Validation Key, clear **Automatically generate at runtime**, and then paste the key you copied in step g.
    j. Minimize the RD Connection window to this RD Web server.
    k. Repeat steps b through e for the second RD Web Access server, ending on the feature view of **Machine Key**.
    l. For the Validation Key, clear **Automatically generate at runtime**, and then paste the key you copied in step g.
    m. Click **Apply**.
    n. Complete this process for the **RDWeb**, **Feed**, **FeedLogon** and **Pages** pages.
    o. Minimize the RD Connection window to the second RD Web Access server, and then maximize the RD Connection window to the first RD Web Access server.
    p. Repeat steps g through n to copy over the Decryption Key.
    q. When validation keys and decryption keys are identical on both RD Web Access servers for the **RDWeb**, **Feed**, **FeedLogon** and **Pages** pages, sign out of all RD Connection windows.

Step 3: Configure load balancing for the RD Web and RD Gateway servers

If you are using Azure infrastructure, you can create an external Azure load balancer; if not, you can set up a separate hardware or software load balancer. Load balancing is key

so that traffic will be evenly distributed the long-lived connections from Remote Desktop clients, through the RD Gateway, to the servers that users will be running their workloads.

> ⓘ **Note**
>
> If your previous server running RD Web and RD Gateway was already set up behind an external load balancer, skip ahead to step 4, select the existing backend pool, and add the new server to the pool.

1. Create an Azure Load Balancer:
   a. In the Azure portal click **Browse** > **Load balancers** > **Add**.
   b. Enter a name, for example **WebGwLB**.
   c. Select **Public** for the **Scheme**.
   d. Under **Public IP address**, select **Choose a public IP address**, and then pick an existing public IP address or create a new one.
   e. Select the appropriate **Subscription**, **Resource Group**, and **Location**.
   f. Click **Create**.
2. Create a probe to monitor which servers are alive:
   a. In the Azure portal, select **Browse** > **Load Balancers**, and then choose the load balancer that you created in the previous step.
   b. Select **All settings** > **Probes** > **Add**.
   c. Enter a name, for example, **HTTPS**, for the probe. Select **TCP** as the **Protocol**, and enter **443** for the **Port**, then click **OK**.
3. Create the HTTPS and UDP load balancing rules:
   a. In **Settings**, click **Load balancing rules**.
   b. Select **Add** for the **HTTPS rule**.
   c. Enter a name for the rule, for example, HTTPS, and select **TCP** for the **Protocol**. Enter **443** for both **Port** and **Backend port**, and click **OK**.
   d. In **Load balancing rules**, click **Add** for the **UDP rule**.
   e. Enter a name for the rule, for example, **UDP**, and select **UDP** for the **Protocol**. Enter **3391** for both **Port** and **Backend port**, and click **OK**.
4. Create the backend pool for the RD Web and RD Gateway servers:
   a. In **Settings**, click **Backend address pools** > **Add**.
   b. Enter a name (for example, **WebGwBackendPool**), then click **Add a virtual machine**.
   c. Choose an availability set (for example, WebGwAvSet), and then click **OK**.
   d. Click **Choose the virtual machines**, select each virtual machine, and then click **Select** > **OK** > **OK**.

Feedback

Was this page helpful? Yes No

Deploy a two-node Storage Spaces Direct scale-out file server for UPD storage in Azure

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Services (RDS) requires a domain-joined file server for user profile disks (UPDs). To deploy a high availability domain-joined scale-out file server (SOFS) in Azure, use Storage Spaces Direct with Windows Server 2016. If you're not familiar with UPDs or Remote Desktop Services, check out Welcome to Remote Desktop Services.

> **Note**
>
> Microsoft just published an Azure template to deploy a Storage Spaces Direct scale-out file server! You can use the template to create your deployment, or use the steps in this article.

We recommend deploying your SOFS with DS-series VMs and premium storage data disks, where there are the same number and size of data disks on each VM. You will need a minimum of two storage accounts.

For small deployments, we recommend a 2-node cluster with a cloud witness, where the volume is mirrored with 2 copies. Grow small deployments by adding data disks. Grow larger deployments by adding nodes (VMs).

These instructions are for a 2-node deployment. The following table shows the VM and disk sizes you'll need to store UPDs for the number of users in your business.

Expand table

Users	Total (GB)	VM	# Disks	Disk type	Disk size (GB)	Configuration
10	50	DS1	2	P10	128	2x(DS1 + 2 P10)
25	125	DS1	2	P10	128	2x(DS1 + 2 P10)
50	250	DS1	2	P10	128	2x(DS1 + 2 P10)
100	500	DS1	2	P20	512	2x(DS1 + 2 P20)

Users	Total (GB)	VM	# Disks	Disk type	Disk size (GB)	Configuration
250	1250	DS1	2	P30	1024	2x(DS1 + 2 P30)
500	2500	DS2	3	P30	1024	2x(DS2 + 3 P30)
1000	5000	DS3	5	P30	1024	2x(DS3 + 5 P30)
2500	12500	DS4	13	P30	1024	2x(DS4 + 13 P30)
5000	25000	DS5	25	P30	1024	2x(DS5 + 25 P30)

Use the following steps to create a domain controller (we called ours "my-dc" below) and two node VMs ("my-fsn1" and "my-fsn2") and configure the VMs to be a 2-node Storage Spaces Direct SOFS.

1. Create a Microsoft Azure subscription.
2. Sign into the Azure portal.
3. Create an Azure storage account in Azure Resource Manager. Create it in a new resource group and use the following configurations:

    - Deployment model: Resource Manager
    - Type of storage account: General purpose
    - Performance tier: Premium
    - Replication option: LRS

4. Set up an Active Directory forest by either using a quickstart template or deploying the forest manually.

    - Deploy using an Azure quickstart template:
        - Create an Azure VM with a new AD forest
        - Create a new AD domain with 2 domain controllers (for high availability)
    - Manually deploy the forest with the following configurations:
        - Create the virtual network in the same resource group as the storage account.
        - Recommended size: DS2 (increase the size if the domain controller will host more domain objects)
        - Use an automatically generated VNet.
        - Follow the steps to install AD DS.

5. Set up the file server cluster nodes. You can do this by deploying the Windows Server 2016 Storage Spaces Direct SOFS cluster Azure template or by following steps 6-11 to deploy manually.
6. To manually set up the file server cluster nodes:
    a. Create the first node:

i. Create a new virtual machine using the Windows Server 2016 image. (Click **New** > **Virtual Machines** > **Windows Server 2016.** Select **Resource Manager**, and then click **Create**.)

ii. Set the basic configuration as follows:

- Name: my-fsn1
- VM disk type SSD
- Use an existing resource group, the one that you created in step 3.

iii. Size: DS1, DS2, DS3, DS4, or DS5 depending on your user needs (see table at beginning of these instructions). Ensure premium disk support is selected.

iv. Settings:

- Storage account: Choose the storage account you created in step 3.
- High Availability - create a new availability set. (Click **High Availability** > **Create new**, and then enter a name (for example, s2d-cluster). Use the default values for **Update domains** and **Fault domains**.)

b. Create the second node. Repeat the step above with the following changes:

- Name: my-fsn2
- High Availability - select the availability set you created above.

7. Attach data disks to the cluster node VMs according to your user needs (as seen in the table above). After the data disks are created and attached to the VM, set **host caching** to **None**.
8. Set IP addresses for all VMs to **static**.
   a. In the resource group, select a VM, and then click **Network interfaces** (under **settings**). Select the listed network interface, and then click **IP Configurations**. Select the listed IP configuration, select **static**, and then click **Save**.
   b. Note the domain controller (my-dc for our example) private IP address (10.x.x.x).
9. Set primary DNS server address on NICs of the cluster node VMs to the my-dc server. Select the VM, and then click **Network Interfaces** > **DNS servers** > **Custom DNS**. Enter the private IP address you noted above, and then click **Save**.
10. Create an Azure storage account to be your cloud witness. (If you use the linked instructions, stop when you get to "Configuring Cloud Witness with Failover Cluster Manager GUI" - we'll do that step below.)
11. Set up the Storage Spaces Direct file server. Connect to a node VM, and then run the following Windows PowerShell cmdlets.

    a. Install Failover Clustering Feature and File Server Feature on the two file server cluster node VMs:

```
PowerShell
```

```
$nodes = ("my-fsn1", "my-fsn2")
icm $nodes {Install-WindowsFeature Failover-Clustering -
IncludeAllSubFeature -IncludeManagementTools}
icm $nodes {Install-WindowsFeature FS-FileServer}
```

b. Validate cluster node VMs and create 2-node SOFS cluster:

PowerShell

```
Test-Cluster -node $nodes
New-Cluster -Name MY-CL1 -Node $nodes –NoStorage –StaticAddress [new
address within your addr space]
```

c. Configure the cloud witness. Use your cloud witness storage account name and access key.

PowerShell

```
Set-ClusterQuorum –CloudWitness –AccountName <StorageAccountName> -
AccessKey <StorageAccountAccessKey>
```

d. Enable Storage Spaces Direct.

PowerShell

```
Enable-ClusterS2D
```

e. Create a virtual disk volume.

PowerShell

```
New-Volume -StoragePoolFriendlyName S2D* -FriendlyName VDisk01 -
FileSystem CSVFS_REFS -Size 120GB
```

To view information about the cluster shared volume on the SOFS cluster, run the following cmdlet:

PowerShell

```
Get-ClusterSharedVolume
```

f. Create the scale-out file server (SOFS):

PowerShell

```
Add-ClusterScaleOutFileServerRole -Name my-sofs1 -Cluster MY-CL1
```

g. Create a new SMB file share on the SOFS cluster.

PowerShell

```
New-Item -Path C:\ClusterStorage\VDisk01\Data -ItemType Directory
New-SmbShare -Name UpdStorage -Path C:\ClusterStorage\VDisk01\Data
```

You now have a share at `\\my-sofs1\UpdStorage`, which you can use for UPD storage when you enable UPD for your users.

Feedback

Was this page helpful? Yes No

Use personal session desktops with Remote Desktop Services

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can deploy server-based personal desktops in a cloud-computing environment by using personal session desktops. (A cloud-computing environment has a separation between the fabric Hyper-V servers and the guest virtual machines, such as Microsoft Azure Cloud or the Microsoft Cloud Platform.) The personal session desktop capability extends the session-based desktop deployment scenario in Remote Desktop Services to create a new type of session collection where each user is assigned to their own personal session host with administrative rights.

Use the following information to create and manage a personal session desktop collection.

Create a personal session desktop collection

Use the New-RDSessionCollection cmdlet to create a personal session desktop collection. The following three parameters provide the configuration information required for personal session desktops:

- **-PersonalUnmanaged** - Specifies the type of session collection that lets you assign users to a personal session host server. If you don't specify this parameter, then the collection is created as a traditional RD Session Host collection, where users are assigned to the next available session host when they sign in.
- **-GrantAdministrativePrivilege** - If you use **-PersonalUnmanaged**, specifies that the user assigned to the session host be given administrative privileges. If you don't use this parameter, users are granted only standard user privileges.
- **-AutoAssignUser** - If you use **-PersonalUnmanaged**, specifies that new users connecting through the RD Connection Broker are automatically assigned to an unassigned session host. If there are no unassigned session hosts in the collection, the user will see an error message. If you don't use this parameter, you have to manually assign users to a session host before they sign in.

Manually assign a user to a personal session host

Use the **Set-RDPersonalSessionDesktopAssignment** cmdlet to manually assign a user to a personal session host server in the collection. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-User <string>

-Name <string>

- **–CollectionName** - specifies the name of the personal session desktop collection. This parameter is required.
- **–ConnectionBroker** - specifies the Remote Desktop Connection Broker (RD Connection Broker) server for your Remote Desktop deployment. If you don't supply a value, the cmdlet uses the fully qualified domain name (FQDN) of the local computer.
- **–User** - specifies the user account to associate with the personal session desktop, in DOMAIN\User format. This parameter is required.
- **–Name** - specifies the name of the session host server. This parameter is required. The session host identified here must be a member of the collection that the **-CollectionName** parameter specifies.

The **Import-RDPersonalSessionDesktopAssignment** cmdlet imports associations between user accounts and personal session desktops from a text file. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Path <string>

–Path specifies the path and file name of a file to import.

Removing a User Assignment from a Personal Session Host

Use the **Remove-RDPersonalSessionDesktopAssignment** cmdlet to remove the association between a personal session desktop and a user. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Force

-Name <string>

-User <string>

–Force forces the command to run without asking for user confirmation.

Query user assignments

Use the **Get-RDPersonalSessionDesktopAssignment** cmdlet to get a list of personal session desktops and associated user accounts. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-User <string>

-Name <string>

You can run the cmdlet to query by collection name, user name, or by session desktop name. If you specify only the **–CollectionName** parameter, the cmdlet returns a list of session hosts and associated users. If you also specify the **–User** parameter, the session host associated with that user is returned. If you provide the **–Name** parameter, the user associated with that session host is returned.

The **Export-RDPersonalPersonalDesktopAssignment** cmdlet exports the current associations between users and personal virtual desktops to a text file. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Path <string>

All new cmdlets support the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Feedback

Was this page helpful? Yes No

Prepare your virtual machines for Remote Desktop

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can install Remote Desktop Services components on physical servers or on virtual machines.

The first step is to create Windows Server virtual machines in Azure. You'll want to create three VMs: one for the RD Session Host, one for the Connection Broker, and one for the RD Web and RD Gateway. To ensure the availability of your RDS deployment, create an availability set (under **High availablility** in the VM creation process) and group multiple VMs in that availability set.

After you create your VMs, use the following steps to prepare them for RDS.

1. Connect to the virtual machine using the Remote Desktop Connection (RDC) client:
   a. In the Azure portal open the Resource groups view, and then click the resource group to use for the deployment.
   b. Select the new RDSH virtual machine (for example, Contoso-Sh1).
   c. Click **Connect** > **Open** to open the Remote Desktop client.
   d. In the client, click **Connect**, and then click **Use another user account**. Enter the user name and password for the local administrator account.
   e. Click **Yes** when warned about the certificate.
2. Enable remote management:
   a. In Server Manager, click **Local Server** > **Remote management current setting (disabled)**.
   b. Select **Enable remote management for this server**.
   c. Click **OK**.
3. Optional: You can temporarily set Windows Update to not automatically download and install updates. This helps prevent changes and system restarts while you deploy the RDSH server.
   a. In Server Manager, click **Local Server** > **Windows Update current setting**.
   b. Select **Advanced options** > **Defer upgrades**.
4. Add the server to the domain:
   a. In Server Manager, click **Local Server** > **Workgroup current setting**.
   b. Click **Change** > **Domain**, and then enter the domain name (for example, Contoso.com).
   c. Enter the domain administrator credentials.

d. Restart the virtual machine.

5. Repeat steps 1 through 4 for the RD Web and GW virtual machine.
6. Repeat steps 1 through 4 for the RD Connection Broker virtual machine.
7. Initialize and format the attached disk on the RD Connection Broker virtual machine:
   a. Connect to the RD Connection Broker virtual machine (step 1 above).
   b. In Server Manager, click **Tools** > **Computer Management**.
   c. Click **Disk Management**.
   d. Select the attached disk, then **MBR (Master Boot Record)**, and then click **OK**.
   e. Right-click the new disk (marked as **Unallocated**) and click **New Simple Volume**.
   f. In the **New Simple Volume** wizard, accept the default values but provide a applicable name for the **Volume label** (like Shares).
8. On the RD Connection Broker virtual machine create file shares for the user profile disks and certificates:
   a. Open File Explorer, click **This PC**, and open the disk that you added for file shares.
   b. Click **Home** and **New Folder**.
   c. Enter a name for the user disks folder, for example, **UserDisks**.
   d. Right-click the new folder and click **Properties** > **Sharing** > **Advanced Sharing**.
   e. Select **Share this folder** and click **Permissions**.
   f. Select **Everyone**, and then click **Remove**. Now click **Add**, enter **Domain Admins**, and click **OK**.
   g. Select **Allow Full Control**, and then click **OK** > **OK** > **Close**.
   h. Repeat steps c. to g. to create a shared folder for certificates.

Feedback

Configure disaster recovery for Remote Desktop Services

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When you deploy Remote Desktop Services into your environment, it becomes a critical part of your infrastructure, particularly the apps and resources that you share with users. If the RDS deployment goes down due to anything from a network failure to a natural disaster, users can't access those apps and resources, and your business is negatively impacted. To avoid this, you can configure a disaster recovery solution that allows you to failover your deployment - if your RDS deployment is unavailable, for whatever reason, there is a backup available to automatically take over.

To keep your RDS deployment running in the case of a single component or machine going down, we recommend configuring your RDS deployment for high availability. You can do this by setting up an RDSH farm and ensuring your Connection Brokers are clustered for high availability.

The disaster recovery solutions we recommend here are to protect your deployment from catastrophic disaster - something that takes down your entire RDS deployment (including redundant roles configured for high availability). If such a disaster hits, having a disaster recovery solution built into your deployment will allow you to failover the entire deployment and quickly get apps and resources up and running for your users.

Use the following information to deploy disaster recovery solutions in RDS:

- Leverage multiple Azure data centers to ensure users can access your RDS deployment, even if one Azure data center goes down (geo-redundancy)
- Deploy Azure Site Recovery to provide failover for RDS components in site-to-site or site-to-Azure failovers

---

Feedback

Create a geo-redundant, multi-data center RDS deployment for disaster recovery

Article • 07/03/2024 •

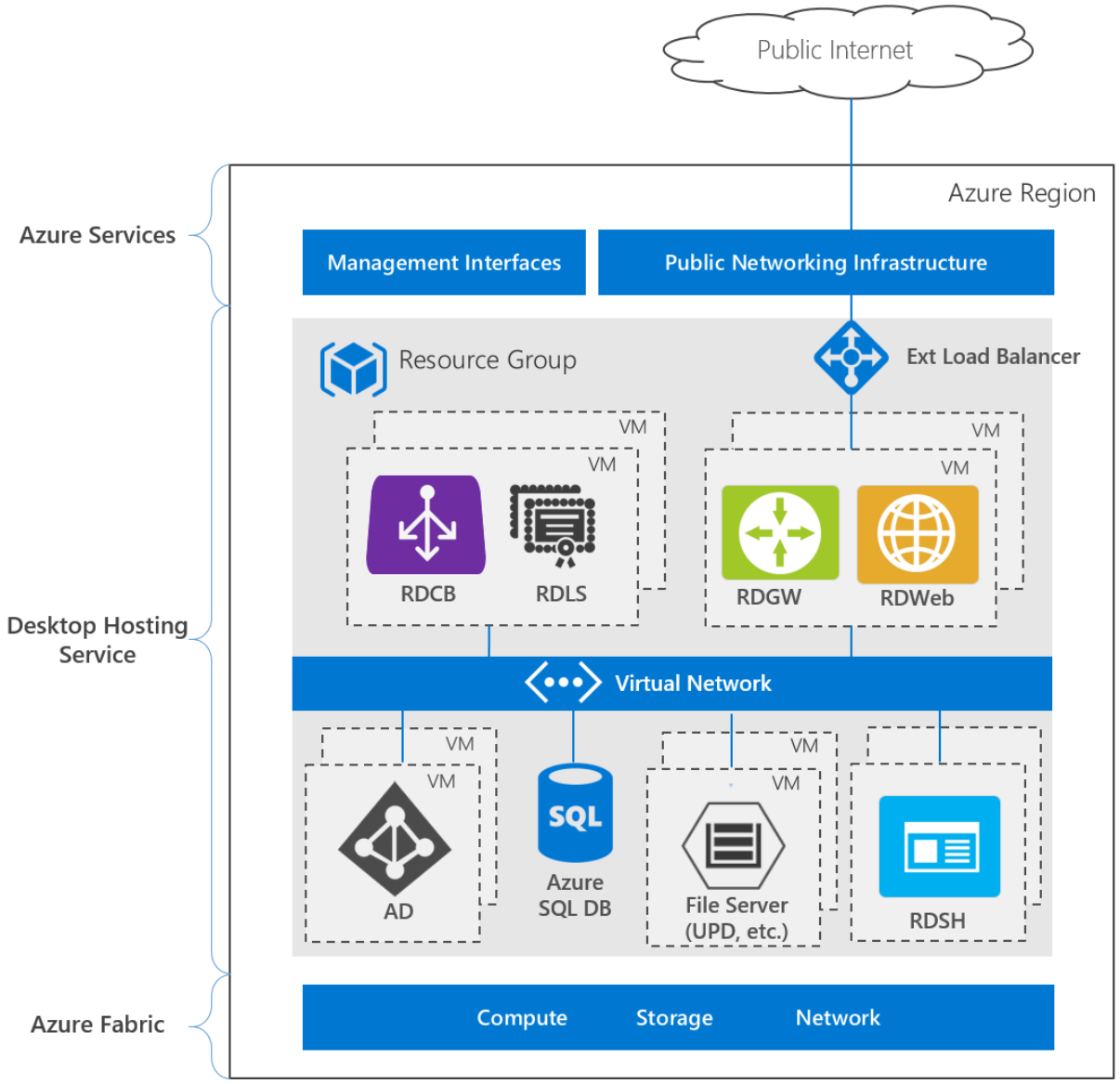
Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can enable disaster recovery for your Remote Desktop Services deployment by leveraging multiple data centers in Azure. Unlike a standard highly available RDS deployment (as outlined in the Remote Desktop Services architecture), which uses data centers in a single Azure region (for example, Western Europe), a multi-data center deployment uses data centers in multiple geographic locations, increasing the availability of your deployment - one Azure data center might be unavailable, but it is unlikely that multiple regions would go down at the same time. By deploying a geo-redundant RDS architecture, you can enable failover in the case of catastrophic failure of an entire region.

You can use the instructions below to leverage Microsoft Azure infrastructure services and RDS to deliver geo-redundant desktop hosting services and Subscriber Access Licenses (SALs) to multiple tenants through the Microsoft Service Provider License Agreement (SPLA) program. You can also use the steps below to create a geo-redundant hosting service for your own employees using RDS User CALs extended rights through Software Assurance.

Logical architecture for high availability - single and multiple regions

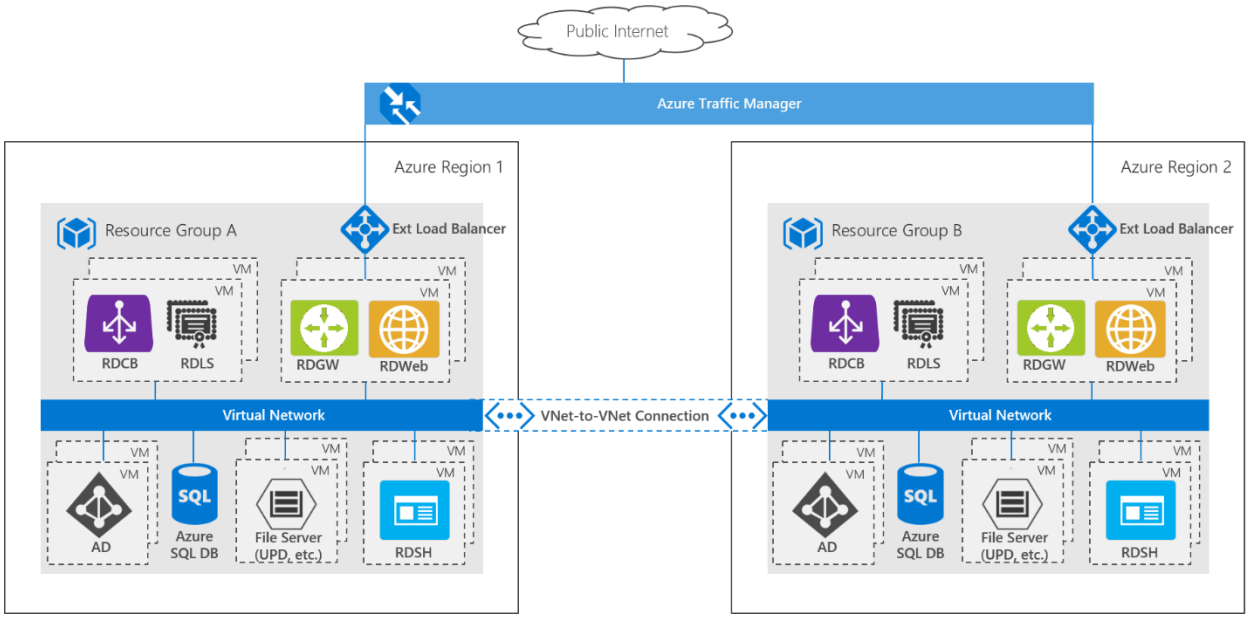
The following image shows the architecture for a highly available deployment in a single Azure region:



The deployment consists of three layers:

- Azure services - the Azure Management interfaces, including the Azure portal and APIs, and public networking services, such as DNS and public IP addressing.
- Desktop hosting service - Virtual machines, networks, storage, Azure services, and Windows Server role services
- Azure Fabric - Windows Server operating systems running the Hyper-V role, used to virtualize physical servers, storage units, network switches, and routers. Using Azure Fabric lets you create VMs, networks, storage, and applications independent from underlying hardware.

In comparison, here is the architecture for a deployment that uses multiple Azure data centers:



The entire RDS deployment is replicated in a second Azure region to create a geo-redundant deployment. This architecture uses an active-passive model, where only one RDS deployment is running at a time. A VNet-to-VNet connection lets the two environments communicate with each other. The RDS deployments are based on a single Active Directory forest/domain, and the AD servers replicate across the two deployments, meaning users can sign into either deployment using the same credentials. User settings and data stored in User Profile Disks (UPD) are stored on a two-node cluster Storage Spaces Direct scale-out file server (SOFS). A second identical Storage Spaces Direct cluster is deployed in the second (passive) region, and Storage Replica is used to replicate the user profiles from the active to passive deployment. Azure Traffic Manager is used to automatically direct end users to whichever deployment is currently active - from the end user perspective, they access the deployment using a single URL and are not aware of which region they end up using.

You *could* create a non-highly available RDS deployment in each region, but if even a single VM is restarted in one region, a failover would occur, increasing the likelihood of failovers occurring with associated performance impacts.

Deployment steps

Create the following resources in Azure to create a geo-redundant multi-data center RDS deployment:

1. Two resource groups in two separate Azure regions. For example RG A (the active deployment, RG stands for "resource group") and RG B (the passive deployment).

2. A highly-available Active Directory deployment in RG A. You can use the New AD Domain with 2 Domain Controllers template to create the deployment.

3. A highly-available RDS deployment in RG A. Use the RDS farm deployment using existing active directory template to create the basic RDS deployment, and then follow the information in Remote Desktop Services - High availability to configure the other RDS components for high availability.

4. A VNet in RG B - make sure to use an address space that does not overlap the deployment in RG A.

5. A VNet-to-VNet connection between the two resource groups.

6. Two AD virtual machines in an availability set in RG B - make sure the VM names are different from the AD VMs in RG A. Deploy two Windows Server 2016 VMs in a single availability set, install the Active Directory Domain Services role, and then promote them to the domain controller in the domain you created in step 1.

7. A second highly-available RDS deployment in RG B.
   a. Use the RDS farm deployment using existing active directory template again, but this time make the following changes. (To customize the template, select it in the gallery, click **Deploy to Azure** and then **Edit template**.)

      i. Adjust the address space of the DNS server private IP to correspond to the VNet in RG B.

      Search for "dnsServerPrivateIp" in variables. Edit the default IP (10.0.0.4) to correspond to the address space you defined in the VNet in RG B.

      ii. Edit the computer names so that they don't collide with those in the deployment in RG A.

      Locate the VMs in the **Resources** section of the template. Change the **computerName** field under **osProfile**. For example, "gateway" can become"gateway-**b**"; "[concat('rdsh-', copyIndex())]" can become "[concat('rdsh-b-', copyIndex())]", and "broker" can become "broker-**b**".

      (You can also change the names of the VMs manually after you run the template.)

   b. As in step 3 above, use the information in Remote Desktop Services - High availability to configure the other RDS components for high availability.

8. A Storage Spaces Direct scale-out file server with Storage Replica across the two deployments. Use the PowerShell script to deploy the template across the resource groups.

> **Note**

You can provision storage manually (instead of using the PowerShell script and template):

a. Deploy a two-node Storage Spaces Direct SOFS in RG A to store your user profile disks (UPDs).
b. Deploy a second, identical Storage Spaces Direct SOFS in RG B - make sure to use the same amount of storage in each cluster.
c. Set up Storage Replica with asynchronous replication between the two.

Enable UPDs

Storage Replica replicates data from a source volume (associated with the primary/active deployment) to a destination volume (associated with the secondary/passive deployment). By design, the destination cluster appears as **Online (No Access)** - Storage Replica dismounts the destination volumes and their drive letters or mount points. This means that enabling UPDs for the secondary deployment by providing the file share path will fail, because the volume is not mounted.

Want to learn more about managing replication? Check out Cluster to cluster Storage Replication.

To enable UPDs on both deployments, do the following:

1. Run the Set-RDSessionCollectionConfiguration cmdlet to enable the user profile disks for the primary (active) deployment - provide a path to the file share on the source volume (which you created in Step 7 in the deployment steps).

2. Reverse the Storage Replica direction so that the destination volume becomes the source volume (this mounts the volume and makes it accessible by the secondary deployment). You can run **Set-SRPartnership** cmdlet to do this. For example:

   PowerShell

   ```
   Set-SRPartnership -NewSourceComputerName "cluster-b-s2d-c" -
   SourceRGName "cluster-b-s2d-c" -DestinationComputerName "cluster-a-s2d-
   c" -DestinationRGName "cluster-a-s2d-c"
   ```

3. Enable the user profile disks in the secondary (passive) deployment. Use the same steps as you did for the primary deployment, in step 1.

4. Reverse the Storage Replica direction again, so the original source volume is again the source volume in the SR Partnership, and the primary deployment can access the file share. For example:

PowerShell

```
Set-SRPartnership -NewSourceComputerName "cluster-a-s2d-c" -
SourceRGName "cluster-a-s2d-c" -DestinationComputerName "cluster-b-s2d-
c" -DestinationRGName "cluster-b-s2d-c"
```

Azure Traffic Manager

Create an Azure Traffic Manager profile, and make sure to select the **Priority** routing method. Set the two endpoints to the public IP addresses of each deployment. Under **Configuration**, change the protocol to HTTPS (instead of HTTP) and the port to 443 (instead of 80). Take note of the **DNS time to live**, and set it appropriately for your failover needs.

Note that Traffic Manager requires endpoints to return 200 OK in response to a GET request in order to be marked as "healthy." The publicIP object created from the RDS templates will function, but do not add a path addendum. Instead, you can give end users the Traffic Manager URL with "/RDWeb" appended, for example: `http://deployment.trafficmanager.net/RDWeb`

By deploying Azure Traffic Manager with the Priority routing method, you prevent end users from accessing the passive deployment while the active deployment is functional. If end users access the passive deployment and the Storage Replica direction hasn't been switched for failover, the user sign-in hangs as the deployment tries and fails to access the file share on the passive Storage Spaces Direct cluster - eventually the deployment will give up and give the user a temporary profile.

Deallocate VMs to save resources

After you configure both deployments, you can optionally shut down and deallocate the secondary RDS infrastructure and RDSH VMs to save cost on these VMs. The Storage Spaces Direct SOFS and AD server VMs must always stay running in the secondary/passive deployment to enable user account and profile synchronization.

When a failover occurs, you'll need to start the deallocated VMs. This deployment configuration has the advantage of being lower cost, but at the expense of fail-over time. If a catastrophic failure occurs in the active deployment, you'll have to manually start the passive deployment, or you'll need an automation script to detect the failure and start the passive deployment automatically. In either case, it may take several minutes to get the passive deployment running and available for users to sign in, resulting in some downtime for the service. This downtime depends on the amount of time it takes to start the RDS infrastructure and RDSH VMs (typically 2-4 minutes, if the

VMs are started in parallel rather than serially), and the time to bring the passive cluster online (which depends on the size of the cluster, typically 2-4 minutes for a 2-node cluster with 2 disks per node).

Active Directory

The Active Directory servers in each deployment are replicas within the same Forest/Domain. Active Directory has a built-in synchronization protocol to keep the four domain controllers in sync. However, there may be some lag so that if a new user is added to one AD server, it may take some time to replicate across all the AD servers in the two deployments. Consequently, be sure to warn users to not try to sign in immediately after being added to the domain.

RD License Server

Provide a per-user RD CAL for each named user that is authorized to access the geo-redundant deployment. Distribute the per user CALs evenly across the two RD License Servers in the active deployment. Then, duplicate these CALs to the two RD License Servers in the passive deployment. Because the CALs are duplicated between the active and passive deployment, at any given time only one deployment can be active with users connecting; otherwise, you violate the license agreement.

Image Management

As you update your RDSH images to provide software updates or new applications, you'll need to separately update the RDSH collections in each deployment to maintain a common user experience across both deployments. You can use the Update RDSH collection template, but note that the passive deployment's RDS infrastructure and RDSH VMs must be running to run the template.

Failover

In the case of the Active-Passive deployment, failover requires you to start the VMs of the secondary deployment. You can do this manually or with an automation script. In the case of a catastrophic failover of the Storage Spaces Direct SOFS, change the Storage Replica partnership direction, so that the destination volume becomes the source volume. For example:

```
PowerShell
```

```
Set-SRPartnership -NewSourceComputerName "cluster-b-s2d-c" -SourceRGName
"cluster-b-s2d-c" -DestinationComputerName "cluster-a-s2d-c" -
DestinationRGName "cluster-a-s2d-c"
```

You can learn more in Cluster to cluster Storage Replication.

Azure Traffic Manager automatically recognizes that the primary deployment failed and that the secondary deployment is healthy (in the RD Gateway VMs have been started in RG B) and directs user traffic to the secondary deployment. Users can use the same Traffic Manager URL to continue working on their remote resources, enjoying a consistent experience. Note that the client DNS cache will not update the record for the duration of the TTL set in Azure Traffic Manager configuration.

Test failover

In a Storage Replica partnership, only one volume (the source) can be active at a time. This means when you switch the SR Partnership direction, the volume in the primary deployment (RG A) becomes the destination of replication and is therefore hidden. Thus, any users connecting to RG A will no longer have access to their UPDs stored on the SOFS in RG A.

To test the failover while allowing users to continue logging in:

1. Start the infrastructure VMs and RDSH VMs in RG B.
2. Switch the SR Partnership direction (cluster-b-s2d-c becomes the source volume).
3. Disable the endpoint of RG A in the Azure Traffic Manager profile to force the ATM to direct traffic to RG B. Alternatively, use a PowerShell script:

   PowerShell

   ```
   Disable-AzureRmTrafficManagerEndpoint -Name publicIpA -Type
   AzureEndpoints -ProfileName MyTrafficManagerProfile -ResourceGroupName
   RGA -Force
   ```

RG B is now the active primary deployment. To switch back to RG A as the primary deployment:

1. Switch the SR Partnership direction (cluster-a-s2d-c becomes the source volume):

   PowerShell

```
Set-SRPartnership -NewSourceComputerName "cluster-a-s2d-c" -
SourceRGName "cluster-a-s2d-c" -DestinationComputerName "cluster-b-s2d-
c" -DestinationRGName "cluster-b-s2d-c"
```

2. Re-enable the endpoint of RG A in the Azure Traffic Manager profile:

PowerShell

```
Enable-AzureRmTrafficManagerEndpoint -Name publicIpA -Type
AzureEndpoints -ProfileName MyTrafficManagerProfile -ResourceGroupName
RGA
```

Considerations for on-premises deployments

While an on-premises deployment couldn't use the Azure Quickstart Templates referenced in this article, you can implement all the infrastructure roles manually. In an on-premises deployment where cost is not driven by Azure consumption, consider using an active-active model for quicker failover.

You can use Azure Traffic Manager with on-premises endpoints, but it requires an Azure subscription. Alternatively, for the DNS provided to end users, give them a CNAME record that simply directs users to the primary deployment. In the case of failover, modify the DNS CNAME record to redirect to the secondary deployment. In this way, the end user uses a single URL, just like with Azure Traffic Manager, that directs the user to the appropriate deployment.

If you are interested in creating an on-premises-to-Azure-site model, consider using Azure Site Recovery.

Feedback

Set up disaster recovery for RDS using Azure Site Recovery

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use Azure Site Recovery to create a disaster recovery solution for your Remote Desktop Services deployment.

Azure Site Recovery is an Azure-based service that provides disaster recovery capabilities by orchestrating replication, failover, and recovery of virtual machines. Azure Site Recovery supports a number of replication technologies to consistently replicate, protect, and seamlessly failover virtual machines and applications to private/public or hoster's clouds.

Use the following information to create and validate the disaster recovery solution.

Disaster recovery deployment options

You can deploy RDS on either physical servers or virtual machines running Hyper-V or VMWare. Azure Site Recovery can protect both on-premises and virtual deployments to either a secondary site or to Azure. The following table shows the different supported RDS deployments in site-to-site and site-to-Azure disaster recvoery scenarios.

Expand table

Deployment type	Hyper-V site-to-site	Hyper-V site-to-Azure	VMWare site-to-Azure	Physical site-to-Azure
Pooled virtual desktop (unmanaged)	Yes	No	No	No
Pooled virtual desktop (managed, no UPD)	Yes	No	No	No
RemoteApps and desktop sessions (no UPD)	Yes	Yes	Yes	Yes

Prerequisites

Before you can configure Azure Site Recovery for your deployment, make sure you meet the following requirements:

- Create an on-premises RDS deployment.
- Add Azure Site Recovery Services vault to your Microsoft Azure subscription.
- If you are going to use Azure as your recovery site, run the Azure Virtual Machine Readiness Assessment tool on your VMs to ensure they are compatible with Azure VMs and Azure Site Recovery Services.

Implementation checklist

We'll cover the various steps to enable Azure Site Recovery Services for your RDS deployment in more detail, but here are the high-level implementation steps.

Expand table

Step 1 - Configure VMs for disaster recovery
Hyper-V - Download the Microsoft Azure Site Recovery Provider. Install it on your VMM server or Hyper-V host. See Prerequisites for replication to Azure by using Azure Site Recovery for information.
VMWare - Configure protection server, configuration server, and target servers
Step 2 - Prepare your resources
Add an Azure Storage account.
Hyper-V - Download the Microsoft Azure Recovery Services agent and install it on Hyper-V host servers.
VMWare - Make sure the mobility service is installed on all VMs.
Enable protection for VMs in VMM cloud, Hyper-V sites, or VMWare sites.
Step 3 - Design your recovery plan.
Map your resources - map on-premises networks to Azure VNETs.
Create the recovery plan.
Test the recovery plan by creating a test failover. Ensure all VMs can access required resources, like Active Directory. Ensure network redirections are configured and working for RDS. For detailed steps on testing your recovery plan, see Run a test failover
Step 4 - Run a disaster recovery drill.
Run a disaster recovery drill using planned and unplanned failovers. Ensure that all VMs have access to required resources, such as Active Directory. Ensure that all VMs have access to required

Step 1 - Configure VMs for disaster recovery
resources, such as Active Directory. For detailed steps on failovers and how to run drills, see Failover in Site Recovery.

Feedback

Was this page helpful? Yes No

Enable disaster recovery of RDS using Azure Site Recovery

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

To ensure that your RDS deployment is adequately configured for disaster recovery, you need to protect all of the components that make up your RDS deployment:

- Active Directory
- SQL Server tier
- RDS components
- Network components

Configure Active Directory and DNS replication

You need Active Directory on the disaster recovery site for your RDS deployment to work. You have two choices based on how complex your RDS deployment is:

- Option 1 - If you have a small number of applications and a single domain controller for your entire on-premises site, and you will be failing over the entire site together, use ASR-Replication to replicate the domain controller to the secondary site (true for both site-to-site and site-to-Azure scenarios).
- Option 2 - If you have a large number of applications and you're running an Active Directory forest, and you'll failover a few applications at a time, set up an additional domain controller on the disaster recovery site (either a secondary site or in Azure).

See Protect Active Directory and DNS with Azure Site Recovery for details on making a domain controller available on the disaster recovery site. For the rest of this guidance, we assume that you've followed those steps and have the domain controller available.

Set up SQL Server replication

See Protect SQL Server using SQL Server disaster recovery and Azure Site Recovery for the steps to set up SQL Server replication.

Enable protection for the RDS application components

Depending on your RDS deployment type you can enable protection for different component VMs (as listed in the table below) in Azure Site Recovery. Configure the relevant Azure Site Recovery elements based on whether your VMs are deployed on Hyper-V or VMWare.

Expand table

Deployment type	Protection steps
Personal virtual desktop (unmanaged)	1. Make sure all virtualization hosts are ready with the RDVH role installed. 2. Connection Broker. 3. Personal desktops. 4. Gold template VM. 5. Web Access, License server, and Gateway server
Pooled virtual desktop (managed with no UPD)	1. All virtualization hosts are ready with the RDVH role installed. 2. Connection Broker. 3. Gold template VM. 4. Web Access, License server, and Gateway server.
RemoteApps and Desktop Sessions (no UPD)	1. Session Hosts. 2. Connection Broker. 3. Web Access, License server, and Gateway server.

Feedback

Was this page helpful? Yes No

Create your disaster recovery plan for RDS

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can create a disaster recovery plan in Azure Site Recovery to automate the failover process. Add all RDS component VMs to the recovery plan.

Use the following steps in Azure to create your recovery plan:

1. Open Azure Site Recovery Vault in the Azure portal, and then click **Recovery Plans**.
2. Click **Create** and enter a name for the plan.
3. Select your **Source** and **Target**. The target is either a secondary RDS site or Azure.
4. Select the VMs that host your RDS components, and then click **OK**.

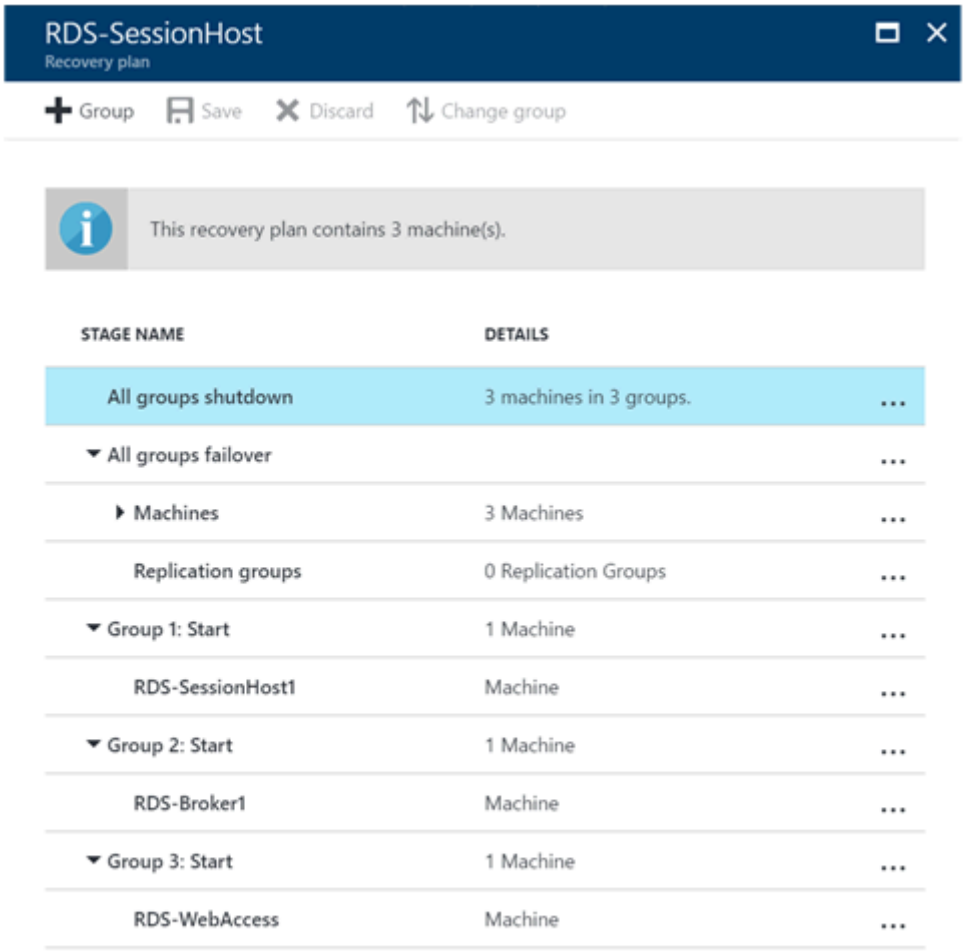
The following sections provide additional information about creating recovery plans for the different types of RDS deployment.

Sessions-based RDS deployment

For an RDS sessions-based deployment, group the VMs so they come up in sequence:

1. Failover group 1 - Session Host VM
2. Failover group 2 - Connection Broker VM
3. Failover group 3 - Web Access VM

Your plan will look something like this:



Pooled desktops RDS deployment

For an RDS deployment with pooled desktops, group the VMs so they come up in sequence, adding manual steps and scripts.

1. Failover group 1 - RDS Connection Broker VM

2. Group 1 manual action - Update DNS

   Run PowerShell in an elevated mode on the Connection Broker VM. Run the following command and wait for a couple of minutes to ensure the DNS is updated with the new value:

   ```
   ipconfig /registerdns
   ```

3. Group 1 script - add Virtualization hosts

   Modify the script below to run for each virtualization host in the cloud. Typically after you add a virtualization host to a Connection Broker, you need to restart the

host. Ensure that the host doesn't have a reboot pending before the script runs, or else it will fail.

```
Broker - broker.contoso.com
Virtualization host - VH1.contoso.com

ipmo RemoteDesktop;
add-rdserver –ConnectionBroker broker.contoso.com –Role RDS-
VIRTUALIZATION –Server VH1.contoso.com
```

4. Failover group 2 - Template VM

5. Group 2 script 1 - Turn off Template VM

   The template VM when recovered to the secondary site will start, but it is a sysprepped VM and cannot start completely. Also RDS requires that the VM be shutdown to create a pooled VM configuration from it. So, we need to turn it off. If you have a single VMM server, the template VM name is the same on the primary and the secondary. Because of that, we use the VM ID as specified by the *Context* variable in the script below. If you have multiple templates, turn them all off.

   PowerShell

   ```
   ipmo virtualmachinemanager;
   Foreach($vm in $VMsAsTemplate)
   {
      Get-SCVirtualMachine -ID $vm | Stop-SCVirtualMachine –Force
   }
   ```

6. Group 2 script 2 - Remove existing pooled VMs

   You need to remove the pooled VMs on the primary site from the Connection Broker so new VMs can be created on the secondary site. In this case you need to specify the exact host on which to create the pooled VM. Note that this will delete the VMs from only the collection.

   PowerShell

   ```
   ipmo RemoteDesktop
   $desktops = Get-RDVirtualDesktop -CollectionName Win8Desktops;
   Foreach($vm in $desktops){
      Remove-RDVirtualDesktopFromCollection -CollectionName Win8Desktops -
   VirtualDesktopName $vm.VirtualDesktopName –Force
   }
   ```

7. Group 2 manual action - Assign new template

   You need to assign the new template to the Connection Broker for the collection so you can create new pooled VMs on the recovery site. Go to the RDS Connection Broker and identify the collection. Edit the properties and specify a new VM image as its template.

8. Group 2 script 3 - Recreate all pooled VMs

   Recreate the pooled VMs on the recovery site through the Connection Broker. In this case, you need to specify the exact host on which to create the pooled VM.

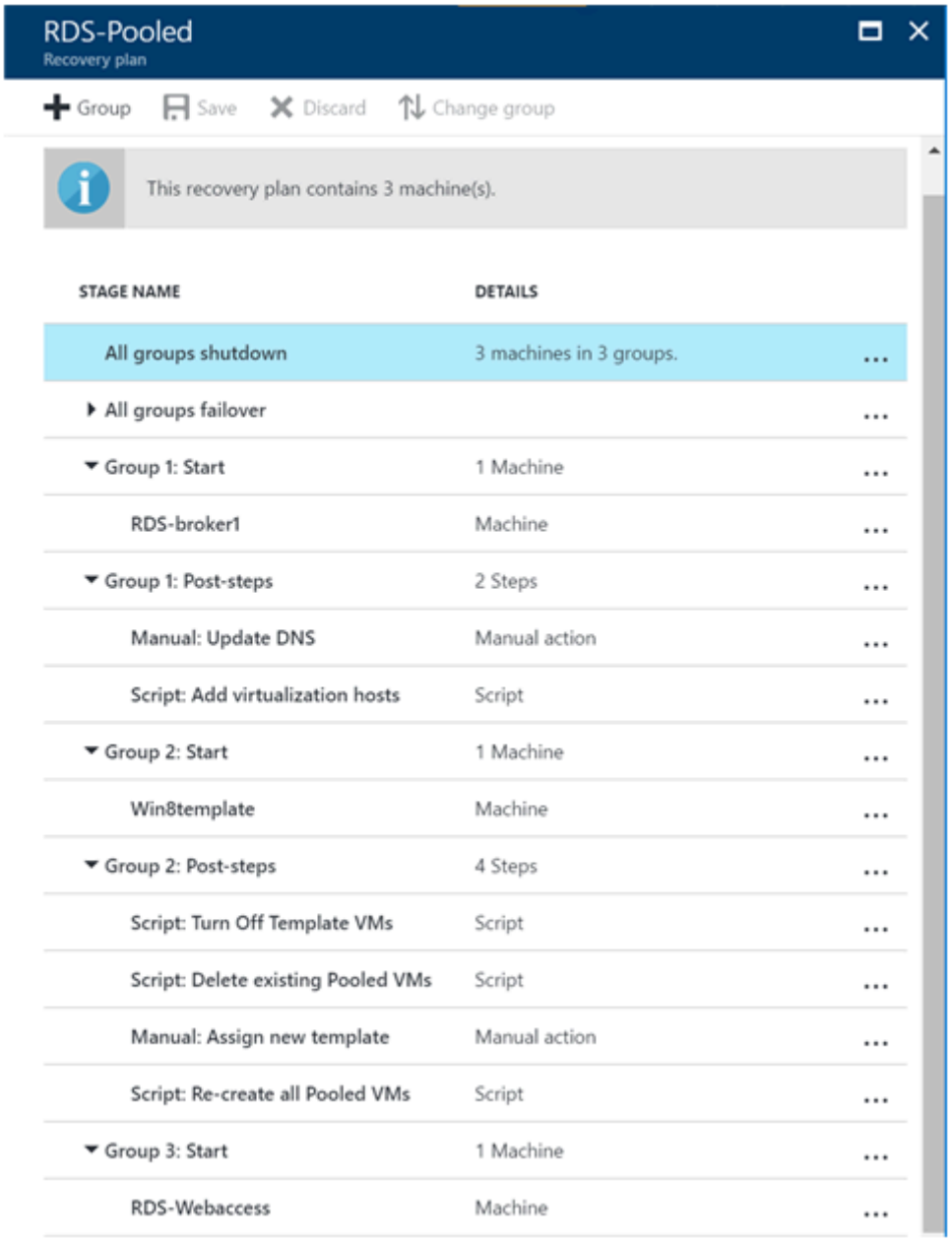
   The pooled VM name needs to be unique, using the prefix and suffix. If the VM name already exists, the script will fail. Also, if the primary side VMs are numbered from 1-5, the recovery site numbering will continue from 6.

   PowerShell

   ```
   ipmo RemoteDesktop;
   Add-RDVirtualDesktopToCollection -CollectionName Win8Desktops -
   VirtualDesktopAllocation @{"RDVH1.contoso.com" = 1}
   ```

9. Failover group 3 - Web Access and Gateway server VM

The recovery plan will look like this:



Personal desktops RDS deployment

For an RDS deployment with personal desktops, group the VMs so they come up in sequence, adding manual steps and scripts.

1. Failover group 1 - RDS Connection Broker VM

2. Group 1 manual action - Update DNS

   Run PowerShell in an elevated mode on the Connection Broker VM. Run the following command and wait for a couple of minutes to ensure the DNS is updated with the new value:

   ```
   ipconfig /registerdns
   ```

3. Group 1 script - Add Virtualization hosts

   Modify the script below to run for each virtualization host in the cloud. Typically after you add a virtualization host to a Connection Broker, you need to restart the host. Ensure that the host doesn't have a reboot pending before the script runs, or else it will fail.

```
PowerShell

Broker - broker.contoso.com
Virtualization host - VH1.contoso.com

ipmo RemoteDesktop;
add-rdserver –ConnectionBroker broker.contoso.com –Role RDS-
VIRTUALIZATION –Server VH1.contoso.com
```

4. Failover group 2 - Template VM

5. Group 2 script 1 - Turn off template VM

   The template VM when recovered to the secondary site will start, but it is a sysprepped VM and cannot start completely. Also RDS requires that the VM be shutdown to create a pooled VM configuration from it. So, we need to turn it off. If you have a single VMM server, the template VM name is the same on the primary and the secondary. Because of that, we use the VM ID as specified by the *Context* variable in the script below. If you have multiple templates, turn them all off.

```
PowerShell

ipmo virtualmachinemanager;
Foreach($vm in $VMsAsTemplate)
{
   Get-SCVirtualMachine -ID $vm | Stop-SCVirtualMachine –Force
}
```

6. Failover group 3 - Personal VMs

7. Group 3 script 1 - Remove existing personal VMs and add them

   Remove the personal VMs on the primary site from the Connection Broker so new VMs can be created on the secondary site. You need to extract the VMs' assignments and re-add the virtual machines to the Connection Broker with the hash of assignments. This will only remove the personal VMs from the collection and re-add them. The personal desktop allocation will be exported and imported back into the collection.

PowerShell

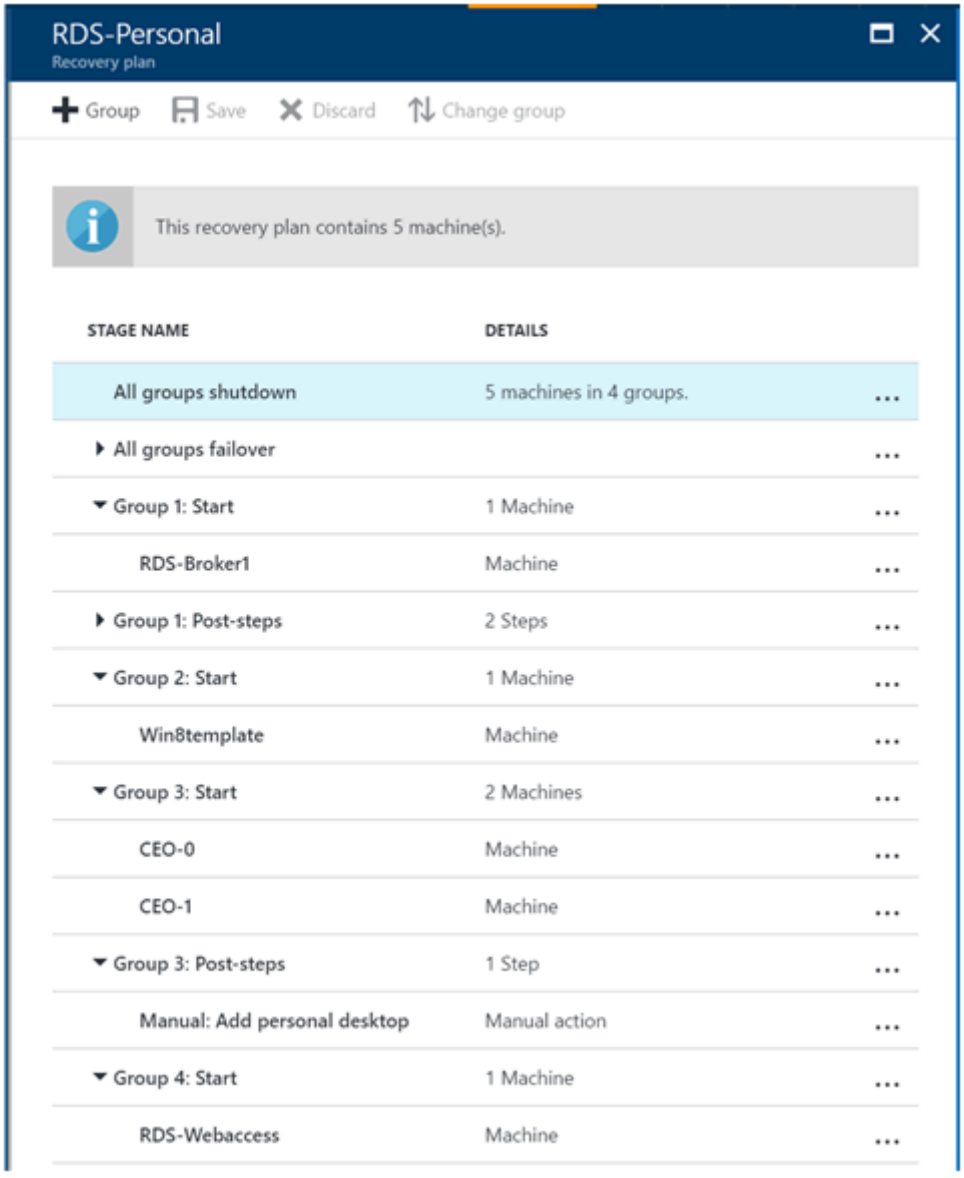
```
ipmo RemoteDesktop
$desktops = Get-RDVirtualDesktop -CollectionName CEODesktops;
Export-RDPersonalVirtualDesktopAssignment -CollectionName CEODesktops -
Path ./Desktopallocations.txt -ConnectionBroker broker.contoso.com

Foreach($vm in $desktops){
  Remove-RDVirtualDesktopFromCollection -CollectionName CEODesktops -
VirtualDesktopName $vm.VirtualDesktopName –Force
}

Import-RDPersonalVirtualDesktopAssignment -CollectionName CEODesktops -
Path ./Desktopallocations.txt -ConnectionBroker broker.contoso.com
```

8. Failover group 3 - Web Access and Gateway server VM

Your plan will look something like this:



Feedback

Was this page helpful? Yes No

Run and tune your Remote Desktop Services environment

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Tuning your deployment takes time and requires instrumentation and monitoring. Use the processes below to refine your Remote Desktop deployment, keep it running and enable scaling out (and in) as needed.

It's a good practice to continually assess the metrics and balance against running costs.

Management and monitoring

Check out Manage users in your RDS collection for information about how to manage access to your desktops and remote resources.

Use **Microsoft Operations Management Suite (OMS)** to monitor Remote Desktop deployments for potential bottlenecks and manage them using one of the following ways:

- **Server Manager**: Use the RD management tool that is built in to Windows Server to manage deployments with up to 500 concurrent remote end-users.
- **PowerShell**: Use the RD PowerShell module, also built into Windows Server, to manage deployments with up to 5000 concurrent remote end-users.

Scale: Bigger, better, faster

With visibility into the deployment, you can control scale with more precision. Easily add or remove Remote Desktop host servers based on scale needs.

Remote Desktop deployments that are built on Azure can make use of Azure services, like Azure SQL, to scale automatically on demand.

Automation: Script for success

Maintaining a running, highly scaled application involves repeating operations on a regular basis. Use Remote Desktop Services PowerShell cmdlets and WMI providers to develop scripts that can be run on multiple deployments when needed. Run Best

Practice Analyzer (BPA) rules for Remote Desktop Services on your deployments to tune your deployments.

Load testing: Avoid surprises

Load test the deployment with both stress tests and simulation of real-life usage. Vary the load size to avoid surprises! Ensure that responsiveness meets user requirements, and that the entire system is resilient. Create load tests with simulation tools, like LoginVSI, that check your deployment's ability to meet the users' needs.

Feedback

Manage your personal desktop session collections

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Use the following information to manage a personal desktop session collection in Remote Desktop Services.

Manually assign a user to a personal session host

Use the **Set-RDPersonalSessionDesktopAssignment** cmdlet to manually assign a user to a personal session host server in the collection. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-User <string>

-Name <string>

- **–CollectionName** - specifies the name of the personal session desktop collection. This parameter is required.
- **–ConnectionBroker** - specifies the Remote Desktop Connection Broker (RD Connection Broker) server for your Remote Desktop deployment. If you don't supply a value, the cmdlet uses the fully qualified domain name (FQDN) of the local computer.
- **–User** - specifies the user account to associate with the personal session desktop, in DOMAIN\User format. This parameter is required.
- **–Name** - specifies the name of the session host server. This parameter is required. The session host identified here must be a member of the collection that the **-CollectionName** parameter specifies.

The **Import-RDPersonalSessionDesktopAssignment** cmdlet imports associations between user accounts and personal session desktops from a text file. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Path <string>

–Path specifies the path and file name of a file to import.

Removing a User Assignment from a Personal Session Host

Use the **Remove-RDPersonalSessionDesktopAssignment** cmdlet to remove the association between a personal session desktop and a user. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Force

-Name <string>

-User <string>

–Force forces the command to run without asking for user confirmation.

Query user assignments

Use the **Get-RDPersonalSessionDesktopAssignment** cmdlet to get a list of personal session desktops and associated user accounts. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-User <string>

-Name <string>

You can run the cmdlet to query by collection name, user name, or by session desktop name. If you specify only the **–CollectionName** parameter, the cmdlet returns a list of session hosts and associated users. If you also specify the **–User** parameter, the session host associated with that user is returned. If you provide the **–Name** parameter, the user associated with that session host is returned.

The **Export-RDPersonalPersonalDesktopAssignment** cmdlet exports the current associations between users and personal virtual desktops to a text file. The cmdlet supports the following parameters:

-CollectionName <string>

-ConnectionBroker <string>

-Path <string>

All new cmdlets support the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Feedback

Remote Desktop IP Virtualization in Windows Server

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

As of Windows Server 2008 R2, Remote Desktop session hosts support per-session and per-program Remote Desktop IP Virtualization for Winsock applications. Remote Desktop assigns individual IP addresses to user sessions to avoid application compatibility issues that can happen when all Remote Desktop users in the same location share the same IP address. This article gives instructions for how to virtualize IP addresses for your organization's Remote Desktop users.

> **Note**
>
> This article's instructions for virtualizing IPs only apply to on-premises environments.

Prerequisites

In order to use IP Virtualization, your system must meet the following requirements:

- Your deployment must run Windows Server 2019 or later.
- You must assign the RD Session Host server role to the machine you use to make the changes.

How to configure IP Virtualization

You can configure IP Virtualization using the Microsoft Management Console (MMC), Group Policy, or running a command in a PowerShell window.

Microsoft Management Console

1. Open the RD Session Host Configuration MMC on the machine you the RD Session Host server role.
2. Go to **Edit settings**.

3. Select **IP Virtualization** and go to **Properties**.
4. Select the **Enable IP virtualization** check box.
5. In the **Select the network adapter to be used for IP Virtualization** field, select the network adapter you want to use for IP Virtualization from the drop-down menu.

   > **Note**
   >
   > IP virtualization currently only supports single-network adapter scenarios. If your server has multiple enabled network adapters, you can only use the adapter you specify in the settings for IP virtualization.

6. When you're finished, select **Apply**.
7. Optionally, to configure IP virtualization for specific programs:
   - Return to **Edit settings**, then select **Add program**.
   - Enter or navigate to the file path of the program you want to use.
   - Select **Open**.
   - Repeat for all programs you want to use.
   - When you're finished, select **Apply**.

Related content

- Remote Desktop Services Virtualization recommendations
- Scale out your Remote Desktop Services deployment by adding an RD Session Host farm
- Manage users in your RDS collection

Feedback

Manage users in your RDS collection

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

As an admin, you can directly manage which users have access to specific collections. This way, you can create one collection with standard applications for information workers, but then create a separate collection with graphics-intensive modeling applications for engineers. There are two primary steps to managing user access in a Remote Desktop Services (RDS) deployment:

1. Create users and groups in Active Directory
2. Assign users and groups to collections

Create your users and groups in Active Directory

In an RDS deployment, Active Directory Domain Services (AD DS) is the source of all users, groups, and other objects in the domain. You can manage Active Directory directly with PowerShell, or you can use built in UI tools that add ease and flexibility. The following steps will guide you to install those tools — if you do not have them already installed — and then use those tools to manage users and groups.

Install AD DS tools

The following steps detail how to install the AD DS tools on a server already running AD DS. Once installed, you can then create users or create groups.

1. Connect to the server running Active Directory Domain Services. For Azure deployments:
   a. In the Azure portal, click **Browse** > **Resource groups**, and then click the resource group for the deployment
   b. Select the AD virtual machine.
   c. Click **Connect** > **Open** to open the Remote Desktop client. If **Connect** is grayed out, the virtual machine might not have a public IP address. To give it one perform the following steps, then try this step again.
      i. Click **Settings** > **Network interfaces**, and then click the corresponding network interface.
      ii. Click **Settings** > **IP address**.

iii. For **Public IP address**, select **Enabled**, and then click **IP address**.
iv. If you have an existing public IP address you want to use, select it from the list. Otherwise, click **Create new**, enter a name, and then click **OK** and **Save**.

d. In the client, click **Connect**, and then click **Use another account**. Enter the user name and password for a domain administrator account.
e. Click **Yes** when asked about the certificate.

2. Install the AD DS tools:
   a. In Server Manager click **Manage** > **Add Roles and Features**.
   b. Click **Role-based or feature-based installation**, and then click the current AD server. Follow the steps until you get to the **Features** tab.
   c. Expand **Remote Server Administration Tools** > **Role Administration Tools** > **AD DS and AD LDS Tools**, and then select **AD DS Tools**.
   d. Select **Restart the destination server automatically if required**, and then click **Install**.

Create a group

You can use AD DS groups to grant access to a set of users that need to use the same remote resources.

1. In Server Manager on the server running AD DS, click **Tools** > **Active Directory Users and Computers**.
2. Expand the domain in the left-hand pane to view its subfolders.
3. Right-click the folder where you want to create the group, and then click **New** > **Group**.
4. Enter an appropriate group name, then select **Global** and **Security**.

Create a user and add to a group

1. In Server Manager on the server running AD DS, click **Tools** > **Active Directory Users and Computers**.
2. Expand the domain in the left-hand pane to view its subfolders.
3. Right-click **Users**, and then click **New** > **User**.
4. Enter, at minimum, a first name and a user logon name.
5. Enter and confirm a password for the user. Set appropriate user options, like **User must change password at next logon**.
6. Add the new user to a group:
   a. In the **Users** folder right-click the new user.
   b. Click **Add to a group**.
   c. Enter the name of the group to which you want to add the user.

Assign users and groups to collections

Now that you've created the users and groups in Active Directory, you can add some granularity regarding who has access to the Remote Desktop collections in your deployment.

1. Connect to the server running the Remote Desktop Connection Broker (RD Connection Broker) role, following the steps described earlier.

2. Add the other Remote Desktop servers to the RD Connection Broker's pool of managed servers:
   a. In Server Manager click **Manage** > **Add Servers**.
   b. Click **Find Now**.
   c. Click each server in your deployment that is running a Remote Desktop Services role, and then click **OK**.

3. Edit a collection to assign access to specific users or groups:
   a. In Server Manager click **Remote Desktop Services** > **Overview**, and then click a specific collection.
   b. Under **Properties**, click **Tasks** > **Edit properties**.
   c. Click **User groups**.
   d. Click **Add** and enter the user or group that you want to have access to the collection. You can also remove users and groups from this window by selecting the user or group you want to remove, and then clicking **Remove**.

   > **Note**
   >
   > The User groups window can never be empty. To narrow the scope of users who have access to the collection, you must first add specific users or groups before removing broader groups.

Feedback

Customize the RDS title “Work Resources” using PowerShell on Windows Server

Article • 07/03/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When using Windows Server to access RemoteApps or desktops through RD WebAccess or the new Remote Desktop app, you may have noticed that the workspace is titled “Work Resources" by default. You can easily change the title by using PowerShell cmdlets.

To change the title, open up a new PowerShell window on the connection broker server and import the RemoteDesktop module with the following command.

PowerShell

```
    Import-Module RemoteDesktop
```

Next, use the Set-RDWorkspace command to change the workspace name.

PowerShell

```
    Set-RDWorkspace [-Name] <string> [-ConnectionBroker <string>]
[<CommonParameters>]
```

For example, you can use the following command to change the workpsace name to "Contoso RemoteApps":

PowerShell

```
    Set-RDWorkspace -Name "Contoso RemoteApps" -ConnectionBroker
broker01.contoso.com
```

If you are running multiple Connection Brokers in High Availability mode, you must run this against the active broker. You can use this command:

PowerShell

```
    Set-RDWorkspace -Name "Contoso RemoteApps" -ConnectionBroker (Get-
```

```
RDConnectionBrokerHighAvailability).ActiveManagementServer
```

For more information about the Set-RDWorkspace cmdlet, see the Set-RDSWorkspace reference.

Feedback

Was this page helpful? Yes No

Use performance counters to diagnose app performance problems on Remote Desktop Session Hosts

Article • 07/03/2024 •

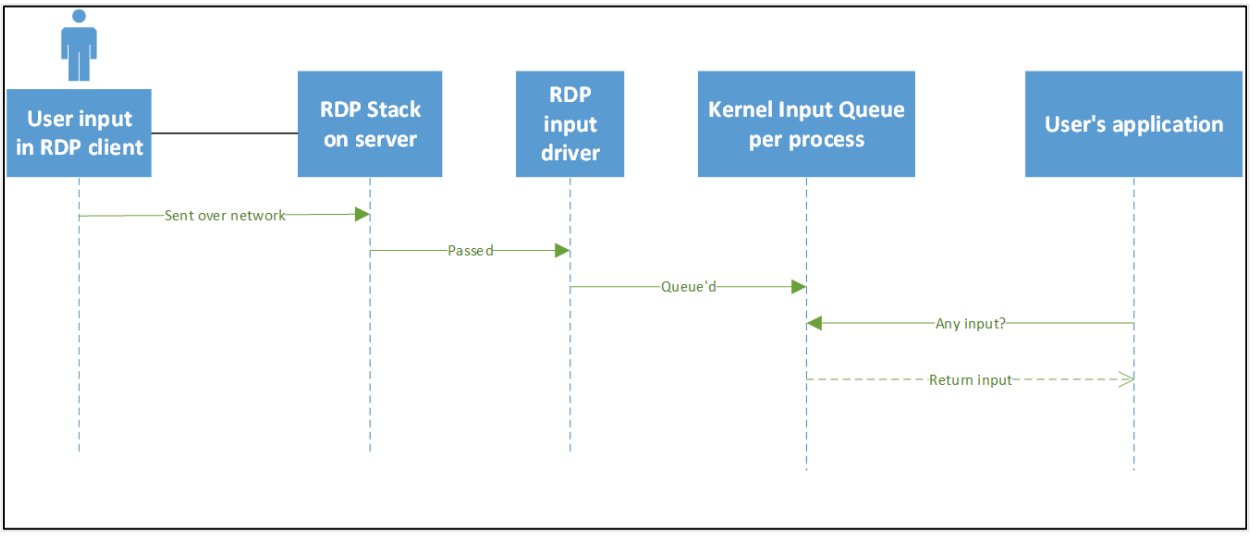
Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Poor application performance is one of the most difficult problems to diagnose, especially for slow or nonresponsive applications. Traditionally, you start your diagnosis by collecting CPU, memory, disk input/output, and other metrics. You then use tools like Windows Performance Analyzer to try to figure out what's causing the problem. Unfortunately, in most situations this data doesn't help you identify the root cause because resource consumption counters have frequent and large variations. This situation makes it difficult to read the data and correlate it with the reported issue.

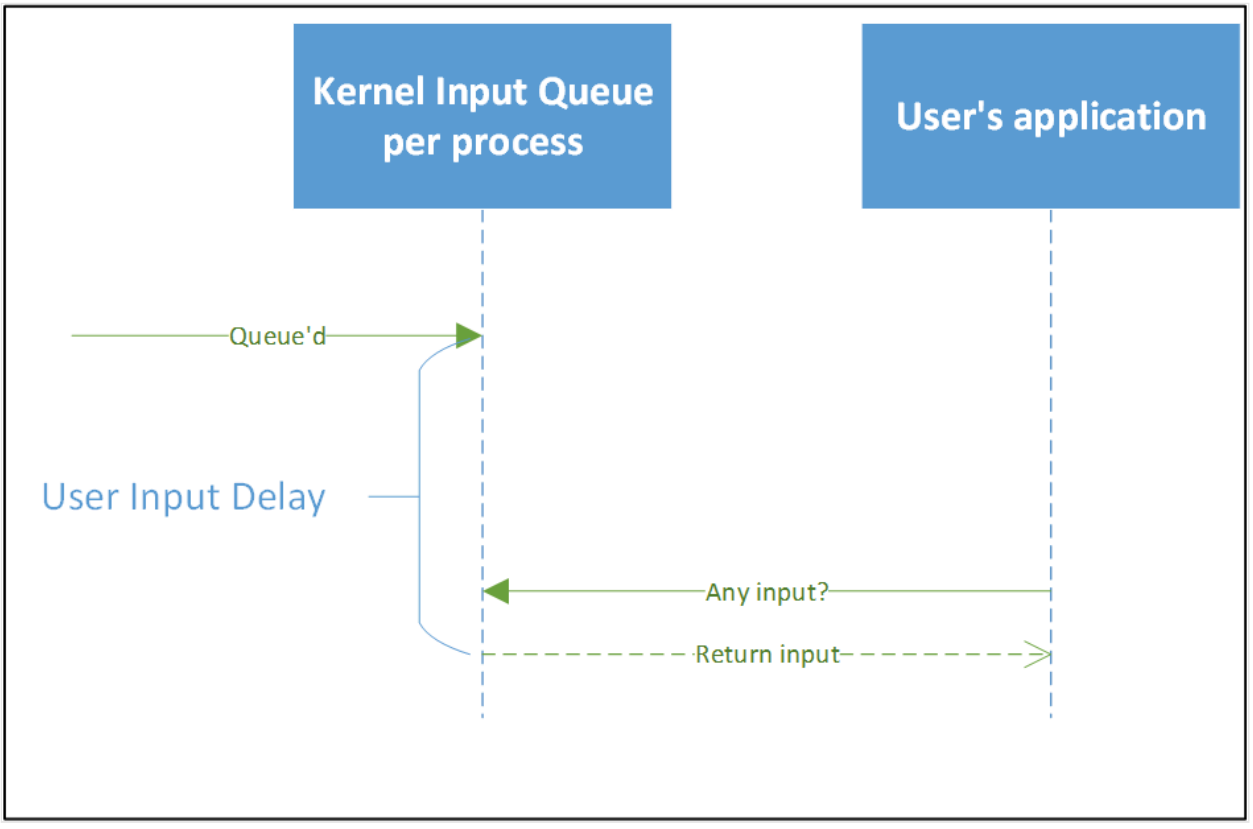
> **Note**
>
> The User Input Delay counter is only compatible with:
>
> - Windows Server 2019 or later
> - Windows 10, version 1809 or later

The User Input Delay counter can help you quickly identify the root cause for bad end user Remote Desktop performance experiences. This counter measures how long any user input, such as mouse or keyboard usage, stays in the queue before a process picks it up. The counter works in both local and remote sessions.

The following image shows a rough representation of user input flow from client to application.



The User Input Delay counter measures the max delta within an interval of time between the input being queued and when the app in a traditional message loop picks it up. A traditional message loop is shown in the following flow chart:



One important detail of this counter is that it reports the maximum user input delay within a configurable interval. This delay is the longest time it takes for an input to reach the application, which can affect the speed of important and visible actions like typing.

For example, in the following table, the user input delay would be reported as 1,000 ms within this interval. The counter reports the slowest user input delay in the interval. The counter reports this delay because the user's perception of "slow" is determined by the slowest input time (the maximum) they experience and not the average speed of all total inputs.

Expand table

Number	0	1	2
Delay	16 ms	20 ms	1,000 ms

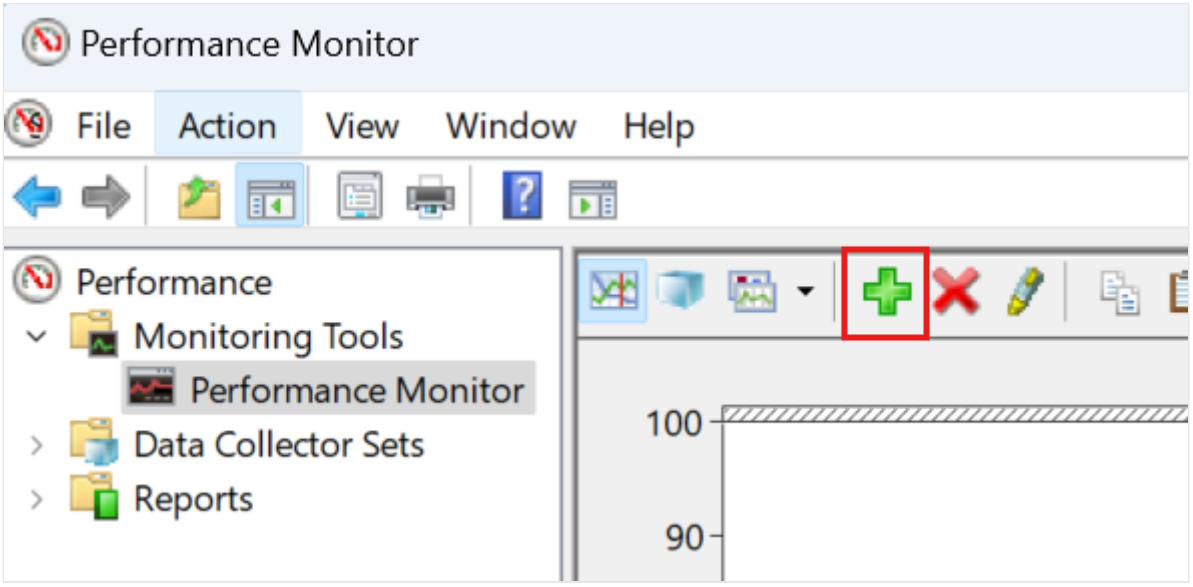
Enable and use the new performance counters

To use these new performance counters, you must first enable a registry key by running this command:

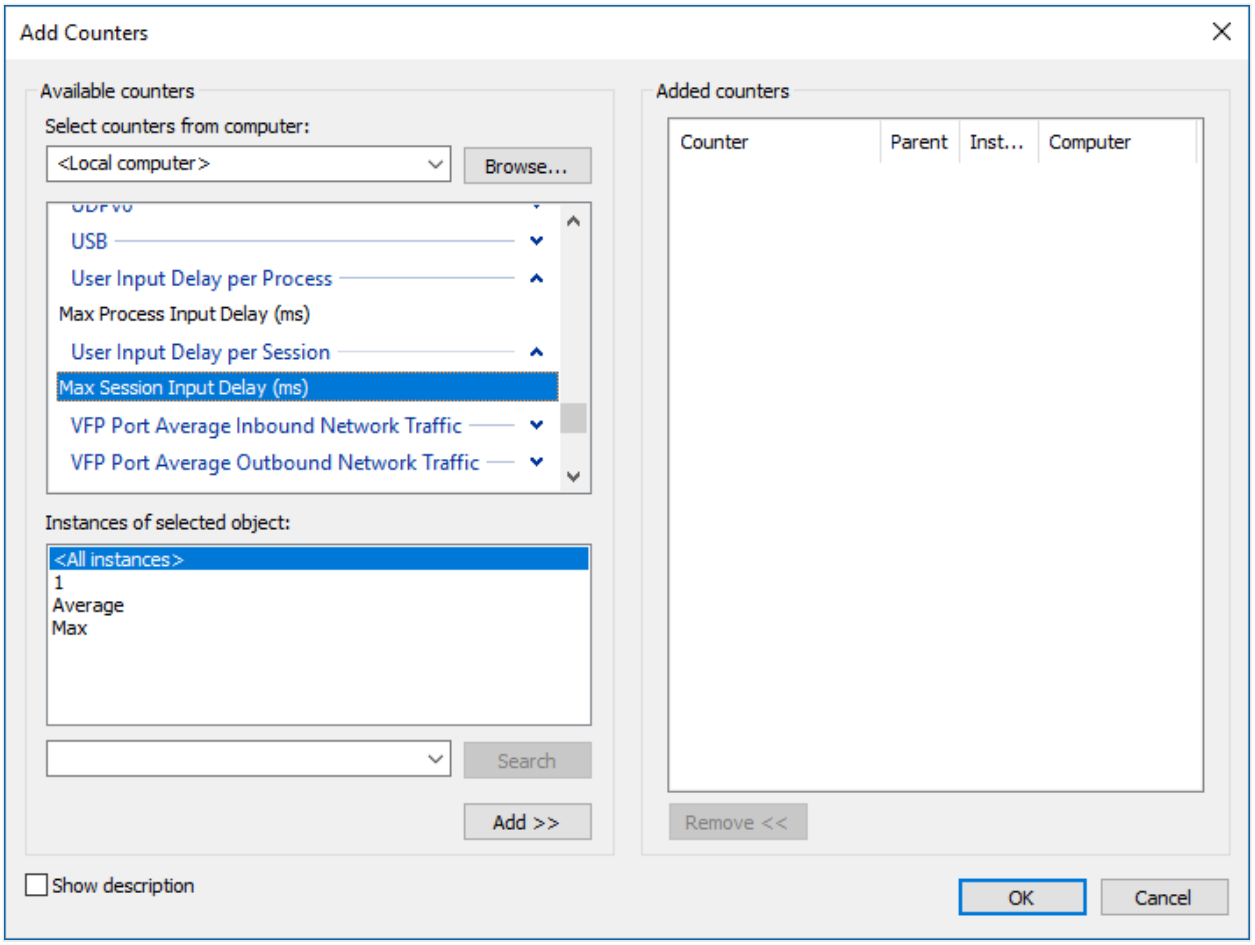
```
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server" /v
"EnableLagCounter" /t REG_DWORD /d 0x1 /f
```

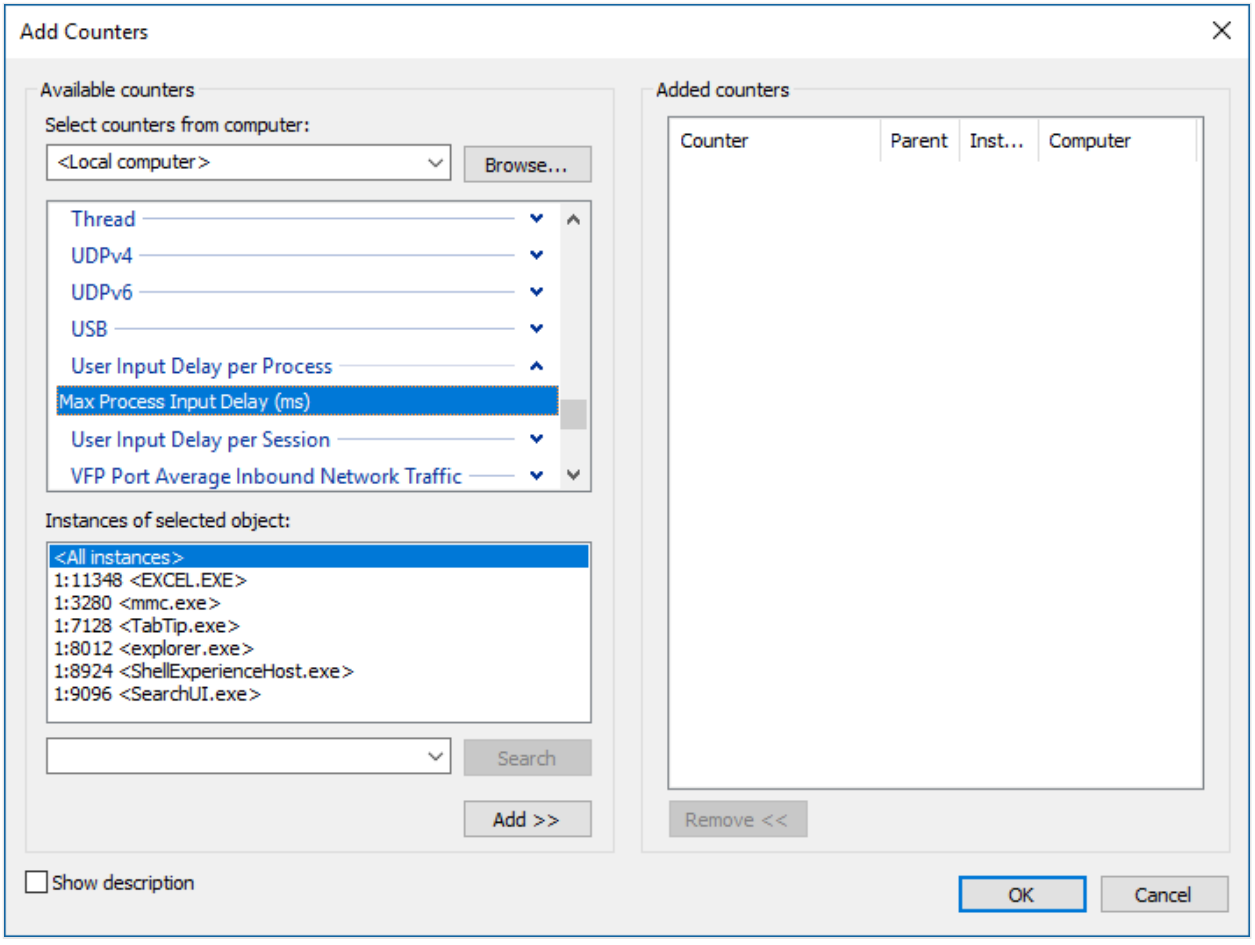
> **Note**
>
> If you use Windows 10, version 1809 or later or Windows Server 2019 or later, you won't need to enable the registry key.

Next, restart the server. Then, open the Performance Monitor, and select the **plus icon (+)**, as shown in the following screenshot:



Next, you should see the **Add Counters** dialog, where you can select **User Input Delay per Process** or **User Input Delay per Session**.



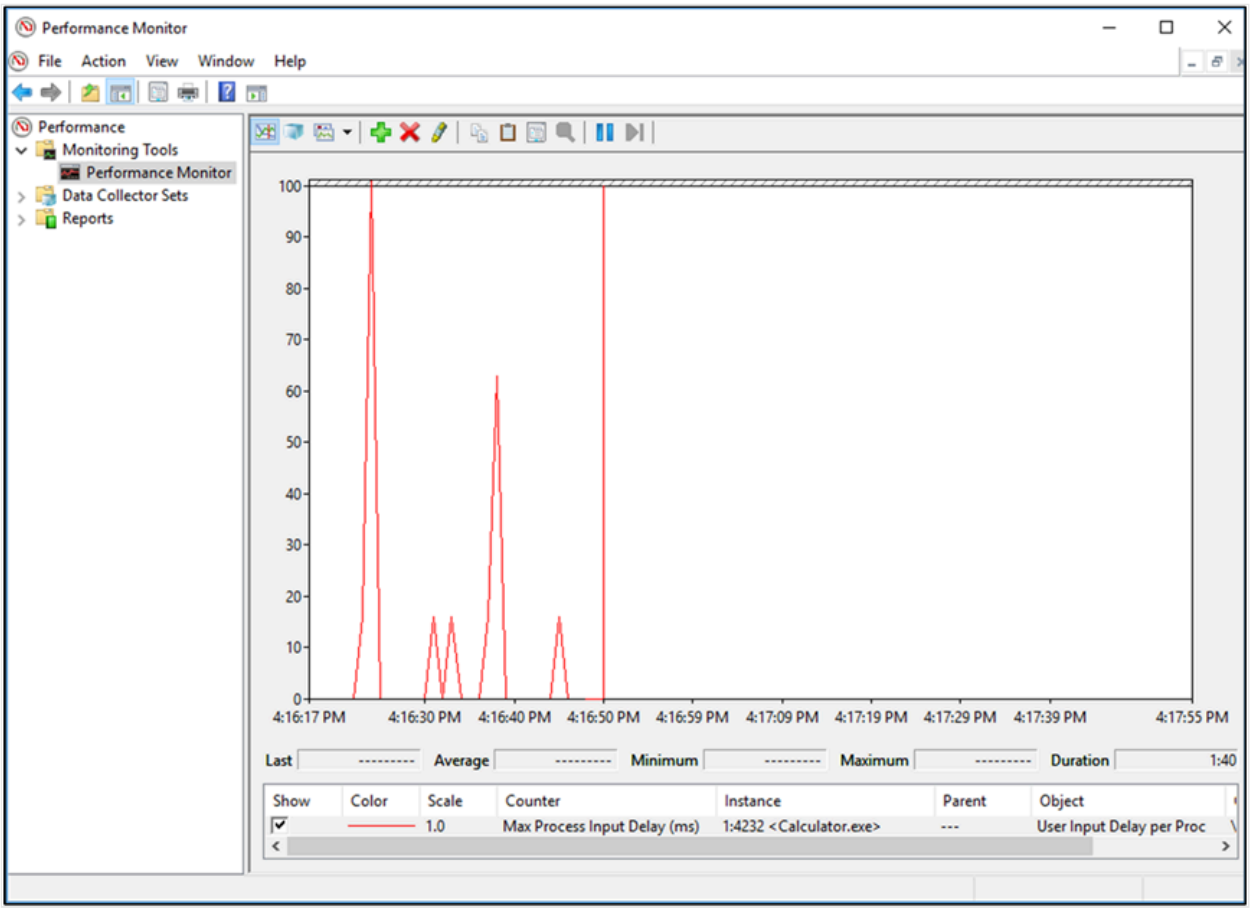


When you select **User Input Delay per Process**, you see the **Instances of the selected object**, in other words, the processes in `SessionID:ProcessID <Process Image>` format.

For example, if the Calculator app is running in a Session ID 1, you see `1:4232 <Calculator.exe>`.

> **Note**
>
> Not all processes are included. You won't see any processes that are running as SYSTEM.

The counter starts reporting user input delay as soon as you add it. The maximum scale is set to 100 (ms) by default.



Next, see the **User Input Delay per Session**. There are instances for each session ID, and their counters show the user input delay of any process within the specified session. In addition, there are two instances called "Max" (the maximum user input delay across all sessions) and "Average" (the average across all sessions).

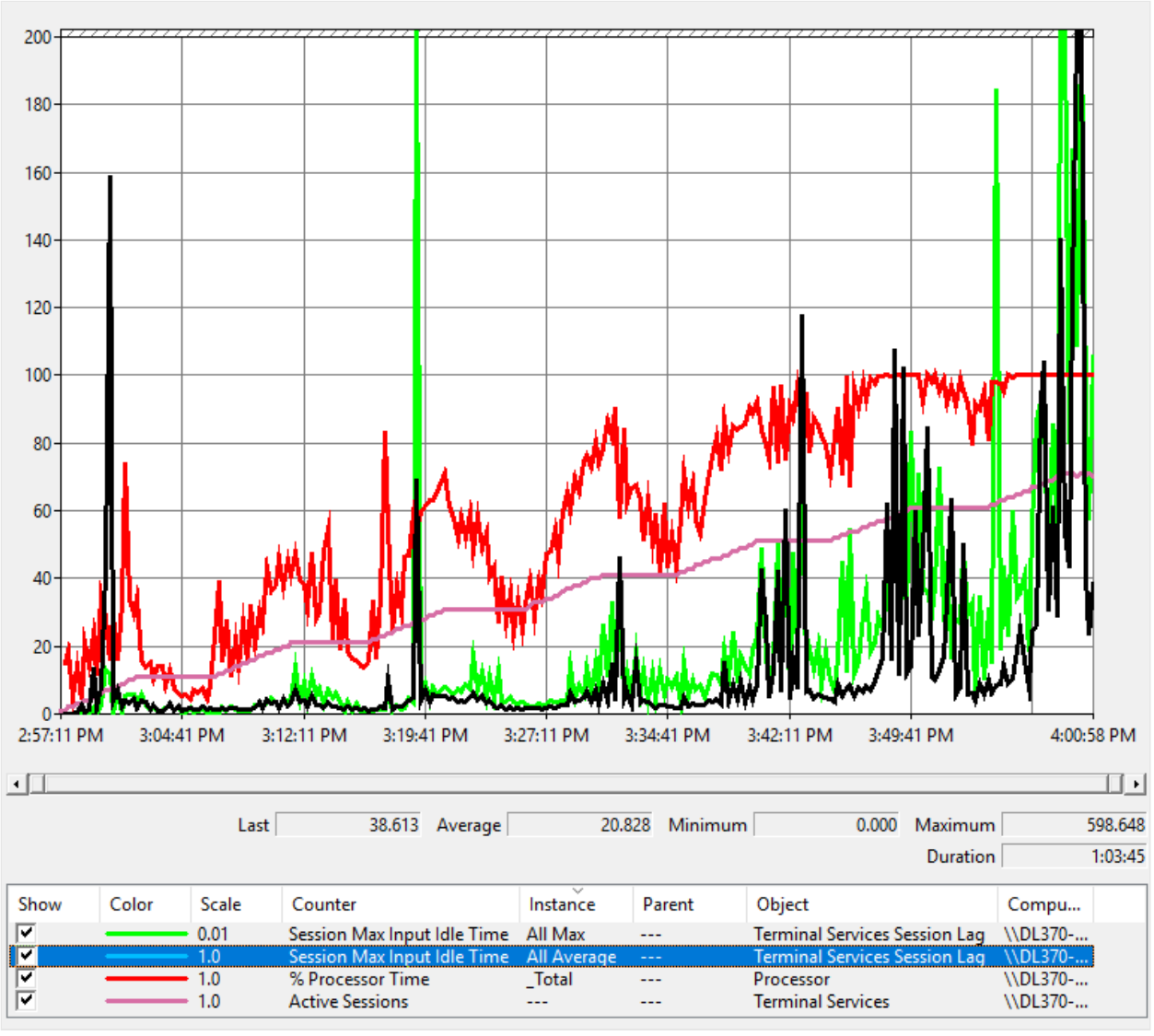
This table shows a visual example of these instances. You can get the same information in Perfmon by switching to the Report graph type.

Expand table

Type of counter	Instance name	Reported delay (ms)
User Input Delay per process	1:4232 <Calculator.exe>	200
User Input Delay per process	2:1000 <Calculator.exe>	16
User Input Delay per process	1:2000 <Calculator.exe>	32
User Input Delay per session	1	200
User Input Delay per session	2	16
User Input Delay per session	Average	108
User Input Delay per session	Max	200

Counters used in an overloaded system

Now let's look at what you see in the report if performance for an app is degraded. The following graph shows readings for users working remotely in Microsoft Word. In this case, the performance degrades over time as more users sign in remotely.



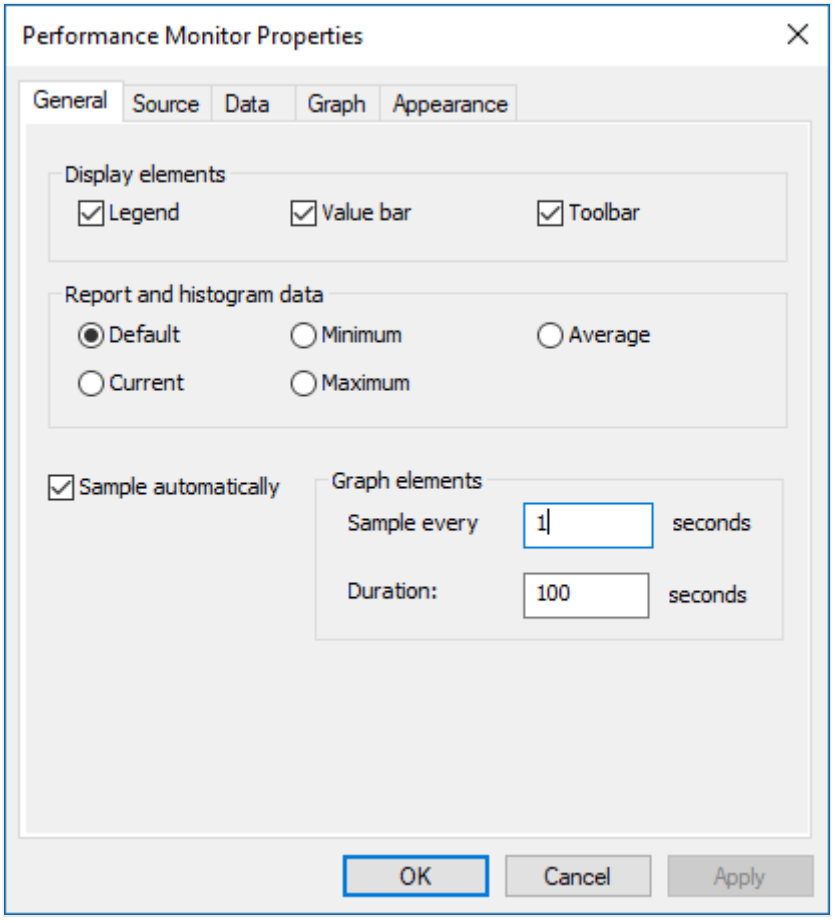
Here's how to read the graph's lines:

- The pink line shows the number of sessions signed in on the server.
- The red line is the CPU usage.
- The green line is the maximum user input delay across all sessions.
- The blue line, displayed as black in this graph, represents average user input delay across all sessions.

There's a correlation between CPU spikes and user input delay. As the CPU gets more usage, the user input delay increases. Also, as more users get added to the system, CPU usage gets closer to 100%, leading to more frequent user input delay spikes. While this counter is useful in cases where the server runs out of resources, it can also track user input delay related to a specific application.

Configuration Options

An important thing to remember when you use this performance counter is that it reports user input delay on an interval of 1,000 ms by default. If you set the performance counter sample interval property, as shown in the following screenshot, to anything different, the reported value will be incorrect.



To fix this issue, you can set the following registry key to match the interval (in milliseconds) that you want to use. For example, if you change Sample every 1 second to

Sample every 5 seconds, you need to set this key to 5000 ms.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server]

"LagCounterInterval"=dword:00005000
```

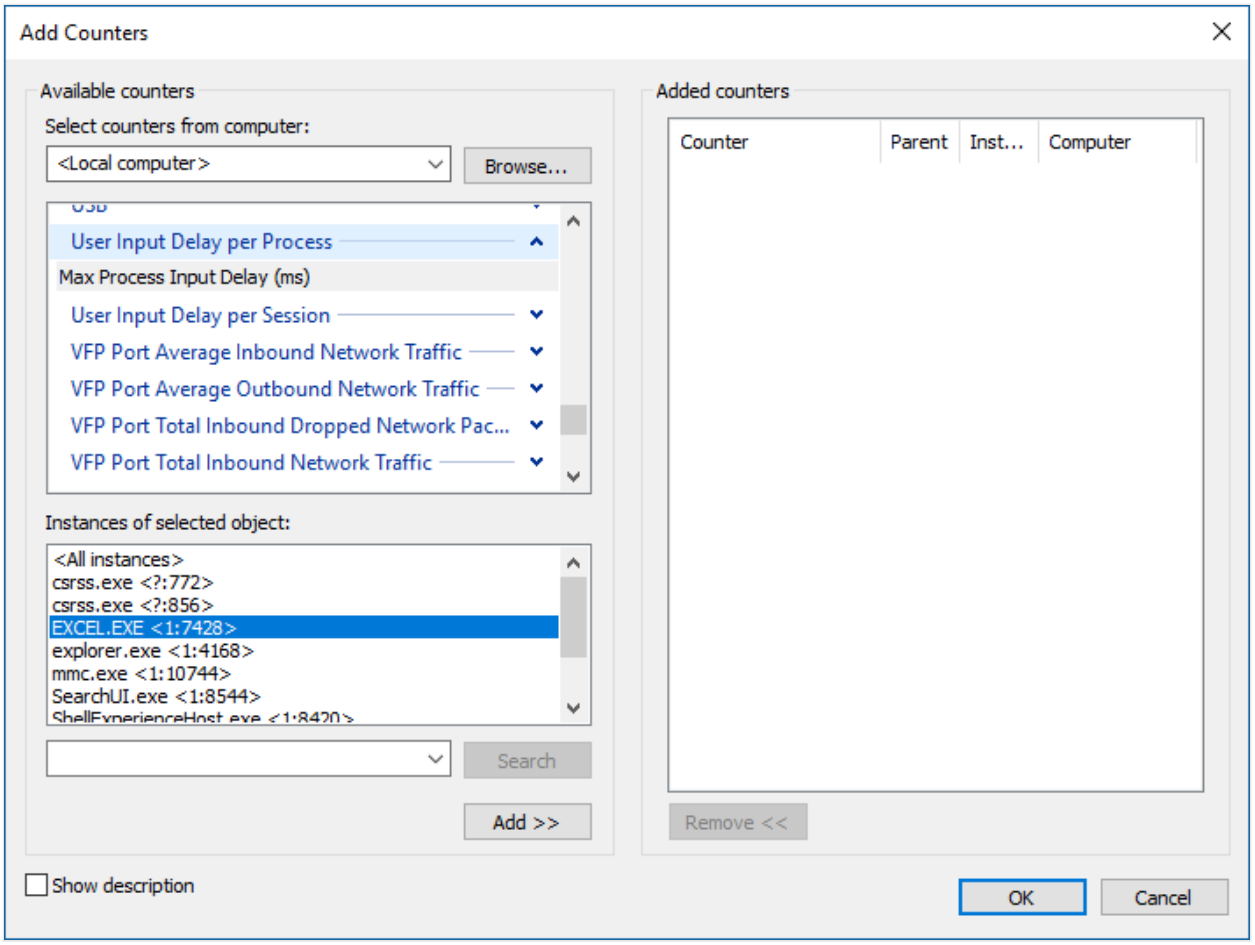
> **Note**
>
> If you use Windows 10, version 1809 or later or Windows Server 2019 or later, you don't need to set LagCounterInterval to fix the performance counter.

We've also added a couple of keys you might find helpful under the same registry key:

`LagCounterImageNameFirst`—set this key to `DWORD 1` (default value 0 or key doesn't exist). This key changes the counter names to "Image Name <SessionID:ProcessId>" for example, "explorer <1:7964>". This change is useful if you want to sort by image name.

`LagCounterShowUnknown`—set this key to `DWORD 1` (default value 0 or key doesn't exist). This key shows any processes that are running as services or SYSTEM. Some processes show up with their session set as "?".

The following image shows what it looks like with both keys on:



Use the new counters with non-Microsoft tools

Monitoring tools can consume this counter by Using Performance Counters.

Share your feedback

You can submit feedback for this feature through the Feedback Hub. Select **Apps** > **All other apps** and include "RDS performance counters—performance monitor" in your post's title.

Feedback

Was this page helpful? Yes No

Optimizing Windows configuration for VDI desktops

08/15/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Windows is tuned out of the box, but you can refine it for your corporate Microsoft Virtual Desktop Infrastructure (VDI) environment. In VDI, many background services and tasks are disabled by default.

This article is a guide to help you optimize your configuration. Some recommendations disable features you might want to use, so weigh the cost and benefit before adjusting a setting for your scenario.

> ⓘ **Note**
>
> Settings not mentioned in this article can stay at their default values, or you can set them based on your requirements and policies. These changes don't affect VDI functionality.

VDI optimization principles

A "full" virtual desktop environment presents a complete desktop session, including apps, to a user over a network. The network can be on-premises, the internet, or both. Some virtual desktop environments use a "base" operating system image, which is the basis for the desktops presented to users for work. There are types of virtual desktop implementations like **persistent**, **non-persistent**, and **desktop session**.

- The persistent type saves changes to the virtual desktop operating system from one session to the next.
- The non-persistent type doesn't save changes to the virtual desktop operating system from one session to the next.
- The desktop session works like sessions on other virtual or physical devices and is accessed over a network.

Set optimization settings on a reference machine. A virtual machine (VM) is ideal for building the VM because it saves state, lets you create checkpoints, and backs up data. Install the default OS to the base VM. Then optimize the base VM by removing unneeded apps, installing updates, deleting temporary files, and applying settings.

Security and stability are top priorities for Microsoft products and services. In virtual desktop environments, security works much like it does on physical devices. Enterprise customers can use the built-in Windows Security services, which work whether the device is connected to the internet or not. For virtual desktop environments not connected to the internet, download security signatures proactively several times a day, because Microsoft can release more than one signature update per day. Provide those signatures to virtual desktop devices and schedule installation during production, whether persistent or non-persistent. This keeps VM protection current.

Some security settings don't apply to virtual desktop environments that aren't connected to the internet or can't use cloud-enabled security. Other settings that standard Windows devices use, like Cloud Experience or the Windows Store, might not be needed. Removing access to unused features reduces footprint, network bandwidth, and attack surface.

Windows uses a monthly update rhythm. Sometimes, virtual desktop admins control the update process by shutting down VMs based on a "master" or "gold" image, unsealing the read-only image, patching it, then resealing and returning it to production. In these cases, virtual desktop devices don't need to check Windows Update. For persistent "personal" virtual desktop devices, standard patching procedures might apply. Organizations can use

Windows Update, Intune, or Microsoft Endpoint Configuration Manager (formerly SCCM) to handle updates and package delivery. Each organization decides the best approach to updating virtual desktop devices while reducing overhead.

Local policy settings and many other settings in this guide can be overridden with domain-based policy. Review the policy settings and remove or skip any that aren't needed for your environment. The settings in this document aim to balance performance optimization in virtual desktop environments while maintaining a quality user experience.

> ⓘ **Note**
>
> You can use a set of **scripts available on GitHub** to automate all the work items in this article. The scripts are easy to customize for your environment and requirements. The main code is PowerShell, and it works by calling input files, which are plain text (now .JSON), along with Local Group Policy Object (LGPO) tool export files. These text files list the apps to remove, services to disable, and more. If you don't want to remove an app or disable a service, edit the corresponding text file and remove the item you don't want to change. There's also an export of local policy settings you can import into your environment machines. It's better to include some settings in the base image than to apply them through group policy, because some settings take effect on the next restart or when a component is first used.

Non-persistent virtual desktop environments

When a non-persistent virtual desktop implementation is based on a base or "gold" image, the optimizations are mostly performed in the base image, and then through local settings and local policies.

With image-based non-persistent (NP) virtual desktop environments, the base image is read-only. When an NP virtual desktop device (VM) is started, a copy of the base image is streamed to the VM. Activity that occurs during startup and thereafter until the next reboot is redirected to a temporary location. Users are provided network locations to store their data. In some cases, the user's profile is merged with the standard VM to provide the user with their settings.

One important aspect of NP virtual desktop that is based on a single image, is servicing. Updates to the operating system (OS) and components of the OS are delivered once per month. With image based virtual desktop environment, there's a set of processes that must be performed to get updates to the image:

- On a given host, all the VMs on that host, based on the base image must be shut down or turned off. This means the users are redirected to other VMs.
- In some implementations, this is referred to as "draining." The virtual machine or session host, when set to draining mode, stops accepting new requests, but continues servicing users currently connected to the device.
- In draining mode, when the last user logs off the device, that device is then ready for servicing operations.
- The base image is then opened and started up. All maintenance activities are then performed, such as OS updates, .NET updates, app updates, and so on.
- Any new settings that need to be applied are applied at this time.
- Any other maintenance is performed at this time.
- The base image is then shut down.
- The base image is sealed and set to go back into production.
- Users are allowed to log back on.

> ⓘ **Note**

Windows performs a set of maintenance tasks, automatically, on a periodic basis. There's a scheduled task that is set to run at 3:00 AM every day by default. This scheduled task performs a list of tasks, including Windows Update cleanup. You can view all the categories of maintenance that take place automatically with this PowerShell command:

```PowerShell
Get-ScheduledTask | Where-Object {$_.Settings.MaintenanceSettings}
```

One of the challenges with non-persistent virtual desktop is that when a user logs off, nearly all the OS activity is discarded. The user's profile and/or state may be saved to a centralized location, but the virtual machine itself discards nearly all changes that were made since last boot. Therefore, optimizations intended for a Windows computer that saves state from one session to the next aren't applicable.

Depending on the architecture of virtual desktop device, things like PreFetch and SuperFetch aren't going to help from one session to the next, as all the optimizations are discarded on VM restart. Indexing may be a partial waste of resources, as would be any disk optimizations such as a traditional defragmentation.

> **Note**
>
> If preparing an image using virtualization, and if connected to the Internet during image creation process, on first logon you should postpone Feature Updates by going to **Settings** > **Windows Update**.

To sysprep or not sysprep

Windows has a built-in capability called the System Preparation Tool, also known as sysprep. Use sysprep to prepare a customized Windows 10 or Windows 11 image for duplication. The sysprep process makes sure the resulting OS is unique and ready for production.

There are reasons for and against running sysprep. In virtual desktop environments, you might want to customize the default user profile, which acts as the profile template for users who sign in using this image. You might want apps installed, but also want to control per-app settings.

The alternative is to use a standard .ISO to install, possibly with an unattended installation answer file, and a task sequence to install or remove applications. You can also use a task sequence to set local policy settings in the image, maybe using the Local Group Policy Object Utility (LGPO) tool.

To learn more about image preparation for Azure, see Prepare a Windows VHD or VHDX to upload to Azure

Supportability

When you change Windows defaults, questions about supportability can come up. After you customize a virtual desktop image (VM or session), track every change in a change log. If you need to troubleshoot, you can isolate an image in a pool and set it up for problem analysis. After you find the root cause, roll out the change to the test environment first, and then to the production workload.

This document intentionally avoids changing system services, policies, or tasks that affect security. Windows servicing comes next. You can't service virtual desktop images outside of maintenance windows, because most servicing events happen during maintenance windows in virtual desktop environments, except for security software

updates. Microsoft's guidance for Windows security in virtual desktop environments is in the Deployment guide for Windows Defender Antivirus in a virtual desktop infrastructure (VDI) environment

Think about supportability when you change default Windows settings. Sometimes, problems that are hard to fix can happen when you change system services, policies, or scheduled tasks to harden or lighten the system. Check the Microsoft Knowledge Base for current known issues about changed default settings. The guidance in this document and the associated script on GitHub are updated based on known issues, if any come up. You can report issues to Microsoft in several ways.

Use your favorite search engine with the terms `"start value" site:support.microsoft.com` to find known issues about default start values for services.

This document and the associated scripts on GitHub don't change any default permissions. If you want to increase your security settings, start with the AaronLocker project. For more information, see AaronLocker overview.

Virtual desktop optimization categories

The following categories are ways in which the virtual desktop can be optimized:

- Universal Windows Platform (UWP) app cleanup
- Optional features cleanup
- Local policy settings
- System services
- Scheduled tasks
- Apply Windows (and other) updates
- Automatic Windows traces
- Windows Defender optimization with VDI
- Client network performance tuning by registry settings
- Other settings from the "Windows Restricted Traffic Limited Functionality Baseline" guidance.
- Disk cleanup

The following sections explain each category in more detail.

Universal Windows Platform (UWP) application cleanup

One of the goals of a virtual desktop image is to be as light as possible with respect to persistent storage. One way to reduce the size of the image is to remove unused UWP applications (apps). With UWP apps, there are the main application files, also known as the payload. There's a small amount of data stored in each user's profile for application-specific settings. There's also a small amount of data in the "All Users" profile.

In addition, all UWP apps are registered at either the user or machine level at some point after startup for the device, and login for the user. The UWP apps, which include the Start Menu and the Windows Shell, perform various tasks at or after installation, and again when a user logs in for the first time, and to a lesser extent at subsequent logins. For all UWP apps, there are occasional evaluations that take place, such as:

- Do you need to update the app to the latest version?
- The app, if pinned to the Start Menu, might have live tile data to download
- Does the app have a cache of data that needs to be updated, such as maps or weather?
- Does the app have persistent data from the user's profile that needs to be presented at login (for example, Sticky Notes)

With a default installation of Windows, it's unlikely that all UWP apps are used by an organization. Therefore, if those apps are removed, there are fewer evaluations that need to take place, less caching, and so on. The second method here's to direct Windows to disable "consumer experiences." This reduces Store activity by having to check for every user what apps are installed, what apps are available, and then to start downloading some UWP apps. The performance savings can be significant when there are hundreds or thousands of users, all start work at approximately the same time, or even starting work at rolling times across time zones.

Connectivity and timing are important factors when it comes to UWP app cleanup. If you deploy your base image to a device with no network connectivity, Windows can't connect to the Microsoft Store and download apps and try to install them while you're trying to uninstall them. This might be a good strategy to allow you time to customize your image, and then update what remains at a later stage of the image creation process.

If you modify your base .WIM that you use to install Windows and remove unneeded UWP apps from the .WIM before you install, the apps don't install and your subsequent profile creation times are shorter. There's a link later in this section with information on how to remove UWP apps from your installation .WIM file.

A good strategy for the virtual desktop environment is to provision the apps you want in the base image, then limit or block access to the Microsoft Store afterward. Store apps are updated periodically in the background on normal computers. The UWP apps can be updated during the maintenance window when other updates are applied.

Delete the payload of UWP apps

UWP apps that aren't needed are still in the file system consuming a small amount of disk space. For apps that aren't needed, the payload of unwanted UWP apps can be removed from the base image using PowerShell commands. If you delete UWP app payloads out of the installation .WIM file using the links provided later in this section, you can start from the beginning with a slim list of UWP apps.

Run the following PowerShell command to enumerate provisioned UWP apps currently running on the local computer:

```
PowerShell
Get-AppxProvisionedPackage -Online
```

UWP apps that are provisioned to a system can be removed during OS installation as part of a task sequence, or later after the OS is installed. This may be the preferred method because it makes the overall process of creating or maintaining an image modular. Once you develop the scripts, if something changes in a subsequent build you edit an existing script rather than repeat the process from scratch.

Then run the following PowerShell command to remove UWP app payloads:

```
PowerShell
Remove-AppxProvisionedPackage -Online - PackageName MyAppxPackage
```

As a final note on this topic, each UWP app should be evaluated for applicability in each unique environment. Install a default installation of Windows 10 or Windows 11, and then note which apps are running and consuming memory. For example, you might remove apps that start automatically, or apps that automatically display information on the Start Menu, such as Weather and News.

> ⓘ **Note**

If you're using the scripts from GitHub, you can easily control which apps are removed before running the script. After downloading the script files, locate the AppxPackage.json file, edit that file, and remove entries for apps that you want to keep, such as Calculator, Sticky Notes, and so on.

Optional features cleanup

This section describes optional features that can be optimized.

Managing optional features with PowerShell

You can manage Windows Optional Features using PowerShell. To enumerate currently installed Windows Features, run the following PowerShell command:

PowerShell

```
Get-WindowsOptionalFeature -Online
```

Using PowerShell, an enumerated Windows Optional Feature can be configured as enabled or disabled, as in the following example:

PowerShell

```
Enabled-WindowsOptionalFeature -Online -FeatureName "DirectPlay" -All
```

Here's an example command that disables the Windows Media Player feature in the virtual desktop image:

PowerShell

```
Disable-WindowsOptionalFeature -Online -FeatureName "WindowsMediaPlayer"
```

Next, you may want to remove the Windows Media Player package. This example command shows you how to find the package name:

PowerShell

```
Get-WindowsPackage -Online -PackageName *media*
```

The output of that command shows something like the following information:

code

```
PackageName             : Microsoft-Windows-MediaPlayer-
Package~31bf3856ad364e35~amd64~~10.0.19041.153
Applicable              : True
Copyright               : Copyright (c) Microsoft Corporation. All Rights Reserved
...
```

If you want to remove the Windows Media Player package (to free up about 60 MB disk space), you can run this command:

PowerShell

```
PS C:\Windows\system32> Remove-WindowsPackage -PackageName Microsoft-Windows-MediaPlayer-
Package~31bf3856ad364e35~amd64~~10.0.19041.153 -Online
```

Enable or disabling Windows features using DISM

You can use the built-in `Dism.exe` tool to enumerate and control Windows Optional Features. A Dism.exe script could be developed and run during an operating system installation task sequence with Features on Demand.

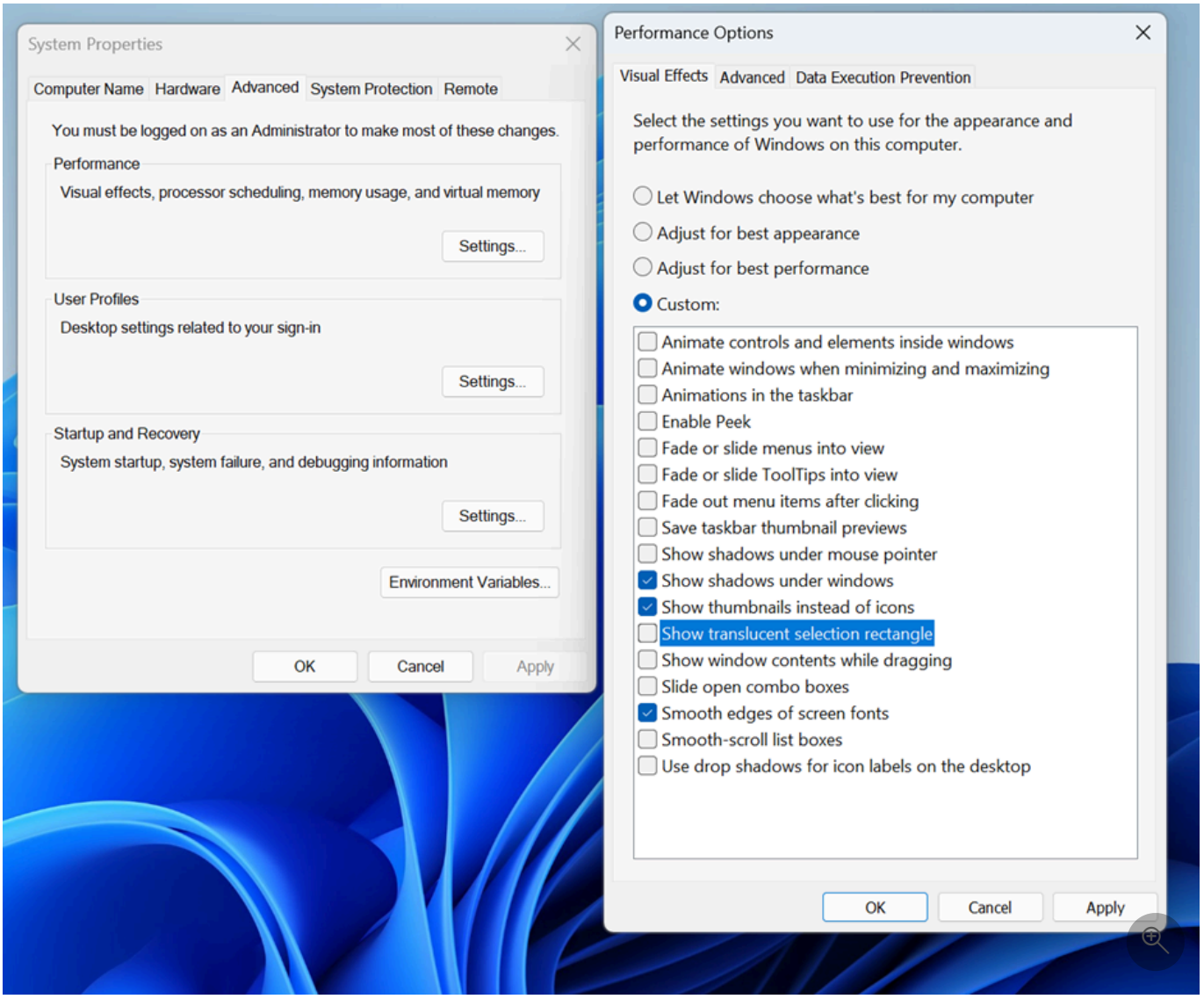
Default user settings

You can customize the Windows registry file at `C:\Users\Default\NTUSER.DAT`. Any setting changes you make to this file are applied to any subsequent user profiles created from a machine running this image. You can control which settings you wish to apply to the default user profile by editing the **DefaultUserSettings.txt** file.

To reduce transmission of graphical data over the virtual desktop infrastructure, you can set the default background to a solid color instead of the default Windows image. You can also set the sign-in screen to be a solid color, and turn off the opaque blurring effect on sign-in.

The following settings are applied to the default user profile registry hive, mainly to reduce animations. If some or all of these settings aren't desired, delete out the settings that you don't wish to apply to new user profiles based on this image. The goal with these settings is to enable the following equivalent settings:

- Show shadows under mouse pointer
- Show shadows under windows
- Smooth edges of screen fonts

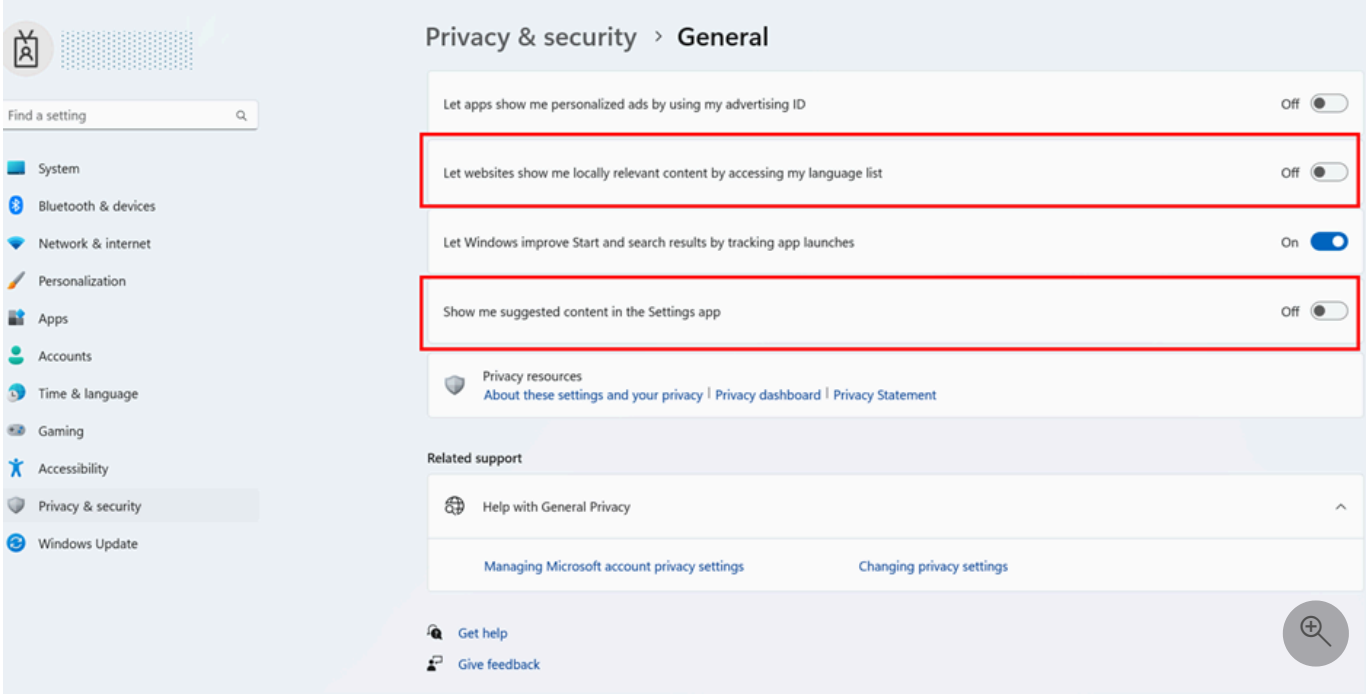


And there's a method to disable the following two privacy settings for any user profile created after you run the optimization:

- Let websites provide locally relevant content by accessing my language list
- Show me suggested content in the Settings app

Optionally, disable the following two privacy settings for any user profile created after you run the optimization:

- Let websites provide locally relevant content by accessing my language list
- Show me suggested content in the Settings app



The following are the optimization settings applied to the default user profile registry hive to optimize performance. This operation is performed by first loading the default user profile registry hive **NTUser.dat**, as the ephemeral key name **Temp**, and then making the following modifications:

regedit

```
Load HKLM\Temp C:\Users\Default\NTUSER.DAT
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer" /v ShellState /t REG_BINARY /d
240000003C28000000000000000000000 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v IconsOnly /t REG_DWORD
/d 1 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ListviewAlphaSelect /t
REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ListviewShadow /t
REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ShowCompColor /t
REG_DWORD /d 1 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v ShowInfoTip /t
REG_DWORD /d 1 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" /v TaskbarAnimations /t
REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\Explorer\VisualEffects" /v VisualFXSetting /t
REG_DWORD /d 3 /f
add "HKLM\Temp\Software\Microsoft\Windows\DWM" /v EnableAeroPeek /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\DWM" /v AlwaysHiberNateThumbnails /t REG_DWORD /d 0 /f
add "HKLM\Temp\Control Panel\Desktop" /v DragFullWindows /t REG_SZ /d 0 /f
add "HKLM\Temp\Control Panel\Desktop" /v FontSmoothing /t REG_SZ /d 2 /f
add "HKLM\Temp\Control Panel\Desktop" /v UserPreferencesMask /t REG_BINARY /d 9032078010000000 /f
add "HKLM\Temp\Control Panel\Desktop\WindowMetrics" /v MinAnimate /t REG_SZ /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\StorageSense\Parameters\StoragePolicy" /v 01
/t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v SubscribedContent-
338393Enabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v SubscribedContent-
353694Enabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v SubscribedContent-
353696Enabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v SubscribedContent-
338388Enabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v SubscribedContent-
338389Enabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" /v
```

```
SystemPaneSuggestionsEnabled /t REG_DWORD /d 0 /f
add "HKLM\Temp\Control Panel\International\User Profile" /v HttpAcceptLanguageOptOut /t REG_DWORD /d 1
/f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.Windows.Ph
otos_8wekyb3d8bbwe" /v Disabled /t REG_DWORD /d 1 /f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.Windows.Ph
otos_8wekyb3d8bbwe" /v DisabledByUser /t REG_DWORD /d 1 /f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.YourPhone_
8wekyb3d8bbwe" /v Disabled /t REG_DWORD /d 1 /f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.YourPhone_
8wekyb3d8bbwe" /v DisabledByUser /t REG_DWORD /d 1 /f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.MicrosoftE
dge_8wekyb3d8bbwe" /v Disabled /t REG_DWORD /d 1 /f
add
"HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\BackgroundAccessApplications\Microsoft.MicrosoftE
dge_8wekyb3d8bbwe" /v DisabledByUser /t REG_DWORD /d 1 /f
add "HKLM\Temp\Software\Microsoft\InputPersonalization" /v RestrictImplicitInkCollection /t REG_DWORD
/d 1 /f
add "HKLM\Temp\Software\Microsoft\InputPersonalization" /v RestrictImplicitTextCollection /t REG_DWORD
/d 1 /f
add "HKLM\Temp\Software\Microsoft\Personalization\Settings" /v AcceptedPrivacyPolicy /t REG_DWORD /d 0
/f
add "HKLM\Temp\Software\Microsoft\InputPersonalization\TrainedDataStore" /v HarvestContacts /t
REG_DWORD /d 0 /f
add "HKLM\Temp\Software\Microsoft\Windows\CurrentVersion\UserProfileEngagement" /v
ScoobeSystemSettingEnabled /t REG_DWORD /d 0 /f
Unload HKLM\Temp
```

Disable settings for Windows apps from starting and running in the background. While not significant on a single device, Windows starts multiple processes for each user session on a given device, or session host. If this functionality is desired as-is, delete the lines in the `DefaultUserSettings.txt` file that include the app names like **Windows.Photos** and/or **MicrosoftEdge**.

Local policy settings

Many optimizations for Windows in a virtual desktop environment can be made using Windows policy. The settings listed in the table in this section can be applied locally to the base/gold image. If the equivalent settings aren't specified in any other way, such as group policy, the settings still apply.

Some decisions may be based on the specifics of the environment.

- Is the virtual desktop environment allowed to access the Internet?
- Is the virtual desktop solution persistent or non-persistent?

The following settings were chosen to not counter or conflict with any setting that has anything to do with security. These settings were chosen to remove settings or disable functionality that may not be applicable to virtual desktop environments.

Expand table

Policy setting	Item	Sub-item	Possible setting and comments
Local Computer Policy \ Computer	N/A	N/A	N/A

Policy setting	Item	Sub-item	Possible setting and comments
Configuration \ Windows Settings \ Security Settings			
Network List Manager policies	All networks properties	Network location	**User can't change location** (This setting is set to prevent the right-hand side pop-up when a new network is detected)
Local Computer Policy \ Computer Configuration \ Administrative Templates \ Control Panel	N/A	N/A	
Control Panel	Allow Online Tips	N/A	**Disabled** (Settings can't contact Microsoft content services to retrieve tips and help content)
Control Panel \ Personalization	Force a specific default lock screen and logon image	N/A	Enabled (This setting allows you to force a specific default lock screen and logon image by entering the path (location) of the image file. The same image is used for both the lock and logon screens. The reason for this recommendation is to reduce bytes transmitted over the network for virtual desktop environments. This setting can be removed or customized for each environment.)
Control Panel\ Regional and Language Options\Handwriting personalization	Turn off automatic learning	N/A	**Enabled** (With this policy setting enabled, automatic learning stops, and any stored data is deleted. Users can't configure this setting in Control Panel)
Local Computer Policy \ Computer Configuration \ Administrative Templates \ Network	N/A	N/A	N/A
Background Intelligent Transfer Service (BITS)	Allow BITS Peer caching	N/A	**Disabled** (This policy setting determines if the Background Intelligent Transfer Service (BITS) peer caching feature is enabled on a specific computer.)
Background Intelligent Transfer Service (BITS)	Don't allow the BITS client to use Windows Branch Cache	N/A	**Enabled** (With this policy setting enabled, the BITS client doesn't use Windows Branch Cache.) The reason for this recommendation is so that virtual desktop devices aren't used for content caching, and the devices aren't allowed to use the network bandwidth.
Background Intelligent Transfer Service (BITS)	Don't allow the computer to act as a	N/A	**Enabled** (With this policy setting enabled, the computer doesn't use the BITS peer caching feature to download files; files

Policy setting	Item	Sub-item	Possible setting and comments
	BITS Peer caching client		are downloaded only from the origin server.)
Background Intelligent Transfer Service (BITS)	Don't allow the computer to act as a BITS Peer caching server	N/A	**Enabled** (With this policy setting enabled, the computer can't cache downloaded files and offer them to its peers.)
BranchCache	Turn on BranchCache	N/A	**Disabled** (With this selection disabled, BranchCache is turned off for all client computers where the policy is applied.)
*Fonts	Enabled Font Providers	N/A	**Disabled** (With this setting disabled, Windows doesn't connect to an online font provider and only enumerates locally installed fonts)
Hotspot Authentication	Enable hotspot Authentication	N/A	**Disabled** (This policy setting defines whether WLAN hotspots are probed for Wireless Internet Service Provider roaming (WISPr) protocol support. With this policy setting disabled, WLAN hotspots aren't probed for WISPr protocol support, and users can only authenticate with WLAN hotspots using a web browser.)
Microsoft Peer-to-Peer Networking Services	Turn off Microsoft Peer-to-Peer Networking Services	N/A	**Enabled** (This setting turns off Microsoft Peer-to-Peer Networking Services in its entirety and causes all dependent applications to stop working. If you enable this setting, peer-to-peer protocols are turned off.)
Network Connectivity Status Indicator (There are other settings in this section that can be used in isolated networks)	Specify passive polling	Disable passive poling (**checkbox**)	**Enabled** (This Policy setting enables you to specify passive polling behavior. NCSI polls various measurements throughout the network stack on a frequent interval to determine if network connectivity is lost. Use the options to control the passive polling behavior.) Disabling NCIS passive polling can improve CPU workload on servers or other machines whose network connectivity is static. This setting may negatively affect Exchange connectivity in Outlook.
Offline Files	Allow or Disallow use of the Offline Files feature	N/A	**Disabled** (This policy setting determines whether the Offline Files feature is enabled. Offline Files saves a copy of network files on the user's computer for use when the computer isn't connected to the network. With this policy setting disabled, Offline Files feature is disabled and users can't enable it.)

Policy setting	Item	Sub-item	Possible setting and comments
*TCPIP Settings\ IPv6 Transition Technologies	Set Teredo State	Disabled State	**Enabled** (With this setting enabled, and set to "Disabled State", no Teredo interfaces are present on the host)
*WLAN Service\ WLAN Settings	Allow Windows to automatically connect to suggested open hot spots, to networks shared by contacts, and to hot spots offering paid services	N/A	**Disabled** (This policy setting determines whether users can enable the following WLAN settings: "Connect to suggested open hotspots," "Connect to networks shared by my contacts," and "Enable paid services." With this policy setting disabled, "Connect to suggested open hotspots," "Connect to networks shared by my contacts," and "Enable paid services" are turned off and users on this device are prevented from enabling them.)
WWAN Service\ Cellular Data Access	Let Windows apps access cellular data	Default for all apps: **Force Deny**	**Enabled** (If you choose the "Force Deny" option, Windows apps aren't allowed to access cellular data and users can't change it.)
Local Computer Policy \ Computer Configuration \ Administrative Templates \ Start Menu and Taskbar	N/A	N/A	
*Notifications	Turn off notifications network usage	N/A	**Enabled** (With this policy setting enabled, applications and system features aren't able to receive notifications from the network from WNS or via notification polling APIs)
Local Computer Policy \ Computer Configuration \ Administrative Templates \ System	N/A	N/A	N/A
Device Installation	Don't send a Windows error report when a generic driver is installed on a device	N/A	**Enabled** (With this policy setting enabled, an error report isn't sent when a generic driver is installed.)
Device Installation	Prevent creation of a system restore point during device activity that would normally prompt creation of a restore point	N/A	**Enabled** (With this policy setting enabled, Windows doesn't create a system restore point when one would normally be created.)
Device Installation	Prevent device metadata retrieval from the Internet	N/A	**Enabled** (This policy setting allows you to prevent Windows from retrieving device metadata from the Internet. With this policy setting enabled, Windows doesn't retrieve device metadata for installed devices from the Internet. This policy

Policy setting	Item	Sub-item	Possible setting and comments
			setting overrides the setting in the Device Installation Settings dialog box (Control Panel > System and Security > System > Advanced System Settings > Hardware tab).)
Device Installation	Turn off "Found New Hardware" balloons during device installation	N/A	**Enabled** (This policy setting allows you to turn off "Found New Hardware" balloons during device installation. With this policy setting enabled, "Found New Hardware" balloons don't appear while a device is being installed.)
Filesystem\NTFS	Short name creation options	Short name creation options: Disabled on all volumes	**Enabled** (These settings provide control over whether or not short names are generated during file creation. Some applications require short names for compatibility, but short names have a negative performance impact on the system. With short names disabled on all volumes, then they aren't generated.)
*Group Policy	Continue experiences on this device	N/A	**Disabled** (This policy setting determines whether the Windows device is allowed to participate in cross-device experiences (continue experiences). Disabling this policy prevents this device from being discoverable by other devices, and thus can't participate in cross-device experiences.)
Internet Communication Management\ Internet Communication settings	Turn off Event Viewer "Events.asp" links	N/A	**Enabled** (This policy setting specifies whether "Events.asp" hyperlinks are available for events within the Event Viewer application.)
Internet Communication Management\ Internet Communication settings	Turn off handwriting personalization data sharing	N/A	**Enabled** (Turns off data sharing from the handwriting recognition personalization tool.)
Internet Communication Management\ Internet Communication settings	Turn off handwriting recognition error reporting	N/A	**Enabled** (Turns off the handwriting recognition error reporting tool.)
Internet Communication Management\ Internet Communication settings	Turn off Help and Support Center Microsoft Knowledge Base search	N/A	**Enabled** (This policy setting specifies whether users can perform a Microsoft Knowledge Base search from the Help and Support Center.)
Internet Communication Management\ Internet	Turn off Internet Connection Wizard if URL connection is	N/A	**Enabled** (This policy setting specifies whether the Internet Connection Wizard can connect to Microsoft to download a list of Internet Service Providers (ISPs).)

Policy setting	Item	Sub-item	Possible setting and comments
Communication settings	referring to Microsoft.com		
Internet Communication Management\ Internet Communication settings	Turn off Internet download for Web publishing and online ordering wizards	N/A	**Enabled** (This policy setting specifies whether Windows should download a list of providers for the web publishing and online ordering wizards.)
Internet Communication Management\ Internet Communication settings	Turn off Internet File Association service	N/A	**Enabled** (This policy setting specifies whether to use the Microsoft Web service for finding an application to open a file with an unhandled file association.)
Internet Communication Management\ Internet Communication settings	Turn off Registration if URL connection is referring to Microsoft.com	N/A	**Enabled** (This policy setting specifies whether the Windows Registration Wizard connects to Microsoft.com for online registration.)
Internet Communication Management\ Internet Communication settings	Turn off Search Companion content file updates	N/A	**Enabled** (This policy setting specifies whether Search Companion should automatically download content updates during local and Internet searches.)
Internet Communication Management\ Internet Communication settings	Turn off the "Order Prints" picture task	N/A	**Enabled** (If you enable this policy setting, the task "Order Prints Online" is removed from Picture Tasks in File Explorer folders.)
Internet Communication Management\ Internet Communication settings	Turn off the "Publish to Web" task for files and folders	N/A	**Enabled* (This policy setting specifies whether the tasks "Publish this file to the Web," "Publish this folder to the Web," and "Publish the selected items to the Web" are available from File and Folder Tasks in Windows folders.)
Internet Communication Management\ Internet Communication settings	Turn off Windows Customer Experience Improvement Program	N/A	**Enabled** (The Windows Customer Experience Improvement Program (CEIP) collects information about your hardware configuration and how you use our software and services to identify trends and usage patterns. If you enable this policy setting, all users are opted out of the Windows CEIP.)
Internet Communication Management\ Internet Communication settings	Turn off Windows Error Reporting	N/A	**Enabled** (This policy setting controls whether or not errors are reported to Microsoft. If you enable this policy setting, users aren't given the option to report errors.)
Internet Communication Management\ Internet Communication settings	Turn off Windows Update device driver searching	N/A	**Enabled** (This policy setting specifies whether Windows searches Windows Update for device drivers when no local drivers for a device are present. If you enable this policy setting, Windows

Policy setting	Item	Sub-item	Possible setting and comments
			Update isn't searched when a new device is installed.)
Logon	Don't display the Getting Started welcome screen at logon	N/A	**Enabled** (With this setting enabled, the welcome screen is hidden from the user logging on to a Windows device.)
Logon	Don't enumerate connected users on domain-joined computers	N/A	**Enabled** (With this setting enabled, the Logon UI doesn't enumerate any connected users on domain-joined computers.)
Logon	Enumerate local users on domain-joined computers	N/A	**Disabled** (With this setting disabled, the Logon UI doesn't enumerate local users on domain-joined computers.)
Logon	Show clear logon background	N/A	**Enabled** (This policy setting disables the acrylic blur effect on logon background image. With this setting enabled, the logon background image shows without blur.)
Logon	Show first sign-in animation	N/A	**Disabled** (This policy setting allows you to control whether users see the first sign-in animation when signing in to the computer for the first time. This applies to both the first user of the computer who completes the initial setup and users who are added to the computer later. It also controls if Microsoft account users are offered the opt-in prompt for services during their first sign-in. With this setting disabled, users don't see the first logon animation and Microsoft account users don't see the opt-in prompt for services.)
Logon	Turn off app notifications on the lock screen	N/A	**Enabled** (This policy setting allows you to prevent app notifications from appearing on the lock screen. With this setting enabled, no app notifications are displayed on the lock screen.)
Power Management	Select an active power plan	Active Power Plan: High Performance	**Enabled** (If you enable this policy setting, specify a power plan from the Active Power Plan list.) With the "Power" service disabled, the Powercfg.cpl UI isn't able to display these power options, and instead returns an RPC error.
Power Management \ Video and Display Settings	Turn on desktop background slideshow (plugged-in)	N/A	**Disabled** (This policy setting allows you to specify if Windows should enable the desktop background slideshow.) With this setting disabled, the desktop background slideshow is disabled. This setting likely has no effect on a VM.

Policy setting	Item	Sub-item	Possible setting and comments
Recovery	Allow restore of system to default state	N/A	**Disabled** (With this setting disabled, the items "Use a system image you created earlier to recover your computer" and "Reinstall Windows" (or "Return your computer to factory condition") in Recovery (in Control Panel) are unavailable.)
*Storage Health	Allow downloading updates to the Disk Failure Prediction Model	N/A	**Disabled** (Updates wouldn't be downloaded for the Disk Failure Prediction Failure Model)
System Restore	Turn off System Restore	N/A	**Enabled** (With this setting enabled, System Restore is turned off, and the System Restore Wizard can't be accessed. The option to configure System Restore or create a restore point through System Protection is also disabled.)
Troubleshooting and Diagnostics\ Scheduled Maintenance	Configure Scheduled Maintenance Behavior	N/A	**Disabled** (Determines whether scheduled diagnostics run to proactively detect and resolve system problems. With this policy setting disabled, Windows can't detect, troubleshoot, or resolve problems on a scheduled basis.)
Troubleshooting and Diagnostics\ Scripted Diagnostics	Troubleshooting: Allow users to access and run Troubleshooting wizards	N/A	**Disabled** (With this setting disabled, users can't access or run the troubleshooting tools from the Control Panel.)
Troubleshooting and Diagnostics\ Scripted Diagnostics	Troubleshooting: Allow users to access online troubleshooting content on Microsoft servers from the Troubleshooting Control Panel (via the Windows Online Troubleshooting Service – WOTS)	N/A	**Disabled** With this setting disabled, users can only access and search troubleshooting content that is available locally on their computers, even if they're connected to the Internet. They're prevented from connecting to the Microsoft servers that host the Windows Online Troubleshooting Service.
Troubleshooting and Diagnostics\ Windows Boot Performance Diagnostics	Configure Scenario Execution Level	N/A	**Disabled** (Determines the execution level for Windows Boot Performance Diagnostics. If you disable this policy setting, Windows can't detect, troubleshoot or resolve any Windows Boot Performance problems that are handled by the DPS.) This setting can be useful during design, test, development, or maintenance phases. This setting could be enabled on an isolated VM or session host, measurements taken, and results noted in event logs under "Microsoft-Windows-Diagnostics-Performance/Operational" Source: Diagnostics-Performance, Task

Policy setting	Item	Sub-item	Possible setting and comments
			Category "Boot Performance Monitoring." **ALSO**: With the DPS service disabled, this setting has no effect, as Windows doesn't log performance data.
Troubleshooting and Diagnostics\ Windows Memory Leak Diagnostics	Configure Scenario Execution Level	N/A	**Disabled** (This policy setting determines whether Diagnostic Policy Service (DPS) diagnoses memory leak problems. With this setting disabled, the DPS isn't able to diagnose memory leak problems.) Many diagnostics modes can be enabled, and tools used such as WPT, though these are done in dev/test/maintenance scenarios and not enabled and used on production VMs or sessions
Troubleshooting and Diagnostics\ Windows Performance PerfTrack	Enable/Disable PerfTrack	N/A	**Disabled** (This policy setting specifies whether to enable or disable tracking of responsiveness events. With this setting disabled, responsiveness events aren't processed.)
Troubleshooting and Diagnostics\ Windows Resource Exhaustion Detection and Resolution	Configure Scenario Execution Level	N/A	**Disabled** (Determines the execution level for Windows Resource Exhaustion Detection and Resolution. With this setting disabled, Windows can't detect, troubleshoot or resolve any Windows Resource Exhaustion problems that are handled by the DPS.)
Troubleshooting and Diagnostics\ Windows Shutdown Performance Diagnostics	Configure Scenario Execution Level	N/A	**Disabled** (Determines the execution level for Windows Shutdown Performance Diagnostics. With this setting disabled, Windows can't detect, troubleshoot or resolve any Windows Shutdown Performance problems that are handled by the DPS.)
Troubleshooting and Diagnostics\ Windows Standby/Resume Performance Diagnostics	Configure Scenario Execution Level	N/A	**Disabled** (Determines the execution level for Windows Standby/Resume Performance Diagnostics. With this setting disabled, Windows can't detect, troubleshoot or resolve any Windows Standby/Resume Performance problems that are handled by the DPS.)
Troubleshooting and Diagnostics\ Windows System Responsiveness Performance Diagnostics	Configure Scenario Execution Level	N/A	**Disabled** (Determines the execution level for Windows System Responsiveness Diagnostics. With this setting disabled, Windows can't detect, troubleshoot or resolve any Windows System Responsiveness problems that are handled by the DPS.)
*User Profiles	Turn off the advertising ID	N/A	**Enabled** (With this setting enabled, the advertising ID is turned off. Apps can't use the ID for experiences across apps)

Policy setting	Item	Sub-item	Possible setting and comments
Local Computer Policy \ Computer Configuration \ Administrative Templates \ Windows Components	N/A	N/A	N/A
*App Privacy	Let Windows apps access diagnostic information about other apps	Default for all apps: Force Deny	**Enabled** (With this setting enabled, and using the "Force Deny" option, Windows apps aren't allowed to get diagnostic information about other apps and employees in your organization can't change it.)
*App Privacy	Let Windows apps access location	Default for all apps: Force Deny	**Enabled** With this setting enabled, and using the "Force Deny" option, Windows apps aren't allowed to access location and users can't change the setting.
*App Privacy	Let Windows apps access motion	Default for all apps: Force Deny	**Enabled** (With this setting enabled, and using the "Force Deny" option, Windows apps aren't allowed to access motion data and users can't change the setting.)
*App Privacy	Let Windows apps access notifications	Default for all apps: Force Deny	**Enabled** (With this setting enabled, and using the "Force Deny" option, Windows apps aren't allowed to access notifications and users can't change the setting)
*App Privacy	Let Windows apps activate with voice	Default for all apps: Force Deny	**Enabled** (This policy setting specifies whether Windows apps can be activated by voice.)
*App Privacy	Let Windows apps activate with voice while the system is locked	Default for all apps: Force Deny	**Enabled** (This policy setting specifies whether Windows apps can be activated by voice while the system is locked.)
*App Privacy	Let Windows apps control radios	Default for all apps: Force Deny	**Enabled** (If you choose the "Force Deny" option, Windows apps don't have access to control radios and employees in your organization can't change it)
Application Compatibility	Turn off Inventory Collector	N/A	**Enabled** (This policy setting controls the state of the Inventory Collector. The Inventory Collector inventories applications, files, devices, and drivers on the system and sends the information to Microsoft. With this policy setting enabled, the Inventory Collector is turned off and data isn't sent to Microsoft. Collection of installation data through the Program Compatibility Assistant is also disabled.)
AutoPlay Policies	Set the default behavior for AutoRun	Don't execute any autorun commands	**Enabled** (This policy setting sets the default behavior for Autorun commands.)

Policy setting	Item	Sub-item	Possible setting and comments
*AutoPlay Policies	Turn off Autoplay	All drives	**Enabled** (If you enable this policy setting, Autoplay is disabled on all drives.)
*Cloud Content	Don't show Windows tips	N/A	**Enabled** (This policy setting prevents Windows tips from being shown to users)
*Cloud Content	Turn off Microsoft consumer experiences	N/A	**Enabled** (With this policy setting enabled, users don't see personalized recommendations from Microsoft and notifications about their Microsoft account)
*Data Collection and Preview Builds	Allow Telemetry	0 – Security [Enterprise Only]	**Enabled** (Setting a value of 0 applies to devices running Enterprise, Education, IoT, or Windows Server editions only, and reduces telemetry sent to the most basic level supported)
Data Collection and Preview Builds	Configure collection of browsing data for Desktop Analytics	Configure telemetry collection: Don't allow sending intranet or internet history	**Enabled** (You can configure Microsoft Edge to send intranet history only, internet history only, or both to Desktop Analytics for enterprise devices with a configured Commercial ID. If disabled or not configured, Microsoft Edge doesn't send browsing history data to Desktop Analytics.)
*Data Collection and Preview Builds	Don't show feedback notifications	N/A	**Enabled** (This policy setting allows an organization to prevent its devices from showing feedback questions from Microsoft.)
Delivery Optimization	Download Mode	Download Mode: Simple (99)	**Enabled** (99 = Simple download mode with no peering. Delivery Optimization downloads using HTTP only and doesn't attempt to contact the Delivery Optimization cloud services.)
Desktop Window Manager	Don't allow window animations	N/A	**Enabled** (This policy setting controls the appearance of window animations such as those found when restoring, minimizing, and maximizing windows. With this policy setting enabled, window animations are turned off.)
Desktop Window Manager	Use solid color for Start background	N/A	**Enabled** (This policy setting controls the Start background visuals. With this policy setting enabled, the Start background uses a solid color.)
Edge UI	Allow edge swipe	N/A	**Disabled** (If you disable this policy setting, users can't invoke any system UI by swiping in from any screen edge.)
Edge UI	Disable help tips	N/A	**Enabled** (If this setting is enabled, Windows doesn't show any help tips to the user.)
File Explorer	Don't show the "new application installed"	N/A	**Enabled** (This policy removes the end-user notification for new application

Policy setting	Item	Sub-item	Possible setting and comments
	notification		associations. These associations are based on file types (for example, TXT files) or protocols (for example, HTTP). If this policy is enabled, no notifications are shown to the end-user)
File History	Turn off File History	N/A	**Enabled** (With this policy setting enabled, File History can't be activated to create regular, automatic backups.)
*Find My Device	Turn On/Off Find My Device	N/A	**Disabled** (When Find My Device is off, the device and its location aren't registered, and the "Find My Device" feature doesn't work. The user can't view the location of the last use of their active digitizer on their device.)
Homegroup	Prevent the computer from joining a homegroup	N/A	**Enabled** (If you enable this policy setting, users can't add computers to a homegroup. This policy setting doesn't affect other network sharing features.)
Internet Information Services	Prevent IIS installation	N/A	**Enabled** (With this policy setting enabled, IIS can't be installed, and you can't install Windows components or applications that require IIS.)
*Location and Sensors	Turn off location	N/A	**Enabled** (With this setting enabled, the location feature is turned off, and all programs on this device are prevented from using location information from the location feature)
Location and Sensors	Turn off sensors	N/A	**Enabled** (This policy setting turns off the sensor feature for this device. With this policy setting enabled, the sensor feature is turned off, and all programs on this computer can't use the sensor feature.)
Locations and Sensors / Windows Location Provider	Turn off Windows Location Provider	N/A	**Enabled** (This policy setting turns off the Windows Location Provider feature for this device.)
*Maps	Turn off Automatic Download and Update of Map Data	N/A	**Enabled** (With this setting enabled, the automatic download and update of map data is turned off.)
*Maps	Turn off unsolicited network traffic on the Offline Maps settings page	N/A	**Enabled** (With this setting enabled, features that generate network traffic on the Offline Maps settings page are turned off. Note: This may turn off the entire settings page)
*Messaging	Allow Message Service Cloud Sync	N/A	**Disabled** (This policy setting allows backup and restore of cellular text messages to Microsoft's cloud services.)
*Microsoft Edge	Allow configuration updates for the Books Library	N/A	**Disabled** (With this setting disabled, Microsoft Edge doesn't automatically

Policy setting	Item	Sub-item	Possible setting and comments
			download updated configuration data for the Books Library.)
*Microsoft Edge	Allow extended telemetry for the Books tab	N/A	**Disabled** (With this setting disabled, Microsoft Edge only sends basic telemetry data, depending on your device configuration.)
Microsoft Edge	Allow Microsoft Edge to pre-launch at Windows startup, when the system is idle, and each time Microsoft Edge is closed	Configure pre-launch: Prevent pre-launching	**Enabled** (With this setting enabled and configured to prevent pre-launch, Microsoft Edge won't pre-launch during Windows sign in, when the system is idle, or each time Microsoft Edge is closed.)
Microsoft Edge	Allow Microsoft Edge to start and load the Start and New Tab page at Windows startup and each time Microsoft Edge is closed	Configure tab preloading: Prevent tab-preloading	**Enabled** (This policy setting lets you decide whether Microsoft Edge can load the Start and New Tab page during Windows sign in and each time Microsoft Edge is closed. By default this setting is to allow preloading. With preloading disabled, Microsoft Edge won't load the Start or New Tab page during Windows sign in and each time Microsoft Edge is closed.)
Microsoft Edge	Allow web content on New Tab page	N/A	**Disabled** (With this setting disabled, Edge opens a new tab with a blank page. If this setting is configured, users can't change the setting.)
*Microsoft Edge	Prevent the First Run webpage from opening on Microsoft Edge	N/A	**Enabled** (users won't see the First Run page when opening Microsoft Edge for the first time)
OneDrive	Prevent OneDrive from generating network traffic until the user signs in to OneDrive	N/A	**Enabled** (Enable this setting to prevent the OneDrive sync client (OneDrive.exe) from generating network traffic (checking for updates, and so on.) until the user signs in to OneDrive or starts syncing files to the local computer)
Online Assistance	Turn off Active Help	N/A	**Enabled** (With this policy setting enabled, active content links aren't rendered. The text is displayed, but there are no clickable links for these elements.)
OOBE	Don't launch privacy settings experience on user logon	N/A	**Enabled** (When logging into a new user account for the first time or after an upgrade in some scenarios, that user may be presented with a screen or series of screens that prompts the user to choose privacy settings for their account. Enable this policy to prevent this experience from launching.)
RSS Feeds	Prevent automatic discovery of feeds and Web Slices	N/A	**Enabled** (This policy setting prevents users from having Microsoft Edge automatically discover whether a feed or

Policy setting	Item	Sub-item	Possible setting and comments
			Web Slice is available for an associated webpage.)
*RSS Feeds	Turn off background synchronization for feeds and Web Slices	N/A	**Enabled** (With this policy setting enabled, the ability to synchronize feeds and Web Slices in the background is turned off.)
*Search	Allow Cortana	N/A	**Disabled** (This policy setting specifies whether Cortana is allowed on the device. When Cortana is off, users are able to use search to find things on the device.)
Search	Allow Cortana above lock screen	N/A	**Disabled** (This policy setting determines whether or not the user can interact with Cortana using speech while the system is locked.)
*Search	Allow search and Cortana to use location	N/A	**Disabled** (This policy setting specifies whether search and Cortana can provide location aware search and Cortana results.)
Search	Control rich previews for attachments	Control Rich Previews for Attachments:**.docx;.xlsx;.txt;.xls**	**Enabled** (Enabling this policy defines a semicolon-delimited list of file extensions which are allowed to have rich attachment previews.) **NOTE**: This setting can be used to limit what types of attachments are previewed, which can also help prevent automatically previewing some potentially dangerous contents types.
Search	Don't allow web search	N/A	**Enabled** (Enabling this policy removes the option of searching the Web from Windows Desktop Search.)
*Search	Don’t search the web or display web results in Search	N/A	**Enabled** (With this policy setting enabled, queries aren't performed on the web and web results aren't displayed when a user performs a query in Search.)
Search	Enable indexing uncached Exchange folders	N/A	**Disabled** (Enabling this policy allows indexing of mail items on a Microsoft Exchange server when Microsoft Outlook isn't running in cached mode. The default behavior for search is to not index uncached Exchange folders. Disabling this policy blocks any indexing of uncached Exchange folders.)
Search	Prevent indexing files in offline files cache	N/A	**Enabled** (If enabled, files on network shares made available offline aren't indexed. Otherwise they're indexed. Disabled by default.)
*Search	Set what information is shared in Search	Anonymous info	**Enabled** (Anonymous info: Share usage information but don't share search

Policy setting	Item	Sub-item	Possible setting and comments
			history, Microsoft account info, or specific location)
Search	Stop indexing if there's limited hard drive space	MB Limit: **5000**	**Enabled** (Enabling this policy prevents indexing from continuing after less than the specified amount of hard drive space is left on the same drive as the index location. Select between 0 and 2147483647 MB.)
Software Protection Platform	Turn off KMS Client Online AVS Validation	N/A	**Enabled** (With this setting enabled, the device doesn't send data to Microsoft regarding its activation state)
*Speech	Allow Automatic Update of Speech Data	N/A	**Disabled** (Specifies whether the device receives updates to the speech recognition and speech synthesis models.)
Store	Turn off the offer to update to the latest version of Windows	N/A	**Enabled** (Enables or disables the Store offer to update to the latest version of Windows. If you enable this setting, the Store application doesn't offer updates to the latest version of Windows.)
Text Input	Improve inking and typing recognition	N/A	**Disabled** (This policy setting controls the ability to send inking and typing data to Microsoft to improve the language recognition and suggestion capabilities of apps and services running on Windows.)
Windows Error Reporting	Disable Windows Error Reporting	N/A	**Enabled** (With this policy setting enabled, Windows Error Reporting doesn't send any problem information to Microsoft. And solution information isn't available in Security and Maintenance in Control Panel.)
Windows Game Recording and Broadcasting	Enables or disables Windows Game Recording and Broadcasting	N/A	**Disabled** (With this setting disabled, Windows Game Recording aren't allowed.)
Windows Ink Workspace	Allow Windows Ink Workspace	Choose one of the following actions: Disabled	**Enabled** (With this setting enabled and sub-setting set to disabled, Windows Ink Workspace functionality is unavailable.)
Windows Installer	Control maximum size of baseline file cache	5	**Enabled** (This policy controls the percentage of disk space available to the Windows Installer baseline file cache. With this policy setting enabled, you can modify the maximum size of the Windows Installer baseline file cache.)
Windows Installer	Turn off creation of System Restore checkpoints	N/A	**Enabled** (With this policy setting enabled, the Windows Installer doesn't generate System Restore checkpoints when installing applications.)

Policy setting	Item	Sub-item	Possible setting and comments
Windows Mobility Center	Turn off Windows Mobility Center	N/A	**Enabled** (With this policy setting enabled, the user is unable to invoke Windows Mobility Center. The Windows Mobility Center UI is removed from all shell entry points and the .exe file doesn't launch it.)
Windows Reliability Analysis	Configure Reliability WMI Providers	N/A	**Disabled** (With this policy setting disabled, Reliability Monitor doesn't display system reliability information, and WMI-capable applications are unable to access reliability information from the listed providers.)
Windows Security \ Notifications	Hide noncritical notifications	N/A	**Enabled** (With this setting enabled, local users only see critical notifications from Windows Security. They don't see other types of notifications, such as regular PC or device health information.)
Windows Update	Turn on Software Notifications	N/A	**Disabled** (This policy setting allows you to control whether users see detailed enhanced notification messages about featured software from the Microsoft Update service. Enhanced notification messages convey the value and promote the installation and use of optional software. This policy setting is intended for use in loosely managed environments in which you allow the end user access to the Microsoft Update service.)
*Windows Update\ Windows Update for Business	Manage preview builds	Set the behavior for receiving preview builds: **Disable preview builds**	**Enabled** (Selecting "Disable preview builds" prevents preview builds from installing on the device. This prevents users from opting into the Windows Insider Program, through Settings -> Update and Security)
*Windows Update\ Windows Update for Business	Select when Preview Builds and Feature Updates are received	Select the Windows readiness level for the updates you want to receive: **Semi-Annual Channel** After a Preview Build or Feature Update is released, defer receiving it for this many days: **365** Pause Preview Builds or Feature Updates starting: **yyyy-mm-dd**	**Enabled** (Enable this policy to specify the level of Preview Build or Feature Updates to receive, and when. Semi-Annual Channel: Receive feature updates when they're released to the general public. When Selecting Semi-Annual Channel: - You can defer receiving Feature Updates for up to 365 days. - To prevent Feature Updates from being received on their scheduled time, you can temporarily pause them. The pause remains in effect for 35 days from the start time provided. - To resume receiving Feature Updates that are paused, clear the start date field.)

Policy setting	Item	Sub-item	Possible setting and comments
Windows Update\ Windows Update for Business	Select when Quality Updates are received	After a quality update is released, defer receiving it for this many days: **30** Pause Quality Updates starting: yyyy-mm-dd	**Enabled** (Enable this policy to specify when to receive quality updates. You can defer receiving quality updates for up to 30 days. To prevent quality updates from being received on their scheduled time, you can temporarily pause quality updates. The pause remains in effect for 35 days or until you clear the start date field. To resume receiving Quality Updates that are paused, clear the start date field.) This recommendation is to help control when updates are applied, and to ensure updates don’t get offered and installed unexpectedly
Local Computer Policy \ User Configuration \ Administrative Templates	N/A	N/A	N/A
Control Panel\ Regional and Language Options	Turn off offer text predictions as I type	N/A	**Enabled** (This policy turns off the offer text predictions as I type option. This doesn't, however, prevent the user or an application from changing the setting programmatically. With this policy setting enabled, the option is locked to not offer text predictions.)
Desktop	Don't add shares of recently opened documents to Network Locations	N/A	**Enabled** (With this setting enabled, shared folders aren't added to Network Locations automatically when you open a document in the shared folder.)
Desktop	Turn off Aero Shake window minimizing mouse gesture	N/A	**Enabled** (Prevents windows from being minimized or restored when the active window is shaken back and forth with the mouse. With this policy enabled, application windows aren't minimized or restored when the active window is shaken back and forth with the mouse.)
Desktop / Active Directory	Maximum size of Active Directory searches	Number of objects returned:**1500**	**Enabled** (Specifies the maximum number of objects the system displays in response to a command to browse or search Active Directory. This setting affects all browse displays associated with Active Directory, such as those in Local Users and Groups, Active Directory Users and Computers, and dialog boxes used to set permissions for user or group objects in Active Directory.)
Start Menu and Taskbar	Don't display or track items in Jump Lists	N/A	**Enabled** (This policy setting allows you to control displaying or tracking items in

Policy setting	Item	Sub-item	Possible setting and comments
	from remote locations		Jump Lists from remote locations.)
Start Menu and Taskbar	Don't search Internet	N/A	**Enabled** (With this policy setting enabled, the Start Menu search box doesn't search for internet history or favorites.)
Start Menu and Taskbar	Don't use the search-based method when resolving shell shortcuts	N/A	**Enabled** (This policy setting prevents the system from conducting a comprehensive search of the target drive to resolve a shortcut.)
Start Menu and Taskbar	Turn off all balloon notifications	N/A	**Enabled** (With this policy setting enabled, no notification balloons are shown to the user.)
Start Menu and Taskbar	Turn off feature advertisement balloon notifications	N/A	**Enabled** (With this policy setting enabled, certain notification balloons that are marked as feature advertisements aren't shown.)
Start Menu and Taskbar	Turn off user tracking	N/A	**Enabled** (With this policy setting enabled, the system doesn't track the programs that the user runs and doesn't display frequently used programs in the Start Menu.)
Start Menu and Taskbar / Notifications	Turn off toast notifications	N/A	**Enabled** (With this policy setting enabled, applications can't raise toast notifications.)
*Start Menu and Taskbar / Notifications	Turn off toast notifications on the lock screen	N/A	**Enabled** (With this policy setting enabled, applications can't raise toast notifications on the lock screen.)
Local Computer Policy / User Configuration	N/A	N/A	N/A
Windows Components / Cloud Content	Configure Windows spotlight on lock screen	N/A	**Disabled** (With this policy disabled, Windows spotlight is turned off and users can't select it as their lock screen. Users see the default lock screen image and are able to select another image, unless you have enabled the "Prevent changing lock screen image" policy.)
*Windows Components / Cloud Content	Don't suggest third-party content in Windows spotlight	N/A	**Enabled** (With this policy enabled, Windows spotlight features like lock screen spotlight, suggested apps in Start menu or Windows tips doesn't suggest apps and content from third-party software publishers. Users may still see suggestions and tips to make them more productive with Microsoft features and apps.)
Windows Components / Cloud Content	Don't use diagnostic data for tailored experiences	N/A	**Enabled** (With this policy setting enabled, Windows doesn't use diagnostic data from this device (this data may include browser, app and feature usage, depending on the "diagnostic data"

Policy setting	Item	Sub-item	Possible setting and comments
			setting value) to customize content shown on lock screen, Windows tips, Microsoft consumer features, and other related features.)
Windows Components / Cloud Content	Turn off all Windows spotlight features	N/A	**Enabled** (Windows spotlight on lock screen, Windows tips, Microsoft consumer features, and other related features are turned off. You should enable this policy setting if your goal is to minimize network traffic from target devices.)
Edge UI	Turn off tracking of app usage	N/A	**Enabled** (This policy setting prevents Windows from keeping track of the apps that are used and searched most frequently. If you enable this policy setting, apps are sorted alphabetically in: - search results - the Search and Share panes - the drop-down app list in the Picker)
File Explorer	Turn off caching of thumbnail pictures	N/A	**Enabled** (With this policy setting enabled, thumbnail views aren't cached.)
File Explorer	Turn off common control and window animations	N/A	**Enabled** (Disabling animations can improve usability for users with some visual disabilities and improve performance and battery life in some scenarios.)
File Explorer	Turn off display of recent search entries in the File Explorer search box	N/A	**Enabled** (Disables suggesting recent queries for the Search Box and prevents entries into the Search Box from being stored in the registry for future references.)
File Explorer	Turn off the caching of thumbnails in hidden thumbs.db files	N/A	**Enabled** (With this policy setting enabled, File Explorer doesn't create, read from, or write to thumbs.db files.)

* Comes from the Windows Restricted Traffic Limited Functionality Baseline.

System services

If you're considering disabling system services to conserve resources, make sure the service isn't a component of some other service. In this paper and with the available GitHub scripts, some services aren't in the list because they can't be disabled in a supported manner.

Most of these recommendations mirror recommendations for Windows Server 2016, installed with the Desktop Experience, based on the instructions in Guidance on disabling system services on Windows Server 2016 with Desktop Experience.

Many services that may seem like good candidates to disable are set to manual service start type. This means that the service doesn't automatically start and start only if an event triggers a request to the service. Services that are

already set to start type manual aren't listed here.

> **Note**
>
> You can enumerate running services with this PowerShell sample code, outputting only the service short name:

```PowerShell
Get-Service | Where-Object {$_.Status -eq 'Running'} | Select-Object -ExpandProperty Name
```

The following table contains some services that may be considered to disable in virtual desktop environments:

Expand table

Windows Service	Service Name	Item	Comment
Cellular Time	autotimesvc	This service sets time based on NITZ messages from a Mobile Network	Virtual desktop environments may not have such devices available. To learn more, see the MB NITZ support article.
GameDVR and Broadcast user service	BcastDVRUserService	This (per-user) service is used for Game Recordings and Live Broadcasts	NOTE: This is a "per-user service", and as such, the template service must be disabled. This user service is used for Game Recordings and Live Broadcasts. To learn more, see the MB NITZ support article.
CaptureService	CaptureService	Enables optional screen capture functionality for applications that call the Windows.Graphics.Capture API.	OneCore capture service: enables optional screen capture functionality for applications that call the Windows.Graphics.Capture API For more information, see the Windows.Graphics.Capture Namespace API docs.
Connected Devices Platform Service	CDPSvc	This service is used for Connected Devices Platform scenarios	Connected Devices Platform Service. To learn more, see the Connected Devices Platform overview article
CDP User Service	CDPUserSvc	N/A	Connected Devices Platform User Service. To learn more, see the Connected Devices Platform Protocol Version 3 article. This user service is used for Connected Devices Platform scenarios This is a "per-user service", and as such, the template service must be disabled (CDPUserSvc).
Optimize drives	defragsvc	Helps the computer run more efficiently by optimizing files on storage drives.	Virtual desktop solutions don't normally benefit from disk optimization. The "drives" are often not traditional drives and often just a temporary storage allocation.
Diagnostic Execution Service	DiagSvc	Executes diagnostic actions for troubleshooting support	Disabling this service disables the ability to run Windows diagnostics Diagnostic Execution Service.
Connected User Experiences and Telemetry	DiagTrack	This service enables features that support in-application and connected user experiences. This service manages the event driven	Consider disabling if on disconnected network. To learn more, see how-to configure Windows diagnostic data in your organization.

Windows Service	Service Name	Item	Comment
		collection and transmission of diagnostic and usage information (used to improve the experience and quality of the Windows Platform) when the diagnostics and usage privacy option settings are enabled under Feedback and Diagnostics.	
Diagnostic Policy Service	DPS	The Diagnostic Policy Service enables problem detection, troubleshooting, and resolution for Windows components. If this service is stopped, diagnostics don't work.	Disabling this service disables the ability to run Windows diagnostics. For more information, see the Windows.System.Diagnostics Namespace reference.
Device Setup Manager	DsmSvc	Enables the detection, download, and installation of device-related software.	If this service is disabled, devices may be configured with outdated software, and may not work correctly. Virtual desktop environments closely control what software is installed and maintain that consistency across the environment.
Data Usage service	DusmSvc	Network data usage, data limit, restrict background data, metered networks.	For more information, see the DUSM schema.
Windows Mobile Hotspot Service	icssvc	Provides the ability to share a cellular data connection with another device.	To learn more, see the NetworkOperatorTetheringAccessPointConfiguration Class reference.
Microsoft Store Install Service	InstallService	Provides infrastructure support for the Microsoft Store.	This service is started on demand and if disabled then installations don't work properly. Consider disabling this service on non-persistent virtual desktop, leave as-is for persistent virtual desktop solutions.
Geolocation Service	Lfsvc	Monitors the current location of the system and manages geofences (a geographical location with associated events).	If you turn off this service, applications are unable to use or receive notifications for geolocation or geofences. To learn more, see the Windows.Devices.Geolocation Namespace reference.
Downloaded Maps Manager	MapsBroker	Windows service for application access to downloaded maps. This service is started on-demand by application accessing downloaded maps.	Disabling this service prevents apps from accessing maps. To learn more, see the Windows.Services.Maps Namespace API docs.
MessagingService	MessagingService	Service supporting text messaging and related functionality.	This is a "per-user service", and as such, the template service must be disabled.
Sync Host	OneSyncSvc	This service synchronizes mail, contacts, calendar, and various other user data.	(UWP) Mail and other applications dependent on this functionality don't work properly when this service isn't running.

Windows Service	Service Name	Item	Comment
			This is a "per-user service", and as such, the template service must be disabled.
Contact Data	PimIndexMaintenanceSvc	Indexes contact data for fast contact searching. If you stop or disable this service, contacts might be missing from your search results.	This is a "per-user service", and as such, the template service must be disabled.
Power	Power	Manages power policy and power policy notification delivery.	Virtual machines have virtually no influence on power properties. If this service is disabled, power management and reporting aren't available. To learn more, see the User-Mode Power Service article.
Payments and NFC/SE Manager	SEMgrSvc	Manages payments and Near Field Communication (NFC) based secure elements.	May not need this service for payments, in the enterprise environment.
Microsoft Windows SMS Router Service	SmsRouter	Routes messages based on rules to appropriate clients.	May not need this service, if other tools are used for messaging, such as Teams. To learn more, see this routing service article.
Superfetch (SysMain)	SysMain	Maintains and improves system performance over time.	Superfetch generally doesn't improve performance in virtual desktop environments for various reasons. The underlying storage is often virtualized and possibly striped across multiple drives. In some virtual desktop solutions, the accumulated user state is discarded when the user logs off. The SysMain feature should be evaluated in each environment.
Update Orchestrator Service	UsoSvc	Manages Windows Updates. If stopped, your devices can't download and install the latest updates.	Virtual desktop devices are often carefully managed with respect to updates. Servicing is performed during maintenance windows. In some cases, an update client may be utilized, such as SCCM. The exception is for security signature updates that are applied at any time, and to any virtual desktop device, in order to maintain up-to-date signatures. If you disable this service, test to ensure that security signatures can still be installed.
Volume Shadow Copy	VSS	Manages and implements Volume Shadow Copies used for backup and other purposes.	If this service is stopped, shadow copies are unavailable for backup and the backup may fail. If this service is disabled, any services that explicitly depend on it fail to start. To learn more, see this volume shadow copy service article.
Diagnostic System Host	WdiSystemHost	The Diagnostic System Host is used by the Diagnostic Policy Service to host diagnostics that need to run in a Local System context. If this service is stopped, any diagnostics that depend on it doesn't function.	Disabling this service disables the ability to run Windows diagnostics
Windows Error Reporting	WerSvc	Allows errors to be reported when programs stop working or responding and allows	With virtual desktop environments, diagnostics are often performed in an "offline" scenario, and not in mainstream production. In addition, some

Windows Service	Service Name	Item	Comment
		existing solutions to be delivered. Also allows logs to be generated for diagnostic and repair services. If this service is stopped, error reporting might not work correctly, and results of diagnostic services and repairs might not be displayed.	customers disable WER anyway. WER incurs a tiny amount of resources for many different things, including failure to install a device, or failure to install an update. To learn more, see Windows Error Reporting.
Windows Search	WSearch	Provides content indexing, property caching, and search results for files, e-mail, and other content.	Disabling this service prevents indexing of e-mail and other things. Test before disabling this service. To learn more, see Windows search service overview.
Xbox Live Auth Manager	XblAuthManager	Provides authentication and authorization services for interacting with Xbox Live.	If this service is stopped, some applications may not operate correctly.
Xbox Live Game Save	XblGameSave	This service syncs save data for Xbox Live save enabled games.	If this service is stopped, game save data doesn't upload to or download from Xbox Live.
Xbox Accessory Management Service	XboxGipSvc	This service manages connected Xbox Accessories.	N/A
Xbox Live Networking Service	XboxNetApiSvc	This service supports the Windows.Networking.XboxLive application programming interface.	N/A

Per-user services in Windows

Per-user services are services created when a user signs into Windows or Windows Server and stopped and deleted when that user signs out. These services run in the security context of the user account - this provides better resource management than the previous approach of running these kinds of services in Explorer, associated with a preconfigured account, or as tasks. For more information, see Per-user services in Windows.

Scheduled tasks

Like other items in Windows, ensure an item isn't needed before disabling a scheduled task. Some tasks in virtual desktop environments, such as **StartComponentCleanup**, may not be desirable to run in production, but may be good to run during a maintenance window on the "gold image" (reference image).

The following list of tasks includes tasks that perform optimizations or data collections on computers that maintain their state across reboots. When a virtual desktop device reboots and discards all changes since last boot, optimizations intended for physical computers aren't helpful.

You can get all the current scheduled tasks, including descriptions, with the following PowerShell code:

```
PowerShell
```

```
Get-ScheduledTask | Select-Object -Property TaskPath,TaskName,State,Description
```

> **Note**
>
> There are several tasks that can't be disabled with a script, even when run on an elevated command prompt. The recommendations here, and in the GitHub scripts don't attempt to disable tasks that can't be disabled with a script.

Expand table

Scheduled Task Name	Description
MNO	Mobile broadband account experience metadata parser
AnalyzeSystem	This task analyzes the system looking for conditions that may cause high energy use
Cellular	Related to cellular devices
Compatibility	Collects program telemetry information if opted-in to the Microsoft Customer Experience Improvement Program.
Consolidator	If the user consents to participate in the Windows Customer Experience Improvement Program, this job collects and sends usage data to Microsoft
Diagnostics	(DiskFootprint in task path) 'DiskFootprint' is the combined contribution of all processes that issue storage I/O in the form of storage reads, writes, and flushes.
FamilySafetyMonitor	Initializes Family Safety monitoring and enforcement.
FamilySafetyRefreshTask	Synchronizes the latest settings with the Microsoft family features service.
MapsToastTask	This task shows various Map-related toasts
Microsoft-Windows-DiskDiagnosticDataCollector	The Windows Disk Diagnostic reports general disk and system information to Microsoft for users participating in the Customer Experience Program.
NotificationTask	Background task for performing per user and web interactions
ProcessMemoryDiagnosticEvents	Schedules a memory diagnostic in response to system events
Proxy	This task collects and uploads autochk SQM data if opted-in to the Microsoft Customer Experience Improvement Program.
QueueReporting	Windows Error Reporting task to process queued reports.
RecommendedTroubleshootingScanner	Check for recommended troubleshooting from Microsoft
RegIdleBackup	Registry Idle Backup Task
RunFullMemoryDiagnostic	Detects and mitigates problems in physical memory (RAM).
Scheduled	The Windows Scheduled Maintenance Task performs periodic maintenance of the computer system by fixing problems automatically or reporting them through Security and Maintenance.
ScheduledDefrag	This task optimizes local storage drives.
SilentCleanup	Maintenance task used by the system to launch a silent auto disk cleanup when running low on free disk space.

Scheduled Task Name	Description
SpeechModelDownloadTask	
Sqm-Tasks	This task gathers information about the Trusted Platform Module (TPM), Secure Boot, and Measured Boot.
SR	This task creates regular system protection points.
StartComponentCleanup	Servicing task that may be better performed during maintenance windows
StartupAppTask	Scans startup entries and raises notification to the user if there are too many startup entries.
SyspartRepair	
WindowsActionDialog	Location Notification
WinSAT	Measures a system's performance and capabilities
XblGameSaveTask	Xbox Live GameSave standby task

Apply Windows (and other) updates

Whether from Microsoft Update, or from your internal resources, apply available updates including Windows Defender signatures. This is a good time to apply other available updates including Microsoft Office if installed, and other software updates. If PowerShell remains in the image you can download the latest available help for PowerShell by running the command `Update-Help`.

Servicing OS and apps

At some point during the image optimization process, available Windows updates should be applied. There's a setting in Windows update settings that can provide more updates. You can find it at **Settings** > **Advanced options**. Once there, set **Give me updates for other uMirosoft products when I update Windows** to **On**.

This would be a good setting in case you're going to install Microsoft applications such as Microsoft Office to the base image. That way Office is up to date when the image is put in service. There are also .NET updates and certain third-party components such as Adobe that have updates available through Windows Update.

One important consideration for non-persistent virtual desktop devices is security updates, including security software definition files. These updates may be released once or more times per day.

For Windows Defender it may be best to allow the updates to occur, even on non-persistent virtual desktop environments. The updates are going to apply nearly every time you sign in, but the updates are small and shouldn't be a problem. Plus, the device won't be behind on updates because only the latest available applies. The same may be true for third-party definition files.

> ⓘ **Note**
>
> Store apps (UWP apps) update through the Windows Store. Modern versions of Office such as Office 365 update through their own mechanisms when directly connected to the Internet, or through management technologies when not.

Windows system startup event traces (AutoLoggers)

Windows is configured by default to collect and save diagnostic data. The purpose is to enable diagnostics, or to record data if further troubleshooting is necessary. Automatic system traces can be found opening **Computer Management** and navigating to **System Tools** > **Performance** > **Data Collector Sets**.

Some of the traces displayed under **Event Trace Sessions** and **Startup Event Trace Sessions** can't and shouldn't be stopped. Others, such as the WiFiSession trace can be stopped. To stop a running trace under **Event Trace Sessions**, right-click the trace and then select **Stop**. Use the following procedure to prevent the traces from starting automatically on startup:

1. Select the **Startup Event Trace Sessions** folder.
2. Find and select the trace file you want to look at to open it.
3. Select the **Trace Session** tab.
4. Uncheck the box labeled **Enabled**.
5. Select **Ok**.

The following table lists some system traces that you should consider disabling in your virtual desktop environments:

Expand table

Name	Comment
Cellcore	Cellular Architecture documentation
CloudExperienceHostOOBE	Plan a Windows Hello for Business deployment.
DiagLog	A log generated by the Diagnostic Policy Service, which is documented in Guidance on disabling system services with Desktop Experience
RadioMgr	Near-field communication (NFC) device drivers
ReadyBoot	ReadyBoot Analysis.
WDIContextLog	WDI Miniport Driver Design Guide.
WiFiDriverIHVSession	User-initiated feedback - normal mode.
WiFiSession	Diagnostic log for WLAN technology. If Wi-Fi isn't implemented, there's no need for this logger
WinPhoneCritical	Diagnostic log for phone (Windows?). If not using phones, no need for this logger

Windows Defender optimization in the virtual desktop environment

For more details about how to optimize Windows Defender in a virtual desktop environment, check out the Deployment guide for Windows Defender Antivirus in a virtual desktop infrastructure (VDI) environment.

The deployment guide contains procedures to service the "gold" virtual desktop image, and how to maintain the virtual desktop clients as they're running. To reduce network bandwidth when virtual desktop devices need to update their Windows Defender signatures, stagger reboots, and schedule reboots during off hours where possible. The Windows Defender signature updates can be contained internally on file shares, and where practical, have those files shares on the same or close networking segments as the virtual desktop devices.

Client network performance tuning by registry settings

There are some registry settings that can increase network performance. This is especially important in environments where the virtual desktop device or physical computer has a workload that is primarily network-based. The settings in this section are recommended to tune performance for the networking workload profile, by setting up extra buffering and caching of things like directory entries and so on.

> ⓘ **Note**
>
> Some settings in this section are registry-based only and should be incorporated in the base image before the image is deployed for production use.

The following settings are documented in Performance tuning guidelines for Windows Server.

DisableBandwidthThrottling

`HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\DisableBandwidthThrottling`

Applies to Windows 10 and Windows 11. The default is **0**. By default, the SMB redirector throttles throughput across high-latency network connections, in some cases to avoid network-related timeouts. Setting this registry value to **1** disables this throttling, enabling higher file transfer throughput over high-latency network connections. Consider setting this value to **1**.

FileInfoCacheEntriesMax

`HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\FileInfoCacheEntriesMax`

Applies to Windows 10 and Windows 11. The default is **64**, with a valid range of 1 to 65536. This value is used to determine the amount of file metadata that can be cached by the client. Increasing the value can reduce network traffic and increase performance when many files are accessed. Try increasing this value to **1024**.

DirectoryCacheEntriesMax

`HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\DirectoryCacheEntriesMax`

Applies to Windows 10 and Windows 11. The default is **16**, with a valid range of 1 to 4096. This value is used to determine the amount of directory information that can be cached by the client. Increasing the value can reduce network traffic and increase performance when large directories are accessed. Consider increasing this value to **1024**.

FileNotFoundCacheEntriesMax

`HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\FileNotFoundCacheEntriesMax`

Applies to Windows 10 and Windows 11. The default is **128**, with a valid range of 1 to 65536. This value is used to determine the amount of file name information that can be cached by the client. Increasing the value can reduce network traffic and increase performance when many file names are accessed. Consider increasing this value to **2048**.

DormantFileLimit

`HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\DormantFileLimit`

Applies to Windows 10 and Windows 11. The default is **1023**. This parameter specifies the maximum number of files that should be left open on a shared resource after the application has closed the file. Where many thousands of clients are connecting to SMB servers, consider reducing this value to **256**.: Windows Server 2022, Windows Server 2019,

You can configure many of these SMB settings by using the Set-SmbClientConfiguration and Set-SmbServerConfiguration Windows PowerShell cmdlets. Registry-only settings can be configured by using Windows PowerShell as well, as in the following example:

PowerShell

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters"
RequireSecuritySignature -Value 0 -Force
```

More settings from the Windows Restricted Traffic Limited Functionality Baseline guidance

Microsoft has released a baseline, created using the same procedures as the Windows Security Baselines, for environments that are either not connected directly to the Internet, or wish to reduce data sent to Microsoft and other services.

The Windows Restricted Traffic Limited Functionality Baseline settings are called out in the group policy table with an asterisk.

Disk cleanup

Disk cleanup can be especially helpful with gold/master image virtual desktop implementations. After the gold/master image is prepared, updated, and configured, one of the last tasks to perform is disk cleanup. The optimization scripts on Github.com have PowerShell code to perform common disk cleanup tasks

> ⓘ **Note**
>
> Disk cleanup settings and are in the Settings category "System" called "Storage." By default, Storage Sense runs when a low disk free space threshold is reached.
>
> To learn more about how to use Storage Sense with Azure custom VHD images, see **Prepare and customize a master VHD image**.
>
> For Azure Virtual Desktop session host that use Windows Enterprise or Windows Enterprise multi-session, we recommend disabling Storage Sense. You can disable Storage Sense in the Settings menu under **Storage**.

Here are suggestions for various disk cleanup tasks. These should all be tested before implementing:

1. Storage Sense may be utilized manually or automatically. For more information on Storage Sense, see Manage drive space with Storage Sense.
2. Manually cleanup temporary files and logs. From an elevated command prompt, run these commands:
   a. `Del C:\*.tmp /s`
   b. `C:\*.etl /s`

c. `C:\*.evtx /s`

```PowerShell
Get-ChildItem -Path c:\ -Include *.tmp, *.dmp, *.etl, *.evtx, thumbcache*.db, *.log -File -Recurse -Force -ErrorAction SilentlyContinue | Remove-Item -ErrorAction SilentlyContinue

Remove-Item -Path $env:ProgramData\Microsoft\Windows\WER\Temp\* -Recurse -Force -ErrorAction SilentlyContinue

Remove-Item -Path $env:ProgramData\Microsoft\Windows\WER\ReportArchive\* -Recurse -Force -ErrorAction SilentlyContinue

Remove-Item -Path $env:ProgramData\Microsoft\Windows\WER\ReportQueue\* -Recurse -Force -ErrorAction SilentlyContinue

Clear-RecycleBin -Force -ErrorAction SilentlyContinue

Clear-BCCache -Force -ErrorAction SilentlyContinue
```

3. Delete any unused profiles on the system by running the following command:

```
wmic path win32_UserProfile where LocalPath="C:\\users\\<users>" Delete
```

For any questions or concerns about the information in this paper, contact your Microsoft account team, research the Microsoft virtual desktop IT Pro blog, post a message to Microsoft Virtual Desktop forums, or contact Microsoft for questions or concerns.

Re-enable Windows Update

If you'd like to enable the use of Windows Update after disabling it, follow these steps:

1. Re-enable group policy settings:
   - Go to **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **System** > **Internet Communication Management** > **Internet Communication settings**.
     - Turn off access to all Windows Update features by changing the setting from **enabled** to **not configured**.
   - Go to **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Update**.
     - Remove access to all Windows Update features by changing the setting from **enabled** to **not configured**.
     - Don't connect to any Windows Update Internet locations by changing the setting from **enabled** to **not configured**.
   - Go to **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Update** > **Windows Update for Business**.
     - Select when Quality Updates are received (change from **enabled** to **not configured**)
   - Go to **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Update** > **Windows Update for Business**.
     - Select when Preview Builds and Feature Updates are received (change from **enabled** to **not configured**)
2. Re-enable services:
   - Change **Update Orchestrator service** from **disabled** to **Automatic (Delayed Start)**.

3. Edit the Windows registry (warning, be careful when editing the registry).

   - Go to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\PolicyState`.
     - Change **DeferQualityUpdates** from '1' to '0'.
   - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\Settings`
     - Delete any existing value for **PausedQualityDate** .
   - Go to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\WAU`
     - Set to **Disabled**.

4. Re-enable scheduled tasks:

   - Go to **Task Scheduler Library** > **Microsoft** > **Windows** > **InstallService** > **ScanForUpdates**.
   - Go to **Task Scheduler Library** > **Microsoft** > **Windows** > **InstallService** > **ScanForUpdatesAsUser**.

5. Restart your device to make all these settings take effect.

6. If you don't want this device offered Feature Updates, go to **Settings** > **Windows Update** > **Advanced options** > **Choose when updates are installed** and manually set the option **A feature update includes new capabilities and improvements. It can be deferred for this many days** to any nonzero value, such as 180, 365, and so on.

More information

Learn more about Microsoft's VDI architecture in the Azure Virtual Desktop documentation.

For help troubleshooting sysprep, see Sysprep fails after you remove or update Microsoft Store apps that include built-in Windows images.

Connect to Remote Desktop Services in Windows Server

Article • 05/19/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can connect to remote desktops and apps running in Remote Desktop Services using Windows App or the Remote Desktop client.

Windows App and the Remote Desktop client are available on many different types of devices on different platforms and form factors, such as desktops and laptops, tablets, smartphones, through a web browser, and virtual reality headsets. This choice provides flexibility and convenience to access desktops and apps from anywhere.

Here's an introductory video for Windows App showing connecting to remote desktops and apps. It combines resources from Remote Desktop Services, remote PC connections, Azure Virtual Desktop, Windows 365, and Microsoft Dev Box.

https://www.youtube-nocookie.com/embed/j0XU59VbKOc

There are many features you can use to enhance your remote experience, such as:

- Multiple monitor support, with custom and dynamic display resolutions and scaling.
- Microsoft Teams optimizations.
- Device redirection, such as webcams, audio, storage devices, and printers.
- Single sign-on to eliminate the need to enter credentials multiple times.
- Sign in with multiple accounts and easily switch between them.

How can I connect to Remote Desktop Services?

First, you need to decide whether to use Windows App or the Remote Desktop client to connect to Remote Desktop Services. Windows App replaces the Remote Desktop client and it's available for all platforms, but you can't currently connect to Remote Desktop Services using Windows App from all platforms, so you might still need to use the Remote Desktop client.

Here's what platforms you can use to connect to Remote Desktop Services from Windows App and the Remote Desktop client:

Expand table

Platform	Windows App	Remote Desktop client
Windows	❌	✅
macOS	✅	✅
iOS/iPadOS	✅	✅
Android	✅	✅
Web browser	❌	✅

Once you choose which app to use, download an install it on your device. If you want to connect from a web browser, contact your administrator for their specific link to the Remote Desktop Web client.

In the following articles, you can find download links for Windows App and the Remote Desktop client, and detailed guidance on how to connect to Remote Desktop Services for each supported platform. You need your user account for Remote Desktop Services to sign in, which is provided by your administrator.

- **Windows App**:
  - macOS
  - iOS/iPadOS
  - Android
- **Remote Desktop client**:
  - Windows
  - macOS
  - iOS/iPadOS
  - Android
  - Web browser

> **Tip**
>
> You should use Windows App to connect to Remote Desktop Services where possible as it replaces the Remote Desktop client.

Related content

- Windows App documentation
- Remote Desktop client documentation

Get started with the web client

Article • 05/06/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Desktop web client lets you use a compatible web browser to access your organization's remote resources (apps and desktops) published to you by your admin. You'll be able to interact with the remote apps and desktops like you would with a local PC no matter where you are, without having to switch to a different desktop PC. Once your admin sets up your remote resources, all you need are your domain, user name, password, the URL your admin sent you, and a supported web browser, and you're good to go.

What you'll need to use the web client

- For the web client, you'll need a PC running Windows, macOS, ChromeOS, or Linux. Mobile devices aren't supported at this time.
- A modern browser like Microsoft Edge, Google Chrome, Apple Safari, or Mozilla Firefox (v55.0 and later).
- The URL your admin sent you.

Start using the Remote Desktop client

To sign in to the client, go to the URL your admin sent you. At the sign-in page, enter your domain and user name in the format `DOMAIN\username`, enter your password, and then select **Sign in**.

> **Note**
>
> By signing in to the web client, you agree that your PC complies with your organization's security policy.

After you sign in, the client will take you to the **All Resources** tab, which contains all items published to you under one or more collapsible groups, such as the "Work Resources" group. You'll see several icons representing the apps, desktops, or folders containing more apps or desktops that the admin makes available to the work group. You can come back to this tab at any time to launch additional resources.

To start using an app or desktop, select the item you want to use, enter the same user name and password you used to sign in to the web client if prompted, and then select **Submit**. You might also be shown a consent dialog to access local resources, like clipboard and printer. You

can choose to not redirect either of these, or select **Allow** to use the default settings. Wait for the web client to establish the connection, and then start using the resource as you would normally.

When you're finished, you can end your session by either selecting the **Sign Out** button in the toolbar at the top of your screen or closing the browser window.

Web client keyboard shortcuts

The following table describes alternate key combinations to inject standard Windows shortcut keys in the remote session.

Expand table

Shortcut key	Description
(Windows) Ctrl+Alt+End (MacOS) fn+control+option+delete	Inject Ctrl+Alt+Del in the remote session.
Alt+F3	Injects Windows key in the remote session.
Alt+Page up	Switches between programs from left to right in the remote session. (Windows shortcut is Alt+Tab.)
Alt+Page down	Switches between programs from right to left in the remote session. (Windows shortcut is Alt+Shift+Tab.)

Printing from the Remote Desktop web client

Follow these steps to print from the web client:

1. Start the printing process as you would normally for the app you want to print from.
2. When prompted to choose a printer, select **Remote Desktop Virtual Printer**.
3. After choosing your preferences, select **Print**.
4. Your browser generates a PDF file of your print job.
5. You can choose to either open the PDF and print its contents to your local printer or save it to your PC for later use.

Transfer files with the web client

Follow these steps to transfer files from your local computer to the remote session:

1. Connect to the remote session.
2. Select the file upload icon in the web client menu.
3. When prompted, select the files you want to upload using the local file explorer.
4. Open the file explorer in your remote session. Your files are uploaded to **Remote Desktop Virtual Drive** > **Uploads**.

To download files from the remote session to your local computer:

1. Connect to the remote session.
2. Open the file explorer in your remote session.
3. Copy the file or files you want to download to **Remote Desktop Virtual Drive** > **Downloads**. There's a file size limit of 255MB.
4. A prompt asks if you want to download the file or files you selected. At this point, you can confirm the download by selecting **Confirm** or cancel it by selecting **Cancel**. If you don't want to see this prompt every time you download files from the current browser, select the check box labeled **Don’t ask me again on this browser** before confirming.
5. Your files are downloaded to your local default downloads folder.

Copy and paste from the Remote Desktop web client

The web client currently supports copying and pasting text only. Files can't be copied or pasted to and from the web client. Additionally, you can only use **Ctrl+C** and **Ctrl+V** to copy and paste text.

Keyboard settings in the remote session

The web client supports using an Input Method Editor (IME) in the remote session in version 1.0.21.16 or later. Before you can use the IME, you must install the language pack for the keyboard you want to use in the remote session on the host virtual machine. To learn more about setting up language packs in the remote session, see Add language packs to a Windows 10 multi-session image.

To select alternative keyboard layout or language:

1. Before you connect to the remote session, go to the web client **Settings** panel.

2. In **Select Remote Keyboard Layout** section, expand the drop-down menu and select the keyboard you want to use in the remote session.

   Azure Virtual Desktop (AVD) web client setting options:

- Auto: This configuration sends KeyCodes on key press, which means the local key is directly sent to the remote machine. For this option, the local machine keyboard layout is important and should match the layout on all the hops taken to the remote machine.
- Remote: This configuration sends Scan Codes to the remote machine. For this option, the local machine keyboard layout isn't as important, but the keyboard layout on all other hops taken to the remote machine should match the selected layout.
- Language specific: If you select a specific language and the language pack is installed on the remote machine, that language will automatically be selected on new Windows sessions only. For example, if you use English UK, you can select it from the drop-down. Make sure to sign out of ALL the Windows user sessions you're trying to connect to. When you open a new session, all the hops should automatically default to using the English UK layout.

  > ⓘ **Note**
  >
  > There's a known issue when using KeyCodes for PowerShell. By selecting a mode on AVD Web Client that uses Scan Codes (either Remote or English UK for example), PowerShell should work as expected.

3. If you're using either an IME-based keyboard or a keyboard with alternate layout, select either **Remote** OR pick any of the languages from the list.
4. Connect to the remote session.

The web client suppresses the local IME window when you're focused on the remote session. If you change the IME settings after you've already connected to the remote session, the setting changes won't have any effect. The web client doesn't support IME input while using a private browsing window. Additionally, IMEs don't work with the Auto setting.

> ⓘ **Note**
>
> If the language pack isn't installed on the host virtual machine, the remote session defaults to the English (United States) keyboard.

Enable native display resolution in remote sessions

The web client supports using native display resolution during remote sessions. In sessions running on a high-DPI display, native resolution can provide higher-fidelity graphics and improved text clarity.

> **Note**
>
> Enabling native display resolution with a high-DPI display might cause increased CPU or network usage.

Native resolution is set to off by default. To turn on native resolution:

1. In your session, go to the upper-right corner of the taskbar and select **Settings**.
2. Set **Enable native display resolution** to **On**.

Get help with the web client

If you've encountered an issue that can't be solved by the information in this article, you can get help with the web client by raising feedback on the web client's Feedback page.

What's new in the Remote Desktop web client

06/11/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article describes the latest updates for the Remote Desktop web client. Updates include new features, improvements, bug fixes, and security enhancements to help you stay productive.

Updates for version 2.1.65.0

Date published: June 6, 2025

- Added a new enhanced graphics decoding setting.
- Bug fixes and security updates.
- New minimum browser requirements will apply in the next release. Your browser must:
  - Be no more than 12 months old on a rolling basis.
  - Support the AVC codec. Most browsers on desktops and laptops support AVC by default.
  - Have WebGL enabled. WebGL is enabled by default on most recent browser versions.

Updates for version 2.1.62.1

Date published: January 30, 2025

- Changed the endpoint that delivers client packages to `*.cdn.office.net`.
- Bug fixes and security updates.

Updates for version 2.1.0.0

Date published: March 21, 2024

- The new web client is now generally available.
- UX improvements added.
- New key features added to this client version.
- Now available for on-premises download.

Updates for version 1.0.28.0

Date published: December 19, 2022

- You can now redirect cameras.
- Updated third-party libraries.
- Accessibility improvements.
- Bug fixes.

Updates for version 1.0.27.0

Date published: March 24, 2022

- Added web client keyboard shortcuts for switching between programs. For more information, see Keyboard shortcuts.
- Support for native resolution on high-DPI devices. For more information, see Enable native display resolution in remote sessions.
- Updated full screen mode icon behavior to disable the icon when you press the F11 key to enter full screen mode.
- Removed support for Internet Explorer and other deprecated browsers.
- Fixed an issue where some keys weren't working correctly on the Japanese keyboard layout.
- Bug fixes and security improvements for file transfer.

Updates for version 1.0.26.0

Date published: December 12, 2021

- Bug fixes.
- Version 1.0.26.0 is the final version of the client that supports Internet Explorer 11 and Windows XP.

Updates for version 1.0.25.0

Date published: July 22, 2021

- Client now has web assembly on supported browsers.
- Added file transfer support.
- Bug fixes.

Updates for 1.0.24.0

Date published: January 6, 2021

> **Important**
>
> Version 1.0.24.0 includes an important security fix. We removed earlier versions of the web client containing this bug.

- Added support for redirecting local microphone input to the remote session.
- Fixed issues with `AltGr` and several other keyboard bugs.
- Accessibility improvements.

Updates for 1.0.22.0

Date published: September 2, 2020

- Users can now move the minimized menu.
- Improved support for 4K and ultra-wide monitors and fixed an issue where copying large amounts of data caused sessions to crash.
- Improved support for using an Input Method Editor in the remote session. To learn more about using an Input Method Editor with the web client, check out Connect to Azure Virtual Desktop with the web client.
- Changed the **All Resources** page UI.
- Fixed several connection sequence failures where web client returned a *General Protocol Error*.
- Fixed keyboard input issues where specific key sequences weren't handled appropriately.
- Accessibility improvements.

Updates for version 1.0.21.0

Date published: November 15, 2019

- Added support for using an Input Method Editor (IME) in the remote session to input complex characters.
- Fixed a regression where users couldn't copy and paste into the remote session on macOS devices.
- Fixed a regression where local Windows Key was sent to the remote session on Firefox.
- Added link to RDWeb password change when enabled by your administrator.

Updates for version 1.0.20.0

Date published: October 18, 2019

- Added support for connections to Windows 7 and Windows Server 2008 R2 hosts.
- Fixed an issue where certain app icons were shown as transparent tiles.
- Fixed connection issues for Internet Explorer browser on Windows 7.
- Fixed unexpected disconnects when the browser was resized.
- Accessibility improvements.
- Updated third-party libraries.

Updates for version 1.0.18.0

Date published: May 14, 2019

- Added Resource Launch Method configuration in the Settings tab, enabling users to either open resources in the browser or download a `.rdp` file to handle with another client. An administrator needs to configure this behavior. Details regarding administrator configurations for this feature can be found in the web client setup documentation.
- Fixed color rendering issues, enabling more vivid colors in your remote session.
- Revised error messages related to remote resource feed errors.
- Added support for more office shortcuts, such as paste special (Ctrl+Alt+V).
- Added keyboard shortcut for users to invoke the Windows Key in the remote session (Alt+F3)
- Updated error message for users attempting to authenticate using an expired password.
- Refreshed feed UI on the All Resources page.
- Resolved overlapping dialogues that occurred during session reconnect.
- Fixed remote resource icon sizing in the resource taskbar.

Updates for version 1.0.11

Date published: February 22, 2019

- Enabled connection to a Remote Desktop Broker without a Remote Desktop Gateway in Windows Server 2019.
- Sorted feeds alphabetically (for example, RemoteApps first, Desktops second).
- Fixed multiple accessibility bugs improving screen reader compatibility.
- Updated our build tools.
- Various bug fixes.

Updates for version 1.0.7

Date published: January 24, 2019

- Offline use on internal networks is now supported.

- Improved rendering on non Microsoft Edge browsers.
- Implemented limit for feed retrieval retry attempts to prevent DoS.
- Fixed accessibility bugs, enabling users with visual disabilities to use the web client.
- Improved error messages displayed to the user for feed errors.
- Added Ctrl + Alt + End (Windows) and fn + control + option + delete (Mac) shortcuts to invoke Ctrl + Alt + Del in remote machine.
- Improved telemetry for crash events.
- Improved our build pipeline and build tools.
- Various bug fixes.

Updates for version 1.0.1

Date published: October 29, 2018

- Added an option to **Capture support information** on the About page to diagnose issues.
- InPrivate mode is now supported.
- Improved support for non-English keyboards.
- Fixed an issue where tooltips with non-English characters showed incorrectly.
- Fixed graphics rendering issue that affected Chrome users.
- Updated time zone redirection with full daylight savings time support.
- Improved the error message for out-of-memory error.
- Various bug fixes.

Updates for version 1.0.0

Date published: July 16, 2018

- Remote Desktop web client is now generally available.
- Admins can globally turn off telemetry for the web client.
- Various bug fixes.

Updates for version 0.9.0

Date published: July 5, 2018

- New sign in experience within the web client.
- No longer prompted for credentials when launching a desktop or app connection (Single sign on).
- Added time zone redirection.
- Various bug fixes.

Updates for version 0.8.1

Date published: May 17, 2018

- Updates to address CredSSP encryption oracle remediation described in CVE-2018-0886.
- Fixed connection failures for some languages when printing is enabled.
- Improved error message when a gateway isn't part of the deployment.
- **Help** and **Feedback** options were added.

Updates for version 0.8.0

Date published: March 28, 2018

- Initial preview release of the web client.
- Copy/paste text through the clipboard with **CTRL+C** and **CTRL+V**.
- Print to a PDF file.
- Localized in 18 languages.

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

Remote Desktop Web Access troubleshooting

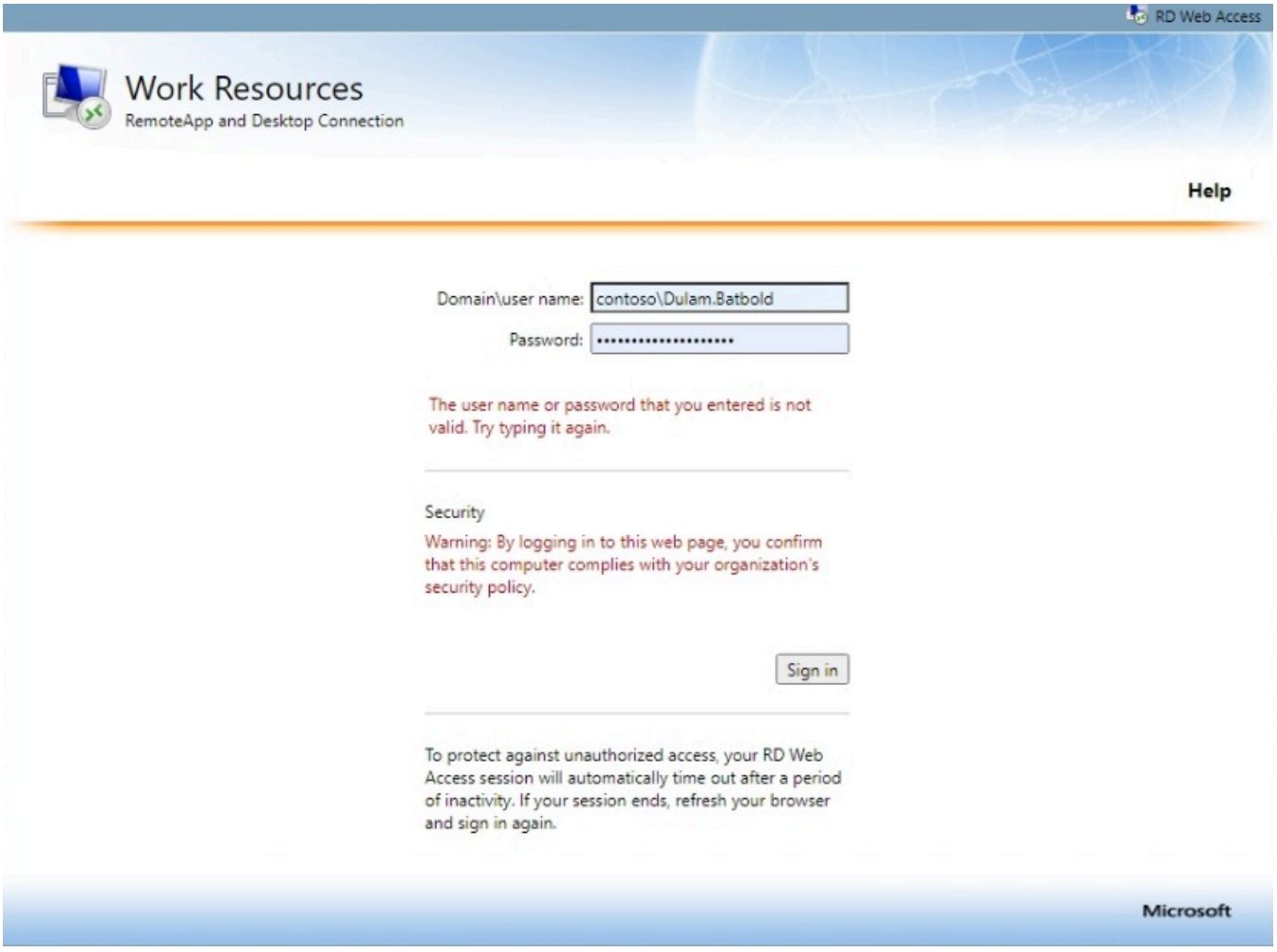
Applies to: Supported versions of Windows Server

This article provides answers to some of the most common questions about Remote Desktop Web Access (RD Web Access). The RD Web Access website enables you to use a Web browser to access RemoteApp and Desktop Connections.

What should I do if I can't sign in?

If you enter an invalid user name or password, you see a message saying: "The user name or password that you entered is not valid. Try typing it again." If your account is locked because of too many incorrect sign in attempts, you see the same message.

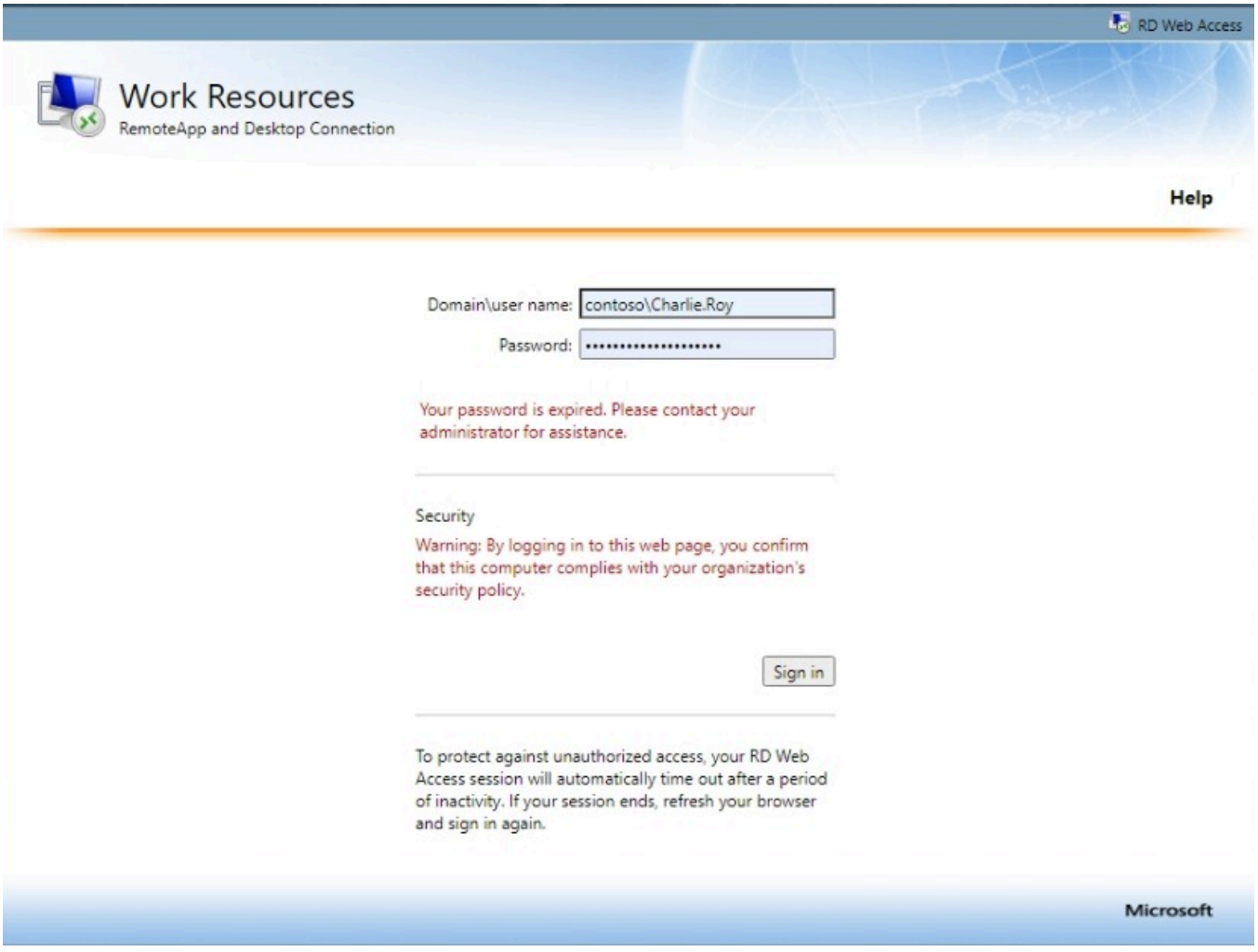
Contact your Active Directory administrator to reset your password, and if needed, unlock your account.



If you see a message that your password doesn't meet the criteria for your domain, contact your Active Directory administrator to help reset your password.

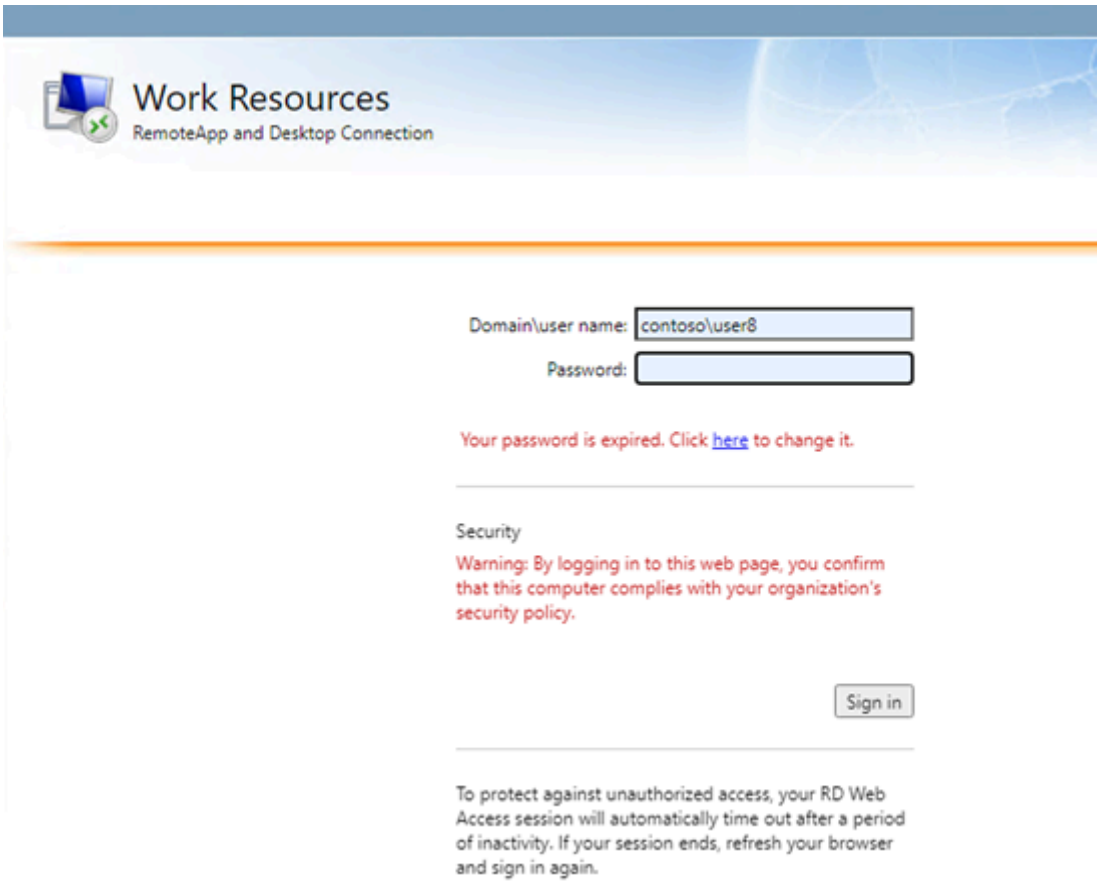
How do I reset my expired password?

If an administrator didn't enable password reset, you see a message saying: "Your password is expired. Please contact your administrator for assistance." You need to contact your Active Directory administrator to reset your password.



If an administrator enabled password reset, follow the link to change your expired password. You need to enter your old password, then your new password, and confirm the new password.

If you're unable to reset your password, and see that your new password doesn't meet the length, complexity, or history requirements of your domain, try choosing a different password, or contact your administrator to learn what the requirements are.



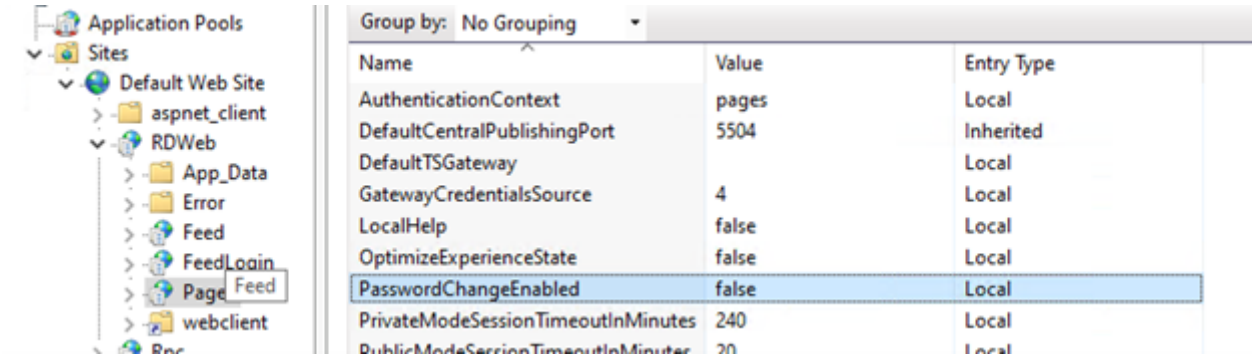
How does an administrator enable password reset?

Administrators can enable remote users to change their own password from within the RD Web access interface if it's expired.

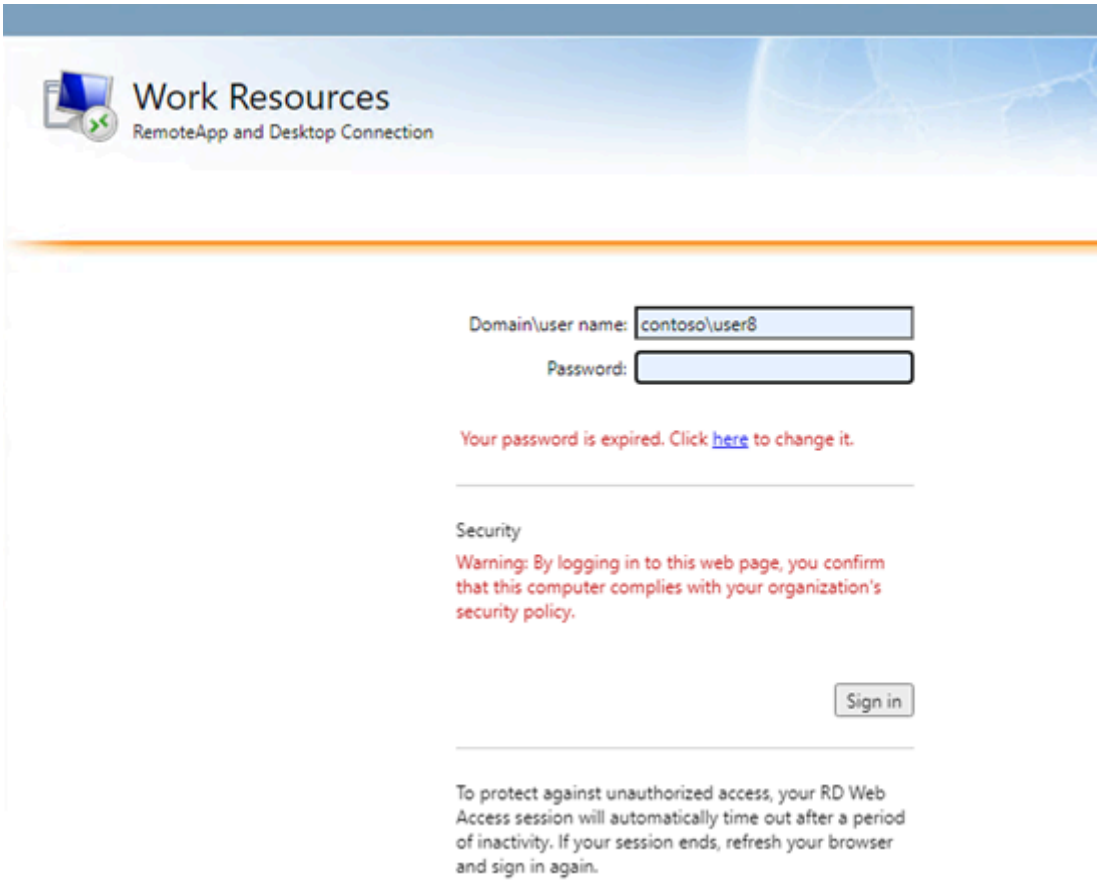
> ⓘ **Important**
>
> This option isn't helpful for users who forgot their password since the old password still needs to be entered before selecting a new password. For forgotten passwords, you must contact your Active Directory administrator.

To set up password reset password for Remote Desktop Web access:

1. Open **Server Manager** on the Server running RD Web Access.
2. In the menu under **Tools**, navigate to **Internet Information Services (IIS) Manager**.
3. Next, locate your server and navigate to **Sites** > **Default Web Site** > **RDWeb** > **Pages**.
4. Select **Application Settings** from the menu. Then, select the setting **PasswordChangeEnabled** and change the value to true.



Now, when users open the RD Web Access page and try to sign in using an expired password, a link appears to reset your password.



When you open the password reset page, there's a user interface where you enter your current password, new password, and confirm the new password.

> ⓘ **Note**
>
> If the remote server is running in Azure, you need create an endpoint for public port 443 in the Azure management portal so that users can access the RD Web Access portal.

How do I change my password?

First, connect to your server using RemoteApp and Desktop Connections. To change your password:

1. Enter Ctrl+Alt+Del.
2. Select **Change a password**.
3. Enter in the old password, then the new password, and confirm the new password.

Contact your administrator if you can't change your password, don't know the password requirements, or you can't sign in.

What is RemoteApp?

Using RemoteApp, you can access programs on a remote computer through Remote Desktop Services. Although the programs are running on a remote computer, RemoteApp programs behave as if they're running on your local computer. For example, a RemoteApp program has its own entry in the taskbar, and you can resize, minimize, or maximize the program window.

How do I start a RemoteApp program?

To start a RemoteApp program or Remote Desktop session, click the program icon in RD Web Access. When you're prompted for user credentials, log on with your network user name and password.

What are the public vs. private computer settings?

If you connect to the RD Web Access website from a public computer, such as a kiosk computer in a public establishment, or from a computer that you share with other users, click This is a public or shared computer. You need to provide both your user name and password each time you sign in to the RD Web Access website.

If you're using a work computer assigned to you and that you don't share with other people, click This is a private computer.

To protect against unauthorized access, RD Web Access sessions automatically end after a period of inactivity. If your RD Web Access session ends, you need to sign in again. The administrator sets how long a session lasts.

Windows App documentation

Use Windows App to securely connect to your Windows devices and apps. You can connect to Azure Virtual Desktop, Windows 365, Microsoft Dev Box, Remote Desktop Services, and remote PCs on a device of your choice.

OVERVIEW
What is Windows App?

GET STARTED
Connect to Windows devices and apps

Get started

Learn how to use Windows App

- Connect to devices and apps
- Device actions
- Configure display settings
- Manage user accounts
- Use keyboard, mouse, touch, and pen
- Redirect local devices, audio, and folders

About Windows App

- What's new in Windows App
- Compare features across platforms and devices

Related products and services

Discover some of the services you can connect to with Windows App.

Azure Virtual Desktop

Windows 365

Microsoft Dev Box

Compare Remote Desktop client features across platforms and devices

Article • 02/27/2025

> **Tip**
>
> This article is shared for services and products that use the Remote Desktop Protocol (RDP) to provide remote access to Windows desktops and apps.
>
> Use the buttons at the top of this article to select what you want to connect to so the article shows the relevant information.

The Remote Desktop client runs on Windows, macOS, iOS and iPadOS, Android and Chrome OS, and in a web browser. However, support for some features differs across these platforms. This article details which features are supported on which platforms. For information about the status of the Remote Desktop client on each platform, such as whether it's still available to download, see Connection matrix and status.

There are two versions of the Remote Desktop client for Windows, which are both supported for connecting to Remote Desktop Services and remote PCs:

- **Remote Desktop Connection**: provided in Windows and is referred to in this article as **Windows (MSTSC)**, after the name of the executable file. It also includes the **RemoteApp and Desktop Connections** Control Panel applet.
- **Remote Desktop app for Windows**: Comes from the Microsoft Store. When installed, the application name is *Remote Desktop*. It's referred to in this article as **Windows (Store)**.

> **Important**
>
> To ensure a seamless experience, you should download Windows App, which replaces the Remote Desktop client.
>
> Starting May 27, 2025, the Remote Desktop app for Windows from the Microsoft Store will no longer be supported or available for download and installation. Users must transition to Windows App to ensure continued access to Windows 365, Azure Virtual Desktop, and Microsoft Dev Box. For more information, see Get started with Windows App to connect to devices and apps.
>
> This announcement doesn't apply to the Remote Desktop client for Windows (MSI), but we recommend those users also transition to Windows App for the best experience.

Experience

The following table compares which Remote Desktop client experience features are supported on which platforms:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Appearance (dark or light)	❌	✅	✅	✅	✅	✅
Integrated apps	✅[1]	❌	❌	❌	❌	❌
Localization	✅	✅	❌	✅	❌	✅
Pin to Start Menu	✅[1]	✅	❌	❌	❌	❌
Search	❌	❌	✅	✅	✅	✅
URI schemes	❌	❌	✅[2]	✅[2]	✅[2]	❌

1. When subscribed to Remote Desktop Services using the **RemoteApp and Desktop Connections** Control Panel applet.
2. Legacy RDP URI scheme only.

The following table provides a description for each of the experience features:

Expand table

Feature	Description
Appearance (dark or light)	Change the appearance of the Remote Desktop client to be light or dark.
Integrated apps	Individual apps using RemoteApp are integrated with the local device as if they're running locally.
Localization	User interface available in languages other than *English (United States)*.
Pin to Start Menu	Pin your favorite devices and apps to the Windows Start Menu for quick access.
Search	Quickly search for devices or apps.
Uniform Resource Identifier (URI) schemes	Start the Remote Desktop client or connect to a remote session with specific parameters and values with a URI.

Display

The following table compares which display features are supported on which platforms:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Dynamic resolution	✗	✓	✓	✓	✓	✓
External monitor	✓	✗	✓	✓	✗	✗
Multiple monitors[1]	✓	✗	✓	✗	✗	✗
Selected monitors	✓	✗	✗	✓	✗	✗
Smart sizing	✓	✓	✓	✗	✗	✗

1. Up to 16 monitors.

The following table provides a description for each of the display features:

Expand table

Feature	Description
Dynamic resolution	The resolution and orientation of local displays is dynamically reflected in the remote session for desktops. If the session is running in *windowed* mode, the desktop is dynamically resized to the size of the window.
External display	Enables the use of an external display for a remote session.
Multiple displays	Enables the remote session to use all local displays. Each display can have a maximum resolution of 8K, with the total combined resolution limited to 32K. These limits depend on factors such as session host specification and network connectivity.
Selected displays	Specifies which local displays to use for the remote session.
Smart sizing	A desktop in *windowed* mode is dynamically scaled to the window's size.

Redirection

The following sections detail the redirection support available on each platform.

> **Tip**
>
> Redirection of some peripheral and resource types needs to be enabled by an administrator before they can be used in a remote session. For more information, see Redirection over the Remote Desktop Protocol, where you can also find links in the Related content section to articles that explain how to configure redirection for specific peripheral and resource types.

Device redirection

The following table shows which local devices you can redirect to a remote session on each platform:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Cameras	✅	❌	✅	✅	✅	✅[1]
Local drive/storage	✅	❌	✅	✅	✅	✅[2]
Microphones	✅	✅	✅	✅	✅	✅
Printers	✅	❌	✅[3]	❌	❌	✅[4]
Scanners[5]	✅	❌	❌	❌	❌	❌
Smart cards	✅	❌	✅	❌	❌	❌
Speakers	✅	✅	✅	✅	✅	✅

1. Camera redirection in a web browser is in preview.
2. Limited to uploading and downloading files through a web browser.
3. The Remote Desktop client on macOS supports the *Publisher Imagesetter* printer driver by default (*Common UNIX Printing System* (CUPS) only). Native printer drivers aren't supported.
4. PDF printing only.
5. High-level redirection of TWAIN scanners isn't supported. You can only redirect USB scanners using opaque low-level redirection. For more information, see Peripheral and resource redirection over the Remote Desktop Protocol.

The following table provides a description for each type of device you can redirect:

Expand table

Device type	Description
Cameras	Redirect a local camera to use with apps like Microsoft Teams.
Local drive/storage	Access local disk drives in a remote session.
Microphones	Redirect a local microphone to use with apps like Microsoft Teams.
Printers	Print from a remote session to a local printer.
Scanners	Access a local scanner in a remote session.
Smart cards	Use smart cards in a remote session.
Speakers	Play audio in the remote session or on local device.

Input redirection

The following table shows which input methods you can redirect:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Keyboard	✅	✅	✅	✅	✅	✅
Keyboard input language	✅	✅	✅	❌	❌	✅[1]
Keyboard shortcuts	✅	✅	✅	✅	✅	✅
Mouse/trackpad	✅	✅	✅	✅	✅	✅
Multi-touch	✅	✅	❌	✅	✅	❌
Pen	✅	❌	❌	✅	✅	✅
Touch	✅	✅	❌	✅	✅	✅

1. Enabled by alternative keyboard layout.

The following table provides a description for each type of input you can redirect:

Expand table

Input type	Description
Keyboard	Redirect keyboard inputs to the remote session.
Mouse/trackpad	Redirect mouse or trackpad inputs to the remote session.
Multi-touch	Redirect multiple touches simultaneously to the remote session.
Pen	Redirect pen inputs, including pressure, to the remote session.
Touch	Redirect touch inputs to the remote session.

Port redirection

The following table shows which ports you can redirect:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Serial	✅	❌	❌	❌	❌	❌
USB	✅	❌	❌	❌	❌	❌

The following table provides a description for each port you can redirect:

Expand table

Port type	Description
Serial	Redirect serial (COM) ports on the local device to the remote session.
USB	Redirect supported USB devices on the local device to the remote session.

Other redirection

The following table shows which other features you can redirect:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Clipboard - bidirectional	✅	✅	✅	✅[1]	✅[2]	✅[2]

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Clipboard - unidirectional[3]	✅	✅	✅	✅	✅	✅
Location	✅[4]	❌	❌	✅	❌	✅
Third-party virtual channel plugins	✅	❌	❌	❌	❌	❌
Time zone	✅	✅	✅	✅	✅	✅
WebAuthn	✅	❌	❌	❌	❌	❌

1. Text and images only.
2. Text only.
3. macOS support is native in the Remote Desktop client. All other platforms require remote session configuration. For more information, see Configure the clipboard transfer direction and types of data that can be copied.
4. From a local device running Windows 11 only.

The following table provides a description for each other redirection feature you can redirect:

Expand table

Feature	Description
Clipboard - bidirectional	Redirect the clipboard on the local device is to the remote session and from the remote session to the local device.
Clipboard - unidirectional	Control the direction in which the clipboard can be used and restrict the types of data that can be copied.
Location	The location of the local device can be available in the remote session.
Third-party virtual channel plugins	Enables third-party virtual channel plugins to extend Remote Desktop Protocol (RDP) capabilities.
Time zone	The time zone of the local device can be available in the remote session.
WebAuthn	Authentication requests in the remote session can be redirected to the local device allowing the use of security devices such as Windows Hello for Business or a security key.

Network

The following table shows which network features are available on each platform:

Expand table

Feature	Windows (MSTSC)	Windows (Store)	macOS	iOS/ iPadOS	Android/ Chrome OS	Web browser
Connection information	✅	❌	✅	❌		✅

The following table provides a description for each network feature:

Expand table

Feature	Description
Connection information	See the connection information of the remote session.

Supported RDP properties

06/20/2025

> **Tip**
>
> This article is shared for services and products that use the Remote Desktop Protocol (RDP) to provide remote access to Windows desktops and apps.

The Remote Desktop Protocol (RDP) has a number of properties you can set to customize the behavior of a remote session, such as for device redirection, display settings, session behavior, and more.

The following sections contain each RDP property available and lists its syntax, description, supported values, the default value, and connections to which services and products you can use them with.

How you use these RDP properties depends on the service or product you're using:

Expand table

Product	Configuration point
Azure Virtual Desktop	Host pool RDP properties. To learn more, see Customize RDP properties for a host pool.
Remote Desktop Services	Session collection RDP properties
Remote PC connections	The `.rdp` file you use to connect to a remote PC.

> **Note**
>
> For each RDP property, replace `<value>` with an allowed value for that property.

Connections

Here are the RDP properties that you can use to configure connections.

`alternate full address`

- **Syntax**: `alternate full address:s:<value>`

- **Description**: Specifies an alternate name or IP address of the remote computer.
- **Supported values**:
  - A valid hostname, IPv4 address, or IPv6 address.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services
  - Remote PC connections

`alternate shell`

- **Syntax**: `alternate shell:s:<value>`
- **Description**: Specifies a program to be started automatically in a remote session as the shell instead of explorer.
- **Supported values**:
  - A valid path to an executable file, such as `C:\Program Files\MyApp\myapp.exe`.
- **Default value**: None.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`authentication level`

- **Syntax**: `authentication level:i:<value>`
- **Description**: Defines the server authentication level settings.
- **Supported values**:
  - `0`: If server authentication fails, connect to the computer without warning.
  - `1`: If server authentication fails, don't establish a connection.
  - `2`: If server authentication fails, show a warning, and choose to connect or refuse the connection.
  - `3`: No authentication requirement specified.
- **Default value**: `3`
- **Applies to**:
  - Remote Desktop Services
  - Remote PC connections

`disableconnectionsharing`

- **Syntax**: `disableconnectionsharing:i:<value>`

- **Description**: Determines whether the client reconnects to any existing disconnected session or initiate a new connection when a new connection is launched.
- **Supported values**:
  - `0`: Reconnect to any existing session.
  - `1`: Initiate new connection.
- **Default value**: `0`
- **Applies to**:
  - Remote Desktop Services

`domain`

- **Syntax**: `domain:s:<value>`
- **Description**: Specifies the name of the Active Directory domain in which the user account that will be used to sign in to the remote computer is located.
- **Supported values**:
  - A valid domain name, such as `CONTOSO`.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services
  - Remote PC connections

`enablecredsspsupport`

- **Syntax**: `enablecredsspsupport:i:<value>`
- **Description**: Determines whether the client will use the Credential Security Support Provider (CredSSP) for authentication if it's available.
- **Supported values**:
  - `0`: RDP won't use CredSSP, even if the operating system supports CredSSP.
  - `1`: RDP will use CredSSP if the operating system supports CredSSP.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`enablerdsaadauth`

- **Syntax**: `enablerdsaadauth:i:<value>`

- **Description**: Determines whether the client will use Microsoft Entra ID to authenticate to the remote PC. When used with Azure Virtual Desktop, this provides a single sign-on experience. This property replaces the property targetisaadjoined.
- **Supported values**:
  - `0`: Connections won't use Microsoft Entra authentication, even if the remote PC supports it.
  - `1`: Connections will use Microsoft Entra authentication if the remote PC supports it.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote PC connections

`full address`

- **Syntax**: `full address:s:<value>`
- **Description**: Specifies the hostname or IP address of the remote computer that you want to connect to.. This is the only mandatory property in a `.rdp` file.
- **Supported values**:
  - A valid hostname, IPv4 address, or IPv6 address.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services
  - Remote PC connections

`gatewaycredentialssource`

- **Syntax**: `gatewaycredentialssource:i:<value>`
- **Description**: Specifies the authentication method used for Remote Desktop gateway connections.
- **Supported values**:
  - `0`: Ask for password (NTLM).
  - `1`: Use smart card.
  - `2`: Use the credentials for the currently signed in user.
  - `3`: Prompt the user for their credentials and use basic authentication.
  - `4`: Allow user to select later.
  - `5`: Use cookie-based authentication.
- **Default value**: `0`
- **Applies to**:
  - Remote Desktop Services

`gatewayhostname`

- **Syntax**: `gatewayhostname:s:<value>`
- **Description**: Specifies the host name of a Remote Desktop gateway.
- **Supported values**:
  - A valid hostname, IPv4 address, or IPv6 address.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services

`gatewayprofileusagemethod`

- **Syntax**: `gatewayprofileusagemethod:i:<value>`
- **Description**: Specifies whether to use the default Remote Desktop gateway settings.
- **Supported values**:
  - `0`: Use the default profile mode, as specified by the administrator.
  - `1`: Use explicit settings, as specified by the user.
- **Default value**: `0`
- **Applies to**:
  - Remote Desktop Services

`gatewayusagemethod`

- **Syntax**: `gatewayusagemethod:i:<value>`
- **Description**: Specifies whether to use a Remote Desktop gateway for the connection.
- **Supported values**:
  - `0`: Don't use a Remote Desktop gateway.
  - `1`: Always use a Remote Desktop gateway.
  - `2`: Use a Remote Desktop gateway if a direct connection can't be made to the RD Session Host.
  - `3`: Use the default Remote Desktop gateway settings.
  - `4`: Don't use a Remote Desktop gateway, bypass gateway for local addresses. Setting this property value to `0` or `4` are effectively equivalent, but `4` enables the option to bypass local addresses.
- **Default value**: `0`
- **Applies to**:
  - Remote Desktop Services

`kdcproxyname`

- **Syntax**: `kdcproxyname:s:<value>`
- **Description**: Specifies the fully qualified domain name of a KDC proxy.
- **Supported values**:
  - A valid path to a KDC proxy server, such as `kdc.contoso.com`.
- **Default value**: None.
- **Applies to**:
  - Azure Virtual Desktop. For more information, see Configure a Kerberos Key Distribution Center proxy.

`promptcredentialonce`

- **Syntax**: `promptcredentialonce:i:<value>`
- **Description**: Determines whether a user's credentials are saved and used for both the Remote Desktop gateway and the remote computer.
- **Supported values**:
  - `0`: Remote session doesn't use the same credentials.
  - `1`: Remote session does use the same credentials.
- **Default value**: `1`
- **Applies to**:
  - Remote Desktop Services

`targetisaadjoined`

- **Syntax**: `targetisaadjoined:i:<value>`
- **Description**: Allows connections to Microsoft Entra joined session hosts using a username and password. This property is only applicable to non-Windows clients and local Windows devices that aren't joined to Microsoft Entra. It is being replaced by the property enablerdsaadauth.
- **Supported values**:
  - `0`: Connections to Microsoft Entra joined session hosts will succeed for Windows devices that meet the requirements, but other connections will fail.
  - `1`: Connections to Microsoft Entra joined hosts will succeed but are restricted to entering user name and password credentials when connecting to session hosts.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop. For more information, see Microsoft Entra joined session hosts in Azure Virtual Desktop.

`username`

- **Syntax**: `username:s:<value>`
- **Description**: Specifies the name of the user account that will be used to sign in to the remote computer.
- **Supported values**:
  - Any valid username.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services

Session behavior

Here are the RDP properties that you can use to configure session behavior.

`autoreconnection enabled`

- **Syntax**: `autoreconnection enabled:i:<value>`
- **Description**: Determines whether the local device will automatically try to reconnect to the remote computer if the connection is dropped, such as when there's a network connectivity interruption.
- **Supported values**:
  - `0`: The local device doesn't automatically try to reconnect.
  - `1`: The local device automatically tries to reconnect.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`bandwidthautodetect`

- **Syntax**: `bandwidthautodetect:i:<value>`
- **Description**: Determines whether or not to use automatic network bandwidth detection.
- **Supported values**:
  - `0`: Don't use automatic network bandwidth detection.
  - `1`: Use automatic network bandwidth detection.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services

- Remote PC connections

`compression`

- **Syntax**: `compression:i:<value>`
- **Description**: Determines whether bulk compression is enabled when transmitting data to the local device.
- **Supported values**:
  - `0`: Disable bulk compression.
  - `1`: Enable RDP bulk compression.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`networkautodetect`

- **Syntax**: `networkautodetect:i:<value>`
- **Description**: Determines whether automatic network type detection is enabled.
- **Supported values**:
  - `0`: Disable automatic network type detection.
  - `1`: Enable automatic network type detection.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`videoplaybackmode`

- **Syntax**: `videoplaybackmode:i:<value>`
- **Description**: Determines whether the connection will use RDP-efficient multimedia streaming for video playback.
- **Supported values**:
  - `0`: Don't use RDP efficient multimedia streaming for video playback.
  - `1`: Use RDP-efficient multimedia streaming for video playback when possible.
- **Default value**: `1`
- **Applies to**:

- Azure Virtual Desktop
- Remote Desktop Services
- Remote PC connections

Device redirection

Here are the RDP properties that you can use to configure device redirection. To learn more, see Redirection over the Remote Desktop Protocol.

`audiocapturemode`

- **Syntax**: `audiocapturemode:i:<value>`
- **Description**: Indicates whether audio input redirection is enabled.
- **Supported values**:
  - `0`: Disable audio capture from a local device.
  - `1`: Enable audio capture from a local device and redirect it to a remote session.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure audio and video redirection over the Remote Desktop Protocol.

`audiomode`

- **Syntax**: `audiomode:i:<value>`
- **Description**: Determines whether the local or remote machine plays audio.
- **Supported values**:
  - `0`: Play sounds on the local device.
  - `1`: Play sounds in a remote session.
  - `2`: Don't play sounds.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure audio and video redirection over the Remote Desktop Protocol.

`camerastoredirect`

- **Syntax**: `camerastoredirect:s:<value>`
- **Description**: Configures which cameras to redirect. This setting uses a semicolon-delimited list of `KSCATEGORY_VIDEO_CAMERA` interfaces of cameras enabled for redirection.
- **Supported values**:
  - `*`: Redirect all cameras.
  - `\\?\usb#vid_0bda&pid_58b0&mi`: Specifies a list of cameras by device instance path, such as this example.
  - `-`: Exclude a specific camera by prepending the symbolic link string.
- **Default value**: None.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure camera, webcam, and video capture redirection over the Remote Desktop Protocol.

`devicestoredirect`

- **Syntax**: `devicestoredirect:s:<value>`
- **Description**: Determines which peripherals that use the Media Transfer Protocol (MTP) or Picture Transfer Protocol (PTP), such as a digital camera, are redirected from a local Windows device to a remote session.
- **Supported values**:
  - `*`: Redirect all supported devices, including ones that are connected later.
  - `\\?\usb#vid_0bda&pid_58b0&mi`: Specifies a list of MTP or PTP peripherals by device instance path, such as this example.
  - `DynamicDevices`: Redirect all supported devices that are connected later.
- **Default value**: `*`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure Media Transfer Protocol and Picture Transfer Protocol redirection on Windows over the Remote Desktop Protocol.

`drivestoredirect`

- **Syntax**: `drivestoredirect:s:<value>`
- **Description**: Determines which fixed, removable, and network drives on the local device will be redirected and available in a remote session.
- **Supported values**:
  - *Empty*: Don't redirect any drives.
  - `*`: Redirect all drives, including drives that are connected later.
  - `DynamicDrives`: Redirect any drives that are connected later.
  - `drivestoredirect:s:C:\;E:\;`: Redirect the specified drive letters for one or more drives, such as this example.
- **Default value**: `*`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure fixed, removable, and network drive redirection over the Remote Desktop Protocol.

`encode redirected video capture`

- **Syntax**: `encode redirected video capture:i:<value>`
- **Description**: Enables or disables encoding of redirected video.
- **Supported values**:
  - `0`: Disable encoding of redirected video.
  - `1`: Enable encoding of redirected video.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure camera, webcam, and video capture redirection over the Remote Desktop Protocol.

`keyboardhook`

- **Syntax**: `keyboardhook:i:<value>`
- **Description**: Determines whether Windows key combinations (`Windows`, `Alt` + `Tab`) are applied to a remote session.
- **Supported values**:
  - `0`: Windows key combinations are applied on the local device.
  - `1`: (Desktop sessions only) Windows key combinations are applied on the remote computer when in focus.
  - `2`: (Desktop sessions only) Windows key combinations are applied on the remote computer in full screen mode only.
  - `3`: (RemoteApp sessions only) Windows key combinations are applied on the RemoteApp when in focus. We recommend you use this value only when publishing the Remote Desktop Connection app (`mstsc.exe`) from the host pool on Azure Virtual Desktop. This value is only supported when using the Windows client.
- **Default value**: `2`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`redirectclipboard`

- **Syntax**: `redirectclipboard:i:<value>`
- **Description**: Determines whether to redirect the clipboard.
- **Supported values**:
  - `0`: Clipboard on local device isn't available in remote session.
  - `1`: Clipboard on local device is available in remote session.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure clipboard redirection over the Remote Desktop Protocol.

`redirectcomports`

- **Syntax**: `redirectcomports:i:<value>`
- **Description**: Determines whether serial or COM ports on the local device are redirected to a remote session.

- **Supported values**:
  - `0`: Serial or COM ports on the local device aren't available in a remote session.
  - `1`: Serial or COM ports on the local device are available in a remote session.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure serial or COM port redirection over the Remote Desktop Protocol.

`redirected video capture encoding quality`

- **Syntax**: `redirected video capture encoding quality:i:<value>`
- **Description**: Controls the quality of encoded video.
- **Supported values**:
  - `0`: High compression video. Quality may suffer when there's a lot of motion.
  - `1`: Medium compression.
  - `2`: Low compression video with high picture quality.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure camera, webcam, and video capture redirection over the Remote Desktop Protocol.

`redirectlocation`

- **Syntax**: `redirectlocation:i:<value>`
- **Description**: Determines whether the location of the local device is redirected to a remote session.
- **Supported values**:
  - `0`: A remote session uses the location of the remote computer or virtual machine.
  - `1`: A remote session uses the location of the local device.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop

    - Remote Desktop Services
    - Remote PC connections

To learn how to use this property, see Configure location redirection over the Remote Desktop Protocol.

`redirectprinters`

- **Syntax**: `redirectprinters:i:<value>`
- **Description**: Determines whether printers available on the local device are redirected to a remote session.
- **Supported values**:
    - `0`: The printers on the local device aren't redirected to a remote session.
    - `1`: The printers on the local device are redirected to a remote session.
- **Default value**: `1`
- **Applies to**:
    - Azure Virtual Desktop
    - Remote Desktop Services
    - Remote PC connections

To learn how to use this property, see Configure printer redirection over the Remote Desktop Protocol.

`redirectsmartcards`

- **Syntax**: `redirectsmartcards:i:<value>`
- **Description**: Determines whether smart card devices on the local device will be redirected and available in a remote session.
- **Supported values**:
    - `0`: Smart cards on the local device aren't redirected to a remote session.
    - `1`: Smart cards on the local device are redirected a remote session.
- **Default value**: `1`
- **Applies to**:
    - Azure Virtual Desktop
    - Remote Desktop Services
    - Remote PC connections

To learn how to use this property, see Configure smart card redirection over the Remote Desktop Protocol.

`redirectwebauthn`

- **Syntax**: `redirectwebauthn:i:<value>`
- **Description**: Determines whether WebAuthn requests from a remote session are redirected to the local device allowing the use of local authenticators (such as Windows Hello for Business and security keys).
- **Supported values**:
  - `0`: WebAuthn requests from a remote session aren't sent to the local device for authentication and must be completed in the remote session.
  - `1`: WebAuthn requests from a remote session are sent to the local device for authentication.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure WebAuthn redirection over the Remote Desktop Protocol.

`usbdevicestoredirect`

- **Syntax**: `usbdevicestoredirect:s:<value>`
- **Description**: Determines which supported USB devices on the client computer are redirected using opaque low-level redirection to a remote session.
- **Supported values**:
  - `*`: Redirect all USB devices that aren't already redirected by high-level redirection.
  - `{*Device Setup Class GUID*}`: Redirect all devices that are members of the specified device setup class.
  - `*USBInstanceID*`: Redirect a specific USB device identified by the instance ID.
- **Default value**: `*`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

To learn how to use this property, see Configure USB redirection on Windows over the Remote Desktop Protocol.

Display settings

Here are the RDP properties that you can use to configure display settings.

`desktop size id`

- **Syntax**: `desktop size id:i:<value>`
- **Description**: Specifies the dimensions of a remote session desktop from a set of predefined options. This setting is overridden if desktopheight and desktopwidth are specified.
- **Supported values**:
  - `0`: 640×480
  - `1`: 800×600
  - `2`: 1024×768
  - `3`: 1280×1024
  - `4`: 1600×1200
- **Default value**: None. Match the local device.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`desktopheight`

- **Syntax**: `desktopheight:i:<value>`
- **Description**: Specifies the resolution height (in pixels) of a remote session.
- **Supported values**:
  - Numerical value between `200` and `8192`.
- **Default value**: None. Match the local device.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`desktopscalefactor`

- **Syntax**: `desktopscalefactor:i:*value*`
- **Description**: Specifies the scale factor of the remote session to make the content appear larger.
- **Supported values**:
  - Numerical value from the following list: `100`, `125`, `150`, `175`, `200`, `250`, `300`, `400`, `500`

- **Default value**: None. Match the local device.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

> ⓘ Note
>
> The `desktopscalefactor` property is being deprecated and will soon be unavailable.

`desktopwidth`

- **Syntax**: `desktopwidth:i:<value>`
- **Description**: Specifies the resolution width (in pixels) of a remote session.
- **Supported values**:
  - Numerical value between `200` and `8192`.
- **Default value**: None. Match the local device.
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`dynamic resolution`

- **Syntax**: `dynamic resolution:i:<value>`
- **Description**: Determines whether the resolution of a remote session is automatically updated when the local window is resized.
- **Supported values**:
  - `0`: Session resolution remains static during the session.
  - `1`: Session resolution updates as the local window resizes.
- **Default value**: `1`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`maximizetocurrentdisplays`

- **Syntax**: `maximizetocurrentdisplays:i:<value>`

- **Description**: Determines which display a remote session uses for full screen on when maximizing. Requires use multimon set to `1`. Only available on Windows App for Windows and the Remote Desktop app for Windows.
- **Supported values**:
  - `0`: Session is full screen on the displays initially selected when maximizing.
  - `1`: Session dynamically is full screen on the displays the session window spans when maximizing.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`screen mode id`

- **Syntax**: `screen mode id:i:<value>`
- **Description**: Determines whether a remote session window appears full screen when you launch the connection.
- **Supported values**:
  - `1`: A remote session appears in a window.
  - `2`: A remote session appears full screen.
- **Default value**: `2`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`selectedmonitors`

- **Syntax**: `selectedmonitors:s:<value>`
- **Description**: Specifies which local displays to use in a remote session. The selected displays must be contiguous. Requires use multimon set to `1`. Only available on Windows App for Windows, the Remote Desktop app for Windows, and the inbox Remote Desktop Connection app on Windows.
- **Supported values**:
  - A comma separated list of machine-specific display IDs. You can retrieve available IDs by running `mstsc.exe /l` from the command line. The first ID listed is set as the primary display in a remote session.
- **Default value**: None. All displays are used.

- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`singlemoninwindowedmode`

- **Syntax**: `singlemoninwindowedmode:i:<value>`
- **Description**: Determines whether a multi display remote session automatically switches to single display when exiting full screen. Requires use multimon set to **1**. Only available on Windows App for Windows and the Remote Desktop app for Windows.
- **Supported values**:
  - `0`: A remote session retains all displays when exiting full screen.
  - `1`: A remote session switches to a single display when exiting full screen.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`smart sizing`

- **Syntax**: `smart sizing:i:<value>`
- **Description**: Determines whether the local device scales the content of the remote session to fit the window size.
- **Supported values**:
  - `0`: The local window content doesn't scale when resized.
  - `1`: The local window content does scale when resized.
- **Default value**: `0`
- **Applies to**:
  - Azure Virtual Desktop
  - Remote Desktop Services
  - Remote PC connections

`use multimon`

- **Syntax**: `use multimon:i:<value>`
- **Description**: Determines whether the remote session will use one or multiple displays from the local device.
- **Supported values**:

    - `0`: A remote session uses a single display.
    - `1`: A remote session uses multiple displays.
- **Default value**: `1`
- **Applies to**:
    - Azure Virtual Desktop
    - Remote Desktop Services
    - Remote PC connections

RemoteApp

Here are the RDP properties that you can use to configure RemoteApp behavior for Remote Desktop Services.

`remoteapplicationcmdline`

- **Syntax**: `remoteapplicationcmdline:s:<value>`
- **Description**: Optional command line parameters for the RemoteApp.
- **Supported values**:
    - Valid command-line parameters for the application.
- **Default value**: None.
- **Applies to**:
    - Remote Desktop Services

`remoteapplicationexpandcmdline`

- **Syntax**: `remoteapplicationexpandcmdline:i:<value>`
- **Description**: Determines whether environment variables contained in the RemoteApp command line parameters should be expanded locally or remotely.
- **Supported values**:
    - `0`: Environment variables should be expanded to the values of the local device.
    - `1`: Environment variables should be expanded to the values of the remote session.
- **Default value**: `1`
- **Applies to**:
    - Remote Desktop Services

`remoteapplicationexpandworkingdir`

- **Syntax**: `remoteapplicationexpandworkingdir:i:<value>`

- **Description**: Determines whether environment variables contained in the RemoteApp working directory parameter should be expanded locally or remotely.
- **Supported values**:
  - `0`: Environment variables should be expanded to the values of the local device.
  - `1`: Environment variables should be expanded to the values of the remote session. The RemoteApp working directory is specified through the shell working directory parameter.
- **Default value**: `1`
- **Applies to**:
  - Remote Desktop Services

`remoteapplicationfile`

- **Syntax**: `remoteapplicationfile:s:<value>`
- **Description**: Specifies a file to be opened in the remote session by the RemoteApp. For local files to be opened, you must also enable drive redirection for the source drive.
- **Supported values**:
  - A valid file path in the remote session.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services

`remoteapplicationicon`

- **Syntax**: `remoteapplicationicon:s:<value>`
- **Description**: Specifies the icon file to be displayed in Windows App or the Remote Desktop app while launching a RemoteApp. If no file name is specified, the client will use the standard Remote Desktop icon. Only `.ico` files are supported.
- **Supported values**:
  - A valid file path to an `.ico` file.
- **Default value**: None.
- **Applies to**:
  - Remote Desktop Services

`remoteapplicationmode`

- **Syntax**: `remoteapplicationmode:i:<value>`
- **Description**: Determines whether a connection is started as a RemoteApp session.
- **Supported values**:
  - `0`: Don't launch a RemoteApp session.

    - `1`: Launch a RemoteApp session.
- **Default value**: `1`
- **Applies to**:
    - Remote Desktop Services

`remoteapplicationname`

- **Syntax**: `remoteapplicationname:s:<value>`
- **Description**: Specifies the name of the RemoteApp in Windows App or the Remote Desktop app while starting the RemoteApp.
- **Supported values**:
    - A valid application display name, for example `Microsoft Excel`.
- **Default value**: None.
- **Applies to**:
    - Remote Desktop Services

`remoteapplicationprogram`

- **Syntax**: `remoteapplicationprogram:s:<value>`
- **Description**: Specifies the alias or executable name of the RemoteApp.
- **Supported values**:
    - A valid application name or alias, for example `EXCEL`.
- **Default value**: None.
- **Applies to**:
    - Remote Desktop Services

Remote Desktop URI scheme

Article • 03/13/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This document defines the format of Uniform Resource Identifiers (URIs) for Remote Desktop. These URI schemes allow for Remote Desktop clients to be invoked with various commands.

ms-rd URI scheme

> **Note**
>
> The ms-rd URI scheme is currently only supported with the Windows Desktop client (MSRDC).

The ms-rd URI provides the option to specify a command for the client and a set of parameters specific to the command using the following format:

```
ms-rd:command?parameters
```

`Parameters` uses the query string format of key=value pair separated by & to provide additional information for the given command:

```
param1=value1&param2=value2&...
```

Commands and parameters

Here's the list of currently supported commands and their corresponding parameters.

Using `ms-rd:` without any commands launches the client.

Subscribe

This command launches the client and starts the subscription process.

Command name: subscribe

Command parameters:

Expand table

Parameter	Description	Values
url	Specifies the Workspace URL.	A valid URL, such as `https://contoso.com`>`.

Example: `ms-rd:subscribe?url=https://contoso.com`

Legacy rdp URI scheme

> **Note**
>
> The following URI scheme is only supported with the clients for macOS, iOS, and Android devices.

Microsoft Remote Desktop uses the URI scheme rdp://query_string to store preconfigured attribute settings that are used when launching the client. The query strings represent a single or set of RDP attributes provided in the URL.

The RDP attributes are separated by the ampersand symbol (&). For example, when connecting to a PC, the string is:

```
rdp://full%20address=s:mypc:3389&audiomode=i:2&disable%20themes=i:1
```

This table gives a complete list of supported attributes that can be used with the iOS, Mac, and Android Remote Desktop clients. (An "x" in the platform column indicates the attribute is supported. The values denoted by chevrons (<>) represent the values that the Remote Desktop client supports.)

Expand table

RDP attribute	Android	Mac	iOS
allow desktop composition=i:<0 or 1>	x	x	x
allow font smoothing=i:<0 or 1>	x	x	x

RDP attribute	Android	Mac	iOS
alternate shell=s:<string>	x	x	x
audiomode=i:<0, 1, or 2>	x	x	x
authentication level=i:<0 or 1>	x	x	x
connect to console=i:<0 or 1>	x	x	x
disable cursor settings=i:<0 or 1>	x	x	x
disable full window drag=i:<0 or 1>	x	x	x
disable menu anims=i:<0 or 1>	x	x	x
disable themes=i:<0 or 1>	x	x	x
disable wallpaper=i:<0 or 1>	x	x	x
drivestoredirect=s:* (this is the only supported value)	x	x	
desktopheight=i:<value in pixels>		x	
desktopwidth=i:<value in pixels>		x	
domain=s:<string>	x	x	x
full address=s:<string>	x	x	x
gatewayhostname=s:<string>	x	x	x
gatewayusagemethod=i:<1 or 2>	x	x	x
prompt for credentials on client=i:<0 or 1>		x	
loadbalanceinfo=s:<string>	x	x	x
redirectprinters=i:<0 or 1>		x	
remoteapplicationcmdline=s:<string>	x	x	x
remoteapplicationmode=i:<0 or 1>	x	x	x
remoteapplicationprogram=s:<string>	x	x	x
shell working directory=s:<string>	x	x	x
Use redirection server name=i:<0 or 1>	x	x	x
username=s:<string>	x	x	x
screen mode id=i:<1 or 2>		x	

RDP attribute	Android	Mac	iOS
session bpp=i:<8, 15, 16, 24, or 32>		x	
use multimon=i:<0 or 1>		x	

Feedback

Was this page helpful? Yes No

Remote Desktop client - supported configuration

Article • 03/13/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Learn which PCs you can access by using supported configurations for Remote Desktop clients.

Supported operating systems for Remote Desktop client connections

You can connect to PCs that run the following Windows operating systems:

- Windows 11 Pro
- Windows 11 Enterprise
- Windows 10 Pro
- Windows 10 Enterprise
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

> **Note**
>
> Windows SKUs that aren't listed in this section, such as Windows 10 Home, aren't compatible with connecting remotely.

Supported operating systems for Remote Desktop Gateway, Web Access and RemoteApp

The following operating systems can serve as Remote Desktop Gateway, Web Access, and RemoteApp:

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

RD Gateway messaging isn't supported

Remote Desktop Client doesn't support RD Gateway messaging. Verify that the Remote Desktop Resource Access Policy (RD RAP) for your RD Gateway server doesn't specify **Only allow computers with support for RD Gateway Messaging**, or you aren't able to connect.

Feedback

Enable Remote Desktop on your PC

06/18/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

You can use Remote Desktop to connect to and control your PC from a remote device by using Windows App or the Microsoft Remote Desktop client. When you allow remote connections to your PC, you can use another device to connect to your PC and have access to all of your apps, files, and network resources as if you were sitting at your desk. If you only need to use your PC locally, there's no need to enable Remote Desktop. Enabling Remote Desktop opens a port on your PC, making it accessible to devices on your local network. Only enable Remote Desktop on trusted networks, such as your home network, and avoid enabling it on devices that require strict access controls. When Remote Desktop is enabled, members of the Administrators group and any users you specify can connect remotely. Make sure all accounts with remote access have strong, unique passwords to help protect your PC.

> **Note**
>
> You can use Remote Desktop to connect to computers running Windows Professional, Enterprise, Education editions, and Windows Server editions. These editions can act as hosts for incoming Remote Desktop connections. However, Windows Home editions can't serve as Remote Desktop hosts, though they can be used as clients to connect to other systems that support Remote Desktop hosting.

If you need to connect to your PC from outside of the network your PC is running on, you can use port forwarding or set up a VPN. To learn more, see Allow access to your PC from outside your PC's network.

Prerequisites

You must be a member of the **Administrators** group or have administrative privileges to change this system setting.

To connect to a remote PC, ensure the following requirements are met:

- The remote PC is powered on and connected to the network.
- Remote Desktop is enabled on the remote PC.
- You have network access to the remote PC (locally or via the Internet).
- Your user account is permitted to connect (the account is in the list of allowed users).
- Remote Desktop connections are allowed through the remote PC's firewall.

Enable Remote Desktop

You can configure your PC for remote access with a few easy steps.

1. Select **Start**, select **Settings**, then select **System**.
2. Select **Remote Desktop**, toggle the option **Enable Remote Desktop** to **On**.
3. Select **Confirm** on the pop-up dialog box to enable Remote Desktop.

By default, the **Make my PC discoverable on private networks to enable automatic connection from a remote device** checkbox is checked.

You can also configure other user accounts to have remote access to your device by following these steps:

1. While in the Remote Desktop settings, select **Select users that can remotely access this PC**.

   In later releases of Windows and Windows Server, select **Remote Desktop users**.

2. Select **Add**, type the username, then select **OK**. More than one account can be added at a time.

Remote connect to a device

To open the **Remote Desktop Connection** app, follow these steps:

1. On your local PC, select **Start**, type **Remote Desktop Connection** and open the app.
2. In the **Remote Desktop Connection** window, enter the computer name or IP address of the remote PC you want to connect to.
3. Select **Connect**, provide the credentials, then select **OK**.

You can also connect to a device using the Windows App. To learn more, see Get started with Windows App to connect to devices and apps.

Why allow connections only with Network Level Authentication?

Network Level Authentication (NLA) adds an extra layer of security to Remote Desktop connections. With NLA enabled, users must authenticate themselves before a remote session is established, reducing the risk of unauthorized access and helping to protect your PC from malicious users and software. Enabling NLA is recommended for most environments, as it ensures only trusted users and devices can connect. However, if you need to connect from

older devices or clients that don't support NLA, you might need to disable this option temporarily. For best security, keep NLA enabled whenever possible.

To learn more, see Configure NLA for RDS Connections.

See also

- Remote Desktop Services overview in Windows Server
- Troubleshoot Remote Desktop connections to an Azure virtual machine
- Troubleshoot authentication errors when you use RDP to connect to Azure VM

Remote Desktop - Allow access to your PC from outside your PC's network

06/26/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

When you connect to your PC by using a Remote Desktop client, you're creating a peer-to-peer connection. This means you need direct access to the PC (sometimes called "the host"). If you need to connect to your PC from outside of the network your PC is running on, you need to enable that access. You have a couple of options: use port forwarding or set up a VPN.

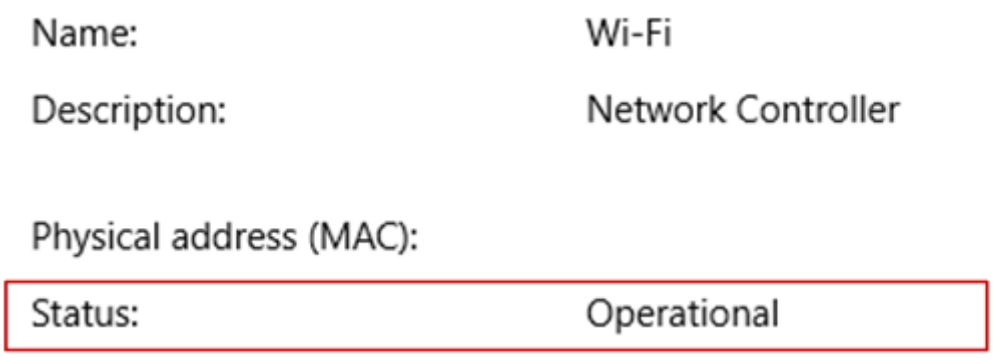
Enable port forwarding on your router

Port forwarding simply maps the port on your router's IP address (your public IP) to the port and IP address of the PC you want to access.

Specific steps for enabling port forwarding depend on the router you're using, so you need to search online for your router's instructions. For a general discussion of the steps, check out How to Set Up Port Forwarding on a Router.

Before you map the port, you need the following information:

- PC internal IP address: Look in **Settings** > **Network & Internet** > **Status** > **View your network properties**. Find the network configuration with an "Operational" status and then get the **IPv4 address**.



- Your public IP address (the router's IP). There are many ways to find your public IP. You can search (in Bing or Google) for "my IP" or view the network properties (for Windows 10 or later).

- Port number being mapped. Typically, the port is 3389 - that's the default port used by Remote Desktop connections.

- Admin access to your router.

  > **⚠ Warning**
  >
  > You're opening your PC up to the internet, which isn't recommended. If you must, make sure you have a strong password set for your PC. It's preferable to **use a VPN**.

After you map the port, you'll be able to connect to your host PC from outside the local network by connecting to the public IP address of your router.

The router's IP address can change - your internet service provider (ISP) can assign you a new IP at any time. To avoid running into this issue, consider using Dynamic DNS. Dynamic DNS (DDNS) lets you connect to the PC using an easy to remember domain name, instead of the IP address. Your router automatically updates the DDNS service with your new IP address, should it change.

Most routers allow you to define which source IP or source network can use port mapping. So, if you know you're only going to connect from work, you can add the IP address for your work network - that lets you avoid opening the port to the entire public internet. If the host you're using to connect uses dynamic IP address, set the source restriction to allow access from the whole range of that particular ISP.

You might also consider setting up a static IP address on your PC so the internal IP address doesn't change. If you do that, the router's port forwarding always points to the correct IP address.

Use a VPN

If you connect to your local area network by using a virtual private network (VPN), you don't have to open your PC to the public internet. Instead, when you connect to the VPN, your RD client acts like it's part of the same network and is able to access your PC. There are many VPN services available - you can find and use whichever works best for you.

Change the Remote Desktop listening port on your computer

06/30/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop allows remote connections to computers running Windows or Windows Server over the Remote Desktop Protocol (RDP), listening on port 3389 by default. For security or configuration purposes, you might want to change this listening port. This article provides step-by-step instructions to modify the listening port using either PowerShell or the Registry Editor.

Prerequisites

Before you begin, make sure you have the following items:

- Administrator access, or equivalent, to the computer you want to connect to.
- A computer with Remote Desktop enabled. For more information, see Enable Remote Desktop.
- A client to test the changes from, such as Remote Desktop Connect (`mstsc.exe`) or Windows App.

Configure the Remote Desktop listening port

The listening port for Remote Desktop is specified in the registry. To change the registry value, here's how to change it using PowerShell or the Registry Editor. Select the relevant tab for the method you prefer.

PowerShell

To change the listening port using PowerShell, follow these steps:

1. Open PowerShell as an administrator.
2. Check the current port by running the following PowerShell command:

```
PowerShell
```

```
Get-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp' -name 'PortNumber'
```

The output is similar to the following example:

PowerShell

```
PortNumber    : 3389
PSPath        :
Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Control\Terminal Server\WinStations\RDP-Tcp
PSParentPath :
Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Control\Terminal Server\WinStations
PSChildName   : RDP-Tcp
PSDrive       : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry
```

3. Change the port by running the following PowerShell command. Be sure to replace `<Port Number>` with the new port number.

PowerShell

```
$portValue = '<Port Number>'

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp' -name 'PortNumber' -Value $portValue
```

Add the new port to the Windows Firewall

If you use the Windows Firewall, you need to add a new inbound rule to allow traffic on the new port. For more information about the different methods you can use to configure the Windows Firewall, see Windows Firewall tools.

> **Important**
>
> If you use any other firewall make sure you or your administrator permit connections to the new port number.

To create new Windows Firewall rules to allow the new port by running the following PowerShell command as an administrator. Be sure to replace `<Port Number>` with the new port number.

PowerShell

```
$portValue = '<Port Number>'

New-NetFirewallRule -DisplayName 'RDPPORTLatest-TCP-In' -Profile Public -Direction
Inbound -Action Allow -Protocol TCP -LocalPort $portValue
New-NetFirewallRule -DisplayName 'RDPPORTLatest-UDP-In' -Profile Public -Direction
Inbound -Action Allow -Protocol UDP -LocalPort $portValue
```

Test the new Remote Desktop listening port

The next time you connect to this computer by using the Remote Desktop Connection or other client, enter the hostname along with the new port. For example, if you changed the port to use 3390 on computer `pc1.contoso.com`, the address is `pc1.contoso.com:3390`.

Related content

- Enable Remote Desktop on your PC
- Allow access to your PC from outside your PC's network
- Frequently asked questions about Remote PC connections

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

Use the Remote Desktop Connection app to connect to a remote PC using single sign-on with Microsoft Entra authentication

07/07/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Desktop Connection app (MSTSC) supports single sign-on with Microsoft Entra authentication, enabling seamless connections to remote PCs without repeated credential entry. This authentication method streamlines remote access by automatically passing through your Microsoft Entra credentials when you're signed in to your local device, improving productivity and user experience.

This article shows you how to configure and use single sign-on with Microsoft Entra authentication in the Remote Desktop Connection app to connect to remote PCs.

Prerequisites

To connect to a remote PC using single sign-on with Microsoft Entra authentication, you need:

- The remote PC and your local device must be running one of the following operating systems:
  - Windows 11 with 2022-10 Cumulative Updates for Windows 11 (KB5018418) or later installed.
  - Windows 10, version 20H2 or later with 2022-10 Cumulative Updates for Windows 10 (KB5018410) or later installed.
  - Windows Server 2022 with 2022-10 Cumulative Update for Microsoft server operating system (KB5018421) or later installed.
- Remote Desktop needs to be enabled in your remote PC. You can follow the steps in Enable Remote Desktop on your PC to enable Remote Desktop.
- The remote PC must be network addressable by hostname, resolving to the IP address of the remote PC. Using an IP address directly isn't supported.
- The remote PC must be Microsoft Entra joined or Microsoft Entra hybrid joined. There's no requirement for the local device to be joined to a domain or Microsoft Entra. As a result, this method allows you to connect to the remote PC from:

    - Microsoft Entra joined or Microsoft Entra hybrid joined devices.
    - Active Directory domain joined devices.
    - Workgroup devices.

- If you're accessing an Azure VM, ensure the Microsoft Entra account is assigned the **Virtual Machine Administrator Login** or **Virtual Machine User Login** role. For more information, see Steps to assign an Azure role.

- If your organization is using Microsoft Entra Conditional Access, your local device must satisfy the Conditional Access requirements to allow connection to the remote computer. Conditional Access policies can be applied to the application **Microsoft Remote Desktop** with ID **a4a365df-50f1-4397-bc59-1a1564b8bb9c** to control access to the remote PC when single sign-on is enabled.

  > **Important**
  >
  > We recommend that you enforce multifactor authentication Conditional Access and configure a periodic reauthentication policy using Sign-in frequency control for added security.

Connect to a remote PC using single sign-on with Microsoft Entra authentication

Here's how to connect to a remote PC using single sign-on with Microsoft Entra authentication

1. Launch the Remote Desktop Connection app on your local device from the Start menu, or by running `mstsc.exe` from a command prompt.

2. Select **Show Options** to expand the Remote Desktop Connection client, then select the **Advanced** tab.

3. Under **User authentication**, check the box **Use a web account to sign in to the remote computer**. This option is equivalent to the `enablerdsaadauth` RDP property. For more information, see Supported RDP properties with Remote Desktop Services.

4. Select the **General** tab and enter the NetBIOS domain name or fully qualified domain name (FQDN) of the remote PC in the **Computer** field. The name you enter must match the hostname of the remote PC in Microsoft Entra ID and be network addressable, resolving to the IP address of the remote PC. Using an IP address directly isn't supported.

5. Select **Connect**.

6. If you're prompted for credentials, your user account in Microsoft Entra ID might be automatically selected. If your account isn't automatically selected, specify the user name for your account in the format `user@domain.com` (the User Principal Name (UPN)).

7. Select **OK** to connect. You're prompted to allow the remote desktop connection when connecting to a new remote PC. Microsoft Entra remembers up to 15 hosts for 30 days before prompting again. If you see this dialogue, select **Yes** to connect.

Session disconnection when locked

The Windows lock screen in the remote session doesn't support Microsoft Entra authentication tokens or passwordless authentication methods like FIDO keys. The lack of support for these authentication methods means that users can't unlock their screens in a remote session. When you try to lock a remote session, either through user action or system policy, the service instead disconnects the session and sends a message to the user explaining they were disconnected.

Disconnecting the session also ensures that when the connection is relaunched after a period of inactivity, Microsoft Entra ID reevaluates the applicable conditional access policies.

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

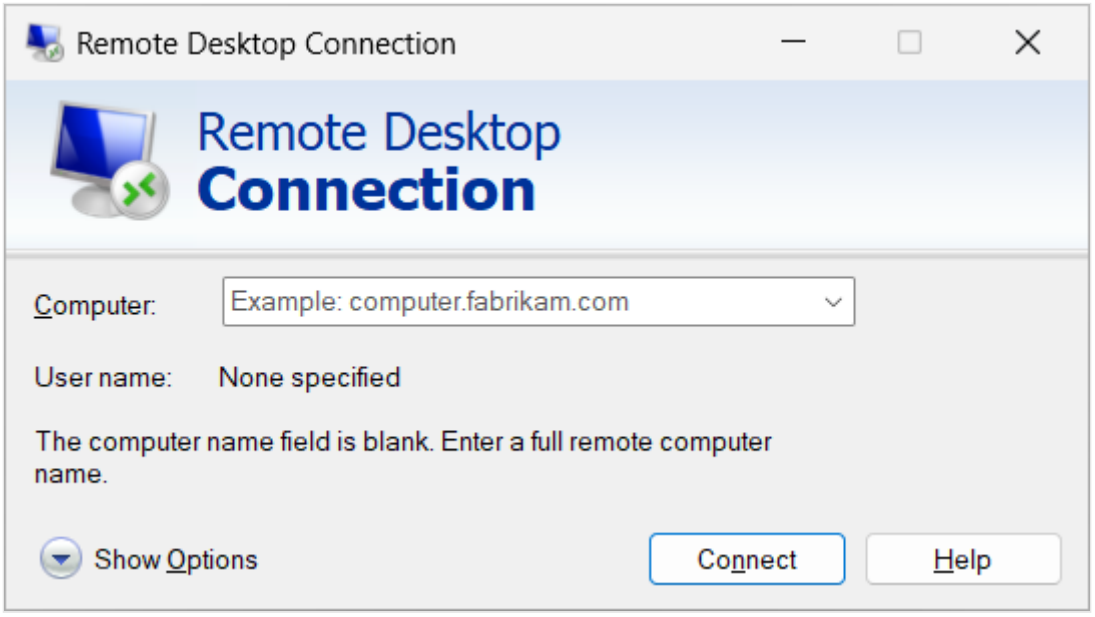
Uninstall and reinstall the built-in Remote Desktop Connection app in Windows

06/06/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Remote Desktop Connection is a built-in Windows app that allows you to connect remotely to other computers. Starting with Windows 11 version 23H2, you can uninstall this app if you no longer need it. This article provides step-by-step instructions on how to uninstall the Remote Desktop Connection app and how to reinstall it later if necessary.

Here's a screenshot of the Remote Desktop Connection app that comes preinstalled in Windows:



> **ⓘ Important**
>
> When you uninstall Remote Desktop Connection, you also become unable to use the **RemoteApp and Desktop Connections** control panel.

Prerequisites

Before you can uninstall Remote Desktop Connection, you must meet the following prerequisites:

- A computer running Windows 11 23H2 or later.

- You must use an account that has administrator privileges on the computer where you want to uninstall the app.

Uninstall the Remote Desktop Connection app

Select one of the following tabs to see how to uninstall the Remote Desktop Connection app using either the graphical user interface (GUI) or the command prompt.

GUI

To uninstall the Remote Desktop Connection app using the GUI:

1. Open the Start menu, then type **Installed apps**. Select the matching system settings entry in the search results to open the **Installed apps** settings page.
2. Find or search for **Remote Desktop Connection**, select the three dots to the right-hand side, then select **Uninstall**.
3. Confirm you want to uninstall the app by selecting **Uninstall**.
4. When prompted, restart your computer to complete the installation.

Reinstall the Remote Desktop Connection app

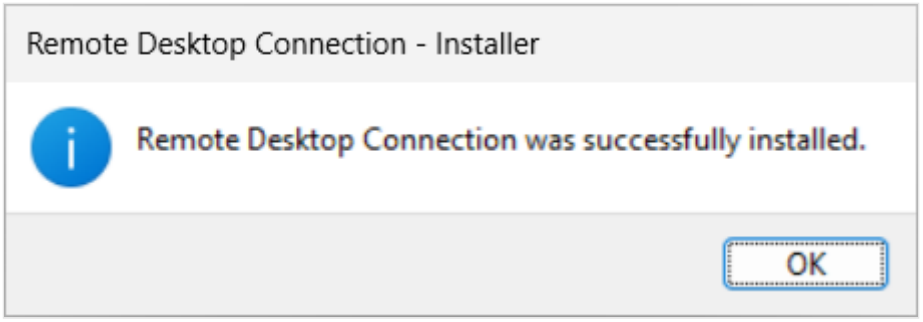
After you uninstall the Remote Desktop Connection app, you can reinstall it by following these instructions:

GUI

To reinstall the Remote Desktop Connection app using the GUI:

1. Download the installer for the Remote Desktop Connection app from the following links and save it somewhere you can remember. Select the version that matches your computer's architecture:
   - Windows 64-bit *(most common)*
   - Windows 32-bit
   - Windows ARM64
2. Open the folder where you downloaded the installer, then double-click the file to run the installer. The installer completes without any user interaction.

3. Once the installation is complete, you get the confirmation message **Remote Desktop Connection was successfully installed**.



4. Open the Start menu, then type **Remote Desktop Connection** to find and launch the app.

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

Understanding security warnings when opening Remote Desktop (RDP) files

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows Server 2012 R2, ✅ Windows Server 2012, ✅ Windows 11, ✅ Windows 10

Starting with the April 2026 security update, the Remote Desktop Connection app shows new security warnings when you open RDP files. This article explains what these warnings mean and how to respond to them safely.

What is Remote Desktop?

Remote Desktop lets you connect to a computer in another location, such as your work PC, over a network connection like the internet. You can see the remote computer's screen, open files, run applications, and use your mouse and keyboard as if you were sitting in front of it.

Risks of RDP files

An RDP file tells the Remote Desktop Connection app how to connect to a remote computer. Depending on its settings, the file can also share parts of your local device, such as your clipboard, drives, or camera, with the remote computer.

Malicious actors misuse this capability by sending RDP files through phishing emails. When a victim opens the file, their device silently connects to a server controlled by the attacker and shares local resources, giving the attacker access to files, credentials, and more.

> **Important**
>
> Never open an RDP file you weren't expecting, even if the email looks legitimate. When in doubt, contact your IT department.

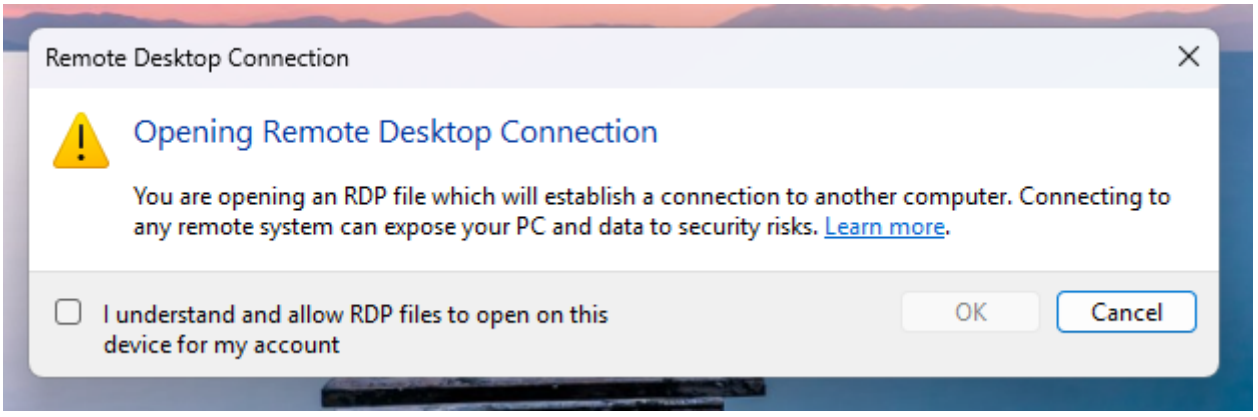
How to stay safe

Pausing to think before you click is the single most effective defense against phishing. Here are practical steps:

- **Don't open unexpected RDP files.** If you receive one you weren't expecting, don't open it - even if the email looks legitimate. Verify with the sender through a separate channel (like a phone call).
- **Check the remote computer address.** If you don't recognize the computer name or address in the dialog, don't connect.
- **Only enable redirections you need.** Leave all others unchecked.
- **Pay attention to the dialog and whether the publisher can be verified.** Verify the publisher even when the file is signed.
- **Report suspicious RDP files to your IT security team.**

The first-launch dialog

The first time you open an RDP file after installing this update, an educational dialog appears. It explains what RDP files are and warns about phishing risks. After you allow RDP file connections in this dialog, it doesn't appear again for your account.



The connection security dialog

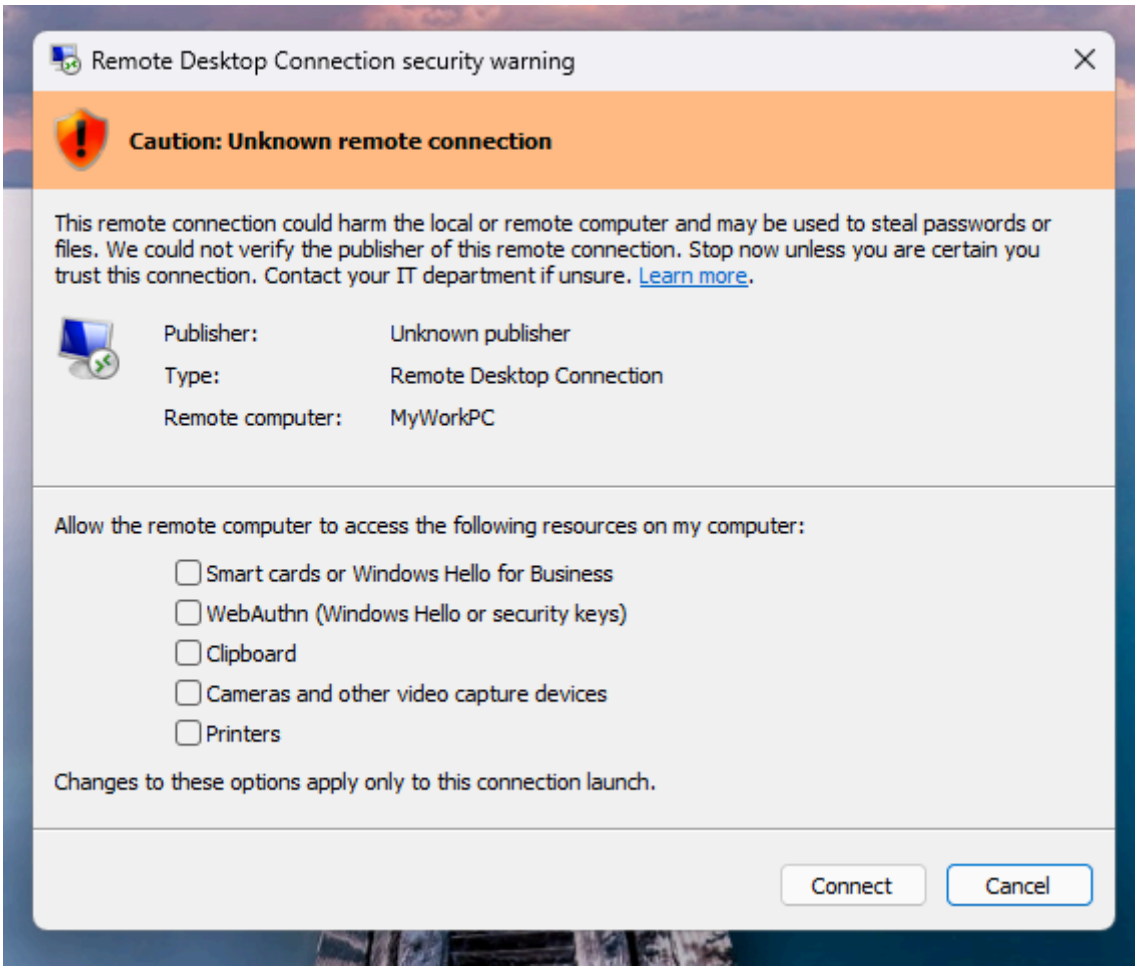
Every time you open an RDP file, a security dialog appears before any connection is made. It shows the remote computer address and a check box for each local resource the file wants to access. Access to all these resources is **off by default** - you must explicitly enable each one.

This dialog exists in two versions depending on whether the publisher of the RDP file can be verified.

RDP files with no verifiable publisher

When an RDP file is **not digitally signed**, there's no way to verify who created it or whether it was tampered with. In this case, the security dialog shows a banner titled **Caution: Unknown**

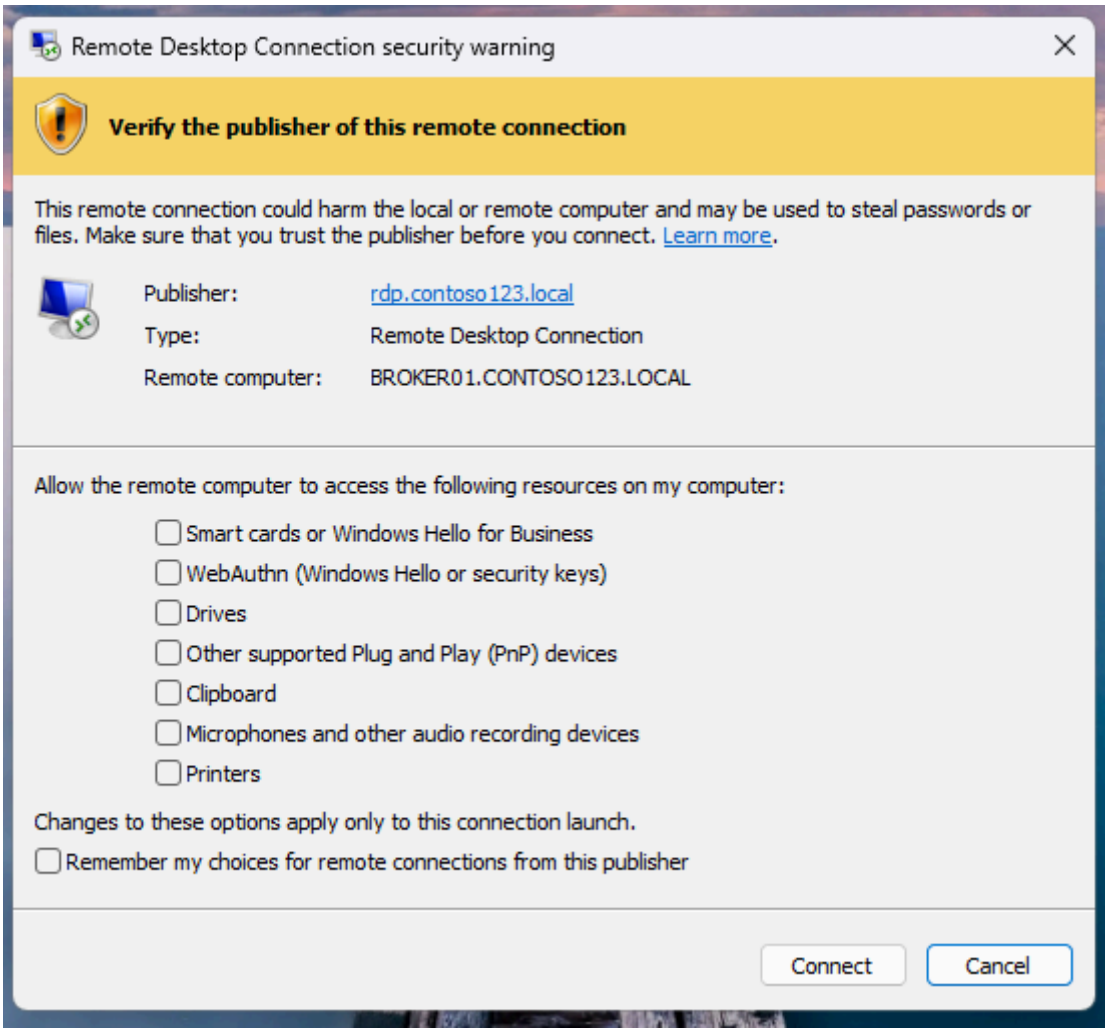
remote connection and sets the **Publisher** field to "Unknown publisher," as the following image shows.



> ⚠ **Warning**
>
> An unsigned RDP file can come from anyone. Treat it with extreme caution, especially if you received it by email or downloaded it from the internet.

RDP files with a verifiable publisher

When a publisher **digitally signs** an RDP file, the signature confirms who created or distributed it. The publisher's name appears in the dialog, and the banner is titled "Verify the publisher of this remote connection," as the following image shows.



A signature confirms the identity of the entity that created the file and that the file wasn't tampered with since it was signed. It doesn't guarantee the file is safe. Cyberattackers can sign files by using names that closely resemble legitimate organizations - for example, "Contoso Security" instead of "Contoso Ltd." Always read the publisher name carefully and verify it matches the organization you expect.

> ⓘ **Note**
>
> Your IT department might configure your computer to trust specific publishers. When an RDP file is signed by a trusted publisher, the experience might differ based on your organization's policies.

Understanding redirections

When you open an RDP file, it can request access to resources on your local device. These requests are called redirections. They share parts of your local device with the remote computer. After this update, all redirections requested by RDP files are **turned off by default** unless you opt into them.

The following list explains each redirection type and the risk it poses. Older versions of Windows support a different set of redirections, so not all of these might be available on your device.

Drives

- **What it does:** Makes your local drives (hard drives, USB drives, network-mapped drives) accessible from the remote computer. The remote computer can read files from and write files to your local drives.
- **Risk:** This redirection is one of the most dangerous. An attacker can:
  - Steal files from your local drives, including documents, databases, and credentials stored on disk.
  - Plant malware on your local drives. For example, the attacker could write a malicious program to your Startup folder, which runs the next time you sign in.
  - Access network shares that are mapped as drives on your device, potentially reaching other systems in your organization.

Clipboard

- **What it does:** Shares the contents of your clipboard (anything you copy and paste) between your device and the remote computer.
- **Risk:** A cyberattacker could read anything you copy on your local device - including passwords, sensitive text, or confidential information. The cyberattacker could also place malicious content in your clipboard, which you might then paste into a local application.

Smart cards or Windows Hello for Business

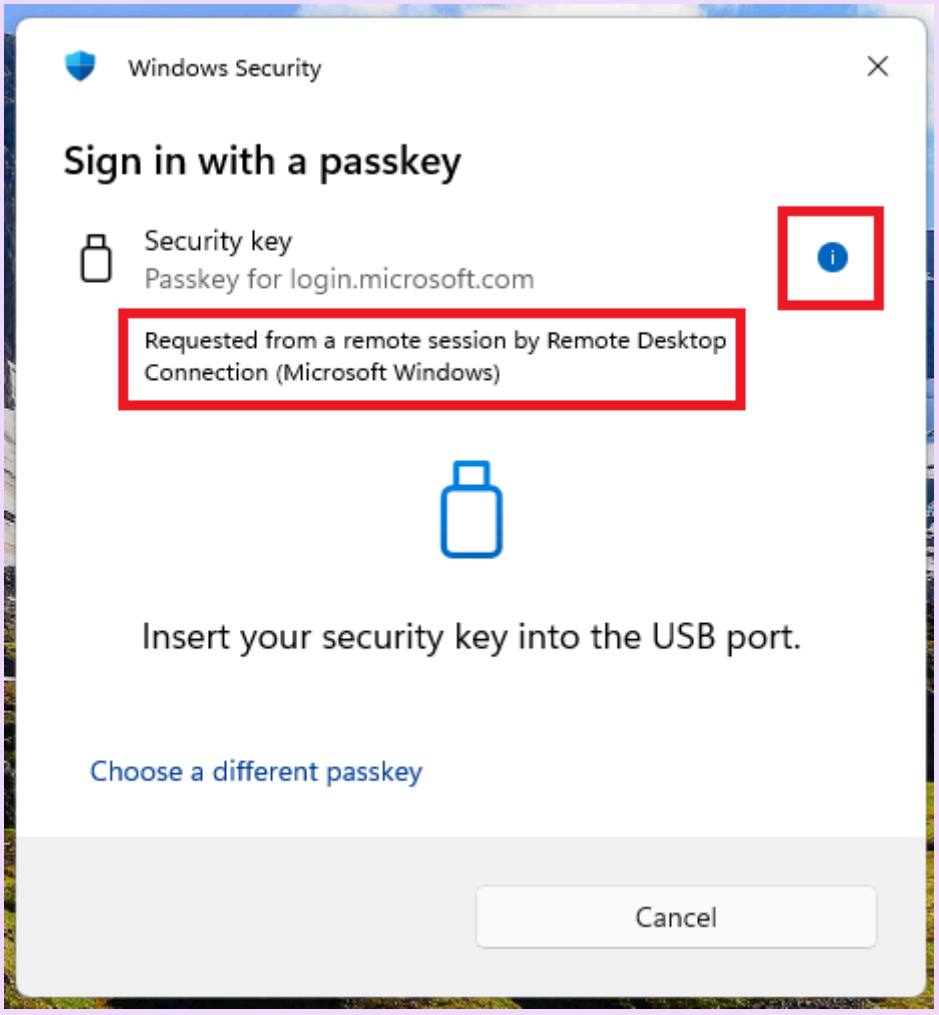
- **What it does:** Allows the remote computer to use smart cards or Windows Hello for Business credentials that you connect or configure on your local device.
- **Risk:** An attacker can use your redirected smart card or Windows Hello for Business credentials to authenticate as you on the remote system or on other systems accessible from the remote computer. This action can lead to unauthorized access to your organization's resources using your identity.

WebAuthn (Windows Hello or security keys)

- **What it does:** Allows the remote computer to use your local FIDO2 security keys or passkeys to complete web authentication challenges.

- **Risk:** Authentication prompts might be redirected from a malicious remote session to the local device and used for phishing.

> **Note**
>
> When a WebAuthn request is redirected through a remote session, Windows displays this information in the authentication prompt. If you see an indication that the request is coming from a remote connection and you didn't expect it, don't approve the request.



Microphones and other audio recording devices

- **What it does:** Shares your local microphone and audio recording devices with the remote computer, so remote applications can record audio from your environment.
- **Risk:** With access to your microphone, an attacker can eavesdrop on conversations, meetings, or other audio in your environment without your knowledge.

Cameras and other video capture devices

- **What it does:** Shares your local cameras and video capture devices with the remote computer, so remote applications can record video from your environment.
- **Risk:** An attacker with access to your camera can see your surroundings, read documents on your desk, observe your screen, or conduct visual surveillance without your awareness.

Location

- **What it does:** Shares your device's geographic location with the remote computer.
- **Risk:** An attacker could determine your physical location, which might be sensitive depending on your role or context (for example, military, law enforcement, or executive personnel).

Printers

- **What it does:** Makes your local printers available from the remote computer, so remote applications can print to your local printers.
- **Risk:** A cyberattacker who controls a remote session could send print jobs to your printers, potentially wasting resources or printing misleading documents that appear to be local.

Ports

- **What it does:** Shares your local serial (COM) and parallel (LPT) ports with the remote computer.
- **Risk:** An attacker could access devices connected to these ports, such as specialized hardware or legacy peripherals, and potentially read or send data through them.

Point-of-service devices

- **What it does:** Shares point-of-service (POS) devices, such as barcode scanners and receipt printers, that you connect to your local device.
- **Risk:** An attacker could interact with POS equipment, potentially interfering with transactions or reading financial data from connected devices.

Other supported Plug and Play (PnP) devices

- **What it does:** Shares other supported Plug and Play devices with the remote computer, including a wide range of USB and other peripherals.

- **Risk:** Depending on the device, an attacker could read data, interact with the device, or use it as a pivot point for further attacks.

Other supported RemoteFX USB devices

- **What it does:** Shares supported USB devices with the remote computer at a low level by using RemoteFX USB redirection. The remote computer gets more direct access to the device than standard redirection.
- **Risk:** RemoteFX USB redirection provides deeper access to USB devices. An attacker could exploit this access to interact with sensitive USB devices, such as authentication tokens, storage devices, or specialized hardware, at a level that bypasses typical application-layer protections.

Frequently asked questions

What if my organization's RDP files show a warning with an unknown publisher?

This warning means your organization's RDP files are unsigned. Contact your IT department - they can sign the files so they show their publisher instead.

Does this update affect connections I start manually from Remote Desktop Connection?

No. This update only affects connections started by opening an RDP file. If you type a computer name directly into Remote Desktop Connection, the experience is unchanged.

I'm seeing this warning from Azure Virtual Desktop or Windows 365. Is it safe?

RDP files from Microsoft services like Azure Virtual Desktop and Windows 365 are typically signed by Microsoft. You shouldn't see the new security dialog when connecting to these services. If you do, don't proceed - contact your IT department to investigate.

I'm an app developer. My app uses the Remote Desktop ActiveX Control. How do I control this dialog?

If your application relies on the Remote Desktop ActiveX Control (`mstscax.dll`), you can use IMsRdpExtendedSettings Property to control the dialog behavior. The

`RedirectionWarningDialogVersion` property lets you configure whether to disable the new version of the security dialog after the update.

I'm an IT administrator. How do I temporarily revert the new security dialog?

> ⚠ **Warning**
>
> If you use Registry Editor incorrectly, you might cause serious problems that might require you to reinstall the operating system. Microsoft can't guarantee that you can solve problems that result from using Registry Editor incorrectly. Use Registry Editor at your own risk.

If the update causes temporary disruptions in your environment, you can revert to the previous dialog behavior by setting a registry value.

1. Select **Start**, type **Registry Editor**, and then open it.
2. Go to and modify the key: `HKLM\Software\Policies\Microsoft\Windows NT\Terminal Services\Client` with the following values:
    - **Name**: RedirectionWarningDialogVersion
    - **Type**: REG_DWORD
    - **Data**: 1

> ⚠ **Warning**
>
> A future Windows update might remove support for this setting, even on older versions of Windows. Plan to transition your environment to work with the new security dialog.

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

---

Last updated on 04/14/2026

Frequently asked questions about the Remote Desktop clients

Summarize this article for me

Now that you set up the Remote Desktop client on your device (Android, Mac, iOS, or Windows), you might have questions. Here are answers to the most commonly asked questions about the Remote Desktop clients.

Most these questions apply to all of the clients, but there are a few client specific items.

If you have more questions that you'd like us to answer, visit the Microsoft Support Community to ask your question.

Setting up

Which PCs can I connect to?

Check out the supported configuration article for information about what PCs you can connect to.

How do I set up a PC for Remote Desktop?

I have my device set up, but I don't think the PC's ready. Help?

The Remote Desktop Setup Wizard walks you through getting your PC ready for remote access. Download and run that tool on your PC to get everything set.

Otherwise, if you prefer to do things manually, read on.

For Windows, complete the following steps:

1. On the device you want to connect to, open **Settings**.
2. Select **System** and then **Remote Desktop**.
3. Use the slider to enable Remote Desktop.

4. In general, it's best to keep the PC awake and discoverable to facilitate connections. Select **Show settings** to go to the power settings for your PC, where you can change this setting.

   > ⓘ **Note**
   >
   > You can't connect to a PC that's asleep or hibernating, so make sure the settings for sleep and hibernation on the remote PC are set to **Never**. (Hibernation isn't available on all PCs.)

Make note of the name of this PC under **How to connect to this PC**. You need this name to configure the clients.

You can grant permission for specific users to access this PC - to do that, select **Select users that can remotely access this PC**. Members of the Administrators group automatically have access.

For Windows 8.1, follow the instructions to allow remote connections in Connect to another desktop using Remote Desktop Connections.

Connection, gateway, and networks

Why can't I connect using Remote Desktop?

Here are some possible solutions to common problems you might encounter when trying to connect to a remote PC. If these solutions don't work, you can find more help on the Microsoft Community website.

- **The remote PC can't be found.** Make sure you have the right PC name, and then check to see if you entered that name correctly. If you still can't connect, try using the IP address of the remote PC instead of the PC name.
- **There's a problem with the network.** Make sure you have internet connection.
- **The Remote Desktop port might be blocked by a firewall.** If you're using Windows Firewall, follow these steps:

  1. Open Windows Firewall.
  2. Select **Allow an app or feature through Windows Firewall**.
  3. Select **Change settings**. You might be asked for an admin password or to confirm your choice.

4. Under **Allowed apps and features**, select **Remote Desktop**, and then tap or select **OK**.

   If you're using a different firewall, make sure the port for Remote Desktop (usually 3389) is open.

- **Remote connections might not be set up on the remote PC.** To fix this problem, scroll back up to How do I set up a PC for Remote Desktop? question in this article.
- **The remote PC might only allow PCs to connect that have Network Level Authentication set up.**
- **The remote PC might be turned off.** You can't connect to a PC that's off, asleep, or hibernating. Make sure the settings for sleep and hibernation on the remote PC are set to **Never** (hibernation isn't available on all PCs.).

Why can't I find or connect to my PC?

Check the following things:

- Is the PC on and awake?
- Did you enter the right name or IP address?

  > **Important**
  >
  > Using the PC name requires your network to resolve the name correctly through DNS. In many home networks, you have to use the IP address instead of the host name to connect.

- Is the PC on a different network? Did you configure the PC to let outside connections through? Check out Allow access to your PC from outside your network for help.
- Are you connecting to a supported Windows version?

Why can't I sign in to a remote PC?

If you can see the sign-in screen of the remote PC but you can't sign in, you might not be added to the Remote Desktop Users Group or to any group with administrator rights on the remote PC. Ask your system admin to do add you to the appropriate group.

Which connection methods are supported for company networks?

If you want to access your office desktop from outside your company network, your company must provide you with a means of remote access. The RD Client currently supports the following connection methods:

- Terminal Server Gateway or Remote Desktop Gateway
- Remote Desktop Web Access
- VPN (through iOS built-in VPN options)

VPN doesn't work

VPN issues can have several causes. The first step is to verify that the VPN works on the same network as your PC or Mac computer. If you can't test with a PC or Mac, you can try to access a company intranet web page with your device's browser.

Other things to check:

- **The 3G network blocks or corrupts VPN.** There are several 3G providers in the world who seem to block or corrupt 3G traffic. Verify VPN connectivity works correctly for over a minute.
- **L2TP or PPTP VPNs.** If you're using L2TP or PPTP in your VPN, set Send All Traffic to ON in the VPN configuration.
- **VPN is misconfigured.** A misconfigured VPN server can be the reason why the VPN connections never worked or stopped working after some time. Ensure testing with the iOS device's web browser or a PC or Mac on the same network if this problem occurs.

How can I test if VPN is working properly?

Verify that VPN is enabled on your device. You can test your VPN connection by going to a webpage on your internal network or using a web service which is only available via the VPN.

How do I configure L2TP or PPTP VPN connections?

If you're using L2TP or PPTP in your VPN, make sure to set **Send all traffic** to **ON** in the VPN configuration.

Web client

Which browsers can I use?

The web client supports Microsoft Edge, Mozilla Firefox (v55.0 and later), Safari, and Google Chrome.

What PCs can I use to access the web client?

The web client supports Windows, macOS, Linux, and ChromeOS. Mobile devices aren't supported at this time.

Can I use the web client in a Remote Desktop deployment without a gateway?

No. The client requires a Remote Desktop Gateway to connect. Don't know what that means? Ask your admin about it.

Does the Remote Desktop web client replace the Remote Desktop Web Access page?

No. The Remote Desktop web client is hosted at a different URL than the Remote Desktop Web Access page. You can use either the web client or the Web Access page to view the remote resources in a browser.

Can I embed the web client in another web page?

This feature isn't supported at the moment.

Monitors, audio, and mouse

How do I use all of my monitors?

To use two or more screens, complete the following steps:

1. Right-click the remote desktop that you want to enable multiple screens for, and then select **Edit**.
2. Enable **Use all monitors** and **Full screen**.

Is bi-directional sound supported?

Bi-directional sound can be configured in the Windows client on a per-connection basis. The relevant settings can be accessed in the **Remote audio** section of the **Local Resources** options tab.

What can I do if the sound doesn't play?

Sign out of the session (don't just disconnect, sign all the way out), and then sign in again.

Mac client - hardware questions

Is retina resolution supported?

Yes, the remote desktop client supports retina resolution.

How do I enable secondary right-click?

In order to make use of right-click inside an open session you have three options:

- Standard PC two button USB mouse
- Apple Magic Mouse: To enable right-click, select **System Preferences** in the dock, select **Mouse**, and then enable **Secondary select**.
- Apple Magic Trackpad or MacBook Trackpad: To enable right-click, select **System Preferences** in the dock, select **Trackpad**, and then enable **Secondary select**.

Is AirPrint supported?

No, the Remote Desktop client doesn't support AirPrint for both Mac or iOS clients.

Why do incorrect characters appear in the session?

If you're using an international keyboard, you might see an issue where the characters that appear in the session don't match the characters you typed on the Mac keyboard.

This problem can occur in the following scenarios:

- You're using a keyboard that the remote session doesn't recognize. When Remote Desktop doesn't recognize the keyboard, it defaults to the language last used with the remote PC.
- You're connecting to a previously disconnected session on a remote PC and that remote PC uses a different keyboard language than the language you're currently trying to use.

You can fix this issue by manually setting the keyboard language for the remote session. See the steps in the next section.

How do language settings affect keyboards in a remote session?

There are many types of Mac keyboard layouts. Some of these layouts are Mac specific or custom layouts for which an exact match might not be available on the version of Windows you're remoting into. The remote session maps your keyboard to the best matching keyboard language available on the remote PC.

If your Mac keyboard layout is set to the PC version of the language keyboard (for example, French – PC) all your keys should be mapped correctly and your keyboard should just work.

If your Mac keyboard layout is set to the Mac version of a keyboard (for example, French) the remote session maps you to the PC version of the French language. Some of the Mac keyboard shortcuts you're used to using on OSX don't work in the remote Windows session.

If your keyboard layout is set to a variation of a language and if the remote session can't map you to that exact variation, the remote session maps you to the closest language. For example, if you have a Canadian-French, the remote session maps your layout to French. Some of the Mac keyboard shortcuts you're used to using on OSX don't work in the remote Windows session.

If your keyboard layout is set to a layout the remote session can't match at all, your remote session defaults to give you the language you last used with that PC. In this case, or in cases where you need to change the language of your remote session to match your Mac keyboard, you can manually set the keyboard language in the remote session to the language that is the closest match to the one you wish to use as follows.

Use the following instructions to change the keyboard layout inside the remote desktop session:

1. From inside the remote session, open Region and Language. Select **Start** > **Settings** > **Time and Language**. Open **Region and Language**.
2. Add the language you want to use. Then close the Region and Language window.
3. Now, in the remote session, you see the ability to switch between languages. (In the right side of the remote session, near the clock.) Select the language you want to switch to (such as **Eng**).

You might need to close and restart the application you're currently using for the keyboard changes to take effect.

Specific errors

Why do I get an "Insufficient privileges" error?

You aren't allowed to access the session you want to connect to. The most likely cause is that you're trying to connect to an admin session. Only administrators are allowed to connect to the console. Verify that the console switch is off in the advanced settings of the remote desktop. If connecting to the console isn't the source of the problem contact your system administrator for further assistance.

Why does the client say that there's no CAL?

When a remote desktop client connects to a Remote Desktop server, the server issues a Remote Desktop Services Client Access License (RDS CAL) stored by the client. Whenever the client connects again it uses its RDS CAL and the server doesn't issue another license. The server issues another license if the RDS CAL on the device is missing or corrupt. When the maximum number of licensed devices is reached, the server doesn't issue new RDS CALs. Contact your network administrator for assistance.

Why did I get an "Access Denied" error?

The "Access Denied" error is a generated by the Remote Desktop Gateway and the result of incorrect credentials during the connection attempt. Verify your username and password. If the connection worked before and the error occurred recently, you possibly changed your Windows user account password and need to update it in the remote desktop settings.

What does the "Failed to parse NTLM challenge" error mean?

Misconfiguration on the remote PC causes this error. Make sure the RDP security level setting on the remote PC is set to "Client Compatible." Talk to your system admin if you need assistance with configuring this setting.

What does "TS_RAP You aren't allowed to connect to the given host" mean?

This error happens when a Resource Authorization Policy on the gateway server stops your user name from connecting to the remote PC. This problem can happen in the following instances:

- The remote PC name is the same as the name of the gateway. Then, when you try to connect to the remote PC, the connection goes to the gateway instead, which you probably don't have permission to access. If you need to connect to the gateway, don't use the external gateway name as PC name. Instead use "localhost" or the IP address (127.0.0.1), or the internal server name.
- Your user account isn't a member of the user group for remote access.