Contents

- Host pool creation
- Session host virtual machine configuration
- Management issues
- Azure Virtual Desktop PowerShell
- Azure Virtual Desktop Agent
- Remote Desktop service connections
- Remote Desktop client issues
- Diagnosing graphics performance issues
- Connections to Azure AD-joined VMs
- Azure Monitor
- Azure Files authorization

Reference

- Security best practices
- Azure CLI
- Estimate Azure Monitor costs
- Supported RDP file settings
- Virtual machine sizing guidance
- Azure command-line interface (CLI)
- Azure example scenarios
- Email discovery

Resources

- PowerShell
- REST API reference
- Experience estimator
- Pricing calculator
- Learning path
- How-to videos
- Tech Community support group
- Azure update roadmap
- Azure Resource Manager templates
- Azure compliance offerings
- Azure Advisor

What is Azure Virtual Desktop?

12/6/2021 • 6 minutes to read • Edit Online

Azure Virtual Desktop is a desktop and app virtualization service that runs on the cloud.

Here's what you can do when you run Azure Virtual Desktop on Azure:

- Set up a multi-session Windows 10 deployment that delivers a full Windows 10 with scalability
- Virtualize Microsoft 365 Apps for enterprise and optimize it to run in multi-user virtual scenarios
- Provide Windows 7 virtual desktops with free Extended Security Updates
- Bring your existing Remote Desktop Services (RDS) and Windows Server desktops and apps to any computer
- Virtualize both desktops and apps
- Manage Windows 10, Windows Server, and Windows 7 desktops and apps with a unified management experience

Introductory video

Learn about Azure Virtual Desktop, why it's unique, and what's new in this video:

https://www.youtube.com/embed/NQFtI3JLtaU

For more videos about Azure Virtual Desktop, see our playlist.

Key capabilities

With Azure Virtual Desktop, you can set up a scalable and flexible environment:

- Create a full desktop virtualization environment in your Azure subscription without running any gateway servers.
- Publish as many host pools as you need to accommodate your diverse workloads.
- Bring your own image for production workloads or test from the Azure Gallery.
- Reduce costs with pooled, multi-session resources. With the new Windows 10 Enterprise multi-session capability, exclusive to Azure Virtual Desktop and Remote Desktop Session Host (RDSH) role on Windows Server, you can greatly reduce the number of virtual machines and operating system (OS) overhead while still providing the same resources to your users.
- Provide individual ownership through personal (persistent) desktops.

You can deploy and manage virtual desktops:

- Use the Azure portal, Azure Virtual Desktop PowerShell and REST interfaces to configure the host pools, create app groups, assign users, and publish resources.
- Publish full desktop or individual remote apps from a single host pool, create individual app groups for different sets of users, or even assign users to multiple app groups to reduce the number of images.
- As you manage your environment, use built-in delegated access to assign roles and collect diagnostics to understand various configuration or user errors.
- Use the new Diagnostics service to troubleshoot errors.
- Only manage the image and virtual machines, not the infrastructure. You don't need to personally manage the Remote Desktop roles like you do with Remote Desktop Services, just the virtual machines in your Azure

subscription.

You can also assign and connect users to your virtual desktops:

- Once assigned, users can launch any Azure Virtual Desktop client to connect to their published Windows desktops and applications. Connect from any device through either a native application on your device or the Azure Virtual Desktop HTML5 web client.
- Securely establish users through reverse connections to the service, so you never have to leave any inbound ports open.

Requirements

There are a few things you need to set up Azure Virtual Desktop and successfully connect your users to their Windows desktops and applications.

We support the following operating systems, so make sure you have the appropriate licenses for your users based on the desktop and apps you plan to deploy:

OS	REQUIRED LICENSE
Windows 10 Enterprise multi-session or Windows 10 Enterprise	Microsoft 365 E3, E5, A3, A5, F3, Business Premium Windows E3, E5, A3, A5
Windows 7 Enterprise	Microsoft 365 E3, E5, A3, A5, F3, Business Premium Windows E3, E5, A3, A5
Windows Server 2012 R2, 2016, 2019, 2022	RDS Client Access License (CAL) with Software Assurance

Your infrastructure needs the following things to support Azure Virtual Desktop:

- An Azure Active Directory.
- A Windows Server Active Directory in sync with Azure Active Directory. You can configure this using Azure AD Connect (for hybrid organizations) or Azure AD Domain Services (for hybrid or cloud organizations).
  - A Windows Server AD in sync with Azure Active Directory. User is sourced from Windows Server AD and the Azure Virtual Desktop VM is joined to Windows Server AD domain.
  - A Windows Server AD in sync with Azure Active Directory. User is sourced from Windows Server AD and the Azure Virtual Desktop VM is joined to Azure AD Domain Services domain.
  - An Azure AD Domain Services domain. User is sourced from Azure Active Directory, and the Azure Virtual Desktop VM is joined to Azure AD Domain Services domain.
- An Azure subscription, parented to the same Azure AD tenant, that contains a virtual network that either contains or is connected to the Windows Server Active Directory or Azure AD DS instance.

User requirements to connect to Azure Virtual Desktop:

- The user must be sourced from the same Active Directory that's connected to Azure AD. Azure Virtual Desktop does not support B2B or MSA accounts.
- The UPN you use to subscribe to Azure Virtual Desktop must exist in the Active Directory domain the VM is joined to.

The Azure virtual machines you create for Azure Virtual Desktop must be:

- Standard domain-joined or Hybrid AD-joined. Azure AD-joined virtual machines are available in preview.
- Running one of the following supported OS images.

> **NOTE**
> If you need an Azure subscription, you can sign up for a one-month free trial. If you're using the free trial version of Azure, you should use Azure AD Domain Services to keep your Windows Server Active Directory in sync with Azure Active Directory.

For a list of URLs you should unblock for your Azure Virtual Desktop deployment to work as intended, see our Required URL list.

Azure Virtual Desktop includes the Windows desktops and apps you deliver to users and the management solution, which is hosted as a service on Azure by Microsoft. Desktops and apps can be deployed on virtual machines (VMs) in any Azure region, and the management solution and data for these VMs will reside in the United States. This may result in data transfer to the United States.

For optimal performance, make sure your network meets the following requirements:

- Round-trip (RTT) latency from the client's network to the Azure region where host pools have been deployed should be less than 150 ms. Use the Experience Estimator to view your connection health and recommended Azure region.
- Network traffic may flow outside country/region borders when VMs that host desktops and apps connect to the management service.
- To optimize for network performance, we recommend that the session host's VMs are located in the Azure region that is closest to the user.

You can see a typical architectural setup of Azure Virtual Desktop for the enterprise in our architecture documentation.

Supported Remote Desktop clients

The following Remote Desktop clients support Azure Virtual Desktop:

- Windows Desktop
- Web
- macOS
- iOS
- Android
- Microsoft Store Client

> **IMPORTANT**
> Azure Virtual Desktop doesn't support the RemoteApp and Desktop Connections (RADC) client or the Remote Desktop Connection (MSTSC) client.

To learn more about URLs you must unblock to use the clients, see the Safe URL list.

Supported virtual machine OS images

Azure Virtual Desktop follows the Microsoft Lifecycle Policy and supports the following x64 operating system images:

- Windows 11 Enterprise multi-session (Preview)
- Windows 11 Enterprise (Preview)
- Windows 10 Enterprise multi-session

- Windows 10 Enterprise
- Windows 7 Enterprise
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

Azure Virtual Desktop doesn't support x86 (32-bit), Windows 10 Enterprise N, Windows 10 LTSB, Windows 10 LTSC, Windows 10 Pro, or Windows 10 Enterprise KN operating system images. Windows 7 also doesn't support any VHD or VHDX-based profile solutions hosted on managed Azure Storage due to a sector size limitation.

Available automation and deployment options depend on which OS and version you choose, as shown in the following table:

OPERATING SYSTEM	AZURE IMAGE GALLERY	MANUAL VM DEPLOYMENT	AZURE RESOURCE MANAGER TEMPLATE INTEGRATION	PROVISION HOST POOLS ON AZURE MARKETPLACE
Windows 11 Enterprise multi-session (Preview)	Yes	Yes	Yes	Yes
Windows 11 Enterprise (Preview)	Yes	Yes	Yes	Yes
Windows 10 Enterprise multi-session, version 1909 and later	Yes	Yes	Yes	Yes
Windows 10 Enterprise, version 1909 and later	Yes	Yes	Yes	Yes
Windows 7 Enterprise	Yes	Yes	No	No
Windows Server 2022	Yes	Yes	No	No
Windows Server 2019	Yes	Yes	No	No
Windows Server 2016	Yes	Yes	Yes	Yes
Windows Server 2012 R2	Yes	Yes	No	No

Next steps

If you're using Azure Virtual Desktop (classic), you can get started with our tutorial at Create a tenant in Azure Virtual Desktop.

If you're using the Azure Virtual Desktop with Azure Resource Manager integration, you'll need to create a host pool instead. Head to the following tutorial to get started.

Create a host pool with the Azure portal

What's new in Azure Virtual Desktop?

12/6/2021 • 33 minutes to read • Edit Online

Azure Virtual Desktop updates regularly. This article is where you'll find out about:

- The latest updates
- New features
- Improvements to existing features
- Bug fixes

This article is updated monthly. Make sure to check back here often to keep up with new updates.

Client updates

Check out these articles to learn about updates for our clients for Azure Virtual Desktop and Remote Desktop Services:

- Windows
- macOS
- iOS
- Android
- Web

Azure Virtual Desktop Agent updates

The Azure Virtual Desktop agent updates at least once per month.

Here's what's changed in the Azure Virtual Desktop Agent:

- Version 1.0.3719.1700: This update was released November 2021 and has the following changes:
  - Updated agent error messages.
  - Fixes an issue with the agent restarting every time the side-by-side stack was updated.
  - General agent improvements.
- Version 1.0.3583.2600: This update was released October 2021 and it fixes an issue where upgrading from Windows 10 to Windows 11 disabled the side-by-side stack.
- Version 1.0.3373.2605: This update was released September 2021 and it fixes an issue with package deregistration getting stuck when using MSIX App Attach.
- Version 1.0.3373.2600: This update was released September 2021 and has the following changes:
  - General agent improvements.
  - Fixes issues with restarting the agent on Windows 7 VMs.
  - Fixes an issue with fields in the WVDAgentHealthStatus table not showing up correctly.
- Version 1.0.3130.2900: This update was released July 2021 and has the following changes:
  - General improvements and bug fixes.
  - Fixes an issue with getting the host pool path for Intune registration.
  - Added logging to better diagnose agent issues.
  - Fixes an issue with orchestration timeouts.
- Version 1.0.3050.2500: This update was released July 2021 and has the following changes:
  - Updated internal monitors for agent health.

    - Updated retry logic for stack health.
- Version 1.0.2990.1500: This update was released April 2021 and has the following changes:
    - Updated agent error messages.
    - Added an exception that prevents you from installing non-Windows 7 agents on Windows 7 VMs.
    - Has updated heartbeat service logic.
- Version 1.0.2944.1400: This update was released April 2021 and has the following changes:
    - Placed links to the Azure Virtual Desktop Agent troubleshooting guide in the event viewer logs for agent errors.
    - Added an additional exception for better error handling.
    - Added the WVDAgentUrlTool.exe that allows customers to check which required URLs they can access.
- Version 1.0.2866.1500: This update was released March 2021 and it fixes an issue with the stack health check.
- Version 1.0.2800.2802: This update was released March 2021 and it has general improvements and bug fixes.
- Version 1.0.2800.2800: This update was released March 2021 and it fixes a reverse connection issue.
- Version 1.0.2800.2700: This update was released February 2021 and it fixes an access denied orchestration issue.

FSLogix updates

Curious about the latest updates for FSLogix? Check out What's new at FSLogix.

November 2021

Here's what changed in November 2021:

Azure Virtual Desktop for Azure Stack HCI

Azure Virtual Desktop for Azure Stack HCI is now in public preview. This feature is for customers who need desktop virtualization for apps that have to stay on-premises for performance and data security reasons. To learn more, see our blog post and the Azure Virtual Desktop for Azure Stack HCI documentation.

Autoscale public preview

We're pleased to introduce the new autoscale feature, which lets you stop or start session hosts automatically based on a schedule you set. Autoscale lets you optimize infrastructure costs by configuring your shared or pooled desktops to only charge for the resources you actually use. You can learn more about the autoscale feature by reading our documentation and watching our Azure Academy video.

Azure Virtual Desktop starter kit for Power Automate

Your organization can now use the Azure Virtual Desktop starter kit to manage its robotic process automation (RPA) workloads. Learn more by reading our documentation.

Tagging with Azure Virtual Desktop

We recently released new documentation about how to configure tags for Azure Virtual Desktop to track and manage costs. For more information, see Tag Azure Virtual Desktop resources.

October 2021

Here's what changed in October 2021:

Azure Virtual Desktop support for Windows 11

Azure Virtual Desktop support for Windows 11 is now generally available for single and multi-session deployments. You can now use Windows 11 images when creating host pools in the Azure portal. For more

information, see our blog post.

RDP Shortpath now generally available

Remote Desktop Protocol (RDP) Shortpath for managed networks is now generally available. RDP Shortpath establishes a direct connection between the Remote Desktop client and the session host. This direct connection reduces dependency on gateways, improves the connection's reliability, and increases the bandwidth available for each user session. For more information, see our blog post.

Screen capture protection updates

Screen capture protection is now supported on the macOS client and the Azure Government and Azure China clouds. For more information, see our blog post.

Azure Active Directory domain join

Azure Active Directory domain join for Azure Virtual Desktop VMs is now available in the Azure Government and Azure China clouds. Microsoft Endpoint Manager (Intune) is currently only supported in the Azure Public cloud. Learn more at Deploy Azure AD-joined virtual machines in Azure Virtual Desktop.

Breaking change in Azure Virtual Desktop Azure Resource Manager template

A breaking change has been introduced into the Azure Resource Manager template for Azure Virtual Desktop. If you're using any code that depends on the change, then you'll need to follow the directions in our blog post to address the issue.

Autoscale (preview) public preview

Autoscale for Azure Virtual Desktop is now in public preview. This feature natively turns your virtual machines (VMs) in pooled host pools on or off based on availability needs. Scheduling when your VMs turn on and off optimizes deployment costs, and this feature also offers flexible scheduling options based on your needs. Once you've configured the required custom Role-Based Access Control (RBAC) role, you can start configuring your scaling plan. For more information, see Autoscale (preview) for Azure Virtual Desktop host pools.

September 2021

Here's what changed in September 2021.

Azure portal updates

You can now use Azure Resource Manager templates for any update you want to apply to your session hosts after deployment. You can access this feature by selecting the **Virtual machines** tab while creating a host pool.

You can also now set host pool, app group, and workspace diagnostic settings while creating host pools instead of afterwards. Configuring these settings during the host pool creation process also automatically sets up reporting data for Azure Virtual Desktop Insights.

Azure Active Directory domain join

Azure Active Directory domain join is now generally available. This service lets you join your session hosts to Azure Active Directory. Domain join also lets you autoenroll into Intune as part of Microsoft Endpoint Manager. You can access this feature in the Azure public cloud, but not the Government cloud or Azure China. For more information, see our blog post.

Azure China

Azure Virtual Desktop is now generally available in the Azure China cloud. For more information, see our blog post.

Automatic migration module tool

With the automatic migration tool, you can move your organization from Azure Virtual Desktop (classic) to Azure Virtual Desktop with just a few PowerShell commands. This feature is currently in public preview, and you can find out more at Automatic migration.

August 2021

Here's what changed in August 2021:

Windows 11 (Preview) for Azure Virtual Desktop

Windows 11 (Preview) images are now available in the Azure Marketplace for customers to test and validate with Azure Virtual Desktop. For more information, see our announcement.

Multimedia redirection is now in public preview

Multimedia redirection gives you smooth video playback while watching videos in your Azure Virtual Desktop web browser and works with Microsoft Edge and Google Chrome. Learn more at our blog post.

Windows Defender Application Control and Azure Disk Encryption support

Azure Virtual Desktop now supports Windows Defender Application Control to control which drivers and applications are allowed to run on Windows virtual machines (VMs), and Azure Disk Encryption, which uses Windows BitLocker to provide volume encryption for the OS and data disks of your VMs. For more information, see our announcement.

Signing into Azure AD using smart cards are now supported in Azure Virtual Desktop

While this isn't a new feature for Azure AD, Azure Virtual Desktop now supports configuring Active Directory Federation Services to sign in with smart cards. For more information, see our announcement.

Screen capture protection is now generally available

Prevent sensitive information from being screen captured by software running on the client endpoints with screen capture protection in Azure Virtual Desktop. Learn more at our blog post.

July 2021

Here's what changed in July 2021:

Azure Virtual Desktop images now include optimized Teams

All available images in the Azure Virtual Desktop image gallery that include Microsoft 365 Apps for Enterprise now have the media-optimized version of Teams for Azure Virtual Desktop pre-installed. For more information, see our announcement.

Azure Active Directory Domain Join for Session hosts is in public preview

You can now join your Azure Virtual Desktop virtual machines (VMs) directly to Azure Active Directory (Azure AD). This feature lets you connect to your VMs from any device with basic credentials. You can also automatically enroll your VMs with Microsoft Endpoint Manager. For certain scenarios, this will help eliminate the need for a domain controller, reduce costs, and streamline your deployment. Learn more at Deploy Azure AD joined virtual machines in Azure Virtual Desktop.

FSLogix version 2105 is now available

FSLogix version 2105 is now generally available. This version includes improved sign-in times and bug fixes that weren't available in the public preview version (version 2105). For more detailed information, you can see the FSLogix release notes and our blog post.

Azure Virtual Desktop in China has entered public preview

With Azure Virtual Desktop available in China, we now have more rounded global coverage that helps organizations support customers in this region with improved performance and latency. Learn more at our announcement page.

The getting started feature for Azure Virtual Desktop

This feature offers a streamlined onboarding experience in the Azure portal to set up your Azure Virtual Desktop environment. You can use this feature to create deployments that meet system requirements for automated

Azure Active Directory Domain Services the simple and easy way. For more information, check out our blog post.

Start VM on connect is now generally available

The start VM on connect feature is now generally available. This feature helps you optimize costs by letting you turn off deallocated or stopped VMs, letting your deployment be flexible with user demands. For more information, see Start Virtual Machine on Connect.

Remote app streaming documentation

We recently announced a new pricing option for remote app streaming for using Azure Virtual Desktop to deliver apps as a service to your customers and business partners. For example, software vendors can use remote app streaming to deliver apps as a software as a service (SaaS) solution that's accessible to their customers. To learn more about remote app streaming, check out our documentation.

From July 14th, 2021 to December 31st, 2021, we're giving customers who use remote app streaming a promotional offer that lets their business partners and customers access Azure Virtual Desktop for no charge. This offer only applies to external user access rights. Regular billing will resume on January 1st, 2022. In the meantime, you can continue to use your existing Windows license entitlements found in licenses like Microsoft 365 E3 or Windows E3. To learn more about this offer, see the Azure Virtual Desktop pricing page.

New Azure Virtual Desktop handbooks

We recently released four new handbooks to help you design and deploy Azure Virtual Desktop in different scenarios:

- Application Management will show you how to modernize application delivery and simplify IT management.
- In Disaster Recovery, learn how to strengthen business resilience by developing a disaster recovery strategy.
- Get more value from Citrix investments with the Citrix Cloud with Azure Virtual Desktop migration guide.
- Get more value from existing VMware investments with the VMware Horizon with Azure Virtual Desktop migration guide.

June 2021

Here's what changed in June 2021:

Windows Virtual Desktop is now Azure Virtual Desktop

To better align with our vision of a flexible cloud desktop and remote application platform, we've renamed Windows Virtual Desktop to Azure Virtual Desktop. Learn more at the announcement post in our blog.

EU, UK, and Canada geographies are now generally available

Metadata service for the European Union, UK, and Canada is now in general availability. These new locations are very important to data sovereignty outside the US. For more information, see our blog post.

The Getting Started tool is now in public preview

We created the Azure Virtual Desktop Getting Started tool to make the deployment process easier for first-time users. By simplifying and automating the deployment process, we hope this tool will help make adopting Azure Virtual Desktop faster and more accessible to a wider variety of users. Learn more at our blog post.

Azure Virtual Desktop pricing calculator updates

We've made some significant updates to improve the Azure Virtual Desktop pricing experience on the Azure pricing calculator, including the following:

- We've updated the service name to Azure Virtual Desktop
- We also updated the layout with the following new items:
  - A Storage section with both managed disk and file storage bandwidth

- A custom section that shows cost-per-user

You can access the pricing calculator at this page.

Single Sign-on (SSO) using Active Directory Federation Services (AD FS)

The AD FS single-sign on feature is now generally available. This feature lets customers use AD FS to give a single sign-on experience for users on the Windows and web clients. For more information, see Configure AD FS single sign-on for Azure Virtual Desktop.

May 2021

Here's what's new for May 2021:

Smart card authentication

We've now officially released the Key Distribution Center (KDC) Proxy Remote Desktop Protocol (RDP) properties. These properties enable Kerberos authentication for the RDP portion of an Azure Virtual Desktop session, which includes permitting Network Level Authentication without a password. Learn more at our blog post.

The web client now supports file transfer

Starting with the public preview version of the web client, version 1.0.24.7 (preview), users can now transfer files between their remote session and local computer. To upload files to the remote session, select the upload icon in the menu at the top of the web client page. To download files, search for **Remote Desktop Virtual Drive** in the Start menu on your remote session. After you've opened your virtual drive, just drag and drop your files into the Downloads folder and the browser will begin downloading the files to your local computer.

Start VM on connect support updates

Start VM on connect (preview) now supports pooled host pools and the Azure Government Cloud. To learn more, read our blog post.

Latency improvements for the United Arab Emirates region

We've expanded our Azure control plane presence to the United Arab Emirates (UAE), so customers in that region can now experience improved latency. Learn more at our Azure Virtual Desktop roadmap.

Ending Internet Explorer 11 support

On September 30th, 2021, the Azure Virtual Desktop web client will no longer support Internet Explorer 11. We recommend you start using the Microsoft Edge browser for your web client and remote sessions instead. For more information, see the announcement in this blog post.

Microsoft Endpoint Manager public preview

We've started the public preview for Microsoft Endpoint Manager support in Windows 10 Enterprise multi-session. This new feature will let you manage your Windows 10 VMs with the same tools as your local devices. Learn more at our Microsoft Endpoint Manger documentation.

FSLogix agent public preview

We have released a public preview of the latest version of the FSLogix agent. Check out our blog post for more information and to submit the form you'll need to access the preview.

May 2021 updates for Teams for Azure Virtual Desktop

For this update, we resolved an issue that caused the screen to remain black while sharing video. We also fixed a mismatch in video resolutions between the session client and the Teams server. Teams on Azure Virtual Desktop should now change resolution and bit rates based on input from the Teams server.

Azure portal deployment updates

We've made the following updates to the deployment process in the Azure portal:

- Added new images (including GEN2) to the drop-down list box of "image" when creating a new Azure Virtual Desktop session host VM.
- You can now configure boot diagnostics for virtual machines when creating a host pool.
- Added a tool tip to the RDP proxy in the advanced host pool RDP properties tab.
- Added an information bubble for the icon path when adding an application from an MSIX package.
- You can no longer do managed boot diagnostics with an unmanaged disk.
- Updated the template for creating a host pool in Azure Resource Manager so that the Azure portal can now support creating host pools with third-party marketplace images.

Single sign-on using Active Directory Federation Services public preview

We've started a public preview for Active Directory Federation Services (AD FS) support for single sign-on (SSO) per host pool. Learn more at Configure AD FS single sign-on for Azure Virtual Desktop.

Enterprise-scale support

We've released an updated section of the Cloud Adoption framework for Enterprise-scale support for Azure Virtual Desktop. For more information, see Enterprise-scale support for the Azure Virtual Desktop construction set.

Customer adoption kit

We've recently released the Azure Virtual Desktop Customer adoption kit to help customers and partners set up Azure Virtual Desktop for their customers. You can download the kit here.

April 2021

Here's what's new for April:

Use the Start VM on Connect feature (preview) in the Azure portal

You can now configure Start VM on Connect (preview) in the Azure portal. With this update, users can access their VMs from the Android and macOS clients. To learn more, see Start VM on Connect.

Required URL Check tool

The Azure Virtual Desktop agent, version 1.0.2944.400 includes a tool that validates URLs and displays whether the virtual machine can access the URLs it needs to function. If any required URLs are accessible, the tool will list them so you can unblock them, if needed. Learn more at our Safe URL list.

Updates to the Azure portal UI for Azure Virtual Desktop

Here's what changed in the latest update of the Azure portal UI for Azure Virtual Desktop:

- Fixed an issue that caused an error to appear when retrieving the session host while drain mode is enabled.
- Upgraded the Portal SDK to version 7.161.0.
- Fixed an issue that caused the resource ID missing error message to appear in the User Sessions tab.
- The Azure portal now shows detailed sub-status messages for session hosts.

April 2021 updates for Teams on Azure Virtual Desktop

Here's what's new for Teams on Azure Virtual Desktop:

- Added hardware acceleration for video processing of outgoing video streams for Windows 10-based clients.
- When joining a meeting with both a front facing camera and a rear facing or external camera, the front facing camera will be selected by default.
- Resolved an issue that made Teams crash on x86-based machines.
- Resolved an issue that caused striations during screen sharing.
- Resolved an issue that prevented meeting members from seeing incoming video or screen sharing.

MSIX app attach is now generally available

MSIX app attach for Azure Virtual Desktop has now come out of public preview and is available to all users. Learn more about MSIX app attach at our TechCommunity announcement.

The macOS client now supports Apple Silicon and Big Sur

The macOS Azure Virtual Desktop client now supports Apple Silicon and Big Sur. The full list of updates is available in What's new in the macOS client.

March 2021

Here's what changed in March 2021.

Updates to the Azure portal UI for Azure Virtual Desktop

We've made the following updates to Azure Virtual Desktop for the Azure portal:

- We've enabled new availability options (availability set and zones) for the workflows to create host pools and add VMs.
- We've fixed an issue where a host with the "Needs assistance" status appeared as unavailable. Now the host will have a warning icon next to it.
- We've enabled sorting for active sessions.
- You can now send messages to or sign out specific users on the host details tab.
- We've changed the maximum session limit field.
- We've added an OU validation path to the workflow to create a host pool.
- You can now use the latest version of the Windows 10 image when you create a personal host pool.

Generation 2 images and Trusted Launch

The Azure Marketplace now has Generation 2 images for Windows 10 Enterprise and Windows 10 Enterprise multi-session. These images will let you use Trusted Launch VMs. Learn more about Generation 2 VMs at Should I create a generation 1 or 2 virtual machine. To learn how to provision Azure Virtual Desktop Trusted Launch VMs, see our TechCommunity post.

FSLogix is now preinstalled on Windows 10 Enterprise multi-session images

Based on customer feedback, we've set up a new version of the Windows 10 Enterprise multi-session image that has an unconfigured version of FSLogix already installed. We hope this makes your Azure Virtual Desktop deployment easier.

Azure Monitor for Azure Virtual Desktop is now in General Availability

Azure Monitor for Azure Virtual Desktop is now generally available to the public. This feature is an automated service that monitors your deployments and lets you view events, health, and troubleshooting suggestions in a single place. For more information, see our documentation or check out our TechCommunity post.

March 2021 updates for Teams on Azure Virtual Desktop

We've made the following updates for Teams on Azure Virtual Desktop:

- We've improved video quality performance on calls and 2x2 mode.
- We've reduced CPU utilization by 5-10% (depending on CPU generation) by using hardware offload of video processing (XVP).
- Older machines can now use XVP and hardware decoding to display more incoming video streams smoothly in 2x2 mode.
- We've updated the WebRTC stack from M74 to M88 for better AV sync performance and fewer transient issues.
- We've replaced our software H264 encoder with OpenH264 (OSS used in Teams on the web), which increased the video quality of the outgoing camera.
- We enabled 2x2 mode for Teams Server for the general public on March 30. 2x2 mode shows up to four

incoming video streams at the same time.

Start VM on Connect public preview

The new host pool setting, Start VM on Connect, is now available in public preview. This setting lets you turn on your VMs whenever you need them. If you want to save costs, you'll need to deallocate your VMs by configuring your Azure Compute settings. For more information, check out our blog post and our documentation.

Azure Virtual Desktop Specialty certification

We've released a beta version of the AZ-140 exam that will let you prove your expertise in Azure Virtual Desktop in Azure. To learn more, check out our TechCommunity post.

February 2021

Here's what changed in February 2021.

Portal experience

We've improved the Azure portal experience in the following ways:

- Bulk drain mode on hosts in the session host grid tab.
- MSIX app attach is now available for public preview.
- Fixed host pool overview info for dark mode.

EU metadata storage now in public preview

We're now hosting a public preview of the Europe (EU) geography as a storage option for service metadata in Azure Virtual Desktop. Customers can choose between West or North Europe when they create their service objects. The service objects and metadata for the host pools will be stored in the Azure geography associated with each region. To learn more, read our blog post announcing the public preview.

Teams on Azure Virtual Desktop plugin updates

We've improved video call quality on the Azure Virtual Desktop plugin by addressing the most commonly reported issues, such as when the screen would suddenly go dark or the video and sound desynchronized. These improvements should increase the performance of single-video view with active speaker switching. We also fixed an issue where hardware devices with special characters weren't available in Teams.

January 2021

Here's what changed in January 2021:

New Azure Virtual Desktop offer

New customers save 30 percent on Azure Virtual Desktop computing costs for D-series and Bs-series virtual machines for up to 90 days when using the native Microsoft solution. You can redeem this offer in the Azure portal before March 31, 2021. Learn more at our Azure Virtual Desktop offer page.

networkSecurityGroupRules value change

In the Azure Resource Manager nested template, we changed the default value for networkSecurityGroupRules from an object to an array. This will prevent any errors if you use managedDisks-customimagevm.json without specifying a value for networkSecurityGroupRules. This wasn't a breaking change and is backward compatible.

FSLogix hotfix update

We've released FSLogix, version 2009 HF_01 (2.9.7654.46150) to solve issues in the previous release (2.9.7621.30127). We recommend you stop using the previous version and update FSLogix as soon as possible.

For more information, see the release notes in What's new in FSLogix.

Azure portal experience improvements

We've made the following improvements to the Azure portal experience:

- You can now add local VM admin credentials directly instead of having to add a local account created with the Active Directory domain join account credentials.
- Users can now list both individual and group assignments in separate tabs for individual users and groups.
- The version number of the Azure Virtual Desktop Agent is now visible in the Virtual Machine overview for host pools.
- Added bulk delete for host pools and application groups.
- You can now enable or disable drain mode for multiple session hosts in a host pool.
- Removed the public IP field from the VM details page.

Azure Virtual Desktop Agent troubleshooting

We recently set up the Azure Virtual Desktop Agent troubleshooting guide to help customers who have encountered common issues.

Microsoft Defender for Endpoint integration

Microsoft Defender for Endpoint integration is now generally available. This feature gives your Azure Virtual Desktop VMs the same investigation experience as a local Windows 10 machine. If you're using Windows 10 Enterprise multi-session, Microsoft Defender for Endpoint will support up to 50 concurrent user connections, giving you the cost savings of Windows 10 Enterprise multi-session and the confidence of Microsoft Defender for Endpoint. For more information, check out our blog post.

Azure Security baseline for Azure Virtual Desktop

We've recently published an article about the Azure security baseline for Azure Virtual Desktop that we'd like to call your attention to. These guidelines include information about how to apply the Azure Security Benchmark, version 2.0 to Azure Virtual Desktop. The Azure Security Benchmark describes the settings and practices we recommend you use to secure your cloud solutions on Azure.

December 2020

Here's what changed in December 2020:

Azure Monitor for Azure Virtual Desktop

The public preview for Azure Monitor for Azure Virtual Desktop is now available. This new feature includes a robust dashboard built on top of Azure Monitor Workbooks to help IT professionals understand their Azure Virtual Desktop environments. Check out the announcement on our blog for more details.

Azure Resource Manager template change

In the latest update, we've removed all public IP address parameter from the Azure Resource Manager template for creating and provisioning host pools. We highly recommend you avoid using public IPs for Azure Virtual Desktop to keep your deployment secure. If your deployment relied on public IPs, you'll need to reconfigure it to use private IPs instead, otherwise your deployment won't work properly.

MSIX app attach public preview

MSIX app attach is another service that began its public preview this month. MSIX app attach is a service that dynamically presents MSIX applications to your Azure Virtual Desktop Session host VMs. Check out the announcement on our blog for more details.

Screen capture protection

This month also marked the beginning of the public preview for screen capture protection. You can use this feature to prevent sensitive information from being captured on the client endpoints. Give screen capture protection a try by going to this page.

Built-in roles

We've added new built-in roles for Azure Virtual Desktop for admin permissions. For more information, see Built-in roles for Azure Virtual Desktop.

Application group limit increase

We've increased the default application group limit per Azure Active Directory tenant to 200 groups.

Client updates for December 2020

We've released new versions of the following clients:

- Android
- macOS
- Windows

For more information about client updates, see Client updates.

November 2020

Azure portal experience

We've fixed two bugs in the Azure portal user experience:

- The Desktop application friendly name is no longer overwritten on the "Add VM" workflow.
- The session host tab will now load if session hosts are part of scale sets.

FSLogix client, version 2009

We've released a new version of the FSLogix client with many fixes and improvements. Learn more at our blog post.

RDP Shortpath public preview

RDP Shortpath introduces direct connectivity to your Azure Virtual Desktop session host using point-to-site and site-to-site VPNs and ExpressRoute. It also introduces the URCP transport protocol. RDP Shortpath is designed to reduce latency and network hops in order to improve user experience. Learn more at Azure Virtual Desktop RDP Shortpath.

Az.DesktopVirtualization, version 2.0.1

We've released version 2.0.1 of the Azure Virtual Desktop cmdlets. This update includes cmdlets that will let you manage MSIX App Attach. You can download the new version at the PowerShell gallery.

Azure Advisor updates

Azure Advisor now has a new recommendation for proximity guidance in Azure Virtual Desktop, and a new recommendation for optimizing performance in depth-first load balanced host pools. Learn more at the Azure website.

October 2020

Here's what changed in October 2020:

Improved performance

- We've optimized performance by reducing connection latency in the following Azure geographies:
  - Switzerland
  - Canada

You can now use the Experience Estimator to estimate the user experience quality in these areas.

Azure Government Cloud availability

The Azure Government Cloud is now generally available. Learn more at our blog post.

Azure Virtual Desktop Azure portal updates

We've made some updates to the Azure Virtual Desktop Azure portal:

- Fixed a resourceID error that prevented users from opening the "Sessions" tab.
- Streamlined the UI on the "Session hosts" tab.
- Fixed the "Defaults," "Usability," and "Restore defaults" settings under RDP properties.
- Made "Remove" and "Delete" functions consistent across all tabs.
- The portal now validates app names in the "Add an app" workflow.
- Fixed an issue where the session host export data wasn't aligned in the columns.
- Fixed an issue where the portal couldn't retrieve user sessions.
- Fixed an issue in session host retrieval that happened when the virtual machine was created in a different resource group.
- Updated the "Session host" tab to list both active and disconnected sessions.
- The "Applications" tab now has pages.
- Fixed an issue where the "requires command line" text didn't display correctly in the "Application list" tab.
- Fixed an issue when the portal couldn't deploy host pools or virtual machines while using the German-language version of the Shared Image Gallery.

Client updates for October 2020

We've released new versions of the clients. See these articles to learn more:

- Windows
- iOS

For more information about the other clients, see Client updates.

September 2020

Here's what changed in September 2020:

- We've optimized performance by reducing connection latency in the following Azure geographies:
  - Germany
  - South Africa (for validation environments only)

You can now use the Experience Estimator to estimate the user experience quality in these areas.

- We released version 1.2.1364 of the Windows Desktop client for Azure Virtual Desktop. In this update, we made the following changes:
  - Fixed an issue where single sign-on (SSO) didn't work on Windows 7.
  - Fixed an issue that caused the client to disconnect when a user who enabled media optimization for Teams tried to call or join a Teams meeting while another app had an audio stream open in exclusive mode.
  - Fixed an issue where Teams didn't enumerate audio or video devices when media optimization for Teams was enabled.
  - Added a "Need help with settings?" link to the desktop settings page.
  - Fixed an issue with the "Subscribe" button that happened when using high-contrast dark themes.
- Thanks to the tremendous help from our users, we've fixed two critical issues for the Microsoft Store Remote Desktop client. We'll continue to review feedback and fix issues as we broaden our phased release of the client to more users worldwide.
- We've added a new feature that lets you change VM location, image, resource group, prefix name, network config as part of the workflow for adding a VM to your deployment in the Azure portal.

- IT Pros can now manage hybrid Azure Active Directory-joined Windows 10 Enterprise VMs using Microsoft Endpoint Manager. To learn more, see our blog post.

August 2020

Here's what changed in August 2020:

- We've improved performance to reduce connection latency in the following Azure regions:
  - United Kingdom
  - France
  - Norway
  - South Korea

  You can use the Experience Estimator to get a general idea of how these changes will affect your users.
- The Microsoft Store Remote Desktop Client (v10.2.1522+) is now generally available! This version of the Microsoft Store Remote Desktop Client is compatible with Azure Virtual Desktop. We've also introduced refreshed UI flows for improved user experiences. This update includes fluent design, light and dark modes, and many other exciting changes. We've also rewritten the client to use the same underlying remote desktop protocol (RDP) engine as the iOS, macOS, and Android clients. This lets us deliver new features at a faster rate across all platforms. Download the client and give it a try!
- We fixed an issue in the Teams Desktop client (version 1.3.00.21759) where the client only showed the UTC time zone in the chat, channels, and calendar. The updated client now shows the remote session's time zone instead.
- Azure Advisor is now a part of Azure Virtual Desktop. When you access Azure Virtual Desktop through the Azure portal, you can see recommendations for optimizing your Azure Virtual Desktop environment. Learn more at Azure Advisor.
- Azure CLI now supports Azure Virtual Desktop (`az desktopvirtualization`) to help you automate your Azure Virtual Desktop deployments. Check out desktopvirtualization for a list of extension commands.
- We've updated our deployment templates to make them fully compatible with the Azure Virtual Desktop Azure Resource Manager interfaces. You can find the templates on GitHub.
- The Azure Virtual Desktop US Gov portal is now in public preview. To learn more, see our announcement.

July 2020

July was when Azure Virtual Desktop with Azure Resource Management integration became generally available.

Here's what changed with this new release:

- The "Fall 2019 release" is now known as "Azure Virtual Desktop (Classic)," while the "Spring 2020 release" is now just "Azure Virtual Desktop." For more information, check out this blog post.

To learn more about new features, check out this blog post.

Autoscaling tool update

The latest version of the autoscaling tool that was in preview is now generally available. This tool uses an Azure automation account and the Azure Logic App to automatically shut down and restart session host VMs within a host pool, reducing infrastructure costs. Learn more at Scale session hosts using Azure Automation.

Azure portal

You can now do the following things with the Azure portal in Azure Virtual Desktop:

- Directly assign users to personal desktop session hosts
- Change the validation environment setting for host pools

Diagnostics

We've released some new prebuilt queries for the Log Analytics workspace. To access the queries, go to **Logs** and under **Category**, select **Azure Virtual Desktop**. Learn more at Use Log Analytics for the diagnostics feature.

Update for Remote Desktop client for Android

The Remote Desktop client for Android now supports Azure Virtual Desktop connections. Starting with version 10.0.7, the Android client features a new UI for improved user experience. The client also integrates with Microsoft Authenticator on Android devices to enable conditional access when subscribing to Azure Virtual Desktop workspaces.

The previous version of Remote Desktop client is now called "Remote Desktop 8." Any existing connections you have in the earlier version of the client will be transferred seamlessly to the new client. The new client has been rewritten to the same underlying RDP core engine as the iOS and macOS clients, faster release of new features across all platforms.

Teams update

We've made improvements to Microsoft Teams for Azure Virtual Desktop. Most importantly, Azure Virtual Desktop now supports audio and video optimization for the Windows Desktop client. Redirection improves latency by creating direct paths between users when they use audio or video in calls and meetings. Less distance means fewer hops, which makes calls look and sound smoother. Learn more at Use Teams on Azure Virtual Desktop.

June 2020

Last month, we introduced Azure Virtual Desktop with Azure Resource Manager integration in preview. This update has lots of exciting new features we'd love to tell you about. Here's what's new for this version of Azure Virtual Desktop.

Azure Virtual Desktop is now integrated with Azure Resource Manager

Azure Virtual Desktop is now integrated into Azure Resource Manager. In the latest update, all Azure Virtual Desktop objects are now Azure Resource Manager resources. This update is also integrated with Azure role-based access control (Azure RBAC). See What is Azure Resource Manager? to learn more.

Here's what this change does for you:

- Azure Virtual Desktop is now integrated with the Azure portal. This means you can manage everything directly in the portal, no PowerShell, web apps, or third-party tools required. To get started, check out our tutorial at Create a host pool with the Azure portal.
- Before this update, you could only publish RemoteApps and Desktops to individual users. With Azure Resource Manager, you can now publish resources to Azure Active Directory groups.
- The earlier version of Azure Virtual Desktop had four built-in admin roles that you could assign to a tenant or host pool. These roles are now in Azure role-based access control (Azure RBAC). You can apply these roles to every Azure Virtual Desktop Azure Resource Manager object, which lets you have a full, rich delegation model.
- In this update, you no longer need to run Azure Marketplace or the GitHub template repeatedly to expand a host pool. All you need to expand a host pool is to go to your host pool in the Azure portal and select **+ Add** to deploy additional session hosts.
- Host pool deployment is now fully integrated with the Azure Shared Image Gallery. Shared Image Gallery

is a separate Azure service that stores VM image definitions, including image versioning. You can also use global replication to copy and send your images to other Azure regions for local deployment.

- Monitoring functions that used to be done through PowerShell or the Diagnostics Service web app have now moved to Log Analytics in the Azure portal. You also now have two options to visualize your reports. You can run Kusto queries and use Workbooks to create visual reports.
- You're no longer required to complete Azure Active Directory (Azure AD) consent to use Azure Virtual Desktop. In this update, the Azure AD tenant on your Azure subscription authenticates your users and provides Azure RBAC controls for your admins.

PowerShell support

We've added new AzWvd cmdlets to the Azure PowerShell Az Module with this update. This new module is supported in PowerShell Core, which runs on .NET Core.

To install the module, follow the instructions in Set up the PowerShell module for Azure Virtual Desktop.

You can also see a list of available commands at the AzWvd PowerShell reference.

For more information about the new features, check out our blog post.

Additional gateways

We've added a new gateway cluster in South Africa to reduce connection latency.

Microsoft Teams on Azure Virtual Desktop (Preview)

We've made some improvements to Microsoft Teams for Azure Virtual Desktop. Most importantly, Azure Virtual Desktop now supports audio and visual redirection for calls. Redirection improves latency by creating direct paths between users when they call using audio or video. Less distance means fewer hops, which makes calls look and sound smoother.

To learn more, see our blog post.

Next steps

Learn about future plans at the Microsoft 365 Azure Virtual Desktop roadmap.

What's new in Azure Monitor for Azure Virtual Desktop?

12/6/2021 • 2 minutes to read • Edit Online

This article describes the changes we make to each new version of Azure Monitor for Azure Virtual Desktop.

If you're not sure which version of Azure Monitor you're currently using, you can find it in the bottom-right corner of your Insights page or configuration workbook. To access your workbook, go to https://aka.ms/azmonwvdi.

How to read version numbers

There are three numbers in each version of Azure Monitor for Azure Virtual Desktop. Here's what each number means:

- The first number is the major version, and is usually used for major releases.
- The second number is the minor version. Minor versions are for backwards-compatible changes such as new features and deprecation notices.
- The third number is the patch version, which is used for small changes that fix incorrect behavior or bugs.

For example, a release with a version number of 1.2.31 is on the first major release, the second minor release, and patch number 31.

When one of the numbers is increased, all numbers after it must change, too. One release has one version number. However, not all version numbers track releases. Patch numbers can be somewhat arbitrary, for example.

Version 1.0.0

Release date: March 21st, 2021.

In this version, we made the following changes:

- We introduced a new visual indicator for high-impact errors and warnings from the Azure Virtual Desktop agent event log on the host diagnostics page.
- We removed five expensive process performance counters from the default configuration. For more information, see our blog post at Updated guidance on Azure Monitor for Azure Virtual Desktop.
- The setup process for Windows Event Log for the configuration workbook is now automated.
- The configuration workbook now supports automated deployment of recommended Windows Event Logs.
- The configuration workbook can now install the Log Analytics agent and setting-preferred workspace for session hosts outside of the resource group's region.
- The configuration workbook now has a tabbed layout for the setup process.
- We introduced versioning with this update.

Next steps

For the general What's New page, see What's New in Azure Virtual Desktop.

To learn more about Azure Monitor for Azure Virtual Desktop, see Use Azure Monitor for Azure Virtual Desktop to monitor your deployment.

Tutorial: Create a tenant in Azure Virtual Desktop (classic)

12/6/2021 • 6 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop (classic), which doesn't support Azure Resource Manager Azure Virtual Desktop objects.

Creating a tenant in Azure Virtual Desktop is the first step toward building your desktop virtualization solution. A tenant is a group of one or more host pools. Each host pool consists of multiple session hosts, running as virtual machines in Azure and registered to the Azure Virtual Desktop service. Each host pool also consists of one or more app groups that are used to publish remote desktop and remote application resources to users. With a tenant, you can build host pools, create app groups, assign users, and make connections through the service.

In this tutorial, learn how to:

- Grant Azure Active Directory permissions to the Azure Virtual Desktop service.
- Assign the TenantCreator application role to a user in your Azure Active Directory tenant.
- Create a Azure Virtual Desktop tenant.

What you need to set up a tenant

Before you start setting up your Azure Virtual Desktop tenant, make sure you have these things:

- The Azure Active Directory tenant ID for Azure Virtual Desktop users.
- A global administrator account within the Azure Active Directory tenant.
  - This also applies to Cloud Solution Provider (CSP) organizations that are creating a Azure Virtual Desktop tenant for their customers. If you're in a CSP organization, you must be able to sign in as global administrator of the customer's Azure Active Directory instance.
  - The administrator account must be sourced from the Azure Active Directory tenant in which you're trying to create the Azure Virtual Desktop tenant. This process doesn't support Azure Active Directory B2B (guest) accounts.
  - The administrator account must be a work or school account.
- An Azure subscription.

You must have the tenant ID, global administrator account, and Azure subscription ready so that the process described in this tutorial can work properly.

Grant permissions to Azure Virtual Desktop

If you have already granted permissions to Azure Virtual Desktop for this Azure Active Directory instance, skip this section.

Granting permissions to the Azure Virtual Desktop service lets it query Azure Active Directory for administrative and end-user tasks.

To grant the service permissions:

1. Open a browser and begin the admin consent flow to the Azure Virtual Desktop server app.

   NOTE

   If you manage a customer and need to grant admin consent for the customer's directory, enter the following URL into the browser and replace {tenant} with the Azure AD domain name of the customer. For example, if the customer's organization has registered the Azure AD domain name of contoso.onmicrosoft.com, replace {tenant} with contoso.onmicrosoft.com.

   ```
   https://login.microsoftonline.com/{tenant}/adminconsent?client_id=5a0aa725-4958-4b0c-80a9-34562e23f3b7&redirect_uri=https%3A%2F%2Frdweb.wvd.microsoft.com%2FRDWeb%2FConsentCallback
   ```

2. Sign in to the Azure Virtual Desktop consent page with a global administrator account. For example, if you were with the Contoso organization, your account might be admin@contoso.com or admin@contoso.onmicrosoft.com.

3. Select **Accept**.

4. Wait for one minute so Azure AD can record consent.

5. Open a browser and begin the admin consent flow to the Azure Virtual Desktop client app.

   NOTE

   If you manage a customer and need to grant admin consent for the customer's directory, enter the following URL into the browser and replace {tenant} with the Azure AD domain name of the customer. For example, if the customer's organization has registered the Azure AD domain name of contoso.onmicrosoft.com, replace {tenant} with contoso.onmicrosoft.com.

   ```
   https://login.microsoftonline.com/{tenant}/adminconsent?client_id=fa4345a4-a730-4230-84a8-7d9651b86739&redirect_uri=https%3A%2F%2Frdweb.wvd.microsoft.com%2FRDWeb%2FConsentCallback
   ```

6. Sign in to the Azure Virtual Desktop consent page as global administrator, as you did in step 2.

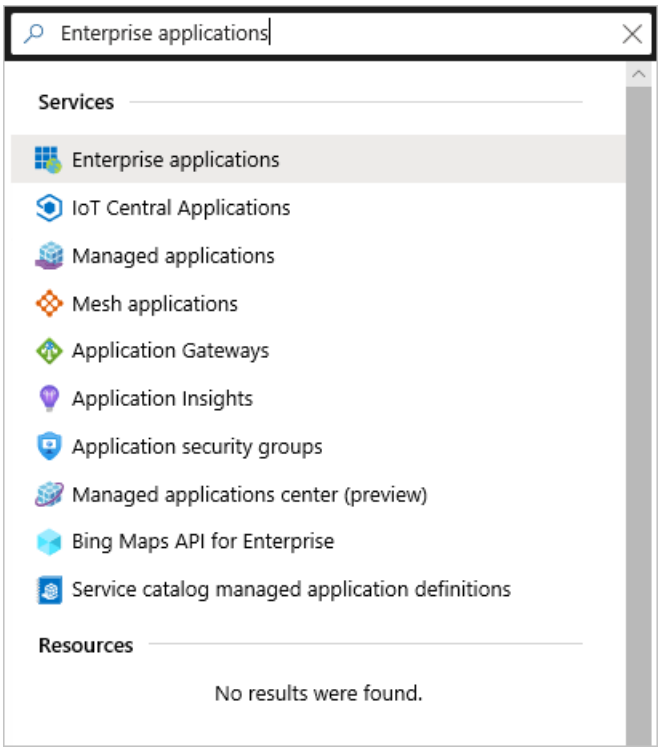
7. Select **Accept**.

Assign the TenantCreator application role

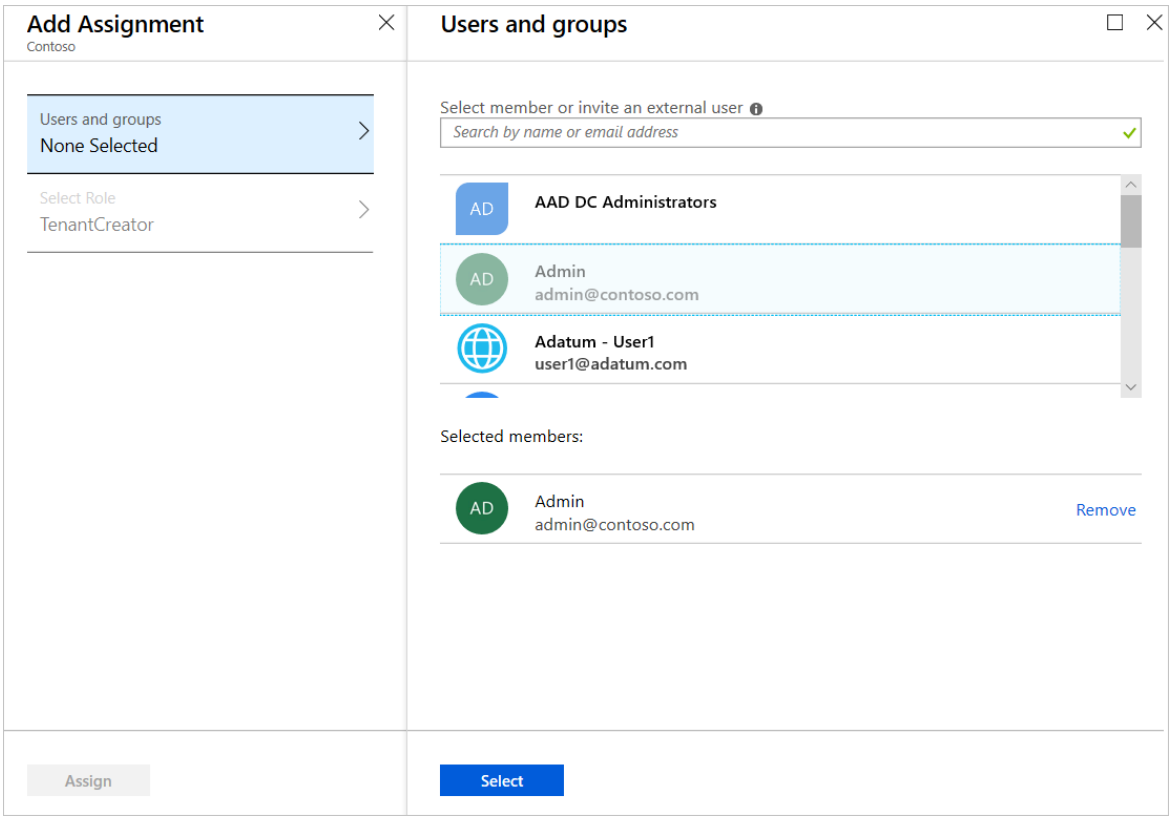
Assigning an Azure Active Directory user the TenantCreator application role allows that user to create a Azure Virtual Desktop tenant associated with the Azure Active Directory instance. You'll need to use your global administrator account to assign the TenantCreator role.

To assign the TenantCreator application role:

1. Go to the Azure portal to manage the TenantCreator application role. Search for and select **Enterprise applications**. If you're working with multiple Azure Active Directory tenants, it's a best practice to open a private browser session and copy and paste the URLs into the address bar.



2. Within **Enterprise applications**, search for **Azure Virtual Desktop**. You'll see the two applications that you provided consent for in the previous section. Of these two apps, select **Azure Virtual Desktop**.
3. Select **Users and groups**. You might see that the administrator who granted consent to the application is already listed with the **Default Access** role assigned. This is not enough to create a Azure Virtual Desktop tenant. Continue following these instructions to add the **TenantCreator** role to a user.
4. Select **Add user**, and then select **Users and groups** in the **Add Assignment** tab.
5. Search for a user account that will create your Azure Virtual Desktop tenant. For simplicity, this can be the global administrator account.
    - If you're using a Microsoft Identity Provider like contosoadmin@live.com or contosoadmin@outlook.com, you might not be able to sign in to Azure Virtual Desktop. We recommend using a domain-specific account like admin@contoso.com or admin@contoso.onmicrosoft.com instead.



> **NOTE**
>
> You must select a user (or a group that contains a user) that's sourced from this Azure Active Directory instance. You can't choose a guest (B2B) user or a service principal.

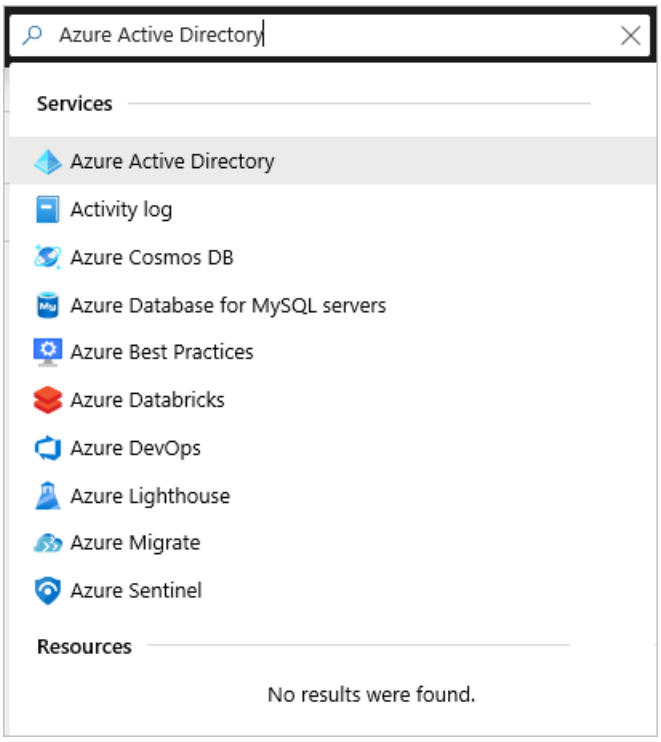
6. Select the user account, choose the **Select** button, and then select **Assign**.
7. On the **Azure Virtual Desktop - Users and groups** page, verify that you see a new entry with the **TenantCreator** role assigned to the user who will create the Azure Virtual Desktop tenant.

Before you continue on to create your Azure Virtual Desktop tenant, you need two pieces of information:

- Your Azure Active Directory tenant ID (or **Directory ID**)
- Your Azure subscription ID

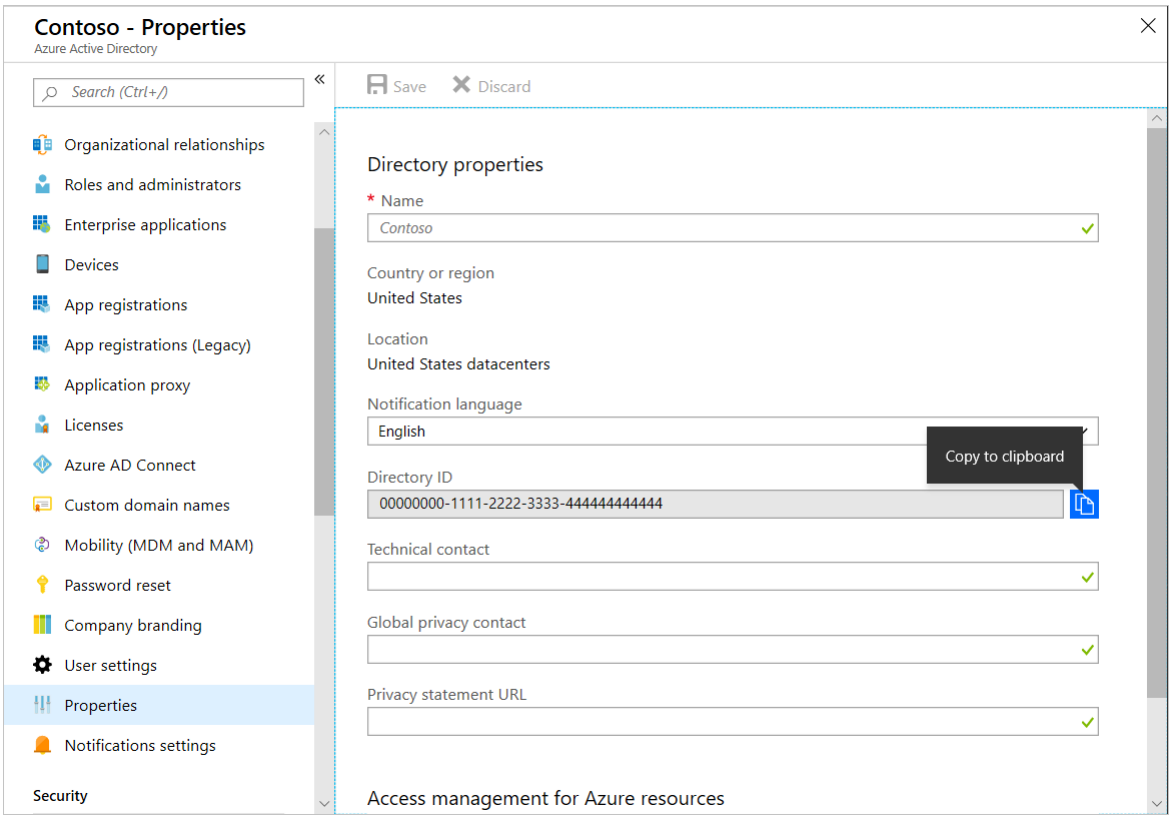
To find your Azure Active Directory tenant ID (or **Directory ID**):

1. In the same Azure portal session, search for and select **Azure Active Directory**.



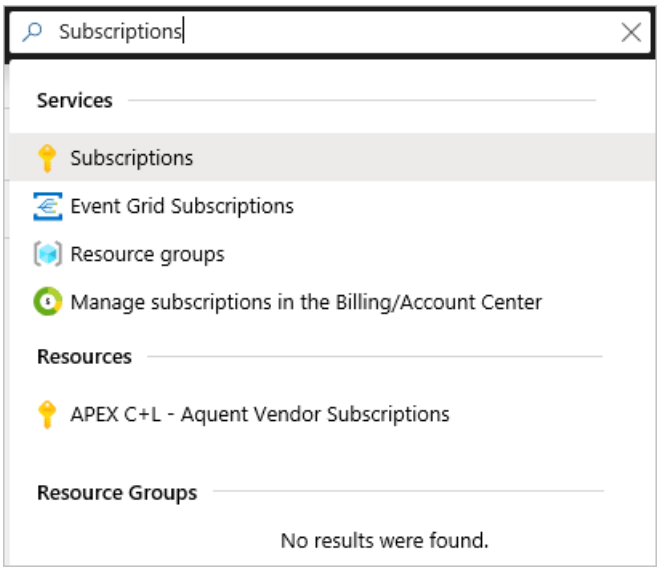
2. Scroll down until you find **Properties**, and then select it.

3. Look for **Directory ID**, and then select the clipboard icon. Paste it in a handy location so you can use it later as the **AadTenantId** value.

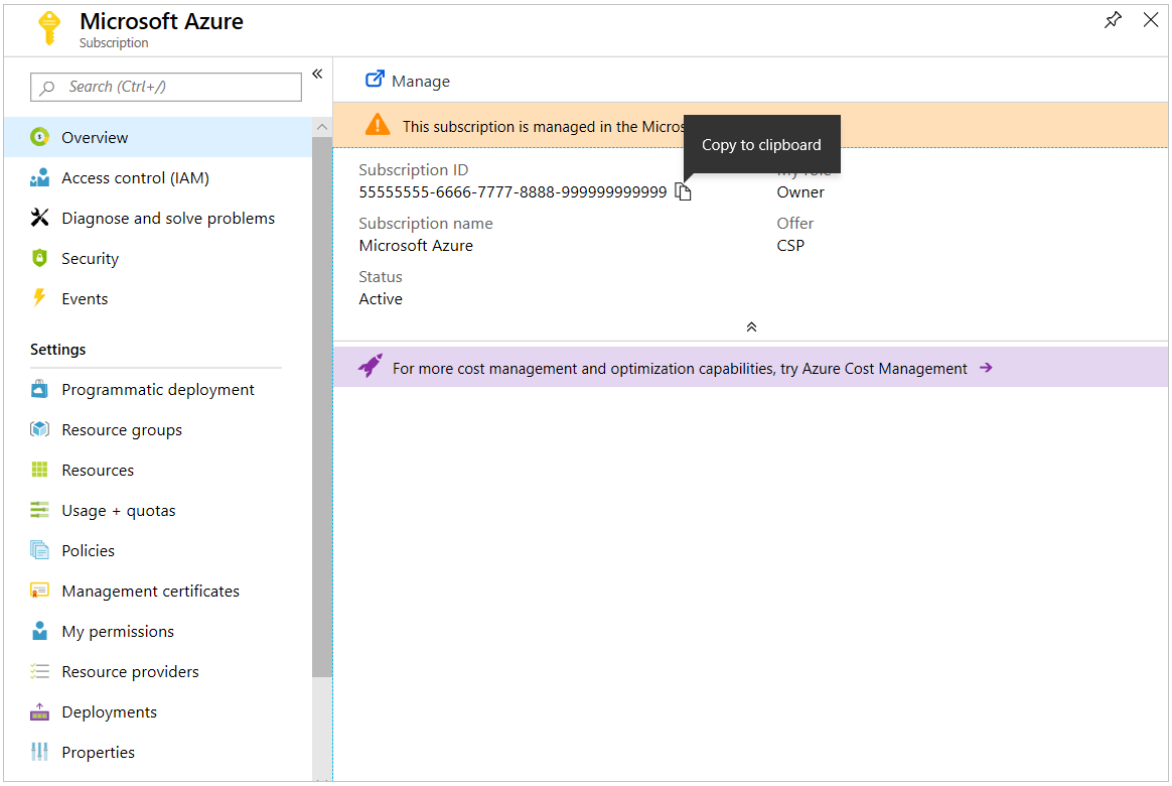


To find your Azure subscription ID:

1. In the same Azure portal session, search for and select **Subscriptions**.



2. Select the Azure subscription you want to use to receive Azure Virtual Desktop service notifications.
3. Look for **Subscription ID**, and then hover over the value until a clipboard icon appears. Select the clipboard icon and paste it in a handy location so you can use it later as the **AzureSubscriptionId** value.



Create a Azure Virtual Desktop tenant

Now that you've granted the Azure Virtual Desktop service permissions to query Azure Active Directory and assigned the TenantCreator role to a user account, you can create a Azure Virtual Desktop tenant.

First, download and import the Azure Virtual Desktop module to use in your PowerShell session if you haven't already.

Sign in to Azure Virtual Desktop by using the TenantCreator user account with this cmdlet:

```
Add-RdsAccount -DeploymentUrl "https://rdbroker.wvd.microsoft.com"
```

After that, create a new Azure Virtual Desktop tenant associated with the Azure Active Directory tenant:

```
New-RdsTenant -Name <TenantName> -AadTenantId <DirectoryID> -AzureSubscriptionId <SubscriptionID>
```

Replace the bracketed values with values relevant to your organization and tenant. The name you choose for your new Azure Virtual Desktop tenant should be globally unique. For example, let's say you're the Azure Virtual Desktop TenantCreator for the Contoso organization. The cmdlet you'd run would look like this:

```
New-RdsTenant -Name Contoso -AadTenantId 00000000-1111-2222-3333-444444444444 -AzureSubscriptionId 55555555-
6666-7777-8888-999999999999
```

It's a good idea to assign administrative access to a second user in case you ever find yourself locked out of your account, or you go on vacation and need someone to act as the tenant admin in your absence. To assign admin access to a second user, run the following cmdlet with `<TenantName>` and `<Upn>` replaced with your tenant name and the second user's UPN.

```
New-RdsRoleAssignment -TenantName <TenantName> -SignInName <Upn> -RoleDefinitionName "RDS Owner"
```

Next steps

After you've created your tenant, you'll need to create a service principal in Azure Active Directory and assign it a role within Azure Virtual Desktop. The service principal will allow you to successfully deploy the Azure Virtual Desktop Azure Marketplace offering to create a host pool. To learn more about host pools, continue to the tutorial for creating a host pool in Azure Virtual Desktop.

Create service principals and role assignments with PowerShell

Deploy Azure Virtual Desktop with the getting started feature

12/6/2021 • 6 minutes to read • Edit Online

The Azure portal's new getting started feature is a quick, easy way to install and configure Azure Virtual Desktop on your deployment.

Requirements

You'll need to meet the following requirements to be able to use getting started:

- An Azure Active Directory (AD) tenant.
- An account with global admin permissions on Azure AD.

  > **NOTE**
  > The getting started feature doesn't currently support MSA, B2B, or guest accounts at this time.

- An active Azure subscription.

  > **NOTE**
  > The getting started feature doesn't currently support accounts with multi-factor authentication.

- An account with **Owner permissions** on the subscription.

If you're using the getting started feature in an environment with Active Directory Domain Services (AD DS), you'll also need to meet these requirements:

- AD DS domain admin credentials.
- You must configure Azure AD connect on your subscription and make sure the "USERS" container is syncing with Azure AD.
- The domain controller in your virtual machine (VM) must not have DSC extensions of type **Microsoft.Powershell.DSC**.

If you're using the getting started feature in an environment without an identity provider, these are the extra requirements you should follow:

- Your AD domain join UPN must not include any keywords that the username guideline list doesn't allow, and you must use a unique user name that's not already in your Azure AD subscription.
- You must create a new host pool to add session hosts you create with the getting started feature. If you try to make a session host in an existing host pool, it won't work.

For subscriptions with Azure AD DS or AD DS

Here's how to use the getting started feature in a subscription that already has Azure AD DS or AD DS:

1. Open the Azure portal.
2. Sign in to Azure and open **Azure Virtual Desktop management**, then select the **Getting started** tab.

This will open the landing page for the getting started feature.

3. Select **Create**.
4. In the **Basic** tab, select the following values:
   - For **Subscription**, go to **How is your subscription configured**, then select **Existing setup**.
   - In the **Location**, select the location where you'll deploy your resources.
   - For **Azure admin UPN**, enter the full user principal name (UPN) of the account with admin permissions in Azure AD and owner permissions in the subscription that you plan to use.
   - For **AD Domain join UPN** enter the full UPN of the account with permissions that you plan to use to join the VMs to your domain.
   - For **Identity**, select either **Azure AD DS** or **AD DS** depending on your environment. What you choose here will affect the input your VMs will need.
5. In the **Virtual machines** tab, select the following values:
   - For **Do you want the users to share this machine?**, select one of the following options depending on your needs:
     - If you want to create a single-session or personal host pool, select **No**.
     - If you want to create a multi-session or pooled host pool, select **Yes (multi-session)**. This will also create an Azure Files storage account joined to either Azure AD DS or AD DS.
   - For **Image type**, select an image from the Azure image gallery, a custom image, or a VHD from a storage blob.
   - For **VM size**, select the size and SKU you want for the VMs you'll deploy.
   - For **Number of VMs**, select how many VMs you want to provision in the host pool.
   - If you're using an existing setup with AD DS, these options will appear:
     - For **Subnet**, select a subnet in the VNET. The subnet you choose must either be in the same location as the identity (AD DS or Azure AD DS) or peered to it.
     - For **Domain controller resource group**, select the resource group where the AD DS VM is either located or peered to. The resource group with the domain controller must be in the same subscription. The get started feature doesn't currently support peered subscriptions at this time.
     - For **Domain controller virtual machine**, enter the name of the VM running your deployment's AD DS.
   - If you want to open the Select Azure AD users or Users group, select the **Assign existing users** check box.
   - If you want to create a validation user account to test your deployment, select the **Create validation user** check box, then enter a username and password in the prompt that appears.

     > **NOTE**
     > Getting started will create the validation user group in the "USERS" container. You must make sure your validation group is synced to Azure AD. If the sync doesn't work, then pre-create the AVDValidationUsers group in an organization unit that is being synced to Azure AD.

For subscriptions without Azure AD DS or AD DS

This section will show you how to use the getting started feature for a subscription without Azure AD DS or AD DS. For reference, these subscriptions are sometimes called "empty" subscriptions.

To deploy Azure Virtual Desktop on a subscription without Azure AD DS or AD DS:

1. Open the Azure portal.
2. Sign in to Azure and open **Azure Virtual Desktop management**, then select the **Getting started** tab. This will open the landing page for the getting started feature.
3. In the **Basic** tab, select the following values:
   - For **Subscription**, select the subscription you want to deploy Azure Virtual Desktop in.
   - For **How is your subscription configured**, select **Empty subscription**. An "empty" subscription is a subscription that doesn't require an identity provider like Azure AD or AD DS.
   - For **Resource group prefix**, enter the prefixes for the resource group you're going to create: *-prerequisite*, *-deployment*, and *-avd*.
   - In **Location**, enter the resource location you want to use for your deployment.
   - For **Azure admin UPN**, enter the full UPN of an account with admin permissions on Azure AD and owner permissions on the subscription.
   - For **AD Domain join UPN**, enter the full UPN for an account that will be added to **AAD DC Administrators** group.

   > **NOTE**
   > The user name for AD Domain join UPN should be a unique one that doesn't already exist in Azure AD. The getting started feature doesn't currently support using existing Azure AD user names for accounts without Azure AD or AD DS.

4. In the **Virtual machines** tab, select the following values:
   - For **Do you want the users to share this machine?**, select one of the following options depending on your needs:
     - If you want to create a single-session or personal host pool, select **No**.
     - If you want to create a multi-session or pooled host pool, select **Yes (multi-session)**. This will also create an Azure Files storage account joined to either Azure AD DS or AD DS.
   - For **Image type**, select an image from the Azure image gallery, a custom image, or a VHD from a storage blob.
   - For **VM size**, select the size and SKU you want for the VMs you'll deploy.
   - For **Number of VMs**, select how many VMs you want to provision in the host pool.
5. In the **Assignments** tab, select the **Create validation user**, then enter a username and password into the **Validation user username** and **Validation user password** fields. The validation user is a user who'll test your deployment once it's ready.

Clean up resources

If after deployment you change your mind and want to remove Azure Virtual Desktop resources from your environment without incurring extra billing costs, you can safely remove them by following the instructions in

this section.

If you created your resources on a subscription with Azure AD DS or AD DS, the feature will have made two resource groups with the prefixes "*-deployment*" and "*-avd*." In the Azure portal, go to **Resource groups** and delete any resource groups with those prefixes to remove the deployment.

If you created your resources on a subscription without Azure AD DS or AD DS, the feature will have made three resource groups with the prefixes *-prerequisite*, *-deployment*, and *-avd*. In the Azure portal, go to **Resource groups** and delete any resource groups with those prefixes to remove the deployment.

Next steps

If you'd like to learn how to deploy Azure Virtual Desktop in a more in-depth way, check out our tutorials for setting up your deployment manually, starting with Create a host pool with the Azure portal.

Tutorial: Create a host pool

12/6/2021 • 14 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article. Any objects you create with Azure Virtual Desktop (classic) can't be managed with the Azure portal.

Host pools are a collection of one or more identical virtual machines (VMs), also known as "session hosts," within Azure Virtual Desktop environments. Each host pool can contain an app group that users can interact with as they would on a physical desktop. If you'd like to learn more about deployment architecture, check out Azure Virtual Desktop environment. If you're an app developer using remote app streaming for Azure Virtual Desktop, your customers or users can use your apps just like local apps on a physical device. For more information how to use Azure Virtual Desktop as an app developer, check out our Azure Virtual Desktop remote app streaming documentation.

NOTE

If you're an app developer using remote app streaming for Azure Virtual Desktop and your app's users are in the same organization as your deployment, you can use your existing Azure tenant to create your host pool. If your users are outside of your organization, then for security reasons you'll need to create separate Azure tenants with at least one host pool for each organization. Learn more about which practices we recommend you follow to keep your deployment secure at Architecture recommendations.

This article will walk you through the setup process for creating a host pool for an Azure Virtual Desktop environment through the Azure portal. This method provides a browser-based user interface to create a host pool in Azure Virtual Desktop, create a resource group with VMs in an Azure subscription, join those VMs to either an Active Directory (AD) domain or Azure Active Directory (Azure AD) tenant, and register the VMs with Azure Virtual Desktop.

Prerequisites

There are two different sets of requirements depending on if you're an IT professional setting up a deployment for your organization or an app developer serving applications to customers.

Requirements for IT professionals

You'll need to enter the following parameters to create a host pool:

- The VM image name
- VM configuration
- Domain and network properties
- Azure Virtual Desktop host pool properties

You'll also need to know the following things:

- Where the source of the image you want to use is. Is it from Azure Gallery or is it a custom image?
- Your domain join credentials.

Requirements for app developers

If you're an app developer who's using remote app streaming for Azure Virtual Desktop to deliver apps to your customers, here's what you'll need to get started:

- If you plan on serving your organization's app to end-users, make sure you actually have that app ready. For more information, see How to host custom apps with Azure Virtual Desktop.
- If existing Azure Gallery image options don't meet your needs, you'll also need to create your own custom image for your session host VMs. To learn more about how to create VM images, see Prepare a Windows VHD or VHDX to upload to Azure and Create a managed image of a generalized VM in Azure.
- Your domain join credentials. If you don't already have an identity management system compatible with Azure Virtual Desktop, you'll need to set up identity management for your host pool. To learn more, see Set up managed identities.

Final requirements

Finally, make sure you've registered the Microsoft.DesktopVirtualization resource provider. If you haven't already, go to **Subscriptions**, select the name of your subscription, and then select **Resource providers**. Search for **DesktopVirtualization**, select **Microsoft.DesktopVirtualization**, and then select **Register**.

If you're an IT professional creating a network, when you create a Azure Virtual Desktop host pool with the Azure Resource Manager template, you can create a virtual machine from the Azure gallery, a managed image, or an unmanaged image. To learn more about how to create VM images, see Prepare a Windows VHD or VHDX to upload to Azure and Create a managed image of a generalized VM in Azure. (If you're an app developer, you don't need to worry about this part.)

Last but not least, if you don't have an Azure subscription already, make sure to create an account before you start following these instructions.

Begin the host pool setup process

- Portal
- Azure CLI

To start creating your new host pool:

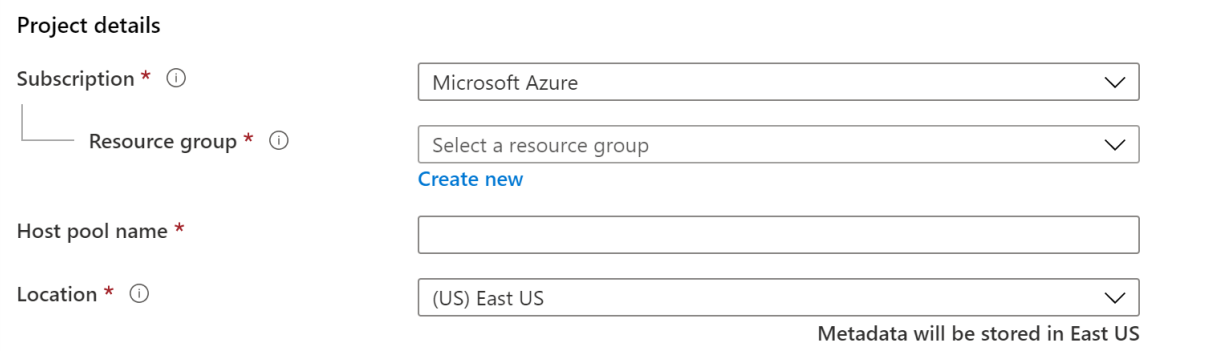
1. Sign in to the Azure portal at https://portal.azure.com.

   > **NOTE**
   > If you're signing in to the US Gov portal, go to https://portal.azure.us/ instead.
   >
   > If you're accessing the Azure China portal, go to https://portal.azure.cn/.

2. Enter **Azure Virtual Desktop** into the search bar, then find and select **Azure Virtual Desktop** under Services.
3. In the **Azure Virtual Desktop** overview page, select **Create a host pool**.
4. In the **Basics** tab, select the correct subscription under Project details.
5. Either select **Create new** to make a new resource group or select an existing resource group from the drop-down menu.
6. Enter a unique name for your host pool.
7. In the Location field, select the region where you want to create the host pool from the drop-down menu.

   The Azure geography associated with the regions you selected is where the metadata for this host pool and its related objects will be stored. Make sure you choose the regions inside the geography you want

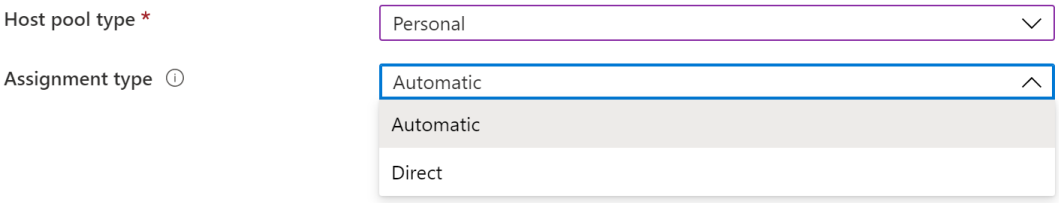
the service metadata to be stored in.



> **NOTE**
> If you want to create your host pool in a supported region outside the US, you'll need to re-register the resource provider. After re-registering, you should see the other regions in the drop-down for selecting the location. Learn how to re-register at our Host pool creation troubleshooting article.

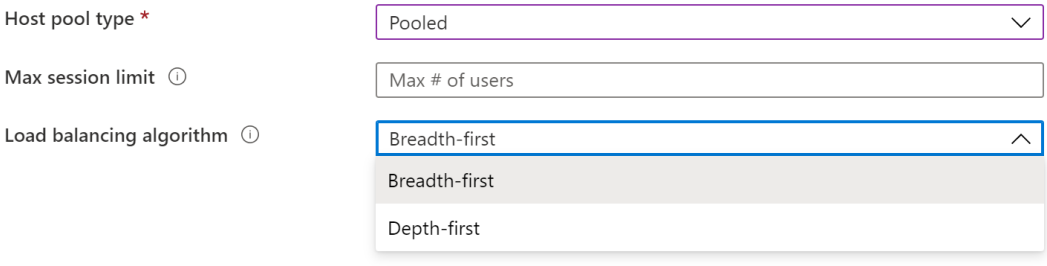
8. Under Host pool type, select whether your host pool will be **Personal** or **Pooled**.

   - If you choose **Personal**, then select either **Automatic** or **Direct** in the Assignment Type field.



9. If you choose **Pooled**, enter the following information:

   - For **Max session limit**, enter the maximum number of users you want load-balanced to a single session host.
   - For **Load balancing algorithm**, choose either breadth-first or depth-first, based on your usage pattern. Learn more about what each of these options means at Host pool load-balancing methods.



10. Select **Next: Virtual Machines >**.
11. If you've already created virtual machines and want to use them with the new host pool, select **No**, select **Next: Workspace >** and jump to the Workspace information section. If you want to create new virtual machines and register them to the new host pool, select **Yes**.

Now that you've created a host pool, let's move on to the next part of the setup process where we create the VM.

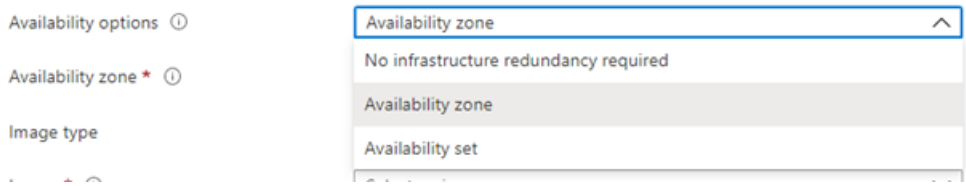
Virtual machine details

Now that we're through the first part, you'll have to set up your VM.

- Portal
- Azure CLI

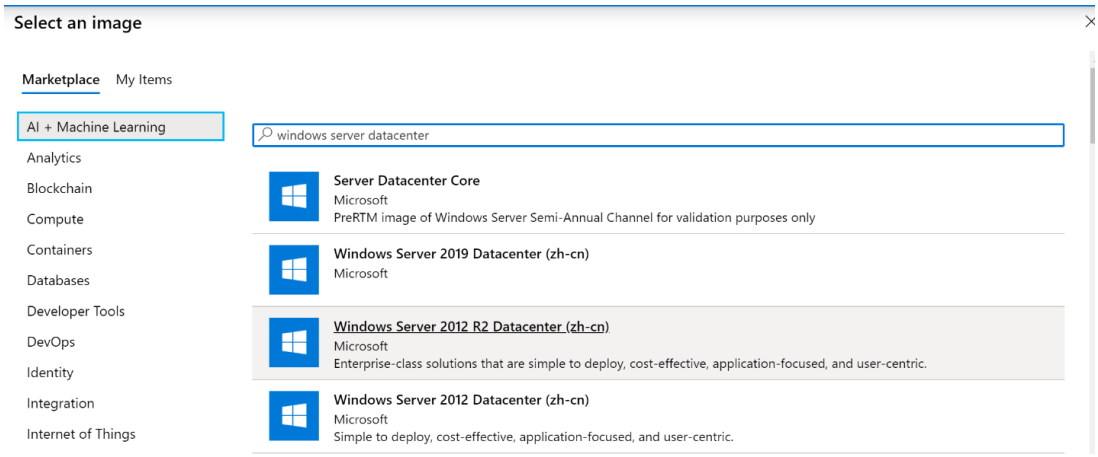
To set up your virtual machine within the Azure portal host pool setup process:

1. Under **Resource group**, choose the resource group where you want to create the virtual machines. This can be a different resource group than the one you used for the host pool.
2. After that, provide a **Name prefix** to name the virtual machines the setup process creates. The suffix will be `-` with numbers starting from 0.
3. Choose the **Virtual machine location** where you want to create the virtual machines. They can be the same or different from the region you selected for the host pool. Keep in mind that VM prices vary by region, and the VM locations should be near their users when possible to maximize performance. Learn more at Data locations for Azure Virtual Desktop.
4. Next, choose the availability option that best suit your needs. To learn more about which option is right for you, see Availability options for virtual machines in Azure and our FAQ.

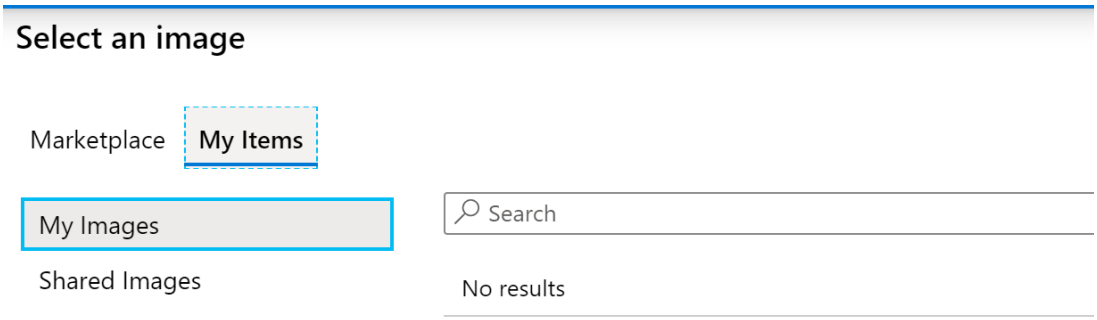


5. Next, choose the image that needs to be used to create the virtual machine. You can choose either **Gallery** or **Storage blob**.
   - If you choose **Gallery**, select one of the recommended images from the drop-down menu:
     - Windows 10 Enterprise multi-session, Version 1909
     - Windows 10 Enterprise multi-session, Version 1909 + Microsoft 365 Apps
     - Windows Server 2019 Datacenter
     - Windows 10 Enterprise multi-session, Version 2004
     - Windows 10 Enterprise multi-session, Version 2004 + Microsoft 365 Apps

     If you don't see the image you want, select **See all images**, which lets you select either another image in your gallery or an image provided by Microsoft and other publishers. Make sure that the image you choose is one of the supported OS images.



   You can also go to **My Items** and choose a custom image you've already uploaded.



- If you choose **Storage Blob**, you can use your own image build through Hyper-V or on an Azure VM. All you have to do is enter the location of the image in the storage blob as a URI.

The image's location is independent of the availability option, but the image's zone resiliency determines whether that image can be used with availability zone. If you select an availability zone while creating your image, make sure you're using an image from the gallery with zone resiliency enabled. To learn more about which zone resiliency option you should use, see the FAQ.

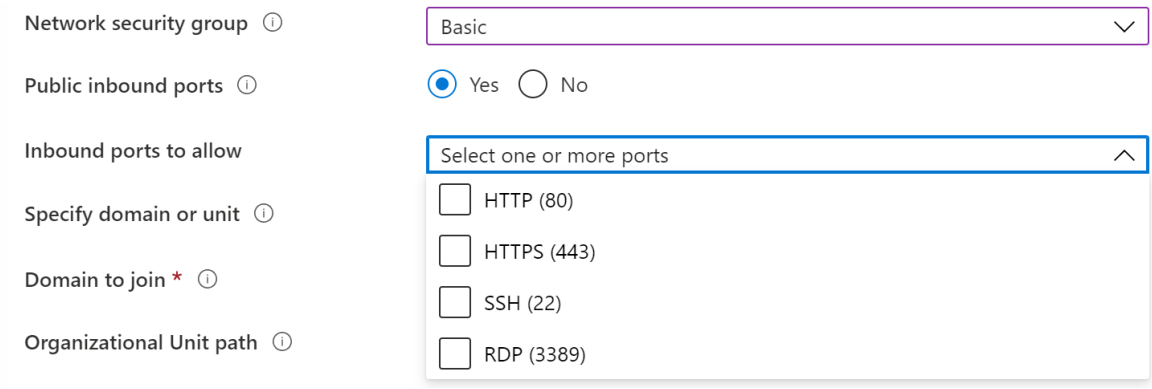
6. After that, choose the **Virtual machine size** you want to use. You can either keep the default size as-is or select **Change size** to change the size. If you select **Change size**, in the window that appears, choose the size of the virtual machine suitable for your workload. To learn more about virtual machine sizes and which size you should choose, see Virtual machine sizing guidelines.
7. Under **Number of VMs**, provide the number of VMs you want to create for your host pool.

   > **NOTE**
   > The setup process can create up to 400 VMs while setting up your host pool, and each VM setup process creates four objects in your resource group. Since the creation process doesn't check your subscription quota, make sure the number of VMs you enter is within the Azure VM and API limits for your resource group and subscription. You can add more VMs after you finish creating your host pool.

8. Choose what kind of OS disks you want your VMs to use: Standard SSD, Premium SSD, or Standard HDD.
9. Under Network and security, select the **Virtual network** and **Subnet** where you want to put the virtual machines you create. Make sure the virtual network can connect to the domain controller, since you'll need to join the virtual machines inside the virtual network to the domain. The DNS servers of the virtual network you selected should be configured to use the IP of the domain controller.
10. Select what kind of security group you want: **Basic**, **Advanced**, or **None**.

    If you select **Basic**, you'll have to select whether you want any inbound port open. If you select **Yes**, choose from the list of standard ports to allow inbound connections to.

    > **NOTE**
    > For greater security, we recommend that you don't open public inbound ports.



If you choose **Advanced**, select an existing network security group that you've already configured.

11. After that, select whether you want the virtual machines to be joined to **Active Directory** or **Azure Active Directory** (Preview).

    - For Active Directory, provide an account to join the domain and choose if you want to join a specific domain and organizational unit.

        - For the AD domain join UPN, enter the credentials for the Active Directory Domain admin of the virtual network you selected. The account you use can't have multifactor authentication (MFA) enabled. When joining to an Azure Active Directory Domain Services (Azure AD DS) domain, the account you use must be part of the Azure AD DC Administrators group and the account password must work in Azure AD DS.

        - To specify a domain, select **Yes**, then enter the name of the domain you want to join. If you want, you can also add a specific organizational unit you want the virtual machines to be in by entering the full path (Distinguished Name) and without quotation marks. If you don't want to specify a domain, select **No**. The VMs will automatically join the domain that matches the suffix of the **AD domain join UPN**.

    - For Azure Active Directory, you can select **Enroll the VM with Intune** to automatically make the VM available for management after it's deployed.

12. Under **Virtual Machine Administrator account**, enter the credentials for the local admin account to be added while creating the VM. You can use this account for management purposes in both AD and Azure AD-joined VMs.

13. Under **Post update custom configuration**, you can enter the location of an Azure Resource Manager template to perform custom configurations on your session hosts after you create them. You'll need to enter the URLs for both the Azure Resource Manager template file and the Azure Resource Manager template parameter file.

    > **NOTE**
    > Azure Virtual Desktop doesn't support provisioning Azure resources in the template.

14. Select **Next: Workspace >**.

With that, we're ready to start the next phase of setting up your host pool: registering your app group to a workspace.

Workspace information

The host pool setup process creates a desktop application group by default. For the host pool to work as intended, you'll need to publish this app group to users or user groups, and you must register the app group to

a workspace.

> **NOTE**
> If you're an app developer trying to publish your organization's apps, you can dynamically attach MSIX apps to user sessions or add your app packages to a custom VM image. See How to serve your custom app with Azure Virtual Desktop for more information.

- Portal
- Azure CLI

To register the desktop app group to a workspace:

1. Select **Yes**.

   If you select **No**, you can register the app group later, but we recommend you get the workspace registration done as soon as you can so your host pool works properly.

2. Next, choose whether you want to create a new workspace or select from existing workspaces. Only workspaces created in the same location as the host pool will be allowed to register the app group to.

3. Optionally, you can select **Next: Tags >**.

   Here you can add tags so you can group the objects with metadata to make things easier for your admins.

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

   > **NOTE**
   > The review + create validation process doesn't check if your password meets security standards or if your architecture is correct, so you'll need to check for any problems with either of those things yourself.

5. Review the information about your deployment to make sure everything looks correct. When you're done, select **Create**.

This starts the deployment process, which creates the following objects:

- Your new host pool.
- A desktop app group.
- A workspace, if you chose to create it.
- If you chose to register the desktop app group, the registration will be completed.
- Virtual machines, if you chose to create them, which are joined to the domain and registered with the new host pool.
- A download link for an Azure Resource Management template based on your configuration.

After that, you're all done!

Run the Azure Resource Manager template to provision a new host pool

If you'd rather use an automated process, download our Azure Resource Manager template to provision your new host pool instead.

NOTE

If you're using an automated process to build your environment, you'll need the latest version of the configuration JSON file. You can find the JSON file here.

Next steps

Now that you've made your host pool, you can populate it with RemoteApp programs. To learn more about how to manage apps in Azure Virtual Desktop, head to our next tutorial:

Manage app groups tutorial

Tutorial: Manage app groups with the Azure portal

12/6/2021 • 5 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

The default app group created for a new Azure Virtual Desktop host pool also publishes the full desktop. In addition, you can create one or more RemoteApp application groups for the host pool. Follow this tutorial to create a RemoteApp app group and publish individual Start menu apps.

NOTE

You can dynamically attach MSIX apps to user sessions or add your app packages to a custom virtual machine (VM) image to publish your organization's apps. Learn more at How to host custom apps with Azure Virtual Desktop.

In this tutorial, learn how to:

- Create a RemoteApp group.
- Grant access to RemoteApp programs.

Create a RemoteApp group

If you've already created a host pool and session host VMs using the Azure portal or PowerShell, you can add application groups from the Azure portal with the following process:

1. Sign in to the Azure portal.

   NOTE

   If you're signing in to the US Gov portal, go to https://portal.azure.us/ instead.

   If you're accessing the Azure China portal, go to https://portal.azure.cn/.

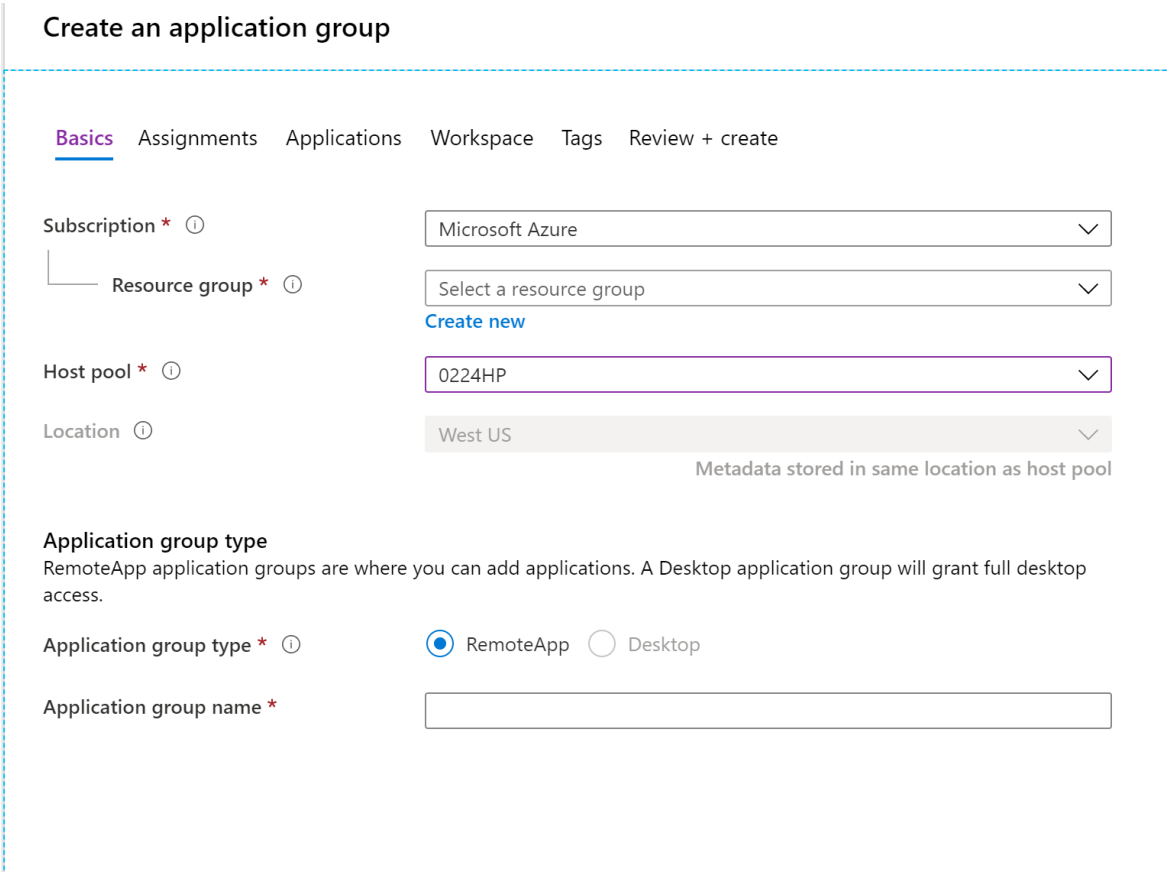
2. Search for and select **Azure Virtual Desktop**.
3. You can add an application group directly or you can add it from an existing host pool. Choose an option below:

   - Select **Application groups** in the menu on the left side of the page, then select **+ Add**.
   - Select **Host pools** in the menu on the left side of the screen, select the name of the host pool, select **Application groups** from the menu on the left side, then select **+ Add**. In this case, the host pool will already be selected on the Basics tab.

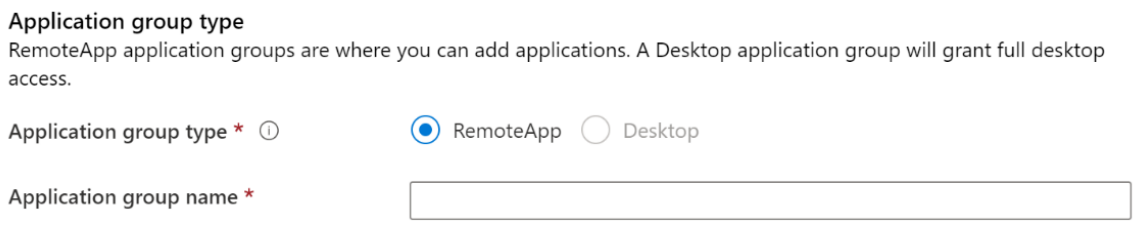
4. On the **Basics** tab, select the **Subscription** and **Resource group** you want to create the app group for. You can also choose to create a new resource group instead of selecting an existing one.
5. Select the **Host pool** that will be associated with the application group from the drop-down menu.

NOTE

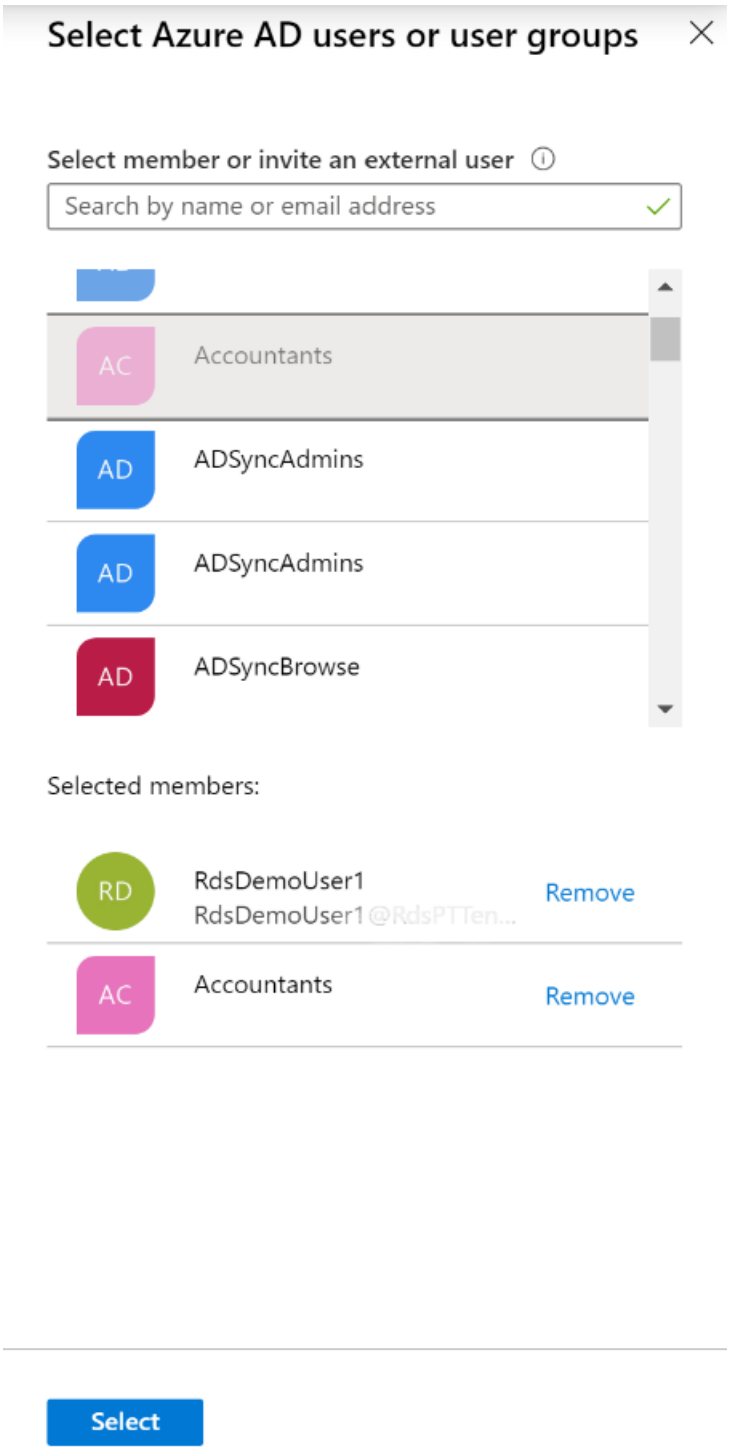
You must select the host pool associated with the application group. App groups have apps or desktops that are served from a session host and session hosts are part of host pools. The app group needs to be associated with a host pool during creation.



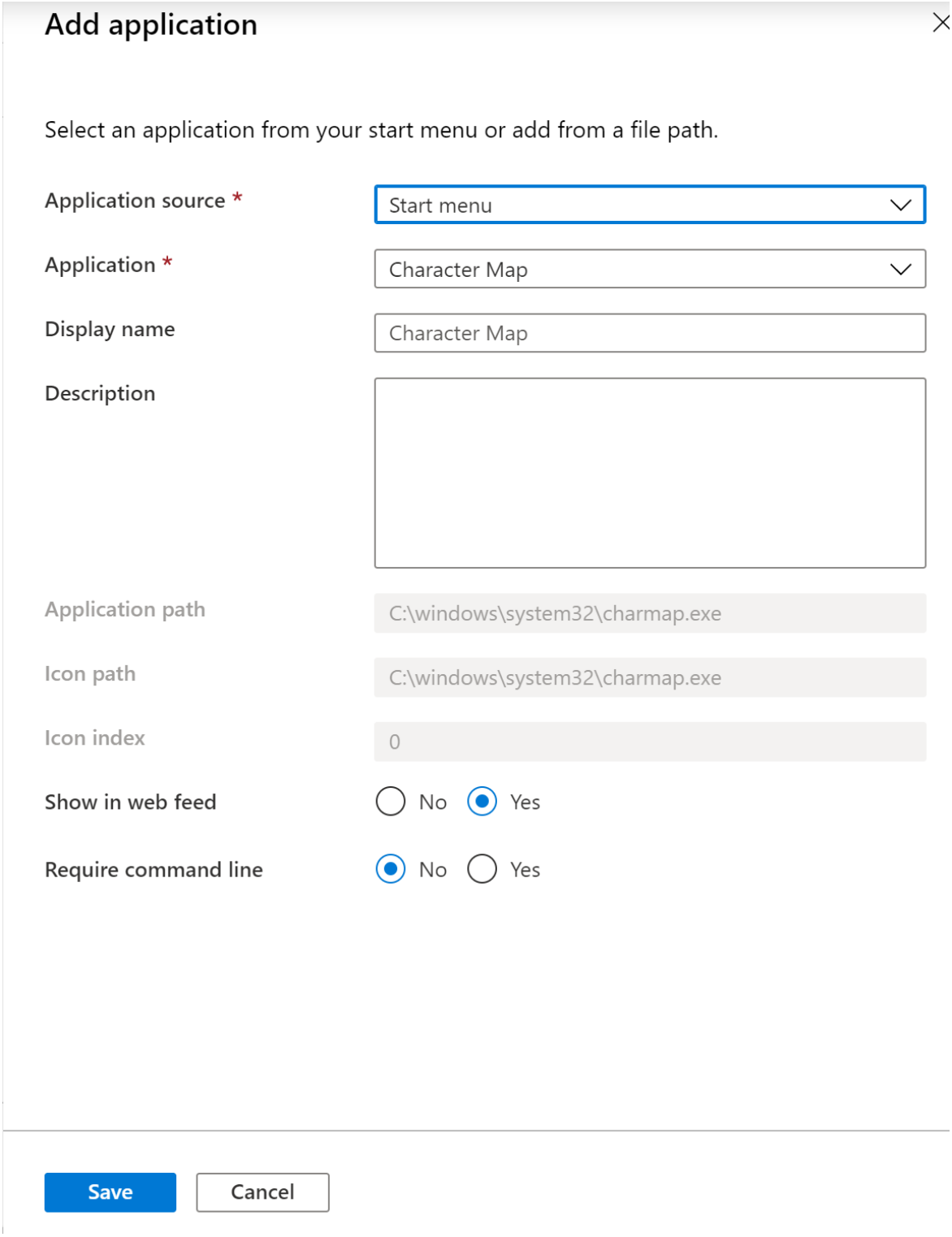
6. Select **RemoteApp** under **Application group type**, then enter a name for your RemoteApp.



7. Select **Next: Assignments >** tab.
8. To assign individual users or user groups to the app group, select **+Add Azure AD users or user groups**.
9. Select the users you want to have access to the apps. You can select single or multiple users and user groups.



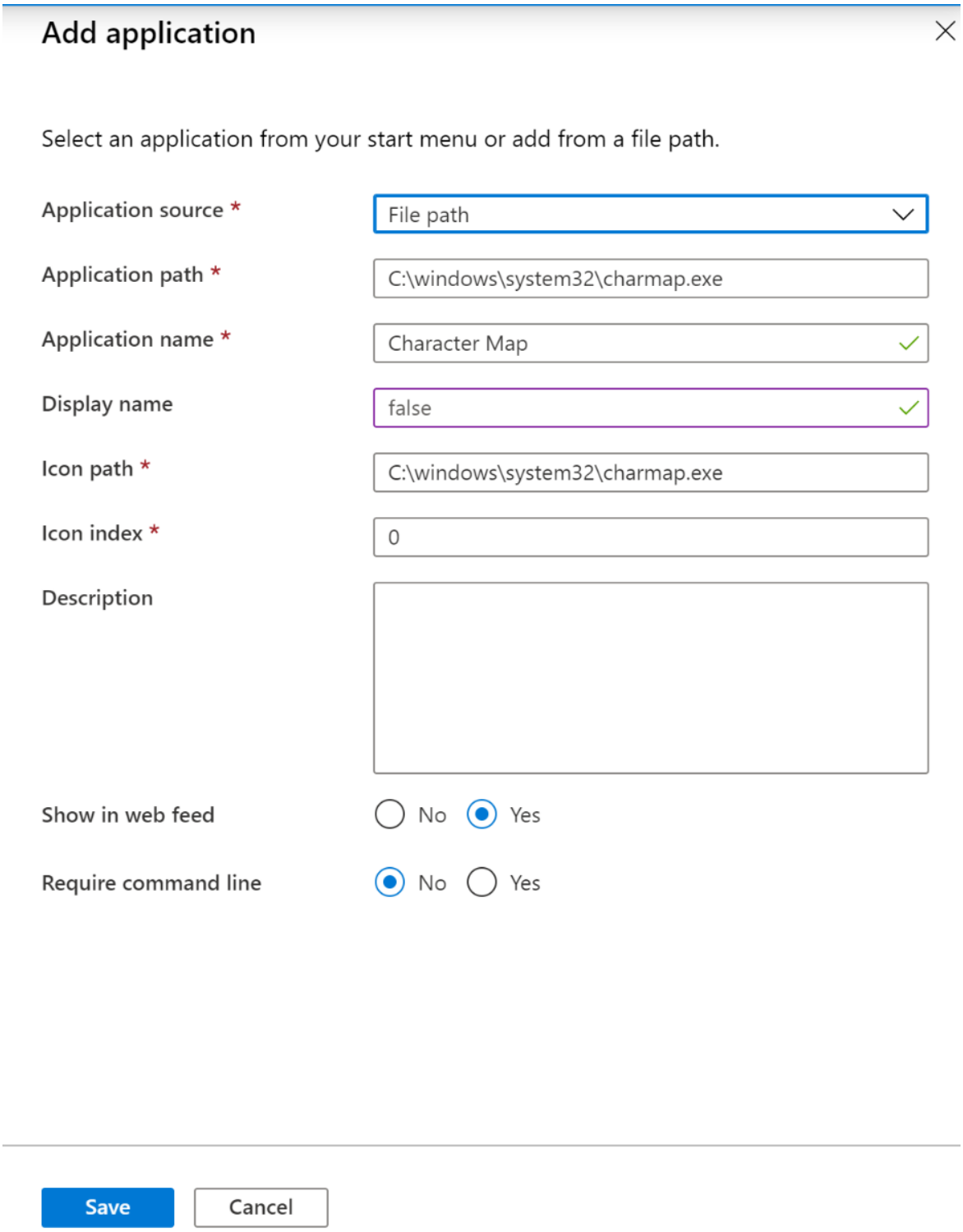
10. Select **Select**.
11. Select **Next: Applications >**, then select **+Add applications**.
12. To add an application from the start menu:
    - Under **Application source**, select **Start menu** from the drop-down menu. Next, under **Application**, choose the application from the drop-down menu.



- In **Display name**, enter the name for the application that will be shown to the user on their client.
- Leave the other options as-is and select **Save**.

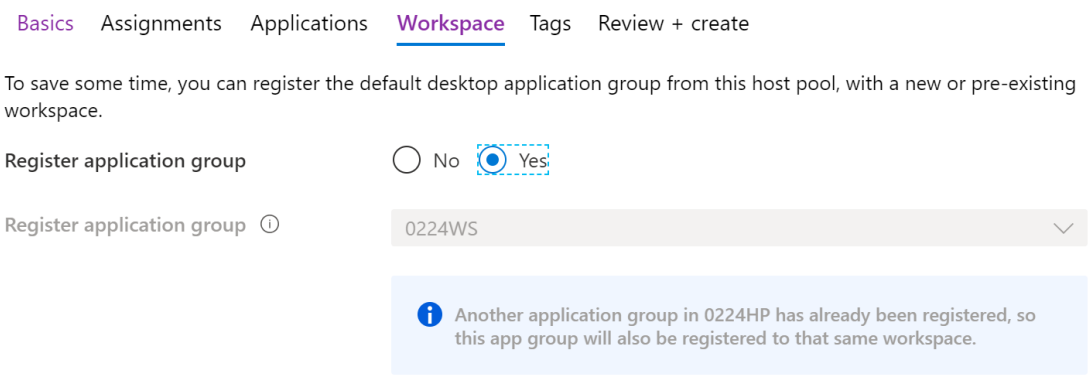
13. To add an application from a specific file path:

    - Under **Application source**, select **File path** from the drop-down menu.
    - In **Application path**, enter the path to the application on the session host registered with the associated host pool.
    - Enter the application's details in the **Application name**, **Display name**, **Icon path**, and **Icon index** fields.
    - Select **Save**.



14. Repeat this process for every application you want to add to the application group.
15. Next, select **Next: Workspace >**.
16. If you want to register the app group to a workspace, select **Yes** for **Register application group**. If you'd rather register the app group at a later time, select **No**.
17. If you select **Yes**, you can select an existing workspace to register your app group to.

> **NOTE**
> You can only register the app group to workspaces created in the same location as the host pool. Also. if you've previously registered another app group from the same host pool as your new app group to a workspace, it will be selected and you can't edit it. All app groups from a host pool must be registered to the same workspace.



18. Optionally, if you want to create tags to make your workspace easy to organize, select **Next: Tags >** and enter your tag names.
19. When you're done, select **Review + create**.
20. Wait a bit for the validation process to complete. When it's done, select **Create** to deploy your app group.

The deployment process will do the following things for you:

- Create the RemoteApp app group.
- Add your selected apps to the app group.
- Publish the app group published to users and user groups you selected.
- Register the app group, if you chose to do so.
- Create a link to an Azure Resource Manager template based on your configuration that you can download and save for later.

> **IMPORTANT**
> You can only create 200 application groups for each Azure Active Directory tenant. We added this limit because of service limitations for retrieving feeds for our users. This limit doesn't apply to app groups created in Azure Virtual Desktop (classic).

Edit or remove an app

To edit or remove an app from an app group:

1. Sign in to the Azure portal.

   > **NOTE**
   > If you're signing in to the US Gov portal, go to https://portal.azure.us/ instead.

2. Search for and select **Azure Virtual Desktop**.
3. You can either add an application group directly or from an existing host pool by choosing one of the following options:

   - To add a new application group directly, select **Application groups** in the menu on the left side of the page, then select the app group you want to edit.
   - To edit an app group in an existing host pool, select **Host pools** in the menu on the left side of the screen, select the name of the host pool, then select **Application groups** in the menu that appears on the left side of the screen, and then select the app group you want to edit.

4. Select **Applications** in the menu on the left side of the page.
5. If you want to remove an application, select the check box next to the application, then select **Remove** from the menu on the top of the page.
6. If you want to edit the details of an application, select the application name. This will open up the editing menu.
7. When you're done making changes, select **Save**.

Next steps

In this tutorial, you learned how to create an app group, populate it with RemoteApp programs, and assign users to the app group. To learn how to create a validation host pool, see the following tutorial. You can use a validation host pool to monitor service updates before rolling them out to your production environment.

Create a host pool to validate service updates

Tutorial: Create a host pool to validate service updates

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Host pools are a collection of one or more identical virtual machines within Azure Virtual Desktop environment. We highly recommend you create a validation host pool where service updates are applied first. Validation host pools let you monitor service updates before the service applies them to your standard or non-validation environment. Without a validation host pool, you may not discover changes that introduce errors, which could result in downtime for users in your standard environment.

To ensure your apps work with the latest updates, the validation host pool should be as similar to host pools in your non-validation environment as possible. Users should connect as frequently to the validation host pool as they do to the standard host pool. If you have automated testing on your host pool, you should include automated testing on the validation host pool.

You can debug issues in the validation host pool with either the diagnostics feature or the Azure Virtual Desktop troubleshooting articles.

NOTE

We recommend that you leave the validation host pool in place to test all future updates.

Create your host pool

You can configure any existing pooled or personal host pool to be a validation host pool. You can also create a new host pool to use for validation by following the instructions in any of these articles:

- Tutorial: Create a host pool with Azure Marketplace or the Azure CLI
- Create a host pool with PowerShell or the Azure CLI

Define your host pool as a validation host pool

- Portal
- Azure PowerShell
- Azure CLI

To use the Azure portal to configure your validation host pool:

1. Sign in to the Azure portal at https://portal.azure.com.
2. Search for and select **Azure Virtual Desktop**.
3. In the Azure Virtual Desktop page, select **Host pools**.
4. Select the name of the host pool you want to edit.
5. Select **Properties**.

6. In the validation environment field, select **Yes** to enable the validation environment.
7. Select **Save** to apply the new settings.

Update schedule

Service updates happen monthly. If there are major issues, critical updates will be provided at a more frequent pace.

If there are any service updates, make sure you have at least a couple of users sign in each day to validate the environment. We recommend you regularly visit our TechCommunity site and follow any posts with WVDUPdate to stay informed about service updates.

Next steps

Now that you've created a validation host pool, you can learn how to use Azure Service Health to monitor your Azure Virtual Desktop deployment.

Set up service alerts

Tutorial: Set up service alerts

12/6/2021 • 2 minutes to read • Edit Online

> **IMPORTANT**
> This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can use Azure Service Health to monitor service issues and health advisories for Azure Virtual Desktop. Azure Service Health can notify you with different types of alerts (for example, email or SMS), help you understand the effect of an issue, and keep you updated as the issue resolves. Azure Service Health can also help you mitigate downtime and prepare for planned maintenance and changes that could affect the availability of your resources.

In this tutorial, you'll learn how to:

- Create and configure service alerts.

To learn more about Azure Service Health, see the Azure Health Documentation.

Create service alerts

This section shows you how to configure Azure Service Health and how to set up notifications, which you can access on the Azure portal. You can set up different types of alerts and schedule them to notify you in a timely manner.

Recommended service alerts

We recommend you create service alerts for the following health event types:

- **Service issue:** Receive notifications on major issues that impact connectivity of your users with the service or with the ability to manage your Azure Virtual Desktop tenant.
- **Health advisory:** Receive notifications that require your attention. The following are some examples of this type of notification:
  - Virtual Machines (VMs) not securely configured as open port 3389
  - Deprecation of functionality

Configure service alerts

To configure service alerts:

1. Sign in to the Azure portal.
2. Select **Service Health**.
3. Follow the instructions in Create activity log alerts on service notifications to set up your alerts and notifications.

Next steps

In this tutorial, you learned how to set up and use Azure Service Health to monitor service issues and health advisories for Azure Virtual Desktop. To learn about how to sign in to Azure Virtual Desktop, continue to the Connect to Azure Virtual Desktop How-tos.

Connect to the Remote Desktop client on Windows 7 and Windows 10

Migrate automatically from Azure Virtual Desktop (classic) (preview)

12/6/2021 • 9 minutes to read • Edit Online

IMPORTANT

The migration module tool for Azure Virtual Desktop is currently in public preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

The migration module tool (preview) lets you migrate your organization from Azure Virtual Desktop (classic) to Azure Virtual Desktop automatically. This article will show you how to use the tool.

Requirements

Before you use the migration module, make sure you have the following things ready:

- An Azure subscription where you'll create new Azure service objects.
- You must be assigned the Contributor role to create Azure objects on your subscription, and the User Access Administrator role to assign users to application groups.
- At least Remote Desktop Services (RDS) Contributor permissions on an RDS tenant or the specific host pools you're migrating.
- The latest version of the Microsoft.RdInfra.RDPowershell PowerShell module
- The latest version of the Az.DesktopVirtualization PowerShell module
- The latest version of the Az.Resources PowerShell module
- Install the migration module in your computer
- PowerShell or PowerShell ISE to run the scripts you'll see in this article. The Microsoft.RdInfra.RDPowershell module doesn't work in PowerShell Core.

IMPORTANT

Migration only creates service objects in the US geography. If you try to migrate your service objects to another geography, it won't work. Also, if you have more than 200 app groups in your Azure Virtual Desktop (classic) deployment, you won't be able to migrate. You'll only be able to migrate if you rebuild your environment to reduce the number of app groups within your Azure Active Directory (Azure AD) tenant.

Prepare your PowerShell environment

First, you'll need to prepare your PowerShell environment for the migration process.

To prepare your PowerShell environment:

1. Before you start, make sure you have the latest version of the Az.Desktop Virtualization and Az.Resources modules by running the following cmdlets:

```
Get-Module Az.Resources
Get-Module Az.DesktopVirtualization
https://www.powershellgallery.com/packages/Az.DesktopVirtualization/
https://www.powershellgallery.com/packages/Az.Resources/
```

If you don't, then install and import the modules by running these cmdlets:

```
Install-module Az.Resources
Import-module Az.Resources
Install-module Az.DesktopVirtualization
Import-module Az.DesktopVirtualization
```

2. Next, uninstall the current RDInfra PowerShell module by running this cmdlet:

```
Uninstall-Module -Name Microsoft.RDInfra.RDPowershell -AllVersions
```

3. After that, install the RDPowershell module with this cmdlet:

```
Install-Module -Name Microsoft.RDInfra.RDPowershell -RequiredVersion 1.0.3414.0 -force
Import-module Microsoft.RDInfra.RDPowershell
```

4. Once you're done installing everything, run this cmdlet to make sure you have the right versions of the modules:

```
Get-Module Microsoft.RDInfra.RDPowershell
```

5. Now, let's install and import the migration module by running these cmdlets:

```
Install-Module -Name PackageManagement -Repository PSGallery -Force
Install-Module -Name PowerShellGet -Repository PSGallery -Force
# Then restart shell
Install-Module -Name Microsoft.RdInfra.RDPowershell.Migration -RequiredVersion 1.0.3725-Prerelease -
AllowPrerelease -AllowClobber
Import-Module <Full path to the location of the migration
module>\Microsoft.RdInfra.RDPowershell.Migration.psd1
```

6. Once you're done, sign into Windows Virtual Desktop (classic) in your PowerShell window:

```
Add-RdsAccount -DeploymentUrl https://rdbroker.wvd.microsoft.com
```

7. Sign in to Azure Resource Manager:

```
Login-AzAccount
```

8. If you have multiple subscriptions, select the one you want to migrate your resources to with this cmdlet:

```
Select-AzSubscription -Subscriptionid <subID>
```

9. Register the Resource Provider in Azure portal for the selected subscription.

10. Finally you'll need to register the provider. There are two ways you can do this:

- If you want to use PowerShell, then run this cmdlet:

  ```
  Register-AzResourceProvider -ProviderNamespace Microsoft.DesktopVirtualization
  ```

- If you'd rather use the Azure portal, open and sign in to the Azure portal, then go to **Subscriptions** and select the name of the subscription you want to use. After that, go to **Resource Provider** > **Microsoft.DesktopVirtualization** and select **Re-register**. You won't see anything change in the UI just yet, but your PowerShell environment should now be ready to run the module.

Migrate Azure Virtual Desktop (classic) resources to Azure Resource Manager

Now that your PowerShell environment is ready, you can begin the migration process.

To migrate your Azure virtual Desktop (classic) resources to Azure Resource Manager:

1. Before you migrate, if you want to understand how the existing Classic resources will get mapped to new Azure Resource Manager resources, run this cmdlet:

   ```
   Get-RdsHostPoolMigrationMapping
   ```

   With **Get-RdsHostPoolMigrationMapping**, you can create a CSV file that maps where your resources will go. For example, if your tenant's name is "Contoso," and you want to store your mapping file in the "contosouser" file, you'd run a cmdlet that looks like this:

   ```
   Get-RdsHostPoolMigrationMapping -Tenant Contoso -HostPool Office -Location EastUS -OutputFile
   'C:\\Users\contosouser\OneDrive - Microsoft\Desktop\mapping.csv'
   ```

2. Next, run the **Start-RdsHostPoolMigration** cmdlet to choose whether to migrate a single host pool or all host pools within a tenant.

   For example:

   ```
   Start-RdsHostPoolMigration -Tenant Contoso -Location WestUS
   ```

   If you want to migrate your resources a specific host pool, then include the host pool name. For example, if you want to move the host pool named "Office," run a command like this:

   ```
   Start-RdsHostPoolMigration -Tenant Contoso -HostPool Office -CopyUserAssignments $false -Location
   EastUS
   ```

   If you don't give a workspace name, the module will automatically create one for you based on the tenant name. However, if you'd prefer to use a specific workspace, you can enter its resource ID like this:

   ```
   Start-RdsHostPoolMigration -Tenant Contoso -HostPool Office -CopyUserAssignments -Location EastUS -
   Workspace <Resource ID of workspacename>
   ```

   If you'd like to use a specific workspace but don't know its resource ID, run this cmdlet:

```
Get-AzWvdWorkspace -WorkspaceName <workspace> -ResourceGroupName <resource group> |fl
```

You'll also need to specify a user assignment mode for the existing user assignments:

- Use **Copy** to copy all user assignments from your old app groups to Azure Resource Manager application groups. Users will be able to see feeds for both versions of their clients.
- Use **None** if you don't want to change the user assignments. Later, you can assign users or user groups to app groups with the Azure portal, PowerShell, or API. Users will only be able to see feeds using the Azure Virtual Desktop (classic) clients.

You can only copy 2,000 user assignments per subscription, so your limit will depend on how many assignments are already in your subscription. The module calculates the limit based on how many assignments you already have. If you don't have enough assignments to copy, you'll get an error message that says "Insufficient role assignment quota to copy user assignments. Rerun command without the -CopyUserAssignments switch to migrate."

3. After you run the commands, it will take up to 15 minutes for the module to create the service objects. If you copied or moved any user assignments, that will add to the time it takes for the module to finish setting everything up.

   After the **Start-RdsHostPoolMigration** cmdlet is done, you should see the following things:

   - Azure service objects for the tenant or host pool you specified
   - Two new resource groups:
     - A resource group called "Tenantname," which contains your workspace.
     - A resource group called "Tenantname_originalHostPoolName," which contains the host pool and desktop app groups.
   - Any users you published to the newly created app groups.
   - Virtual machines will be available in both existing and new host pools to avoid user downtime during the migration process. This lets users connect to the same user session.

   Since these new Azure service objects are Azure Resource Manager objects, the module can't set Role-based Access Control (RBAC) permissions or diagnostic settings on them. Therefore, you'll need to update the RBAC permissions and settings for these objects manually.

   Once the module validates the initial user connections, you can also publish the app group to more users or user groups, if you'd like.

   > **NOTE**
   > After migration, if you move app groups to a different resource group after assigning permissions to users, it will remove all RBAC roles. You'll need to reassign users RBAC permissions all over again.

4. If you want to delete all Azure Virtual Desktop (classic) service objects, run **Complete-RdsHostPoolMigration** to finish the migration process. This cmdlet will delete all Azure Virtual Desktop (classic) objects, leaving only the new Azure objects. Users will only be able to see the feed for the newly created app groups on their clients. Once this command is done, you can safely delete the Azure Virtual Desktop (classic) tenant to finish the process.

   For example:

```
Complete-RdsHostPoolMigration -Tenant Contoso -Location EastUS
```

If you want to complete a specific host pool, you can include the host pool name in the cmdlet. For example, if you want to complete a host pool named "Office," you'd use a command like this:

```
Complete-RdsHostPoolMigration -Tenant Contoso -HostPool Office -Location EastUS
```

This will delete all service objects created by Azure Virtual Desktop (classic). You will be left with just the new Azure objects and users will only be able to see the feed for the newly created app groups on their clients. Once you are done finalizing your migration, you need to explicitly delete the tenant in Azure Virtual Desktop (classic).

5. If you've changed your mind about migrating and want to revert the process, run the **Revert-RdsHostPoolMigration** cmdlet.

   For example:

   ```
   Revert-RdsHostPoolMigration -Tenant Contoso -Location EastUS
   ```

   If you'd like to revert a specific host pool, you can include the host pool name in the command. For example, if you want to revert a host pool named "Office," then you'd enter something like this:

   ```
   Revert-RdsHostPoolMigration -Tenant Contoso -HostPool Office -Location EastUS
   ```

   This cmdlet will delete all newly created Azure service objects. Your users will only see the feed for Azure Virtual Desktop (classic) objects in their clients.

   However, the cmdlet won't delete the workspace the module created or its associated resource group. You'll need to manually delete those items to get rid of them.

6. If you don't want to delete your Azure Virtual Desktop (classic) service objects yet but do want to test migration, you can run **Set-RdsHostPoolHidden**.

   For example:

   ```
   Set-RdsHostPoolHidden -Tenant Contoso -Hostpool Office -Hidden $true -Location WestUS
   ```

   Setting the status to "true" will hide the Azure Virtual Desktop (classic) resources. Setting it to "false" will reveal the resources to your users.

   The *-Hostpool* parameter is optional. You can use this parameter if there's a specific Azure Virtual Desktop (classic) host pool you want to hide.

   This cmdlet will hide the Azure Virtual Desktop (classic) user feed and service objects instead of deleting them. However, this is usually only used for testing and doesn't count as a completed migration. To complete your migration, you'll need to run the **Complete-RdsHostPoolMigration** command. Otherwise, revert your deployment by running **Revert-RdsHostPoolMigration**.

Troubleshoot automatic migration

This section explains how to solve commonly encountered issues in the migration module.

I can't access the tenant

First, try these two things:

- Make sure your admin account has the required permissions to access the tenant.
- Try running **Get-RdsTenant** on the tenant.

If those two things work, try running the **Set-RdsMigrationContext** cmdlet to set the RDS Context and ADAL Context for your migration:

1. Create the RDS Context by running the **Add-RdsAccount** cmdlet.
2. Find the RDS Context in the global variable *$rdMgmtContext*.
3. Find the ADAL Context in the global variable *$AdalContext*.
4. Run **Set-RdsMigrationContext** with the variables you found in this format:

```
Set-RdsMigrationContext -RdsContext <rdscontext> -AdalContext <adalcontext>
```

Next steps

If you'd like to learn how to migrate your deployment manually instead, see Migrate manually from Azure Virtual Desktop (classic).

Once you've migrated, get to know how Azure Virtual Desktop works by checking out our tutorials. Learn about advanced management capabilities at Expand an existing host pool and Customize RDP properties.

To learn more about service objects, check out Azure Virtual Desktop environment.

Migrate manually from Azure Virtual Desktop (classic)

12/6/2021 • 3 minutes to read • Edit Online

Azure Virtual Desktop (classic) creates its service environment with PowerShell cmdlets, REST APIs, and service objects. An "object" in a Azure Virtual Desktop service environment is a thing that Azure Virtual Desktop creates. Service objects include tenants, host pools, application groups, and session hosts.

However, Azure Virtual Desktop (classic) isn't integrated with Azure. Without Azure integration, any objects you create aren't automatically managed by the Azure portal because they're not connected to your Azure subscription.

The recent major update of Azure Virtual Desktop marks a shift in the service towards full Azure integration. Objects you create in Azure Virtual Desktop are automatically managed by the Azure portal.

In this article, we'll explain why you should consider migrating to the latest version of Azure Virtual Desktop. After that, we'll tell you how to manually migrate from Azure Virtual Desktop (classic) to the latest update of Azure Virtual Desktop.

Why migrate?

Major updates can be inconvenient, especially ones you have to do manually. However, there are some reasons why you can't automatically migrate:

- Existing service objects made with the classic release don't have any representation in Azure. Their scope doesn't extend beyond the Azure Virtual Desktop service.
- With the latest update, the service's application ID was changed to remove consent for apps the way it did for Azure Virtual Desktop (classic). You won't be able to create new Azure objects with Azure Virtual Desktop unless they're authenticated with the new application ID.

Despite the hassle, migrating away from the classic version is still important. Here's what you can do after you migrate:

- Manage Azure Virtual Desktop through the Azure portal.
- Assign Azure Active Directory (AD) user groups to application groups.
- Use the improved Log Analytics feature to troubleshoot your deployment.
- Use Azure-native role-based access control (Azure RBAC) to manage administrative access.

When should I migrate?

When asking yourself if you should migrate, you should also take into account your deployment's current and future situation.

There are a few scenarios in particular where we recommend you manually migrate:

- You have a test host pool setup with a small number of users.
- You have a production host pool setup with a small number of users, but plan to eventually ramp up to hundreds of users.
- You have a simple setup that can be easily replicated. For example, if your VMs use a gallery image.

> **IMPORTANT**
> If you're using an advanced configuration that took a long time to stabilize or has a lot of users, we don't recommend manually migrating.

Prepare for migration

Before you get started, you'll need to make sure your environment is ready to migrate.

Here's what you need to start the migration process:

- An Azure subscription where you'll create new Azure service objects.
- Make sure you're assigned to the following roles:
  - Contributor
  - User Access Administrator

  The Contributor role lets you create Azure objects on your subscription, and the User Access Administrator role lets you assign users to application groups.

How to migrate manually

Now that you've prepared for the migration process, it's time to actually migrate.

To migrate manually from Azure Virtual Desktop (classic) to Azure Virtual Desktop:

1. Follow the instructions in Create a host pool with the Azure portal to create all high-level objects with the Azure portal.
2. If you want to bring over the virtual machines you're already using, follow the instructions in Register the virtual machines to the Azure Virtual Desktop host pool to manually register them to the new host pool you created in step 1.
3. Create new RemoteApp app groups.
4. Publish users or user groups to the new desktop and RemoteApp app groups.
5. Update your Conditional Access policy to allow the new objects by following the instructions in Set up multi-factor authentication.

To prevent downtime, you should first register your existing session hosts to the Azure Resource Manager-integrated host pools in small groups at a time. After that, slowly bring your users over to the new Azure Resource Manager-integrated app groups.

Next steps

If you'd like to learn how to migrate your deployment automatically instead, go to Migrate automatically from Azure Virtual Desktop (classic).

Once you've migrated, get to know how Azure Virtual Desktop works by checking out our tutorials. Learn about advanced management capabilities at Expand an existing host pool and Customize RDP properties.

To learn more about service objects, check out Azure Virtual Desktop environment.

Connect with the Windows Desktop client

12/6/2021 • 2 minutes to read • Edit Online

You can access Azure Virtual Desktop resources on devices with Windows 11, Windows 10, Windows 10 IoT Enterprise, and Windows 7 using the Windows Desktop client.

> **IMPORTANT**
>
> This method doesn't support Windows 8 or Windows 8.1.
>
> This method only supports Azure Resource Manager objects. To support objects without Azure Resource Manager, see Connect with Windows Desktop (classic) client.
>
> This method also doesn't support the RemoteApp and Desktop Connections (RADC) client or the Remote Desktop Connection (MSTSC) client.

Install the Windows Desktop client

Download the client based on your Windows version:

- Windows 64-bit
- Windows 32-bit
- Windows ARM64

During installation to determine access, select either:

- **Install just for you**
- **Install for all users of this machine** (requires admin rights)

To launch the client after installation, use the **Start** menu and search for **Remote Desktop**.

Subscribe to a Workspace

To subscribe to a Workspace, choose to either:

- Use a work or school account and have the client discover the resources available for you
- Use the specific URL of the resource

To launch the resource once subscribed, go to the **Connection Center** and double-click the resource.

> **TIP**
>
> To launch a resource from the **Start** menu, you can find the folder with the Workspace name or enter the resource name in the search bar.

Use a user account

1. Select **Subscribe** from the main page.
2. Sign in with your user account when prompted.

The resources grouped by workspace will appear in the **Connection Center**.

> **NOTE**
>
> The Windows client automatically defaults to Azure Virtual Desktop (classic).
>
> However, if the client detects additional Azure Resource Manager resources, it adds them automatically or notifies the user that they're available.

Use a specific URL

1. Select **Subscribe with URL** from the main page.
2. Enter either the *Workspace URL* or an *email address*:
   - For **Workspace URL**, use the URL provided by your admin.

AVAILABLE RESOURCES	URL
Azure Virtual Desktop (classic)	`https://rdweb.wvd.microsoft.com/api/feeddiscovery/webfeeddiscovery.`
Azure Virtual Desktop	`https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery`
Azure Virtual Desktop (US Gov)	`https://rdweb.wvd.azure.us/api/arm/feeddiscovery`
Azure Virtual Desktop (China)	`https://rdweb.wvd.azure.cn/api/arm/feeddiscovery`

   - For **Email**, use your email address.

   The client will find the URL associated with your email, provided your admin has enabled email discovery.
3. Select **Next**.
4. Sign in with your user account when prompted.

The resources grouped by workspace will appear in the **Connection Center**.

Next steps

To learn more about how to use the client, check out Get started with the Windows Desktop client.

If you're an admin interested in learning more about the client's features, check out Windows Desktop client for admins.

Connect to Azure Virtual Desktop with the web client

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

The web client lets you access your Azure Virtual Desktop resources from a web browser without the lengthy installation process.

NOTE

The web client doesn't currently have mobile OS support.

Supported operating systems and browsers

IMPORTANT

As of September 30, 2021, the Azure Virtual Desktop web client no longer supports Internet Explorer. We recommend that you use Microsoft Edge to connect to the web client instead. For more information, see our blog post.

While any HTML5-capable browser should work, we officially support the following operating systems and browsers.

BROWSER	SUPPORTED OS	NOTES
Microsoft Edge	Windows	
Apple Safari	macOS	
Mozilla Firefox	Windows, macOS, Linux	Version 55 or later
Google Chrome	Windows, macOS, Linux, Chrome OS	

Access remote resources feed

In a browser, navigate to the Azure Resource Manager-integrated version of the Azure Virtual Desktop web client at https://rdweb.wvd.microsoft.com/arm/webclient and sign in with your user account.

NOTE

If you're using Azure Virtual Desktop (classic) without Azure Resource Manager integration, connect to your resources at https://rdweb.wvd.microsoft.com/webclient instead.

If you're using the US Gov portal, use https://rdweb.wvd.azure.us/arm/webclient/index.html.

To connect to the Azure China portal, use https://rdweb.wvd.azure.cn/arm/webclient/index.html.

NOTE

If you've already signed in with a different Azure Active Directory account than the one you want to use for Azure Virtual Desktop, you should either sign out or use a private browser window.

After signing in, you should now see a list of resources. You can launch resources by selecting them like you would a normal app in the **All Resources** tab.

Next steps

To learn more about how to use the web client, check out Get started with the Web client.

Connect to Azure Virtual Desktop with the Android client

12/6/2021 • 2 minutes to read • Edit Online

Applies to: Android 4.1 and later, Chromebooks with ChromeOS 53 and later.

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can access Azure Virtual Desktop resources from your Android device with our downloadable client. You can also use the Android client on Chromebook devices that support the Google Play Store. This guide will tell you how to set up the Android client.

Install the Android client

To get started, download and install the client on your Android device.

Subscribe to a feed

Subscribe to the feed provided by your admin to get the list of managed resources you can access on your Android device.

To subscribe to a feed:

1. In the Connection Center, tap **+**, and then tap **Workspaces**.
2. Enter the feed URL into the **Feed URL** field. The feed URL can be either a URL or an email address.
   - If you use a URL, use the one your admin gave you, normally https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery.
   - To use email, enter your email address. The client will search for a URL associated with your email address if your admin configured the server that way.
   - To connect through the US Gov portal, use https://rdweb.wvd.azure.us/api/arm/feeddiscovery.
3. Tap **NEXT**.
4. Provide your credentials when prompted.
   - For **User name**, give the user name with permission to access resources.
   - For **Password**, give the password associated with the user name.
   - You may also be prompted to provide additional factors if your admin configured authentication that way.

After subscribing, the Connection Center should display the remote resources.

Once subscribed to a feed, the feed's content will update automatically on a regular basis. Resources may be added, changed, or removed based on changes made by your administrator.

Next steps

To learn more about how to use the Android client, check out Get started with the Android client.

Connect to Azure Virtual Desktop with the macOS client

12/6/2021 • 2 minutes to read • Edit Online

Applies to: macOS 10.12 or later

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can access Azure Virtual Desktop resources from your macOS devices with our downloadable client. This guide will tell you how to set up the client.

Install the client

To get started, download and install the client on your macOS device.

Subscribe to a feed

Subscribe to the feed your admin gave you to get the list of managed resources available to you on your macOS device.

To subscribe to a feed:

1. Select **Add Workspace** on the main page to connect to the service and retrieve your resources.
2. Enter the Feed URL. This can be a URL or email address:
   - If you use a URL, use the one your admin gave you. Normally, the URL is https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery.
   - To use email, enter your email address. This tells the client to search for a URL associated with your email address if your admin configured the server that way.
   - To connect through the US Gov portal, use https://rdweb.wvd.azure.us/api/arm/feeddiscovery.
3. Select **Add**.
4. Sign in with your user account when prompted.

After you've signed in, you should see a list of available resources.

Once you've subscribed to a feed, the feed's content will update automatically on a regular basis. Resources may be added, changed, or removed based on changes made by your administrator.

Next steps

To learn more about the macOS client, check out the Get started with the macOS client documentation.

Connect to Azure Virtual Desktop with the iOS client

12/6/2021 • 2 minutes to read • Edit Online

Applies to: iOS 13.0 or later. Compatible with iPhone, iPad, and iPod touch.

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can access Azure Virtual Desktop resources from your iOS device with our downloadable client. This guide will tell you how to set up the iOS client.

Install the iOS client

To get started, download and install the client on your iOS device.

Subscribe to a feed

Subscribe to the feed provided by your admin to get the list of managed resources you can access on your iOS device.

To subscribe to a feed:

1. In the Connection Center, tap **+**, and then tap **Add Workspace**.
2. Enter the feed URL into the **Feed URL** field. The feed URL can be either a URL or an email address.
   - If you use a URL, use the one your admin gave you. Normally, the URL is https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery.
   - To use email, enter your email address. This tells the client to search for a URL associated with your email address if your admin configured the server that way.
   - To connect through the US Gov portal, use https://rdweb.wvd.azure.us/api/arm/feeddiscovery.
3. Tap **Next**.
4. Provide your credentials when prompted.
   - For **User name**, give the user name with permission to access resources.
   - For **Password**, give the password associated with the user name.
   - You may also be prompted to provide additional factors if your admin configured authentication that way.
5. Tap **Save**.

After this, the Connection Center should display the remote resources.

Once subscribed to a feed, the feed's content will update automatically on a regular basis. Resources may be added, changed, or removed based on changes made by your administrator.

Next steps

To learn more about how to use the iOS client, check out the Get started with the iOS client documentation.

Connect with the Microsoft Store client

12/6/2021 • 2 minutes to read • Edit Online

> Applies to: Windows 10.

You can access Azure Virtual Desktop resources on devices with Windows 10.

Install the Microsoft Store client

You can install the client for the current user, which doesn't require admin rights. Alternatively, your admin can install and configure the client so that all users on the device can access it.

Once installed, the client can be launched from the Start menu by searching for Remote Desktop.

To get started, download and install the client from the Microsoft Store.

Subscribe to a workspace

Subscribe to the workspace provided by your admin to get the list of managed resources you can access on your PC.

To subscribe to a workspace:

1. In the Connection Center screen, tap **+Add**, then tap **Workspaces**.
2. Enter the Workspace URL into the Workspace URL field provided by your admin. The workspace URL can be either a URL or an email address.
   - If you're using a Workspace URL, use the URL your admin gave you.
   - If you're connecting from Azure Virtual Desktop, use one of the following URLs depending on which version of the service you're using:
     - Azure Virtual Desktop (classic): `https://rdweb.wvd.microsoft.com/api/feeddiscovery/webfeeddiscovery.aspx`.
     - Azure Virtual Desktop: `https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery`.
3. Tap **Subscribe**.
4. Provide your credentials when prompted.
5. After subscribing, the workspaces should be displayed in the Connection Center.

Workspaces may be added, changed, or removed based on changes made by your administrator.

Next steps

To learn more about how to use the Microsoft Store client, check out Get started with the Microsoft Store client.

Thin client support

12/6/2021 • 2 minutes to read • Edit Online

You can access Azure Virtual Desktop resources from your thin client devices with the web client or the following supported clients, provided by our partners. We're working with a number of partners to enable supported Azure Virtual Desktop clients on other platforms.

Connect with your thin client device

The following partners have approved Azure Virtual Desktop clients.

PARTNER	PARTNER DOCUMENTATION	PARTNER SUPPORT
10ZiG	10ZiG client documentation	10ZiG support
Dell	Dell client documentation	Dell support
HP	HP client documentation	HP support
IGEL	IGEL client documentation	IGEL support
NComputing	NComputing client documentation	NComputing support
Stratodesk	Stratodesk client documentation	Stratodesk support

Next steps

Check out our documentation for the following clients:

- Windows Desktop client
- Web client
- Android client
- macOS client
- iOS client

Configure device redirections

12/6/2021 • 2 minutes to read • Edit Online

Configuring device redirections for your Azure Virtual Desktop environment allows you to use printers, USB devices, microphones and other peripheral devices in the remote session. Some device redirections require changes to both Remote Desktop Protocol (RDP) properties and Group Policy settings.

Supported device redirections

Each client supports different device redirections. Check out Compare the clients for the full list of supported device redirections for each client.

Customizing RDP properties for a host pool

To learn more about customizing RDP properties for a host pool using PowerShell or the Azure portal, check out RDP properties. For the full list of supported RDP properties, see Supported RDP file settings.

Setup device redirections

You can use the following RDP properties and Group Policy settings to configure device redirections.

Audio input (microphone) redirection

Set the following RDP property to configure audio input redirection:

- `audiocapturemode:i:1` enables audio input redirection.
- `audiocapturemode:i:0` disables audio input redirection.

Audio output (speaker) redirection

Set the following RDP property to configure audio output redirection:

- `audiomode:i:0` enables audio output redirection.
- `audiomode:i:1` or `audiomode:i:2` disable audio output redirection.

Camera redirection

Set the following RDP property to configure camera redirection:

- `camerastoredirect:s:*` redirects all cameras.
- `camerastoredirect:s:` disables camera redirection.

> **NOTE**
> Even if the `camerastoredirect:s:` property is disabled, local cameras may be redirected through the `devicestoredirect:s:` property. To fully disable camera redirection set `camerastoredirect:s:` and either set `devicestoredirect:s:` or define some subset of plug and play devices that does not include any camera.

You can also redirect specific cameras using a semicolon-delimited list of KSCATEGORY_VIDEO_CAMERA interfaces, such as `camerastoredirect:s:\?\usb#vid_0bda&pid_58b0&mi`.

Clipboard redirection

Set the following RDP property to configure clipboard redirection:

- `redirectclipboard:i:1` enables clipboard redirection.
- `redirectclipboard:i:0` disables clipboard redirection.

COM port redirections

Set the following RDP property to configure COM port redirection:

- `redirectcomports:i:1` enables COM port redirection.
- `redirectcomports:i:0` disables COM port redirection.

USB redirection

First, set the following RDP property to enable USB device redirection:

- `usbdevicestoredirect:s:*` enables USB device redirection.
- `usbdevicestoredirect:s:` disables USB device redirection.

Second, set the following Group Policy on the user's local device:

- Navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Connection Client** > **RemoteFX USB Device Redirection**.
- Select **Allows RDP redirection of other supported RemoteFX USB devices from this computer**.
- Select the **Enabled** option, and then select the **Administrators and Users in RemoteFX USB Redirection Access Rights** box.
- Select **OK**.

Plug and play device redirection

Set the following RDP property to configure plug and play device redirection:

- `devicestoredirect:s:*` enables redirection of all plug and play devices.
- `devicestoredirect:s:` disables redirection of plug and play devices.

You can also select specific plug and play devices using a semicolon-delimited list, such as `devicestoredirect:s:root\*PNP0F08`.

Local drive redirection

Set the following RDP property to configure local drive redirection:

- `drivestoredirect:s:*` enables redirection of all disk drives.
- `drivestoredirect:s:` disables local drive redirection.

You can also select specific drives using a semicolon-delimited list, such as `drivestoredirect:s:C:;E:;`.

To enable web client file transfer, set `drivestoredirect:s:*`. If you set any other value for this RDP property, web client file transfer will be disabled.

Printer redirection

Set the following RDP property to configure printer redirection:

- `redirectprinters:i:1` enables printer redirection.
- `redirectprinters:i:0` disables printer redirection.

Smart card redirection

Set the following RDP property to configure smart card redirection:

- `redirectsmartcards:i:1` enables smart card redirection.
- `redirectsmartcards:i:0` disables smart card redirection.

Set up the PowerShell module for Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager integration.

The Azure Virtual Desktop PowerShell module is integrated into the Azure PowerShell module. This article will tell you how to set up the PowerShell module so you can run cmdlets for Azure Virtual Desktop.

Set up your PowerShell environment

To get started with using the module, first install the latest version of PowerShell Core. Azure Virtual Desktop cmdlets currently only work with PowerShell Core.

Next, you'll need to install the DesktopVirtualization module to use in your PowerShell session.

Run the following PowerShell cmdlet in elevated mode to install the module:

```
Install-Module -Name Az.DesktopVirtualization
```

NOTE

If this cmdlet doesn't work, try running it again with elevated permissions.

Next, run the following cmdlet to connect to Azure:

```
Connect-AzAccount
```

IMPORTANT

If you're connecting to the US Gov portal, run this cmdlet instead:

```
Connect-AzAccount -EnvironmentName AzureUSGovernment
```

To connect to the Azure China portal, run this cmdlet:

```
Connect-AzAccount -EnvironmentName AzureChinaCloud
```

Signing into your Azure account requires a code that's generated when you run the Connect cmdlet. To sign in, go to https://microsoft.com/devicelogin, enter the code, then sign in using your Azure admin credentials.

```
Account SubscriptionName TenantId Environment

------- ---------------- -------- -----------

Youradminupn subscriptionname AzureADTenantID AzureCloud
```

This will sign you directly into the subscription that is default for your admin credentials.

Change the default subscription

If you want to change the default subscription after you've signed in, run this cmdlet:

```
Select-AzSubscription -Subscription <preferredsubscriptionname>
```

You can also select one from a list using the Out-GridView cmdlet:

```
Get-AzSubscription | Out-GridView -PassThru | Select-AzSubscription
```

When you select a new subscription to use, you don't need to specify that subscription's ID in cmdlets you run afterwards. For example, the following cmdlet retrieves a specific session host without needing the subscription ID:

```
Get-AzWvdSessionHost -HostPoolName <hostpoolname> -Name <sessionhostname> -ResourceGroupName
<resourcegroupname>
```

You can also change subscriptions on a per-cmdlet basis by adding the desired subscription name as a parameter. The next cmdlet is the same as the previous example, except with the subscription ID added as a parameter to change which subscription the cmdlet uses.

```
Get-AzWvdSessionHost -HostPoolName <hostpoolname> -Name <sessionhostname> -ResourceGroupName
<resourcegroupname> -SubscriptionId <subscriptionGUID>
```

Get locations

The location parameter is mandatory for all **New-AzWVD** cmdlets that create new objects.

Run the following cmdlet to get a list of locations your subscription supports:

```
Get-AzLocation
```

The output for **Get-AzLocation** will look like this:

```
Location : eastasia

DisplayName : East Asia

Providers : {Microsoft.RecoveryServices, Microsoft.ManagedIdentity,
Microsoft.SqlVirtualMachine, microsoft.insightsΓÇª}

Location : southeastasia

DisplayName : Southeast Asia

Providers : {Microsoft.RecoveryServices, Microsoft.ManagedIdentity,
Microsoft.SqlVirtualMachine, microsoft.insightsΓÇª}

Location : centralus

DisplayName : Central US

Providers : {Microsoft.RecoveryServices, Microsoft.DesktopVirtualization,
Microsoft.ManagedIdentity, Microsoft.SqlVirtualMachineΓÇª}

Location : eastus

DisplayName : East US

Providers : {Microsoft.RecoveryServices, Microsoft.DesktopVirtualization,
Microsoft.ManagedIdentity, Microsoft.SqlVirtualMachineΓÇª}
```

Once you know your account's location, you can use it in a cmdlet. For example, here's a cmdlet that creates a host pool in the "southeastasia" location:

```
New-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -WorkspaceName <workspacename>
-Location “southeastasia”
```

Next steps

Now that you've set up your PowerShell module, you can run cmdlets to do all sorts of things in Azure Virtual Desktop. Here are some of the places you can use your module:

- Run through our Azure Virtual Desktop tutorials to set up your very own Azure Virtual Desktop environment.
- Create a host pool with PowerShell
- Configure the Azure Virtual Desktop load-balancing method
- Configure the personal desktop host pool assignment type
- And much more!

If you run into any issues, check out our PowerShell troubleshooting article for help.

Create an Azure Virtual Desktop host pool with PowerShell or the Azure CLI

12/6/2021 • 6 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Host pools are a collection of one or more identical virtual machines within Azure Virtual Desktop tenant environments. Each host pool can be associated with multiple RemoteApp groups, one desktop app group, and multiple session hosts.

Create a host pool

- Azure PowerShell
- Azure CLI

If you haven't already done so, follow the instructions in Set up the PowerShell module.

Run the following cmdlet to sign in to the Azure Virtual Desktop environment:

```
New-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -WorkspaceName <workspacename>
-HostPoolType <Pooled|Personal> -LoadBalancerType <BreadthFirst|DepthFirst|Persistent> -Location <region> -
DesktopAppGroupName <appgroupname>
```

This cmdlet will create the host pool, workspace and desktop app group. Additionally, it will register the desktop app group to the workspace. You can either create a workspace with this cmdlet or use an existing workspace.

Run the next cmdlet to create a registration token to authorize a session host to join the host pool and save it to a new file on your local computer. You can specify how long the registration token is valid by using the *-ExpirationTime* parameter.

NOTE

The token's expiration date can be no less than an hour and no more than one month. If you set *-ExpirationTime* outside of that limit, the cmdlet won't create the token.

```
New-AzWvdRegistrationInfo -ResourceGroupName <resourcegroupname> -HostPoolName <hostpoolname> -
ExpirationTime $((get-date).ToUniversalTime().AddDays(1).ToString('yyyy-MM-ddTHH:mm:ss.fffffffZ'))
```

For example, if you want to create a token that expires in two hours, run this cmdlet:

```
New-AzWvdRegistrationInfo -ResourceGroupName <resourcegroupname> -HostPoolName <hostpoolname> -
ExpirationTime $((get-date).ToUniversalTime().AddHours(2).ToString('yyyy-MM-ddTHH:mm:ss.fffffffZ'))
```

After that, run this cmdlet to add Azure Active Directory users to the default desktop app group for the host

pool.

```
New-AzRoleAssignment -SignInName <userupn> -RoleDefinitionName "Desktop Virtualization User" -ResourceName
<hostpoolname+"-DAG"> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Run this next cmdlet to add Azure Active Directory user groups to the default desktop app group for the host pool:

```
New-AzRoleAssignment -ObjectId <usergroupobjectid> -RoleDefinitionName "Desktop Virtualization User" -
ResourceName <hostpoolname+"-DAG"> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Run the following cmdlet to export the registration token to a variable, which you will use later in Register the virtual machines to the Azure Virtual Desktop host pool.

```
$token = Get-AzWvdRegistrationInfo -ResourceGroupName <resourcegroupname> -HostPoolName <hostpoolname>
```

Create virtual machines for the host pool

Now you can create an Azure virtual machine that can be joined to your Azure Virtual Desktop host pool.

You can create a virtual machine in multiple ways:

- Create a virtual machine from an Azure Gallery image
- Create a virtual machine from a managed image
- Create a virtual machine from an unmanaged image

> **NOTE**
> If you're deploying a virtual machine using Windows 7 as the host OS, the creation and deployment process will be a little different. For more details, see Deploy a Windows 7 virtual machine on Azure Virtual Desktop.

After you've created your session host virtual machines, apply a Windows license to a session host VM to run your Windows or Windows Server virtual machines without paying for another license.

Prepare the virtual machines for Azure Virtual Desktop agent installations

You need to do the following things to prepare your virtual machines before you can install the Azure Virtual Desktop agents and register the virtual machines to your Azure Virtual Desktop host pool:

- You must domain-join the machine. This allows incoming Azure Virtual Desktop users to be mapped from their Azure Active Directory account to their Active Directory account and be successfully allowed access to the virtual machine.
- You must install the Remote Desktop Session Host (RDSH) role if the virtual machine is running a Windows Server OS. The RDSH role allows the Azure Virtual Desktop agents to install properly.

To successfully domain-join, do the following things on each virtual machine:

1. Connect to the virtual machine with the credentials you provided when creating the virtual machine.
2. On the virtual machine, launch **Control Panel** and select **System**.

3. Select **Computer name**, select **Change settings**, and then select **Change...**
4. Select **Domain** and then enter the Active Directory domain on the virtual network.
5. Authenticate with a domain account that has privileges to domain-join machines.

> **NOTE**
> If you're joining your VMs to an Azure Active Directory Domain Services (Azure AD DS) environment, ensure that your domain join user is also a member of the AAD DC Administrators group.

> **IMPORTANT**
> We recommend that you don't enable any policies or configurations that disable Windows Installer. If you disable Windows Installer, the service won't be able to install agent updates on your session hosts, and your session hosts won't function properly.

Register the virtual machines to the Azure Virtual Desktop host pool

Registering the virtual machines to a Azure Virtual Desktop host pool is as simple as installing the Azure Virtual Desktop agents.

To register the Azure Virtual Desktop agents, do the following on each virtual machine:

1. Connect to the virtual machine with the credentials you provided when creating the virtual machine.
2. Download and install the Azure Virtual Desktop Agent.
   - Download the Azure Virtual Desktop Agent.
   - Run the installer. When the installer asks you for the registration token, enter the value you got from the **Get-AzWvdRegistrationInfo** cmdlet.
3. Download and install the Azure Virtual Desktop Agent Bootloader.
   - Download the Azure Virtual Desktop Agent Bootloader.
   - Run the installer.

> **IMPORTANT**
> To help secure your Azure Virtual Desktop environment in Azure, we recommend you don't open inbound port 3389 on your VMs. Azure Virtual Desktop doesn't require an open inbound port 3389 for users to access the host pool's VMs. If you must open port 3389 for troubleshooting purposes, we recommend you use just-in-time VM access. We also recommend you don't assign your VMs to a public IP.

Update the agent

You'll need to update the agent if you're in one of the following situations:

- You want to migrate a previously registered session host to a new host pool
- The session host doesn't appear in your host pool after an update

To update the agent:

1. Sign in to the VM as an administrator.
2. Go to **Services**, then stop the **Rdagent** and **Remote Desktop Agent Loader** processes.
3. Next, find the agent and bootloader MSIs. They'll either be located in the **C:\DeployAgent** folder or

whichever location you saved it to when you installed it.

4. Find the following files and uninstall them:

   - Microsoft.RDInfra.RDAgent.Installer-x64-ver x.x.x
   - Microsoft.RDInfra.RDAgentBootLoader.Installer-x64

   To uninstall these files, right-click on each file name, then select **Uninstall**.

5. Optionally, you can also remove the following registry settings:

   - Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\RDInfraAgent
   - Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\RDAgentBootLoader

6. Once you've uninstalled these items, this should remove all associations with the old host pool. If you want to reregister this host to the service, follow the instructions in Register the virtual machines to the Azure Virtual Desktop host pool.

Next steps

Now that you've made a host pool, you can populate it with RemoteApps. To learn more about how to manage apps in Azure Virtual Desktop, see the Manage app groups tutorial.

Manage app groups tutorial

Deploy Azure AD-joined virtual machines in Azure Virtual Desktop

12/6/2021 • 5 minutes to read • Edit Online

This article will walk you through the process of deploying and accessing Azure Active Directory joined virtual machines in Azure Virtual Desktop. Azure AD-joined VMs remove the need to have line-of-sight from the VM to an on-premises or virtualized Active Directory Domain Controller (DC) or to deploy Azure AD Domain services (Azure AD DS). In some cases, it can remove the need for a DC entirely, simplifying the deployment and management of the environment. These VMs can also be automatically enrolled in Intune for ease of management.

Supported configurations

The following configurations are currently supported with Azure AD-joined VMs:

- Personal desktops with local user profiles.
- Pooled desktops used as a jump box. In this configuration, users first access the Azure Virtual Desktop VM before connecting to a different PC on the network. Users shouldn't save data on the VM.
- Pooled desktops or apps where users don't need to save data on the VM. For example, for applications that save data online or connect to a remote database.

User accounts can be cloud-only or hybrid users from the same Azure AD tenant.

Known limitations

The following known limitations may impact access to your on-premises or Active Directory domain-joined resources and should be considered when deciding whether Azure AD-joined VMs are right for your environment. We currently recommend Azure AD-joined VMs for scenarios where users only need access to cloud-based resources or Azure AD-based authentication.

- Azure Virtual Desktop (classic) doesn't support Azure AD-joined VMs.
- Azure AD-joined VMs don't currently support external users.
- Azure AD-joined VMs only supports local user profiles at this time.
- Azure AD-joined VMs can't access Azure Files file shares for FSLogix or MSIX app attach. You'll need Kerberos authentication to access either of these features.
- The Windows Store client doesn't currently support Azure AD-joined VMs.
- Azure Virtual Desktop doesn't currently support single sign-on for Azure AD-joined VMs.

Deploy Azure AD-joined VMs

You can deploy Azure AD-joined VMs directly from the Azure portal when you create a new host pool or expand an existing host pool. To deploy an Azure AD-joined VM, open the **Virtual Machines** tab, then select whether to join the VM to Active Directory or Azure Active Directory. Selecting **Azure Active Directory** gives you the option to enroll VMs with Intune automatically, which lets you easily manage Windows 10 Enterprise and Windows 10 Enterprise multi-session VMs. Keep in mind that the Azure Active Directory option will only join VMs to the same Azure AD tenant as the subscription you're in.

NOTE

- Host pools should only contain VMs of the same domain join type. For example, AD-joined VMs should only be with other AD VMs, and vice-versa.
- The host pool VMs must be Windows 10 single-session or multi-session, version 2004 or later.
- Managing Azure Virtual Desktop session hosts using Microsoft Endpoint Manager (Intune) is currently only supported in the Azure Public cloud.

Assign user access to host pools

After you've created your host pool, you must assign users access to let them access their resources. To grant access to resources, add each user to the app group. Follow the instructions in Manage app groups to assign user access to apps and desktops. We recommend that you use user groups instead of individual users wherever possible.

For Azure AD-joined VMs, you'll need to do two extra things on top of the requirements for Active Directory or Azure Active Directory Domain Services-based deployments:

- Assign your users the **Virtual Machine User Login** role so they can sign in to the VMs.
- Assign administrators who need local administrative privileges the **Virtual Machine Administrator Login** role.

To grant users access to Azure AD-joined VMs, you must configure role assignments for the VM. You can assign the **Virtual Machine User Login** or **Virtual Machine Administrator Login** role either on the VMs, the resource group containing the VMs, or the subscription. We recommend assigning the Virtual Machine User Login role to the same user group you used for the app group at the resource group level to make it apply to all the VMs in the host pool.

Access Azure AD-joined VMs

This section explains how to access Azure AD-joined VMs from different Azure Virtual Desktop clients.

Connect using the Windows Desktop client

The default configuration supports connections from Windows 10 using the Windows Desktop client. You can use your credentials, smart card, Windows Hello for Business certificate trust or Windows Hello for Business key trust with certificates to sign in to the session host. However, to access the session host, your local PC must meet one of the following conditions:

- The local PC is Azure AD-joined to the same Azure AD tenant as the session host
- The local PC is hybrid Azure AD-joined to the same Azure AD tenant as the session host
- The local PC is running Windows 10, version 2004 and later, and is Azure AD registered to the same Azure AD tenant as the session host

To enable access from Windows devices not joined to Azure AD, add **targetisaadjoined:i:1** as a custom RDP property to the host pool. These connections are restricted to entering user name and password credentials when signing in to the session host.

Connect using the other clients

To access Azure AD-joined VMs using the web, Android, macOS and iOS clients, you must add **targetisaadjoined:i:1** as a custom RDP property to the host pool. These connections are restricted to entering user name and password credentials when signing in to the session host.

Enabling MFA for Azure AD joined VMs

You can enable multifactor authentication for Azure AD-joined VMs by setting a Conditional Access policy on the Azure Virtual Desktop app. For connections to succeed, you must disable the legacy per-user multifactor

authentication. If you don't want to restrict signing in to strong authentication methods like Windows Hello for Business, you'll also need to exclude the Azure Windows VM Sign-In app from your Conditional Access policy.

User profiles

You can use FSLogix profile containers with Azure AD-joined VMs when you store them on Azure Files. For more information, see Create a profile container with Azure Files and Azure AD.

Next steps

Now that you've deployed some Azure AD joined VMs, you can sign in to a supported Azure Virtual Desktop client to test it as part of a user session. If you want to learn how to connect to a session, check out these articles:

- Connect with the Windows Desktop client
- Connect with the web client
- Troubleshoot connections to Azure AD-joined VMs

Deploy a Windows 7 virtual machine on Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

The process to deploy a Windows 7 virtual machine (VM) on Azure Virtual Desktop is slightly different than for VMs running later versions of Windows. This guide will tell you how to deploy Windows 7.

Prerequisites

Before you start, follow the instructions in Create a host pool with PowerShell to create a host pool. If you're using the portal, follow the instructions in steps 1 through 9 of Create a host pool using the Azure portal. After that, select **Review + Create** to create an empty host pool.

Configure a Windows 7 virtual machine

Once you've done the prerequisites, you're ready to configure your Windows 7 VM for deployment on Azure Virtual Desktop.

To set up a Windows 7 VM on Azure Virtual Desktop:

1. Sign in to the Azure portal and either search for the Windows 7 Enterprise image or upload your own customized Windows 7 Enterprise (x64) image.
2. Deploy one or multiple virtual machines with Windows 7 Enterprise as its host operating system. Make sure the virtual machines allow Remote Desktop Protocol (RDP) (the TCP/3389 port).
3. Connect to the Windows 7 Enterprise host using the RDP and authenticate with the credentials you defined while configuring your deployment.
4. Add the account you used while connecting to the host with RDP to the "Remote Desktop User" group. If you don't add the account, you might not be able to connect to the VM after you join it to your Active Directory domain.
5. Go to Windows Update on your VM.
6. Install all Windows Updates in the Important category.
7. Install all Windows Updates in the Optional category (excluding language packs). This process installs the Remote Desktop Protocol 8.0 update (KB2592687) that you need to complete these instructions.
8. Open the Local Group Policy Editor and navigate to **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Remote Session Environment**.
9. Enable the Remote Desktop Protocol 8.0 policy.
10. Join this VM to your Active Directory domain.

11. Restart the virtual machine by running the following command:

    ```
    shutdown /r /t 0
    ```

12. Follow the instructions here to get a registration token.

    - If you'd rather use the Azure portal, you can also go to the Overview page of the host pool you want to add the VM to and create a token there.

13. Download the Azure Virtual Desktop Agent for Windows 7.

14. Download the Azure Virtual Desktop Agent Manager for Windows 7.

15. Open the Azure Virtual Desktop Agent installer and follow the instructions. When prompted, give the registration key you created in step 12.

16. Open the Azure Virtual Desktop Agent Manager and follow the instructions.

17. Optionally, block the TCP/3389 port to remove direct Remote Desktop Protocol access to the VM.

18. Optionally, confirm that your .NET framework is at least version 4.7.2. Updating your framework is especially important if you're creating a custom image.

Next steps

Your Azure Virtual Desktop deployment is now ready to use. Download the latest version of the Azure Virtual Desktop client to get started.

For a list of known issues and troubleshooting instructions for Windows 7 on Azure Virtual Desktop, see our troubleshooting article at Troubleshoot Windows 7 virtual machines in Azure Virtual Desktop.

Configure graphics processing unit (GPU) acceleration for Azure Virtual Desktop

12/6/2021 • 6 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Azure Virtual Desktop supports GPU-accelerated rendering and encoding for improved app performance and scalability. GPU acceleration is particularly crucial for graphics-intensive apps.

Follow the instructions in this article to create a GPU optimized Azure virtual machine, add it to your host pool, and configure it to use GPU acceleration for rendering and encoding. This article assumes you already have a Azure Virtual Desktop tenant configured.

Select an appropriate GPU optimized Azure virtual machine size

Select one of Azure's NV-series, NVv3-series, or NVv4-series VM sizes. These are tailored for app and desktop virtualization and enable most apps and the Windows user interface to be GPU accelerated. The right choice for your host pool depends on a number of factors, including your particular app workloads, desired quality of user experience, and cost. In general, larger and more capable GPUs offer a better user experience at a given user density, while smaller and fractional-GPU sizes allow more fine-grained control over cost and quality. Consider NV series VM retirement when selecting VM, details on NV retirement

NOTE

Azure's NC, NCv2, NCv3, ND, and NDv2 series VMs are generally not appropriate for Azure Virtual Desktop session hosts. These VMs are tailored for specialized, high-performance compute or machine learning tools, such as those built with NVIDIA CUDA. They do not support GPU acceleration for most apps or the Windows user interface.

Create a host pool, provision your virtual machine, and configure an app group

Create a new host pool using a VM of the size you selected. For instructions, see Tutorial: Create a host pool with the Azure portal.

Azure Virtual Desktop supports GPU-accelerated rendering and encoding in the following operating systems:

- Windows 10 version 1511 or newer
- Windows Server 2016 or newer

NOTE

Multi-session OS is not specifically listed however NV instances GRID license supports 25 concurrent users, see NV-series

You must also configure an app group, or use the default desktop app group (named "Desktop Application Group") that's automatically created when you create a new host pool. For instructions, see Tutorial: Manage app

groups for Azure Virtual Desktop.

Install supported graphics drivers in your virtual machine

To take advantage of the GPU capabilities of Azure N-series VMs in Azure Virtual Desktop, you must install the appropriate graphics drivers. Follow the instructions at Supported operating systems and drivers to install drivers. Only drivers distributed by Azure are supported.

- For Azure NV-series or NVv3-series VMs, only NVIDIA GRID drivers, and not NVIDIA CUDA drivers, support GPU acceleration for most apps and the Windows user interface. If you choose to install drivers manually, be sure to install GRID drivers. If you choose to install drivers using the Azure VM extension, GRID drivers will automatically be installed for these VM sizes.
- For Azure NVv4-series VMs, install the AMD drivers provided by Azure. You may install them automatically using the Azure VM extension, or you may install them manually.

After driver installation, a VM restart is required. Use the verification steps in the above instructions to confirm that graphics drivers were successfully installed.

Configure GPU-accelerated app rendering

By default, apps and desktops running in multi-session configurations are rendered with the CPU and do not leverage available GPUs for rendering. Configure Group Policy for the session host to enable GPU-accelerated rendering:

1. Connect to the desktop of the VM using an account with local administrator privileges.
2. Open the Start menu and type "gpedit.msc" to open the Group Policy Editor.
3. Navigate the tree to **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Remote Session Environment**.
4. Select policy **Use hardware graphics adapters for all Remote Desktop Services sessions** and set this policy to **Enabled** to enable GPU rendering in the remote session.

Configure GPU-accelerated frame encoding

Remote Desktop encodes all graphics rendered by apps and desktops (whether rendered with GPU or with CPU) for transmission to Remote Desktop clients. When part of the screen is frequently updated, this part of the screen is encoded with a video codec (H.264/AVC). By default, Remote Desktop does not leverage available GPUs for this encoding. Configure Group Policy for the session host to enable GPU-accelerated frame encoding. Continuing the steps above:

> **NOTE**
> GPU-accelerated frame encoding is not available in NVv4-series VMs.

1. Select policy **Configure H.264/AVC hardware encoding for Remote Desktop connections** and set this policy to **Enabled** to enable hardware encoding for AVC/H.264 in the remote session.

   > **NOTE**
   > In Windows Server 2016, set option **Prefer AVC Hardware Encoding** to **Always attempt**.

2. Now that the group policies have been edited, force a group policy update. Open the Command Prompt and type:

```
gpupdate.exe /force
```

3. Sign out from the Remote Desktop session.

Configure fullscreen video encoding

> **NOTE**
> Fullscreen video encoding can be enabled even without a GPU present.

If you often use applications that produce a high-frame rate content, such as 3D modeling, CAD/CAM and video applications, you may choose to enable a fullscreen video encoding for a remote session. Fullscreen video profile provides a higher frame rate and better user experience for such applications at expense of network bandwidth and both session host and client resources. It is recommended to use GPU-accelerated frame encoding for a full-screen video encoding. Configure Group Policy for the session host to enable fullscreen video encoding. Continuing the steps above:

1. Select policy **Prioritize H.264/AVC 444 Graphics mode for Remote Desktop connections** and set this policy to **Enabled** to force H.264/AVC 444 codec in the remote session.
2. Now that the group policies have been edited, force a group policy update. Open the Command Prompt and type:

   ```
   gpupdate.exe /force
   ```

3. Sign out from the Remote Desktop session.

Verify GPU-accelerated app rendering

To verify that apps are using the GPU for rendering, try any of the following:

- For Azure VMs with a NVIDIA GPU, use the `nvidia-smi` utility as described in Verify driver installation to check for GPU utilization when running your apps.
- On supported operating system versions, you can use the Task Manager to check for GPU utilization. Select the GPU in the "Performance" tab to see whether apps are utilizing the GPU.

Verify GPU-accelerated frame encoding

To verify that Remote Desktop is using GPU-accelerated encoding:

1. Connect to the desktop of the VM using Azure Virtual Desktop client.
2. Launch the Event Viewer and navigate to the following node: **Applications and Services Logs** > **Microsoft** > **Windows** > **RemoteDesktopServices-RdpCoreCDV** > **Operational**
3. To determine if GPU-accelerated encoding is used, look for event ID 170. If you see "AVC hardware encoder enabled: 1" then GPU encoding is used.

Verify fullscreen video encoding

To verify that Remote Desktop is using fullscreen video encoding:

1. Connect to the desktop of the VM using Azure Virtual Desktop client.
2. Launch the Event Viewer and navigate to the following node: **Applications and Services Logs** > **Microsoft** > **Windows** > **RemoteDesktopServices-RdpCoreCDV** > **Operational**

3. To determine if fullscreen video encoding is used, look for event ID 162. If you see "AVC Available: 1 Initial Profile: 2048" then AVC 444 is used.

Next steps

These instructions should have you up and running with GPU acceleration on one session host (one VM). Some additional considerations for enabling GPU acceleration across a larger host pool:

- Consider using a VM extension to simplify driver installation and updates across a number of VMs. Use the NVIDIA GPU Driver Extension for VMs with NVIDIA GPUs, and use the AMD GPU Driver Extension for VMs with AMD GPUs.
- Consider using Active Directory Group Policy to simplify group policy configuration across a number of VMs. For information about deploying Group Policy in the Active Directory domain, see Working with Group Policy Objects.

Expand an existing host pool with new session hosts

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

As you ramp up usage within your host pool, you may need to expand your existing host pool with new session hosts to handle the new load.

This article will tell you how you can expand an existing host pool with new session hosts.

What you need to expand the host pool

Before you start, make sure you've created a host pool and session host virtual machines (VMs) using one of the following methods:

- Azure portal
- Create a host pool with PowerShell

You'll also need the following information from when you first created the host pool and session host VMs:

- VM size, image, and name prefix
- Domain join administrator credentials
- Virtual network name and subnet name

Add virtual machines with the Azure portal

To expand your host pool by adding virtual machines:

1. Sign in to the Azure portal.
2. Search for and select **Azure Virtual Desktop**.
3. In the menu on the left side of the screen, select **Host pools**, then select the name of the host pool you want to add virtual machines to.
4. Select **Session hosts** from the menu on the left side of the screen.
5. Select **+Add** to start creating your host pool.
6. Ignore the the Basics tab and instead select the **VM details** tab. Here you can view and edit the details of the virtual machine (VM) you want to add to the host pool.
7. Select the resource group you want to create the VMs under, then select the region. You can choose the current region you're using or a new region.
8. Enter the number of session hosts you want to add to your host pool into **Number of VMs**. For example, if you're expanding your host pool by five hosts, enter **5**.

> **NOTE**
> Although it's possible to edit the image and prefix of the VMs, we don't recommended editing them if you have VMs with different images in the same host pool. Edit the image and prefix only if you plan on removing VMs with older images from the affected host pool.

9. For the **virtual network information**, select the virtual network and subnet to which you want the virtual machines to be joined to. You can select the same virtual network your existing machines currently use or choose a different one that's more suitable to the region you selected in step 7.
10. For the **Domain to join**, select if you want to join the virtual machines to Active Directory or Azure Active Directory. Selecting **Enroll the VM with Intune** automatically enrolls the virtual machines in Intune. All virtual machines in a host pool should be joined to the same domain or Azure AD tenant.
11. For the **AD domain join UPN**, enter an Active Directory domain username and password associated with the domain you selected. These credentials will be used to join the virtual machines to the Active Directory domain.

    > **NOTE**
    > Ensure your admin names comply with info given here. And that there is no MFA enabled on the account.

12. For the **Virtual Machine Administrator account**, enter the local administrator account information you want to use for all virtual machines.
13. Select the **Tags** tab if you have any tags that you want to group the virtual machines with. Otherwise, skip this tab.
14. Select the **Review + Create** tab. Review your choices, and if everything looks fine, select **Create**.

Next steps

Now that you've expanded your existing host pool, you can sign in to a Azure Virtual Desktop client to test them as part of a user session. You can connect to a session with any of the following clients:

- Connect with the Windows Desktop client
- Connect with the web client
- Connect with the Android client
- Connect with the macOS client
- Connect with the iOS client

Manage app groups using PowerShell or the Azure CLI

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

The default app group created for a new Azure Virtual Desktop host pool also publishes the full desktop. In addition, you can create one or more RemoteApp application groups for the host pool. Follow this tutorial to create a RemoteApp app group and publish individual **Start** menu apps.

In this tutorial, learn how to:

- Create a RemoteApp group.
- Grant access to RemoteApp programs.

Prerequisites

- Azure PowerShell
- Azure CLI

This article assumes you've followed the instructions in Set up the PowerShell module to set up your PowerShell module and sign in to your Azure account.

Create a RemoteApp group

- Azure PowerShell
- Azure CLI

To create a RemoteApp group with PowerShell:

1. Run the following PowerShell cmdlet to create a new empty RemoteApp app group.

```
New-AzWvdApplicationGroup -Name <appgroupname> -ResourceGroupName <resourcegroupname> -
ApplicationGroupType "RemoteApp" -HostPoolArmPath
'/subscriptions/SubscriptionId/resourcegroups/ResourceGroupName/providers/Microsoft.DesktopVirtualiza
tion/hostPools/HostPoolName'-Location <azureregion>
```

2. (Optional) To verify that the app group was created, you can run the following cmdlet to see a list of all app groups for the host pool.

```
Get-AzWvdApplicationGroup -Name <appgroupname> -ResourceGroupName <resourcegroupname>
```

3. Run the following cmdlet to get a list of **Start** menu apps on the host pool's virtual machine image. Write down the values for **FilePath**, **IconPath**, **IconIndex**, and other important information for the application that you want to publish.

```
Get-AzWvdStartMenuItem -ApplicationGroupName <appgroupname> -ResourceGroupName <resourcegroupname> |
Format-List | more
```

The output should show all the Start menu items in a format like this:

```
AppAlias            : access
CommandLineArgument :
FilePath            : C:\Program Files\Microsoft Office\root\Office16\MSACCESS.EXE
FriendlyName        :
IconIndex           : 0
IconPath            : C:\Program Files\Microsoft Office\Root\VFS\Windows\Installer\{90160000-000F-
0000-1000-0000000FF1CE}\accicons.exe
Id                  :
/subscriptions/resourcegroups/providers/Microsoft.DesktopVirtualization/applicationgroups/startmenuit
ems/Access
Name                : 0301RAG/Access
Type                : Microsoft.DesktopVirtualization/applicationgroups/startmenuitems

AppAlias            : charactermap
CommandLineArgument :
FilePath            : C:\windows\system32\charmap.exe
FriendlyName        :
IconIndex           : 0
IconPath            : C:\windows\system32\charmap.exe
Id                  :
/subscriptions/resourcegroups/providers/Microsoft.DesktopVirtualization/applicationgroups/startmenuit
ems/Character Map
Name                : 0301RAG/Character Map
Type                : Microsoft.DesktopVirtualization/applicationgroups/startmenuitems
```

4. Run the following cmdlet to install the application based on `AppAlias`. `AppAlias` becomes visible when you run the output from step 3.

   ```
   New-AzWvdApplication -AppAlias <appalias> -GroupName <appgroupname> -Name <remoteappname> -
   ResourceGroupName <resourcegroupname> -CommandLineSetting <DoNotAllow|Allow|Require>
   ```

5. (Optional) Run the following cmdlet to publish a new RemoteApp program to the application group created in step 1.

   ```
   New-AzWvdApplication -GroupName <appgroupname> -Name <remoteappname> -ResourceGroupName
   <resourcegroupname> -Filepath <filepath> -IconPath <iconpath> -IconIndex <iconindex> -
   CommandLineSetting <DoNotAllow|Allow|Require>
   ```

6. To verify that the app was published, run the following cmdlet.

   ```
   Get-AzWvdApplication -GroupName <appgroupname> -ResourceGroupName <resourcegroupname>
   ```

7. Repeat steps 1–5 for each application that you want to publish for this app group.

8. Run the following cmdlet to grant users access to the RemoteApp programs in the app group.

   ```
   New-AzRoleAssignment -SignInName <userupn> -RoleDefinitionName "Desktop Virtualization User" -
   ResourceName <appgroupname> -ResourceGroupName <resourcegroupname> -ResourceType
   'Microsoft.DesktopVirtualization/applicationGroups'
   ```

Next steps

If you came to this How-to guide from our tutorials, check out Create a host pool to validate service updates. You can use a validation host pool to monitor service updates before rolling them out to your production environment.

Delete a host pool

12/6/2021 • 2 minutes to read • Edit Online

All host pools created in Azure Virtual Desktop are attached to session hosts and app groups. To delete a host pool, you need to delete its associated app groups and session hosts. Deleting an app group is fairly simple, but deleting a session host is more complicated. When you delete a session host, you need to make sure it doesn't have any active user sessions. All user sessions on the session host should be logged off to prevent users from losing data.

- Portal
- Azure PowerShell
- Azure CLI

To delete a host pool in the Azure portal:

1. Sign in to the Azure portal.
2. Search for and select **Azure Virtual Desktop**.
3. Select **Host pools** in the menu on the left side of the page, then select the name of the host pool you want to delete.
4. On the menu on the left side of the page, select **Application groups**.
5. Select all application groups in the host pool you're going to delete, then select **Remove**.
6. Once you've removed the app groups, go to the menu on the left side of the page and select **Overview**.
7. Select **Remove**.
8. If there are session hosts in the host pool you're deleting, you'll see a message asking for your permission to continue. Select **Yes**.
9. The Azure portal will now remove all session hosts and delete the host pool. The VMs related to the session host won't be deleted and will remain in your subscription.

Next steps

To learn how to create a host pool, check out these articles:

- Create a host pool with the Azure portal
- Create a host pool with PowerShell

To learn how to configure host pool settings, check out these articles:

- Customize Remote Desktop Protocol properties for a host pool
- Configure the Azure Virtual Desktop load-balancing method
- Configure the personal desktop host pool assignment type

Create a profile container for a host pool using a file share

12/6/2021 • 3 minutes to read • Edit Online

The Azure Virtual Desktop service offers FSLogix profile containers as the recommended user profile solution. We don't recommend using the User Profile Disk (UPD) solution, which will be deprecated in future versions of Azure Virtual Desktop.

This article will tell you how to set up a FSLogix profile container share for a host pool using a virtual machine-based file share. We strongly recommend using Azure Files instead of file shares. For more FSLogix documentation, see the FSLogix site.

> **NOTE**
> If you're looking for comparison material about the different FSLogix Profile Container storage options on Azure, see Storage options for FSLogix profile containers.

Create a new virtual machine that will act as a file share

When creating the virtual machine, be sure to place it on either the same virtual network as the host pool virtual machines or on a virtual network that has connectivity to the host pool virtual machines. You can create a virtual machine in multiple ways:

- Create a virtual machine from an Azure Gallery image
- Create a virtual machine from a managed image
- Create a virtual machine from an unmanaged image

After creating the virtual machine, join it to the domain by doing the following things:

1. Connect to the virtual machine with the credentials you provided when creating the virtual machine.
2. On the virtual machine, launch **Control Panel** and select **System**.
3. Select **Computer name**, select **Change settings**, and then select **Change...**
4. Select **Domain** and then enter the Active Directory domain on the virtual network.
5. Authenticate with a domain account that has privileges to domain-join machines.

Prepare the virtual machine to act as a file share for user profiles

The following are general instructions about how to prepare a virtual machine to act as a file share for user profiles:

1. Add the Azure Virtual Desktop Active Directory users to an Active Directory security group. This security group will be used to authenticate the Azure Virtual Desktop users to the file share virtual machine you just created.
2. Connect to the file share virtual machine.
3. On the file share virtual machine, create a folder on the **C drive** that will be used as the profile share.
4. Right-click the new folder, select **Properties**, select **Sharing**, then select **Advanced sharing...**.
5. Select **Share this folder**, select **Permissions...**, then select **Add...**.
6. Search for the security group to which you added the Azure Virtual Desktop users, then make sure that

group has **Full Control**.

7. After adding the security group, right-click the folder, select **Properties**, select **Sharing**, then copy down the **Network Path** to use for later.

For more information about permissions, see the FSLogix documentation.

Configure the FSLogix profile container

To configure the virtual machines with the FSLogix software, do the following on each machine registered to the host pool:

1. Connect to the virtual machine with the credentials you provided when creating the virtual machine.
2. Launch an internet browser and navigate to this link to download the FSLogix agent.
3. Navigate to either \\Win32\Release or \\X64\Release in the .zip file and run **FSLogixAppsSetup** to install the FSLogix agent. To learn more about how to install FSLogix, see Download and install FSLogix.
4. Navigate to **Program Files** > **FSLogix** > **Apps** to confirm the agent installed.
5. From the start menu, run **RegEdit** as an administrator. Navigate to **Computer\HKEY_LOCAL_MACHINE\software\FSLogix**.
6. Create a key named **Profiles**.
7. Create the following values for the Profiles key:

NAME	TYPE	DATA/VALUE
Enabled	DWORD	1
VHDLocations	Multi-String Value	"Network path for file share"

IMPORTANT

To help secure your Azure Virtual Desktop environment in Azure, we recommend you don't open inbound port 3389 on your VMs. Azure Virtual Desktop doesn't require an open inbound port 3389 for users to access the host pool's VMs. If you must open port 3389 for troubleshooting purposes, we recommend you use just-in-time VM access.

Create a profile container with Azure NetApp Files and AD DS

12/6/2021 • 8 minutes to read • Edit Online

We recommend using FSLogix profile containers as a user profile solution for the Azure Virtual Desktop service. FSLogix profile containers store a complete user profile in a single container and are designed to roam profiles in non-persistent remote computing environments like Azure Virtual Desktop. When you sign in, the container dynamically attaches to the computing environment using a locally supported virtual hard disk (VHD) and Hyper-V virtual hard disk (VHDX). These advanced filter-driver technologies allow the user profile to be immediately available and appear in the system exactly like a local user profile. To learn more about FSLogix profile containers, see FSLogix profile containers and Azure files.

You can create FSLogix profile containers using Azure NetApp Files, an easy-to-use Azure native platform service that helps customers quickly and reliably provision enterprise-grade SMB volumes for their Azure Virtual Desktop environments. To learn more about Azure NetApp Files, see What is Azure NetApp Files?

This guide will show you how to set up an Azure NetApp Files account and create FSLogix profile containers in Azure Virtual Desktop.

This article assumes you already have host pools set up and grouped into one or more tenants in your Azure Virtual Desktop environment. To learn how to set up tenants, see Create a tenant in Azure Virtual Desktop and our Tech Community blog post.

The instructions in this guide are specifically for Azure Virtual Desktop users. If you're looking for more general guidance for how to set up Azure NetApp Files and create FSLogix profile containers outside of Azure Virtual Desktop, see the Set up Azure NetApp Files and create an NFS volume quickstart.

> **NOTE**
> This article doesn't cover best practices for securing access to the Azure NetApp Files share.

> **NOTE**
> If you're looking for comparison material about the different FSLogix Profile Container storage options on Azure, see Storage options for FSLogix profile containers.

Prerequisites

Before you can create an FSLogix profile container for a host pool, you must:

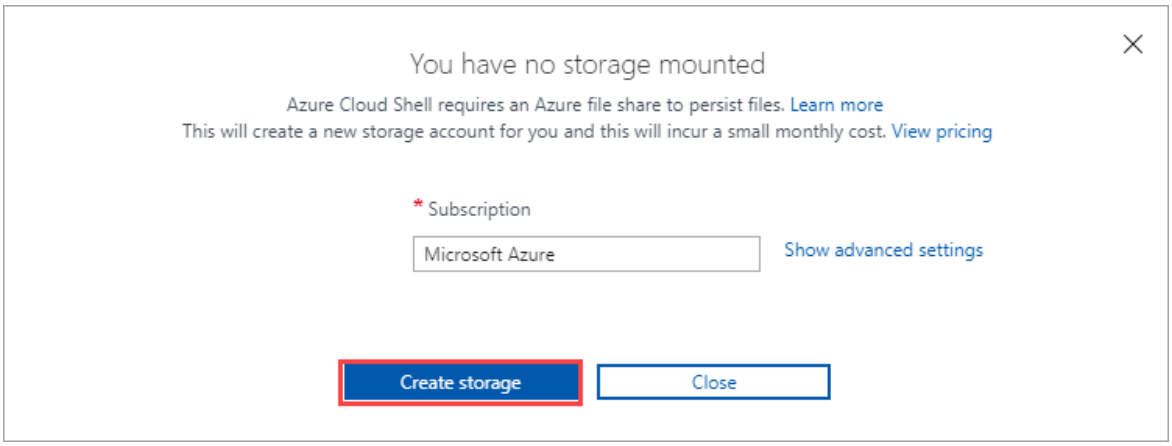
- Set up and configure Azure Virtual Desktop
- Provision a Azure Virtual Desktop host pool

Set up your Azure NetApp Files account

To get started, you need to set up an Azure NetApp Files account.

1. Sign in to the Azure portal. Make sure your account has contributor or administrator permissions.
2. Select the **Azure Cloud Shell icon** to the right of the search bar to open Azure Cloud Shell.

3. Once Azure Cloud Shell is open, select **PowerShell**.
4. If this is your first time using Azure Cloud Shell, create a storage account in the same subscription you keep your Azure NetApp Files and Azure Virtual Desktop.

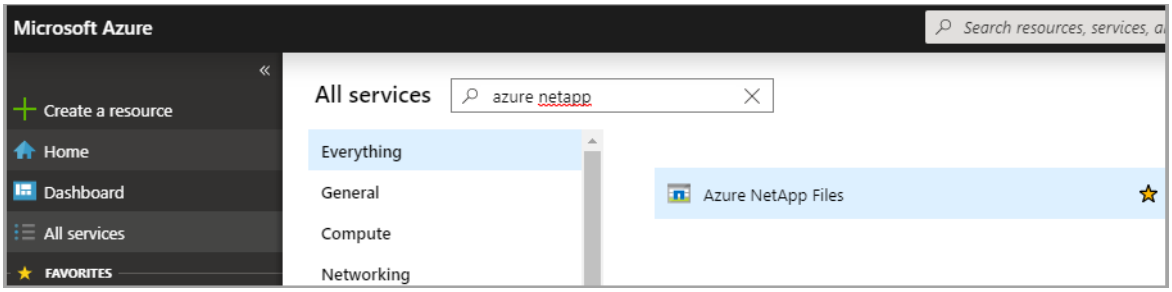


5. Once Azure Cloud Shell loads, run the following two cmdlets.

```
az account set --subscription <subscriptionID>
```

```
az provider register --namespace Microsoft.NetApp --wait
```

6. In the left side of the window, select **All services**. Enter **Azure NetApp Files** into the search box that appears at the top of the menu.



7. Select **Azure NetApp Files** in the search results, then select **Create**.
8. Select the **Add** button.
9. When the **New NetApp account** tab opens, enter the following values:
   - For **Name**, enter your NetApp account name.
   - For **Subscription**, select the subscription for the storage account you set up in step 4 from the drop-down menu.
   - For **Resource group**, either select an existing resource group from the drop-down menu or create a new one by selecting **Create new**.
   - For **Location**, select the region for your NetApp account from the drop-down menu. This region must be the same region as your session host VMs.

> **NOTE**
> Azure NetApp Files currently doesn't support mounting of a volume across regions.

10. When you're finished, select **Create** to create your NetApp account.

Create a capacity pool

Next, create a new capacity pool:

1. Go to the Azure NetApp Files menu and select your new account.
2. In your account menu, select **Capacity pools** under Storage service.
3. Select **Add pool**.
4. When the **New capacity pool** tab opens, enter the following values:
   - For **Name**, enter a name for the new capacity pool.
   - For **Service level**, select your desired value from the drop-down menu. We recommend **Premium** for most environments.

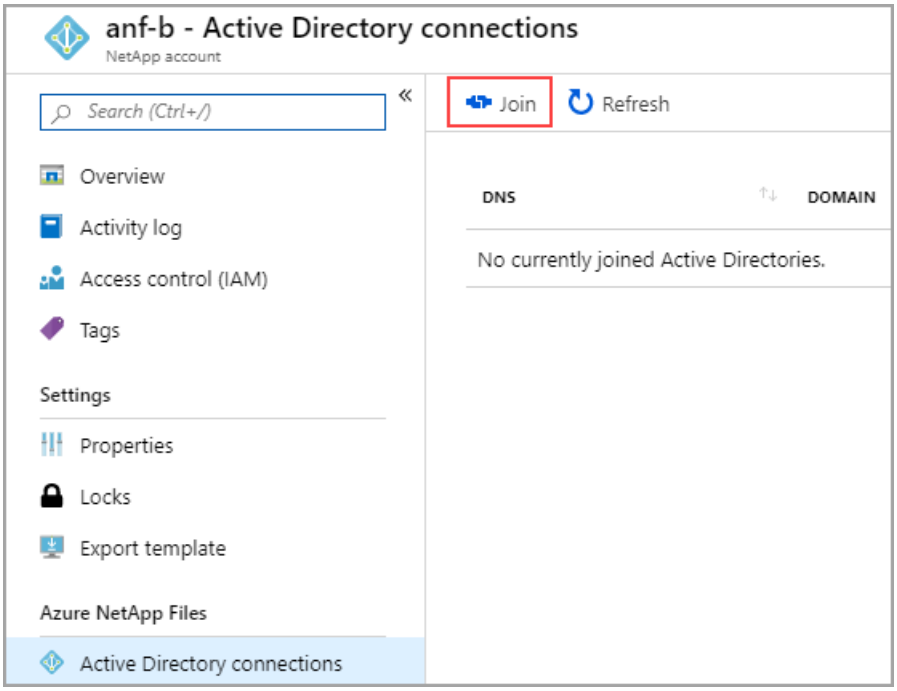
     > **NOTE**
     > The Premium setting provides the minimum throughput available for a Premium Service level, which is 256 MBps. You may need to adjust this throughput for a production environment. Final throughput is based on the relationship described in Throughput limits.

   - For **Size (TiB)**, enter the capacity pool size that best fits your needs. The minimum size is 4 TiB.
5. When you're finished, select **OK**.

Join an Active Directory connection

After that, you need to join an Active Directory connection.

1. Select **Active Directory connections** in the menu on the left side of the page, then select the **Join** button to open the **Join Active Directory** page.



2. Enter the following values in the **Join Active Directory** page to join a connection:
   - For **Primary DNS**, enter the IP address of the DNS server in your environment that can resolve the domain name.
   - For **Domain**, enter your fully qualified domain name (FQDN).
   - For **SMB Server (Computer Account) Prefix**, enter the string you want to append to the computer

account name.

- For **Username**, enter the name of the account with permissions to perform domain join.
- For **Password**, enter the account's password.

Create a new volume

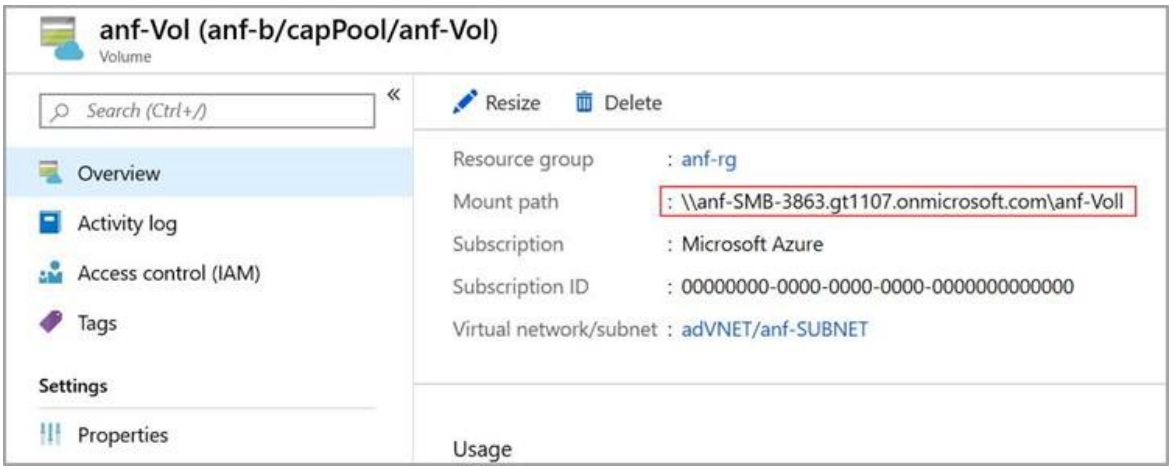
Next, you'll need to create a new volume.

1. Select **Volumes**, then select **Add volume**.
2. When the **Create a volume** tab opens, enter the following values:
   - For **Volume name**, enter a name for the new volume.
   - For **Capacity pool**, select the capacity pool you just created from the drop-down menu.
   - For **Quota (GiB)**, enter the volume size appropriate for your environment.
   - For **Virtual network**, select an existing virtual network that has connectivity to the domain controller from the drop-down menu.
   - Under **Subnet**, select **Create new**. Keep in mind that this subnet will be delegated to Azure NetApp Files.
3. Select **Next: Protocol >>** to open the Protocol tab and configure your volume access parameters.

Configure volume access parameters

After you create the volume, configure the volume access parameters.

1. Select **SMB** as the protocol type.
2. Under Configuration in the **Active Directory** drop-down menu, select the same directory that you originally connected in Join an Active Directory connection. Keep in mind that there's a limit of one Active Directory per subscription.
3. In the **Share name** text box, enter the name of the share used by the session host pool and its users.
4. Select **Review + create** at the bottom of the page. This opens the validation page. After your volume is validated successfully, select **Create**.
5. At this point, the new volume will start to deploy. Once deployment is complete, you can use the Azure NetApp Files share.
6. To see the mount path, select **Go to resource** and look for it in the Overview tab.



Configure FSLogix on session host virtual machines (VMs)

This section is based on Create a profile container for a host pool using a file share.

1. Download the FSLogix agent .zip file while you're still remoted in the session host VM.
2. Unzip the downloaded file.
3. In the file, go to **x64** > **Releases** and run **FSLogixAppsSetup.exe**. The installation menu will open.
4. If you have a product key, enter it in the Product Key text box.
5. Select the check box next to **I agree to the license terms and conditions**.
6. Select **Install**.
7. Navigate to **C:\Program Files\FSLogix\Apps** to confirm the agent installed.
8. From the Start menu, run **RegEdit** as administrator.
9. Navigate to **Computer\HKEY_LOCAL_MACHINE\software\FSLogix**.
10. Create a key named **Profiles**.
11. Create a value named **Enabled** with a **REG_DWORD** type set to a data value of **1**.
12. Create a value named **VHDLocations** with a **Multi-String** type and set its data value to the URI for the Azure NetApp Files share.
13. Create a value named **DeleteLocalProfileWhenVHDShouldApply** with a DWORD value of 1 to avoid problems with existing local profiles before you sign in.

> **WARNING**
> Be careful when creating the DeleteLocalProfileWhenVHDShouldApply value. When the FSLogix Profiles system determines a user should have an FSLogix profile, but a local profile already exists, Profile Container will permanently delete the local profile. The user will then be signed in with the new FSLogix profile.

Assign users to session host

1. Open **PowerShell ISE** as administrator and sign in to Azure Virtual Desktop.
2. Run the following cmdlets:

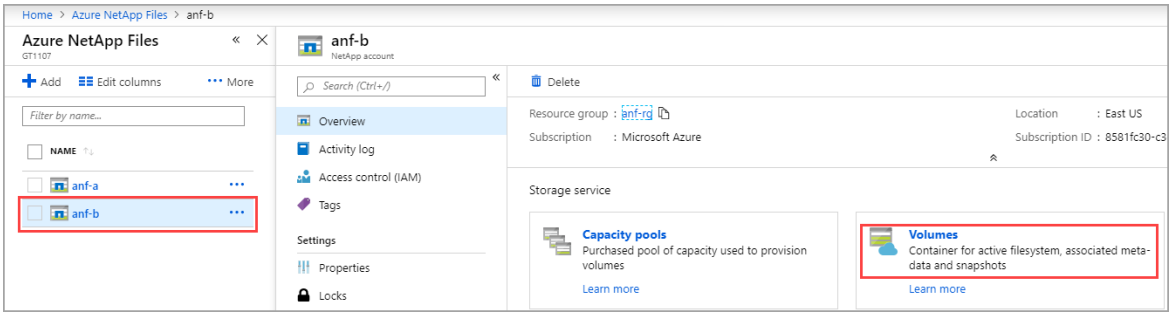
```
Import-Module Microsoft.RdInfra.RdPowershell
# (Optional) Install-Module Microsoft.RdInfra.RdPowershell
$brokerurl = "https://rdbroker.wvd.microsoft.com"
Add-RdsAccount -DeploymentUrl $brokerurl
```

3. When prompted for credentials, enter the credentials for the user with the Tenant Creator or RDS Owner/RDS Contributor roles on the Azure Virtual Desktop tenant.
4. Run the following cmdlets to assign a user to a Remote Desktop group:

```
$wvdTenant = "<your-wvd-tenant>"
$hostPool = "<wvd-pool>"
$appGroup = "Desktop Application Group"
$user = "<user-principal>"
Add-RdsAppGroupUser $wvdTenant $hostPool $appGroup $user
```

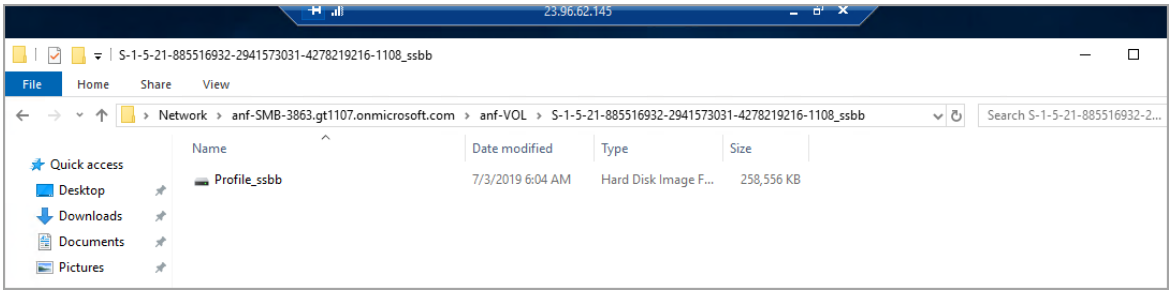
Make sure users can access the Azure NetApp File share

1. Open your internet browser and go to https://rdweb.wvd.microsoft.com/arm/webclient.
2. Sign in with the credentials of a user assigned to the Remote Desktop group.
3. Once you've established the user session, sign in to the Azure portal with an administrative account.
4. Open **Azure NetApp Files**, select your Azure NetApp Files account, and then select **Volumes**. Once the Volumes menu opens, select the corresponding volume.



5. Go to the **Overview** tab and confirm that the FSLogix profile container is using space.
6. Connect directly to any VM part of the host pool using Remote Desktop and open the **File Explorer.** Then navigate to the **Mount path** (in the following example, the mount path is \\anf-SMB-3863.gt1107.onmicrosoft.com\anf-VOL).

   Within this folder, there should be a profile VHD (or VHDX) like the one in the following example.



Next steps

You can use FSLogix profile containers to set up a user profile share. To learn how to create user profile shares with your new containers, see Create a profile container for a host pool using a file share.

You can also create an Azure Files file share to store your FSLogix profile in. To learn more, see Create an Azure Files file share with a domain controller.

Upload MSIX images to Azure NetApp Files in Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

This article describes how to upload MSIX images to Azure NetApp Files in Azure Virtual Desktop.

Requirements

Before you can start uploading the images, you'll need to set up Azure NetApp Files if you haven't already.

To set up Azure NetApp Files, you'll need the following things:

- An Azure account with contributor or administrator access
- A virtual machine (VM) or physical machine joined to Active Directory Domain Services (AD DS), and permissions to access it
- An Azure Virtual Desktop host pool made of domain-joined session hosts. Each session host must be in the same region as the region you create your Azure NetApp files in. For more information, see regional availability. If your existing session hosts aren't in one of the available regions, you'll need to create new ones.

Start using Azure NetApp Files

To start using Azure NetApp Files:

1. Set up your Azure NetApp Files account by following the instructions in Set up your Azure NetApp Files account.
2. Create a capacity pool by following the instructions in Set up a capacity pool.
3. Join an Azure Active Directory (Azure AD) connection by following the instructions in Join an Active Directory connection.
4. Create a new volume by following the instructions in Create a new volume and Configure volume access parameters.
5. Make sure your connection to the Azure NetApp Files share works by following the instructions in Make sure users can access the Azure NetApp Files share.

Upload an MSIX image to the Azure NetApp file share

Now that you've set up your Azure NetApp Files share, you can start uploading images to it.

To upload an MSIX image to your Azure NetApp Files share:

1. In each session host, install the certificate that you signed the MSIX package with. Make sure to store the certificates in the folder named **Trusted People**.
2. Copy the MSIX image you want to add to the Azure NetApps Files share.
3. Go to **File Explorer** and enter the mount path, then paste the MSIX image into the mount path folder.

Your MSIX image should now be accessible to your session hosts when they add an MSIX package using the Azure portal or PowerShell.

Next steps

Now that you've created an Azure NetApp Files share, here are some resources about what you can use it for in Azure Virtual Desktop:

- Create a profile container with Azure NetApp Files and AD DS
- Storage options for FSLogix profile containers in Azure Virtual Desktop
- Create replication peering for Azure NetApp Files

Create a profile container with Azure Files and Azure AD DS

12/6/2021 • 6 minutes to read • Edit Online

This article will show you how to create an FSLogix profile container with Azure Files and Azure Active Directory Domain Services (AD DS).

Prerequisites

This article assumes you've already set up an Azure AD DS instance. If you don't have one yet, follow the instructions in Create a basic managed domain first, then return here.

Add Azure AD DS admins

To add additional admins, you create a new user and grant them permissions.

To add an admin:

1. Select **Azure Active Directory** from the sidebar, then select **All users**, and then select **New user**.
2. Enter the user details into the fields.
3. In the Azure Active Directory pane on the left side of the screen, select **Groups**.
4. Select the **AAD DC Administrators** group.
5. In the left pane, select **Members**, then select **Add members** in the main pane. This will show a list of all users available in Azure AD. Select the name of the user profile you just created.

Set up an Azure Storage account

Now it's time to enable Azure AD DS authentication over Server Message Block (SMB).

To enable authentication:

1. If you haven't already, set up and deploy a general-purpose v2 Azure Storage account by following the instructions in Create an Azure Storage account.
2. Once you've finished setting up your account, select **Go to resource**.
3. Select **Configuration** from the pane on the left side of the screen, then enable **Azure Active Directory authentication for Azure Files** in the main pane. When you're done, select **Save**.
4. Select **Overview** in the pane on the left side of the screen, then select **Files** in the main pane.
5. Select **File share** and enter the **Name** and **Quota** into the fields that appear on the right side of the screen.

Assign access permissions to an identity

Other users will need access permissions to access your file share. To do this, you'll need to assign each user a role with the appropriate access permissions.

To assign users access permissions:

1. From the Azure portal, open the file share you created in Set up an Azure Storage account.
2. Select **Access Control (IAM)**.
3. Select **Add a role assignment**.
4. In the **Add role assignment** tab, select the appropriate built-in role from the role list. You'll need to at least select **Storage File Data SMB Share Contributor** for the account to get proper permissions.
5. For **Assign access to**, select **Azure Active Directory user, group, or service principal**.
6. Select a name or email address for the target Azure Active Directory identity.
7. Select **Save**.

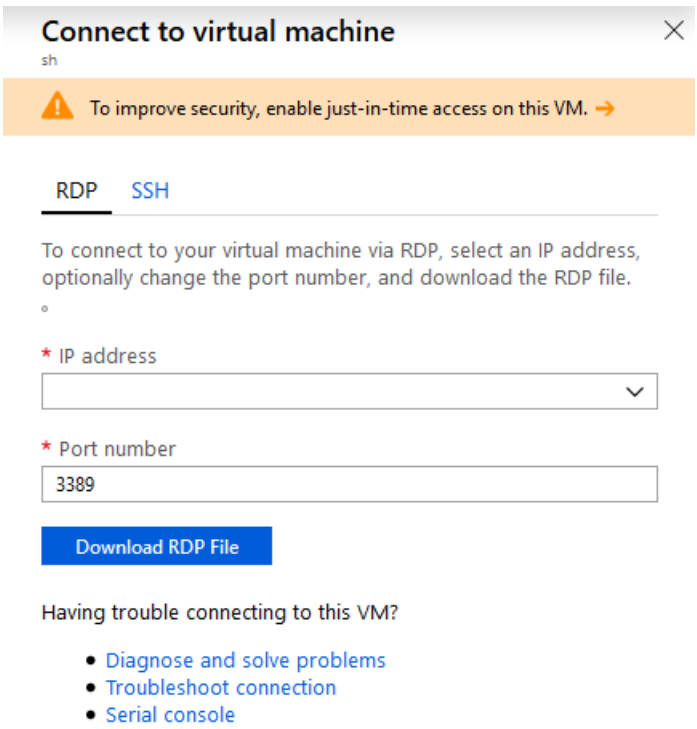
Get the Storage Account access key

Next, you'll need to get the access key for your Storage Account.

To get the Storage Account access key:

1. From the Azure portal sidebar, select **Storage accounts**.
2. From the list of storage accounts, select the account for which you enabled Azure AD DS and created the custom roles in steps above.
3. Under **Settings**, select **Access keys** and copy the key from **key1**.
4. Go to the **Virtual Machines** tab and locate any VM that will become part of your host pool.
5. Select the name of the virtual machine (VM) under **Virtual Machines (adVM)** and select **Connect**

   This will download an RDP file that will let you sign in to the VM with its own credentials.



6. When you've signed in to the VM, run a command prompt as an administrator.
7. Run the following command:

```
net use <desired-drive-letter>: \\<storage-account-name>.file.core.windows.net\<share-name>
/user:Azure\<storage-account-name> <storage-account-key>
```

- Replace `<desired-drive-letter>` with a drive letter of your choice (for example, `y:`).
- Replace all instances of `<storage-account-name>` with the name of the storage account you specified earlier.
- Replace `<share-name>` with the name of the share you created earlier.
- Replace `<storage-account-key>` with the storage account key from Azure.

For example:

```
net use y: \\fsprofile.file.core.windows.net\share HDZQRoFP2BBmoYQ=(truncated)= /user:Azure\fsprofile
```

8. Run the following commands to allow your Azure Virtual Desktop users to create their own profile container while blocking access to the profile containers from other users.

```
icacls <mounted-drive-letter>: /grant <user-email>:(M)
icacls <mounted-drive-letter>: /grant "Creator Owner":(OI)(CI)(IO)(M)
icacls <mounted-drive-letter>: /remove "Authenticated Users"
icacls <mounted-drive-letter>: /remove "Builtin\Users"
```

- Replace `<mounted-drive-letter>` with the letter of the drive you used to map the drive.
- Replace `<user-email>` with the UPN of the user or Active Directory group that contains the users that will require access to the share.

For example:

```
icacls <mounted-drive-letter>: /grant john.doe@contoso.com:(M)
icacls <mounted-drive-letter>: /grant "Creator Owner":(OI)(CI)(IO)(M)
icacls <mounted-drive-letter>: /remove "Authenticated Users"
icacls <mounted-drive-letter>: /remove "Builtin\Users"
```

Create a profile container

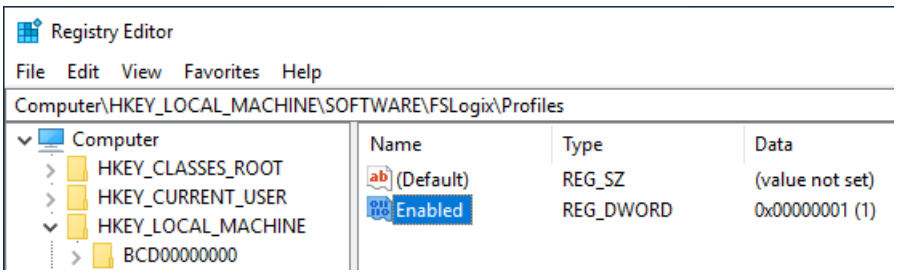
Now that your profiles are ready to go, let's create a FSLogix profile container.

To configure a FSLogix profile container:

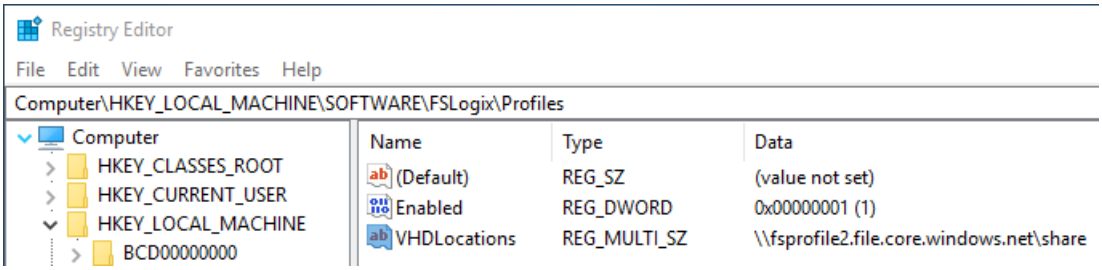
1. Sign in to the session host VM you configured at the beginning of this article, then download and install the FSLogix agent.
2. Unzip the FSLogix agent file you downloaded and go to **x64** > **Releases**, then open **FSLogixAppsSetup.exe**.
3. Once the installer launches, select **I agree to the license terms and conditions.** If applicable, provide a new key.
4. Select **Install**.
5. Open **Drive C**, then go to **Program Files** > **FSLogix** > **Apps** to make sure the FSLogix agent was properly installed.

> **NOTE**
> If there are multiple VMs in the host pool, you'll need to repeat steps 1 through 5 for each VM.

6. Run **Registry Editor** (RegEdit) as an administrator.
7. Navigate to **Computer** > **HKEY_LOCAL_MACHINE** > **software** > **FSLogix**, right-click on **FSLogix**, select **New**, and then select **Key**.
8. Create a new key named **Profiles**.
9. Right-click on **Profiles**, select **New**, and then select **DWORD (32-bit) Value**. Name the value **Enabled** and set the **Data** value to **1**.



10. Right-click on **Profiles**, select **New**, and then select **Multi-String Value**. Name the value **VHDLocations** and set enter the URI for the Azure Files share `\\fsprofile.file.core.windows.net\share` as the Data value.



Assign users to a session host

Now you'll need to assign users to your session host.

To assign users:

1. Run Windows PowerShell as an administrator, then run the following cmdlet to sign in to Azure Virtual Desktop with PowerShell:

```
Import-Module Microsoft.RdInfra.RdPowershell

#Optional
Install-Module Microsoft.RdInfra.RdPowershell

$brokerurl = "https://rdbroker.wvd.microsoft.com"

Add-RdsAccount -DeploymentUrl $brokerurl
```

   When prompted for credentials, enter the same user that was granted the TenantCreator, RDS Owner, or RDS Contributor role on the Azure Virtual Desktop tenant.

2. Run the following cmdlets to assign the user to the remote desktop group:

```
$tenant = "<your-wvd-tenant>"

$pool1 = "<wvd-pool>"

$appgroup = "Desktop Application Group"

$user1 = "<user-principal>"

Add-RdsAppGroupUser $tenant $pool1 $appgroup $user1
```

Like the earlier cmdlets, make sure to replace `<your-wvd-tenant>`, `<wvd-pool>`, and `<user-principal>` with the relevant values.

For example:

```
$pool1 = "contoso"

$tenant = "contoso"

$appgroup = "Desktop Application Group"

$user1 = "jane.doe@contoso.com"

Add-RdsAppGroupUser $tenant $pool1 $appgroup $user1
```

Make sure your profile works

Now all you have to do is make sure the profile you created exists and works as intended.

To verify your profile:

1. Open a browser and go to the Azure Virtual Desktop web client.
2. Sign in with the user account assigned to the Remote Desktop group.
3. Once the user session has been established, open the Azure portal and sign in with an administrative account.
4. From the sidebar, select **Storage accounts**.
5. Select the storage account you configured as the file share for your session host pool and enabled with Azure AD DS.
6. Select the **Files** icon, then expand your share.

   If everything's set up correctly, you should see a **Directory** with a name that's formatted like this: `<user SID>-<username>`.

Next steps

If you're looking for alternate ways to create FSLogix profile containers, check out the following articles:

- Create a profile container for a host pool using a file share.
- Create an FSLogix profile container for a host pool using Azure NetApp Files

You can find more detailed information about concepts related to FSlogix containers for Azure files in FSLogix profile containers and Azure files.

Create a profile container with Azure Files and AD DS

12/6/2021 • 6 minutes to read • Edit Online

In this article, you'll learn how to create an Azure file share authenticated by a domain controller on an existing Azure Virtual Desktop host pool. You can use this file share to store storage profiles.

This process uses Active Directory Domain Services (AD DS), which is an on-prem directory service. If you're looking for information about how to create an FSLogix profile container with Azure AD DS, see Create an FSLogix profile container with Azure Files.

Prerequisites

Before you get started, make sure your domain controller is synchronized to Azure and resolvable from the Azure virtual network (VNET) your session hosts are connected to.

Set up a storage account

First, you'll need to set up an Azure Files storage account.

To set up a storage account:

1. Sign in to the Azure portal.
2. Search for **storage account** in the search bar.
3. Select **+Add**.
4. Enter the following information into the **Create storage account** page:
    - Create a new resource group.
    - Enter a unique name for your storage account.
    - For **Location**, we recommend you choose the same location as the Azure Virtual Desktop host pool.
    - For **Performance**, select **Standard**. (Depending on your IOPS requirements. For more information, see Storage options for FSLogix profile containers in Azure Virtual Desktop.)
    - For **Account type**, select **StorageV2** or **FileStorage** (only available if Performance tier is Premium).
    - For **Replication**, select **Locally-redundant storage (LRS)**.
5. When you're done, select **Review + create**, then select **Create**.

If you need more detailed configuration instructions, see Regional availability.

Create an Azure file share

Next, you'll need to create an Azure file share.

To create a file share:

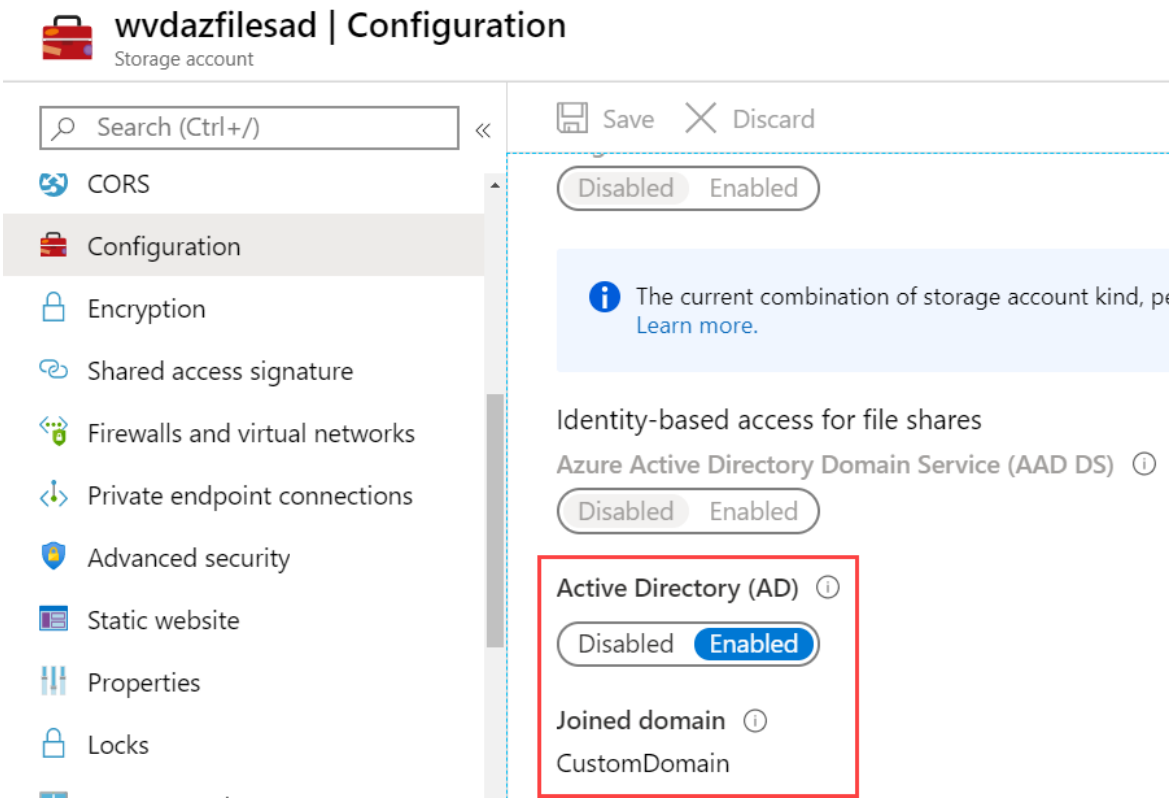
1. Select **Go to resource**.
2. On the Overview page, select **File shares**.
3. Select **+File shares**, create a new file share named **profiles**, then either enter an appropriate quota or leave the field blank for no quota.

4. Select **Create**.

Enable Active Directory authentication

Next, you'll need to enable Active Directory (AD) authentication. To enable this policy, you'll need to follow this section's instructions on a machine that's already domain-joined. To enable authentication, follow these instructions on the VM running the domain controller:

1. Remote Desktop Protocol into the domain-joined VM.
2. Follow the instructions in Enable AD DS authentication for your Azure file shares to install the AzFilesHybrid module and enable authentication.
3. Open the Azure portal, open your storage account, select **Configuration**, then confirm **Active Directory (AD)** is set to **Enabled**.



Assign Azure RBAC permissions to Azure Virtual Desktop users

All users that need to have FSLogix profiles stored on the storage account must be assigned the Storage File Data SMB Share Contributor role.

Users signing in to the Azure Virtual Desktop session hosts need access permissions to access your file share. Granting access to an Azure File share involves configuring permissions both at the share level as well as on the NTFS level, similar to a traditional Windows share.

To configure share level permissions, assign each user a role with the appropriate access permissions. Permissions can be assigned to either individual users or an Azure AD group. To learn more, see Assign access permissions to an identity.

> **NOTE**
> The accounts or groups you assign permissions to should have been created in the domain and synchronized with Azure AD. Accounts created in Azure AD won't work.

To assign Azure role-based access control (Azure RBAC) permissions:

1. Open the Azure portal.
2. Open the storage account you created in Set up a storage account.
3. Select **File shares**, then select the name of the file share you plan to use.
4. Select **Access Control (IAM)**.
5. Select **Add a role assignment**.
6. In the **Add role assignment** tab, select **Storage File Data SMB Share Elevated Contributor** for the administrator account.

   To assign users permissions for their FSLogix profiles, follow these same instructions. However, when you get to step 5, select **Storage File Data SMB Share Contributor** instead.
7. Select **Save**.

Assign users permissions on the Azure file share

Once you've assigned Azure RBAC permissions to your users, next you'll need to configure the NTFS permissions.

You'll need to know two things from the Azure portal to get started:

- The UNC path.
- The storage account key.

Get the UNC path

Here's how to get the UNC path:

1. Open the Azure portal.
2. Open the storage account you created in Set up a storage account.
3. Select **Settings**, then select **Properties**.
4. Copy the **Primary File Service Endpoint** URI to the text editor of your choice.
5. After copying the URI, do the following things to change it into the UNC:
   - Remove `https://` and replace with `\\`
   - Replace the forward slash `/` with a back slash `\`.
   - Add the name of the file share you created in Create an Azure file share to the end of the UNC.

     For example: `\\customdomain.file.core.windows.net\<fileshare-name>`

Get the storage account key

To get the storage account key:

1. Open the Azure portal.
2. Open the storage account you created in Set up a storage account.
3. On the **Storage account** tab, select **Access keys**.
4. Copy **key1** or **key2** to a file on your local machine.

Configure NTFS permissions

To configure your NTFS permissions:

1. Open a command prompt on a domain-joined VM.
2. Run the following command to mount the Azure file share and assign it a drive letter:

   ```
   net use <desired-drive-letter>: <UNC-path> <SA-key> /user:Azure\<SA-name>
   ```

3. Run the following command to review the access permissions to the Azure file share:

   ```
   icacls <mounted-drive-letter>:
   ```

   Replace `<mounted-drive-letter>` with the letter of the drive you mapped to.

   Both *NT Authority\Authenticated Users* and *BUILTIN\Users* have certain permissions by default. These default permissions let these users read other users' profile containers. However, the permissions described in Configure storage permissions for use with Profile Containers and Office Containers don't let users read each others' profile containers.

4. Run the following commands to allow your Azure Virtual Desktop users to create their own profile container while blocking access to their profile containers from other users.

   ```
   icacls <mounted-drive-letter>: /grant <user-email>:(M)
   icacls <mounted-drive-letter>: /grant "Creator Owner":(OI)(CI)(IO)(M)
   icacls <mounted-drive-letter>: /remove "Authenticated Users"
   icacls <mounted-drive-letter>: /remove "Builtin\Users"
   ```

   - Replace <mounted-drive-letter> with the letter of the drive you used to map the drive.
   - Replace <user-email> with the UPN of the user or Active Directory group that contains the users that will require access to the share.

   For example:

   ```
   icacls <mounted-drive-letter>: /grant john.doe@contoso.com:(M)
   icacls <mounted-drive-letter>: /grant "Creator Owner":(OI)(CI)(IO)(M)
   icacls <mounted-drive-letter>: /remove "Authenticated Users"
   icacls <mounted-drive-letter>: /remove "Builtin\Users"
   ```

Configure FSLogix on session host VMs

This section will show you how to configure a VM with FSLogix. You'll need to follow these instructions every time you configure a session host. Before you start configuring, follow the instructions in Download and install FSLogix. There are several options available that ensure the registry keys are set on all session hosts. You can set these options in an image or configure a group policy.

To configure FSLogix on your session host VM:

1. RDP to the session host VM of the Azure Virtual Desktop host pool.
2. Download and install FSLogix.
3. Follow the instructions in Configure profile container registry settings:

   - Navigate to **Computer** > **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **FSLogix**.
   - Create a **Profiles** key.

- Create **Enabled, DWORD** with a value of 1.
- Create **VHDLocations, MULTI_SZ**.
- Set the value of **VHDLocations** to the UNC path you generated in Get the UNC path.

4. Restart the VM.

Testing

Once you've installed and configured FSLogix, you can test your deployment by signing in with a user account that's been assigned an app group or desktop on the host pool. Make sure the user account you sign in with has permission on the file share.

If the user has signed in before, they'll have an existing local profile that will be used during this session. To avoid creating a local profile, either create a new user account to use for tests or use the configuration methods described in Tutorial: Configure Profile Container to redirect User Profiles.

To check your permissions on your session:

1. Start a session on Azure Virtual Desktop.
2. Open the Azure portal.
3. Open the storage account you created in Set up a storage account.
4. Select **Create a share** on the Create an Azure file share page.
5. Make sure a folder containing the user profile now exists in your files.

For additional testing, follow the instructions in Make sure your profile works.

Next steps

To troubleshoot FSLogix, see this troubleshooting guide.

Create a profile container with Azure Files and Azure Active Directory (preview)

12/6/2021 • 12 minutes to read • Edit Online

IMPORTANT

Storing FSLogix profiles on Azure Files for Azure Active Directory (AD)-joined VMs is currently in public preview. This preview version is provided without a service level agreement, and is not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see Supplemental Terms of Use for Microsoft Azure Previews.

In this article, you'll learn how to create an Azure Files share to store FSLogix profiles that can be accessed by hybrid user identities authenticated with Azure Active Directory (AD). Azure AD users can now access an Azure file share using Kerberos authentication. This configuration uses Azure AD to issue the necessary Kerberos tickets to access the file share with the industry-standard SMB protocol. Your end-users can access Azure file shares over the internet without requiring a line-of-sight to domain controllers from Hybrid Azure AD-joined and Azure AD-joined VMs.

In this article, you'll learn how to:

- Configure an Azure storage account for authentication using Azure AD.
- Configure the permissions on an Azure Files share.
- Configure your session hosts to store FSLogix user profiles on Azure Files.

Prerequisites

The Azure AD Kerberos functionality is only available on the following operating systems:

- Windows 11 Enterprise single or multi-session.
- Windows 10 Enterprise single or multi-session, versions 2004 or later with the latest cumulative updates installed, especially the KB5007253 - 2021-11 Cumulative Update Preview for Windows 10.
- Windows Server, version 2022 with the latest cumulative updates installed, especially the KB5007254 - 2021-11 Cumulative Update Preview for Microsoft server operating system version 21H2.

The user accounts must be hybrid user identities, which means you'll also need Active Directory Domain Services (AD DS) and Azure AD Connect. You must create these accounts in Active Directory and sync them to Azure AD.

To assign Azure Role-Based Access Control (RBAC) permissions for the Azure file share to a user group, you must create the group in Active Directory and sync it to Azure AD.

IMPORTANT

This feature is currently only supported in the Azure Public cloud.

Configure your Azure storage account

Start by creating an Azure Storage account if you don't already have one.

> **NOTE**
> Your Azure Storage account can't authenticate with both Azure AD and a second method like Active Directory Domain Services (AD DS) or Azure AD DS. You can only use one authentication method.

Follow the instructions in the following sections to configure Azure AD authentication, configure the Azure AD service principal, and set the API permission for your storage account.

Configure Azure AD authentication on your Azure Storage account

- Install the Azure Storage PowerShell module. This module provides management cmdlets for Azure Storage resources. It's required to create storage accounts, enable Azure AD authentication on the storage account, and retrieve the storage account's Kerberos keys. To install the module, open PowerShell and run the following command:

  ```
  Install-Module -Name Az.Storage
  ```

- Install the Azure AD PowerShell module. This module provides management cmdlets for Azure AD administrative tasks such as user and service principal management. To install this module, open PowerShell, then run the following command:

  ```
  Install-Module -Name AzureAD
  ```

  For more information, see Install the Azure AD PowerShell module.

- Set variables for both storage account name and resource group name by running the following PowerShell cmdlets, replacing the values with the ones relevant to your environment.

  ```
  $resourceGroupName = "<MyResourceGroup>"
  $storageAccountName = "<MyStorageAccount>"
  ```

- Enable Azure AD authentication on your storage account by running the following PowerShell cmdlets:

  ```
  Connect-AzAccount
  $Subscription =  $(Get-AzContext).Subscription.Id;
  $ApiVersion = '2021-04-01'

  $Uri =
  ('https://management.azure.com/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Storage/stora
  geAccounts/{2}?api-version={3}' -f $Subscription, $ResourceGroupName, $StorageAccountName,
  $ApiVersion);

  $json =
     @{properties=@{azureFilesIdentityBasedAuthentication=@{directoryServiceOptions="AADKERB"}}};
  $json = $json | ConvertTo-Json -Depth 99

  $token = $(Get-AzAccessToken).Token
  $headers = @{ Authorization="Bearer $token" }

  try {
      Invoke-RestMethod -Uri $Uri -ContentType 'application/json' -Method PATCH -Headers $Headers -Body
  $json;
  } catch {
      Write-Host $_.Exception.ToString()
      Write-Error -Message "Caught exception setting Storage Account directoryServiceOptions=AADKERB:
  $_" -ErrorAction Stop
  }
  ```

- Generate the kerb1 storage account key for your storage account by running the following PowerShell command:

```
New-AzStorageAccountKey -ResourceGroupName $resourceGroupName -Name $storageAccountName -KeyName
kerb1 -ErrorAction Stop
```

Configure the Azure AD service principal and application

To enable Azure AD authentication on a storage account, you need to create an Azure AD application to represent the storage account in Azure AD. This configuration won't be available in the Azure portal during public preview. To create the application using PowerShell, follow these steps:

- Set the password (service principal secret) based on the Kerberos key of the storage account. The Kerberos key is a password shared between Azure AD and Azure Storage. Kerberos derives the password's value from the first 32 bytes of the storage account's kerb1 key. To set the password, run the following cmdlets:

```
$kerbKey1 = Get-AzStorageAccountKey -ResourceGroupName $resourceGroupName -Name $storageAccountName -
ListKerbKey | Where-Object { $_.KeyName -like "kerb1" }
$aadPasswordBuffer =
[System.Linq.Enumerable]::Take([System.Convert]::FromBase64String($kerbKey1.Value), 32);
$password = "kk:" + [System.Convert]::ToBase64String($aadPasswordBuffer);
```

- Connect to Azure AD and retrieve the tenant information by running the following cmdlets:

```
Connect-AzureAD
$azureAdTenantDetail = Get-AzureADTenantDetail;
$azureAdTenantId = $azureAdTenantDetail.ObjectId
$azureAdPrimaryDomain = ($azureAdTenantDetail.VerifiedDomains | Where-Object {$_._Default -eq
$true}).Name
```

- Generate the service principal names for the Azure AD service principal by running these cmdlets:

```
$servicePrincipalNames = New-Object string[] 3
$servicePrincipalNames[0] = 'HTTP/{0}.file.core.windows.net' -f $storageAccountName
$servicePrincipalNames[1] = 'CIFS/{0}.file.core.windows.net' -f $storageAccountName
$servicePrincipalNames[2] = 'HOST/{0}.file.core.windows.net' -f $storageAccountName
```

- Create an application for the storage account by running this cmdlet:

```
$application = New-AzureADApplication -DisplayName $storageAccountName -IdentifierUris
$servicePrincipalNames -GroupMembershipClaims "All";
```

- Create a service principal for the storage account by running this cmdlet:

```
$servicePrincipal = New-AzureADServicePrincipal -AccountEnabled $true -AppId $application.AppId -
ServicePrincipalType "Application";
```

- Set the password for the storage account's service principal by running the following cmdlets.

```
$Token =
([Microsoft.Open.Azure.AD.CommonLibrary.AzureSession]::AccessTokens['AccessToken']).AccessToken
$apiVersion = '1.6'
$Uri = ('https://graph.windows.net/{0}/{1}/{2}?api-version={3}' -f $azureAdPrimaryDomain,
'servicePrincipals', $servicePrincipal.ObjectId, $apiVersion)
$json = @'
{
  "passwordCredentials": [
  {
    "customKeyIdentifier": null,
    "endDate": "<STORAGEACCOUNTENDDATE>",
    "value": "<STORAGEACCOUNTPASSWORD>",
    "startDate": "<STORAGEACCOUNTSTARTDATE>"
  }]
}
'@
$now = [DateTime]::UtcNow
$json = $json -replace "<STORAGEACCOUNTSTARTDATE>", $now.AddDays(-1).ToString("s")
  $json = $json -replace "<STORAGEACCOUNTENDDATE>", $now.AddMonths(12).ToString("s")
$json = $json -replace "<STORAGEACCOUNTPASSWORD>", $password
$Headers = @{'authorization' = "Bearer $($Token)"}
try {
  Invoke-RestMethod -Uri $Uri -ContentType 'application/json' -Method Patch -Headers $Headers -Body
$json
  Write-Host "Success: Password is set for $storageAccountName"
} catch {
  Write-Host $_.Exception.ToString()
  Write-Host "StatusCode: " $_.Exception.Response.StatusCode.value
  Write-Host "StatusDescription: " $_.Exception.Response.StatusDescription
}
```

Set the API permissions on the newly created application

You can configure the API permissions from the Azure portal by following these steps:

1. Open **Azure Active Directory**.
2. Select **App registrations** on the left pane.
3. Select **All Applications**.
4. Select the application with the name matching your storage account.
5. Select **API permissions** in the left pane.
6. Select **+ Add a permission**.
7. Select **Microsoft Graph** at the top of the page.
8. Select **Delegated permissions**.
9. Select **openid** and **profile** under the **OpenID** permissions group.
10. Select **User.Read** under the **User** permission group.
11. Select **Add permissions** at the bottom of the page.
12. Select **Grant admin consent for "DirectoryName"**.

Configure your Azure Files share

To get started, create an Azure Files share under your storage account to store your FSLogix profiles if you haven't already.

Follow the instructions in the following sections to configure the share-level and directory-level permissions on your Azure Files share to provide the right level of access to your users.

Assign share-level permissions

You must grant your users access to the file share before they can use it. There are two ways you can assign share-level permissions: either assign them to specific Azure AD users or user groups, or you can assign them to

all authenticated identities as a default share-level permission. To learn more about assigning share-level permissions, see Assign share-level permissions to an identity.

All users that need to have FSLogix profiles stored on the storage account you're using must be assigned the **Storage File Data SMB Share Contributor** role.

> **IMPORTANT**
> Azure Virtual Desktop currently only supports assigning specific permissions to hybrid users and user groups. Users and user groups must be managed in Active Directory and synced to Azure AD using Azure AD Connect.

Assign directory level access permissions

To prevent users from accessing the user profile of other users, you must also assign directory-level permissions. This section provides the steps to configure the permissions. Learn more about the recommended list of permissions for FSLogix profiles at Configure the storage permissions for profile containers

> **IMPORTANT**
> Without proper directory level permissions in place, a user can delete the user profile or access the personal information of a different user. It's important to make sure users have proper permissions to prevent accidental deletion from happening.

You can set permissions (ACLs) for files and directories using either the icacls command-line utility or Windows Explorer. The system you use to configure the permissions must meet the following requirements:

- The version of Windows meets the supported OS requirements defined in the Prerequisites section.
- Is Azure AD-joined or Hybrid Azure AD-joined to the same Azure AD tenant as the storage account.
- Has line-of-sight to the domain controller.
- Is domain-joined to your Active Directory (Windows Explorer method only).

During the public preview, configuring permissions using Windows Explorer also requires storage account configuration. You can skip this configuration step when using icacls.

To configure your storage account:

1. On a device that's domain-joined to the Active Directory, install the ActiveDirectory PowerShell module if you haven't already.
2. Set the storage account's ActiveDirectoryProperties to support the Shell experience. Because Azure AD doesn't currently support configuring ACLs in Shell, it must instead rely on Active Directory. To configure Shell, run the following command in PowerShell:

```
function Set-StorageAccountAadKerberosADProperties {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory=$true, Position=0)]
        [string]$ResourceGroupName,

        [Parameter(Mandatory=$true, Position=1)]
        [string]$StorageAccountName,

        [Parameter(Mandatory=$false, Position=2)]
        [string]$Domain
    )

    $AzContext = Get-AzContext;
    if ($null -eq $AzContext) {
        Write-Error "No Azure context found.  Please run Connect-AzAccount and then retry." -
```

```
ErrorAction Stop;
    }

    $AdModule = Get-Module ActiveDirectory;
     if ($null -eq $AdModule) {
        Write-Error "Please install and/or import the ActiveDirectory PowerShell module." -
ErrorAction Stop;
    }

    if ([System.String]::IsNullOrEmpty($Domain)) {
        $domainInformation = Get-ADDomain
        $Domain = $domainInformation.DnsRoot
    } else {
        $domainInformation = Get-ADDomain -Server $Domain
    }

    $domainGuid = $domainInformation.ObjectGUID.ToString()
    $domainName = $domainInformation.DnsRoot
    $domainSid = $domainInformation.DomainSID.Value
    $forestName = $domainInformation.Forest
    $netBiosDomainName = $domainInformation.DnsRoot
    $azureStorageSid = $domainSid + "-123454321";

    Write-Verbose "Setting AD properties on $StorageAccountName in $ResourceGroupName : `
        EnableActiveDirectoryDomainServicesForFile=$true, ActiveDirectoryDomainName=$domainName, `
        ActiveDirectoryNetBiosDomainName=$netBiosDomainName,
ActiveDirectoryForestName=$($domainInformation.Forest) `
        ActiveDirectoryDomainGuid=$domainGuid, ActiveDirectoryDomainSid=$domainSid, `
        ActiveDirectoryAzureStorageSid=$azureStorageSid"

    $Subscription =  $AzContext.Subscription.Id;
    $ApiVersion = '2021-04-01'

    $Uri =
('https://management.azure.com/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Storage/stora
geAccounts/{2}?api-version={3}' `
        -f $Subscription, $ResourceGroupName, $StorageAccountName, $ApiVersion);

    $json=
        @{
            properties=
                @{azureFilesIdentityBasedAuthentication=
                    @{directoryServiceOptions="AADKERB";
                        activeDirectoryProperties=@{domainName="$($domainName)";
                                                    netBiosDomainName="$($netBiosDomainName)";
                                                    forestName="$($forestName)";
                                                    domainGuid="$($domainGuid)";
                                                    domainSid="$($domainSid)";
                                                    azureStorageSid="$($azureStorageSid)"}
                                                    }
                    }
        };

    $json = $json | ConvertTo-Json -Depth 99

    $token = $(Get-AzAccessToken).Token
    $headers = @{ Authorization="Bearer $token" }

    try {
        Invoke-RestMethod -Uri $Uri -ContentType 'application/json' -Method PATCH -Headers $Headers -
Body $json
    } catch {
        Write-Host $_.Exception.ToString()
        Write-Host "Error setting Storage Account AD properties.  StatusCode:"
$_.Exception.Response.StatusCode.value__
        Write-Host "Error setting Storage Account AD properties.  StatusDescription:"
$_.Exception.Response.StatusDescription
        Write-Error -Message "Caught exception setting Storage Account AD properties: $_" -
ErrorAction Stop
```

```
    }
}
```

3. Call the function by running the following PowerShell cmdlets:

```
Connect-AzAccount
Set-StorageAccountAadKerberosADProperties -ResourceGroupName $resourceGroupName -StorageAccountName
$storageAccountName
```

Enable Azure AD Kerberos functionality by configuring the group policy or registry value in the following list:

- Group policy:
  `Administrative Templates\System\Kerberos\Allow retrieving the Azure AD Kerberos Ticket Granting Ticket during logon`
- Registry value:
  `reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters /v CloudKerberosTicketRetrievalEnabled /t REG_DWORD /d 1`

Next, make sure you can retrieve a Kerberos Ticket Granting Ticket (TGT) by following these instructions:

1. Open a command window.
2. Run the following command:

```
dsregcmd /RefreshPrt
```

3. Lock and then unlock your device using the same user account.
4. In the command window, run the following commands:

```
klist purge
klist get krbtgt
```

5. Confirm you have a Kerberos TGT by looking for an item with a server property of `krbtgt/KERBEROS.MICROSOFTONLINE.COM @ KERBEROS.MICROSOFTONLINE.COM`.
6. Verify you can mount the network share by running the following command in your command window:

```
net use <DriveLetter>: \\<storage-account-name>.file.core.windows.net\<fIle-share-name>
```

Finally, follow the instructions in Configure directory and file level permissions to finish configuring your permissions with icacls or Windows Explorer.

Configure the session hosts

To access Azure file shares from an Azure AD-joined VM for FSLogix profiles, you must configure the session hosts. To configure session hosts:

1. Enable the Azure AD Kerberos functionality by configuring the group policy or registry value with the values in the following list. Once you've configured those values, restart your system to make the changes take effect.

   - Group policy:
     `Administrative Templates\System\Kerberos\Allow retrieving the Azure AD Kerberos Ticket Granting Ticket during logon`

- Registry value:

  ```
  reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters /v
  CloudKerberosTicketRetrievalEnabled /t REG_DWORD /d 1
  ```

2. When you use Azure AD with a roaming profile solution like FSLogix, the credential keys in Credential Manager must belong to the profile that's currently loading. This will let you load your profile on many different VMs instead of being limited to just one. To enable this setting, create a new registry value by running the following command:

   ```
   reg add HKLM\Software\Policies\Microsoft\AzureADAccount /v LoadCredKeyFromProfile /t REG_DWORD /d 1
   ```

> **NOTE**
>
> The session hosts don't need network line-of-sight to the domain controller.

Configure FSLogix on the session host

This section will show you how to configure a VM with FSLogix. You'll need to follow these instructions every time you configure a session host. There are several options available that ensure the registry keys are set on all session hosts. You can set these options in an image or configure a group policy.

To configure FSLogix:

1. Update or install FSLogix on your session host, if needed.
2. Follow the instructions in Configure profile container registry settings to create the **Enabled** and **VHDLocations** registry values. Set the value of **VHDLocations** to `\\<Storage-account-name>.file.core.windows.net\<file-share-name>`.

Test your deployment

Once you've installed and configured FSLogix, you can test your deployment by signing in with a user account that's been assigned to an application group on the host pool. The user account you sign in with must have permission to use the file share.

If the user has signed in before, they'll have an existing local profile that the service will use during this session. To avoid creating a local profile, either create a new user account to use for tests or use the configuration methods described in Tutorial: Configure profile container to redirect user profiles to enable the *DeleteLocalProfileWhenVHDShouldApply* setting.

Finally, test the profile to make sure that it works:

1. Open the Azure portal and sign in with an administrative account.
2. From the sidebar, select **Storage accounts**.
3. Select the storage account you configured for your session host pool.
4. From the sidebar, select **File shares**.
5. Select the file share you configured to store the profiles.
6. If everything's set up correctly, you should see a directory with a name that's formatted like this: `<user SID>_<username>`.

Next steps

- To troubleshoot FSLogix, see this troubleshooting guide.

- To configure FSLogix profiles on Azure Files with Azure Active Directory Domain Services, see Create a profile container with Azure Files and Azure AD DS.
- To configure FSLogix profiles on Azure Files with Active Directory Domain Services, see Create a profile container with Azure Files and AD DS.

Install Microsoft Office using FSLogix application containers

12/6/2021 • 2 minutes to read • Edit Online

You can install Microsoft Office quickly and efficiently by using an FSLogix application container as a template for the other virtual machines (VMs) in your host pool.

Here's why using an FSLogix app container can help make installation faster:

- Offloading your Office apps to an app container reduces the requirements for your C drive size.
- Snapshots or backups of your VM takes less resources.
- Having an automated pipeline through updating a single image makes updating your VMs easier.
- You only need one image to install Office (and other apps) onto all the VMs in your Azure Virtual Desktop deployment.

This article will show you how to set up an FSLogix application container with Office.

Requirements

You'll need the following things to set up the rule editor:

- a VM running Windows without Office installed
- a copy of Office
- a copy of FSLogix installed on your deployment
- a network share that all VMs in your host pool have read-only access to

Install Office

To install Office on your VHD or VHDX, enable the Remote Desktop Protocol in your VM, then follow the instructions in Install Office on a VHD master image. When installing, make sure you're using the correct licenses.

> **NOTE**
> Azure Virtual Desktop requires Share Computer Activation (SCA).

Install FSLogix

To install FSLogix and the Rule Editor, follow the instructions in Download and install FSLogix.

Create and prepare a VHD to store Office

Next, you'll need to create and prepare a VHD image to use the Rule Editor on:

1. Open a command prompt as an administrator. and run the following command:

```
taskkill /F /IM MicrosoftEdge.exe /T
```

> **NOTE**
> Make sure to keep the blank spaces you see in this command.

2. Next, run the following command:

```
sc queryex type=service state=all | find /i "ClickToRunSvc"
```

   If you find the service, restart the VM before continuing with step 3.

```
net stop ClickToRunSvc
```

3. After that, go to **Program Files** > **FSLogix** > **Apps** and run the following command to create the target VHD:

```
frx moveto-vhd -filename <path to network share>\office.vhdx -src "C:\Program Files\Microsoft Office"
-size-mbs 5000
```

   The VHD you create with this command should contain the C:\Program Files\Microsoft Office folder.

> **NOTE**
> If you see any errors, uninstall Office and start over from step 1.

Configure the Rule Editor

Now that you've prepared your image, you'll need to configure the Rule Editor and create a file to store your rules in.

1. Go to **Program Files** > **FSLogix** > **Apps** and run **RuleEditor.exe**.
2. Select **File** > **New** > **Create** to make a new rule set, then save that rule set to a local folder.
3. Select **Blank Rule Set**, then select **OK**.
4. Select the **+** button. This will open the **Add Rule** window. This will change the options in the **Add Rule** dialog.
5. From the drop-down menu, select **App Container (VHD) Rule**.
6. Enter **C:\Program Files\Microsoft Office** into the **Folder** field.
7. For the **Disk file** field, select **<path>\office.vhd** from the **Create target VHD** section.
8. Select **OK**.
9. Go to the working folder at **C:\Users\<username>\Documents\FSLogix Rule Sets** and look for the .frx and .fxa files. You need to move these files to the Rules folder located at **C:\Program Files\FSLogix\Apps\Rules** in order for the rules to start working.
10. Select **Apply Rules to System** for the rules to take effect.

> **NOTE**
>
> You'll need to apply the app rule files will need to all session hosts.

Next steps

If you want to learn more about FSLogix, check out our FSLogix documentation.

Authorize an account for Azure Files

12/6/2021 • 4 minutes to read • Edit Online

This article will show you how to authorize an Azure Virtual Desktop host pool to use Azure Files.

Requirements

Before you get started, you'll need the following things:

- An Active Directory Domain Services (AD DS) account synced to Azure Active Directory (Azure AD)
- Permissions to create a group in AD DS
- A storage account and the permissions needed to create a new storage account, if necessary
- A virtual machine (VM) or physical machine joined to AD DS that you have permission to access
- An Azure Virtual Desktop host pool in which all session hosts have been domain joined

Create a security group in Active Directory Domain Services

First, you'll need to create a security group in AD DS. This security group will be used in later steps to grant share-level and New Technology File System (NTFS) file share permissions.

> **NOTE**
> If you have an existing security group you'd prefer to use, select the name of that group instead of creating a new group.

To create a security group:

1. Open a remote session with the VM or physical machine joined to AD DS that you want to add to the security group.
2. Open **Active Directory Users and Computers**.
3. Under the domain node, right-click the name of your machine. In the drop-down menu, select **New** > **Group**.
4. In the **New Object – Group** window, enter the name of the new group, then select the following values:
   - For **Group scope**, select **Global**
   - For **Group type**, select **Security**
5. Right-click on the new group and select **Properties**.
6. In the **Properties** window, select the **Members** tab.
7. Select **Add...**.
8. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** window, select **Object Types...** > **Computers**. When you're finished, select **OK**.
9. In the **Enter the object names to select** window, enter the names of all session hosts you want to include in the security group.
10. Select **Check Names**, then select the name of the session host you want to use from the list that appears.

11. Select **OK**, then select **Apply**.

> **NOTE**
> New security groups may take up to 1 hour to sync with Azure AD.

Create a storage account

If you haven't created a storage account already, follow the directions in Create a storage account first. When you create a new storage account, make sure to also create a new file share.

> **NOTE**
> If you're creating a **Premium** storage account make sure **Account Kind** is set to **FileStorage**.

Get RBAC permissions

To get RBAC permissions:

1. Select the storage account you want to use.
2. Select **Access Control (IAM)**, then select **Add**. Next, select **Add role assignments** from the drop-down menu.
3. In the **Add role assignment** screen, select the following values:
   - For **Role**, select **Storage File Data SMB Share Contributor**.
   - For **Assign access to**, select **User, Group, or Service Principal**.
   - For **Subscription**, select **Based on your environment**.
   - For **Select**, select the name of the Active Directory group that contains your session hosts.
4. Select **Save**.

Join your storage account to AD DS

Next, you'll need to join storage account to AD DS. To join your account to AD DS:

1. Open a remote session in a VM or physical machine joined to AD DS.

   > **NOTE**
   > Run the script using an on-premises AD DS credential that is synced to your Azure AD. The on-premises AD DS credential must have either storage account owner or contributor Azure role permissions.

2. Download and unzip the latest version on AzFilesHybrid.
3. Open **PowerShell** in elevated mode.
4. Run the following cmdlet to set the execution policy:

   ```
   Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser
   ```

5. Next, go to the folder where you unzipped AzfileHybrid and run this command:

```
.\\CopyToPSPath.ps1
```

6. After that, import the AzFilesHybrid module by running this cmdlet:

```
Import-Module -Name AzFilesHybrid
```

7. Next, run this cmdlet to connect to Azure AD:

```
Connect-AzAccount
```

8. Set the following parameters, making sure to replace the placeholders with the values relevant to your scenario:

```
$SubscriptionId = "<your-subscription-id-here>"

$ResourceGroupName = "<resource-group-name-here>"

$StorageAccountName = "<storage-account-name-here>"
```

9. Finally, run this command:

```
Join-AzStorageAccountForAuth `

-ResourceGroupName $ResourceGroupName `

-StorageAccountName $StorageAccountName `

-DomainAccountType "ComputerAccount" `

-OrganizationalUnitDistinguishedName "<ou-here>" `

-EncryptionType "'RC4','AES256'"
```

Get NTFS-level permissions

In order to authenticate with AD DS computer accounts against an Azure Files storage account, we must also assign NTFS-level permissions in addition to the RBAC permission we set up earlier.

To assign NTFS permissions:

1. Open the Azure portal and navigate to the storage account that we added to AD DS.
2. Select **Access keys** and copy the value in the **Key1** field.
3. Start a remote session in the VM or physical machine joined to AD DS.
4. Open a command prompt in elevated mode.
5. Run the following command, with the placeholders replaced with the values relevant to your deployment:

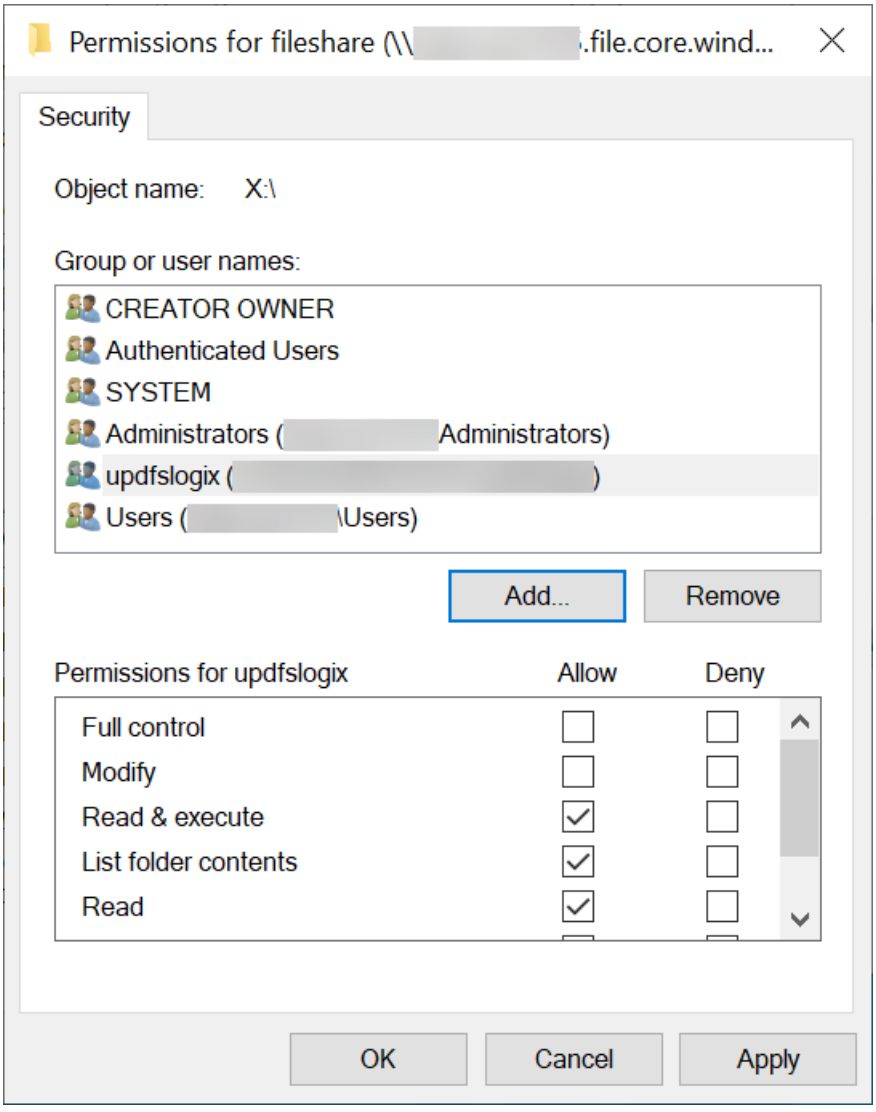
```
net use <desired-drive-letter>:
\\<storage-account-name>.file.core.windows.net\<share-name>
/user:Azure\<storage-account-name> <storage-account-key>
```

> **NOTE**
> When you run this command, the output should say "The command completed successfully." If not, check your input and try again.

6. Open **File Explorer** and find the drive letter you used in the command in step 5.
7. Right-click the drive letter, then select **Properties** > **Security** from the drop-down menu.
8. Select **Edit**, then select **Add...**.

> **NOTE**
> Make sure that domain name matches your AD DS domain name. If it doesn't, then that means the storage account hasn't been domain joined. You'll need to use a domain-joined account in order to continue.

9. If prompted, enter your admin credentials.
10. In the **Select Users, Computers, Service Accounts, or Groups** window, enter the name of the group from Create a security group in Active Directory Domain Services.
11. Select **OK**. After that, confirm the group has the **Read & execute** permission. If the group has permissions, the "Allow" check box should be selected, as shown in the following image:



12. Add the Active Directory group with the computer accounts with **Read & execute** permissions to the

security group.

13. Select **Apply**. If you see a Windows Security prompt, select **Yes** to confirm your changes.

Next steps

If you run into any issues after setup, check out our Azure Files troubleshooting article.

Customize Remote Desktop Protocol (RDP) properties for a host pool

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Customizing a host pool's Remote Desktop Protocol (RDP) properties, such as multi-monitor experience and audio redirection, lets you deliver an optimal experience for your users based on their needs. If you'd like to change the default RDP file properties, you can customize RDP properties in Azure Virtual Desktop by either using the Azure portal or by using the *-CustomRdpProperty* parameter in the **Update-AzWvdHostPool** cmdlet.

See supported RDP file settings for a full list of supported properties and their default values.

Default RDP file properties

RDP files have the following properties by default:

RDP PROPERTY	FOR BOTH DESKTOP AND REMOTEAPP
Multi-monitor mode	Enabled
Drive redirections enabled	Drives, clipboard, printers, COM ports, smart cards, devices, and usbdevicestore
Remote audio mode	Play locally
VideoPlayback	Enabled
EnableCredssp	Enabled

NOTE

- Multi-monitor mode is only enabled for Desktop app groups and will be ignored for RemoteApp app groups.
- All default RDP file properties are exposed in the Azure Portal.
- By default, the CustomRdpProperty field is null in the Azure portal. A null CustomRdpProperty field will apply all default RDP properties to your host pool. An empty CustomRdpProperty field will not apply any default RDP properties to your host pool.

Prerequisites

Before you begin, follow the instructions in Set up the Azure Virtual Desktop PowerShell module to set up your PowerShell module and sign in to Azure.

Configure RDP properties in the Azure portal

To configure RDP properties in the Azure portal:

1. Sign in to Azure at https://portal.azure.com.
2. Enter **Azure Virtual Desktop** into the search bar.
3. Under Services, select **Azure Virtual Desktop**.
4. At the Azure Virtual Desktop page, select **host pools** in the menu on the left side of the screen.
5. Select **the name of the host pool** you want to update.
6. Select **RDP Properties** in the menu on the left side of the screen.
7. Set the property you want.
   - Alternatively, you can open the **Advanced** tab and add your RDP properties in a semicolon-separated format like the PowerShell examples in the following sections.
8. When you're done, select **Save** to save your changes.

The next sections will tell you how to edit custom RDP properties manually in PowerShell.

Add or edit a single custom RDP property

To add or edit a single custom RDP property, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -CustomRdpProperty
<property>
```

To check if the cmdlet you just ran updated the property, run this cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | format-list Name,
CustomRdpProperty

Name               : <hostpoolname>
CustomRdpProperty : <customRDPpropertystring>
```

For example, if you were checking for the "audiocapturemode" property on a host pool named 0301HP, you'd enter this cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName 0301rg -Name 0301hp | format-list Name, CustomRdpProperty

Name               : 0301HP
CustomRdpProperty : audiocapturemode:i:1;
```

Add or edit multiple custom RDP properties

To add or edit multiple custom RDP properties, run the following PowerShell cmdlets by providing the custom RDP properties as a semicolon-separated string:

```
$properties="<property1>;<property2>;<property3>"
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -CustomRdpProperty
$properties
```

You can check to make sure the RDP property was added by running the following cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | format-list Name,
CustomRdpProperty

Name              : <hostpoolname>
CustomRdpProperty : <customRDPpropertystring>
```

Based on our earlier cmdlet example, if you set up multiple RDP properties on the 0301HP host pool, your cmdlet would look like this:

```
Get-AzWvdHostPool -ResourceGroupName 0301rg -Name 0301hp | format-list Name, CustomRdpProperty

Name              : 0301HP
CustomRdpProperty : audiocapturemode:i:1;audiomode:i:0;
```

Reset all custom RDP properties

You can reset individual custom RDP properties to their default values by following the instructions in Add or edit a single custom RDP property, or you can reset all custom RDP properties for a host pool by running the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -CustomRdpProperty ""
```

To make sure you've successfully removed the setting, enter this cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | format-list Name,
CustomRdpProperty

Name              : <hostpoolname>
CustomRdpProperty : <CustomRDPpropertystring>
```

Next steps

Now that you've customized the RDP properties for a given host pool, you can sign in to a Azure Virtual Desktop client to test them as part of a user session. These next how-to guides will tell you how to connect to a session using the client of your choice:

- Connect with the Windows Desktop client
- Connect with the web client
- Connect with the Android client
- Connect with the macOS client
- Connect with the iOS client

Configure the Azure Virtual Desktop load-balancing method

12/6/2021 • 2 minutes to read • Edit Online

Configuring the load-balancing method for a host pool allows you to adjust the Azure Virtual Desktop environment to better suit your needs.

> **NOTE**
> This does not apply to a persistent desktop host pool because users always have a 1:1 mapping to a session host within the host pool.

Prerequisites

This article assumes you've followed the instructions in Set up the Azure Virtual Desktop PowerShell module to download and install the PowerShell module and sign in to your Azure account.

Configure breadth-first load balancing

Breadth-first load balancing is the default configuration for new non-persistent host pools. Breadth-first load balancing distributes new user sessions across all available session hosts in the host pool. When configuring breadth-first load balancing, you may set a maximum session limit per session host in the host pool.

To configure a host pool to perform breadth-first load balancing without adjusting the maximum session limit, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -LoadBalancerType
'BreadthFirst'
```

After that, to make sure you've set the breadth-first load balancing method, run the following cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | format-list Name,
LoadBalancerType

Name             : hostpoolname
LoadBalancerType : BreadthFirst
```

To configure a host pool to perform breadth-first load balancing and to use a new maximum session limit, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -LoadBalancerType
'BreadthFirst' -MaxSessionLimit ###
```

Configure depth-first load balancing

Depth-first load balancing distributes new user sessions to an available session host with the highest number of connections but has not reached its maximum session limit threshold.

IMPORTANT

When configuring depth-first load balancing, you must set a maximum session limit per session host in the host pool.

To configure a host pool to perform depth-first load balancing, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -LoadBalancerType
'DepthFirst' -MaxSessionLimit ###
```

NOTE

The depth-first load balancing algorithm distributes sessions to session hosts based on the maximum session host limit (`-MaxSessionLimit`). This parameter's default value is `999999` , which is also the highest possible number you can set this variable to. This parameter is required when you use the depth-first load balancing algorithm. For the best possible user experience, make sure to change the maximum session host limit parameter to a number that best suits your environment.

To make sure the setting has updated, run this cmdlet:

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | format-list Name,
LoadBalancerType, MaxSessionLimit

Name             : hostpoolname
LoadBalancerType : DepthFirst
MaxSessionLimit  : 6
```

Configure load balancing with the Azure portal

You can also configure load balancing with the Azure portal.

To configure load balancing:

1. Sign into the Azure portal at https://portal.azure.com.
2. Search for and select **Azure Virtual Desktop** under Services.
3. In the Azure Virtual Desktop page, select **Host pools**.
4. Select the name of the host pool you want to edit.
5. Select **Properties**.
6. Enter the **Max session limit** into the field and select the **load balancing algorithm** you want for this host pool in the drop-down menu.
7. Select **Save**. This applies the new load balancing settings.

Configure the personal desktop host pool assignment type

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can configure the assignment type of your personal desktop host pool to adjust your Azure Virtual Desktop environment to better suit your needs. In this topic, we'll show you how to configure automatic or direct assignment for your users.

NOTE

The instructions in this article only apply to personal desktop host pools, not pooled host pools, since users in pooled host pools aren't assigned to specific session hosts.

Prerequisites

This article assumes you've already downloaded and installed the Azure Virtual Desktop PowerShell module. If you haven't, follow the instructions in Set up the PowerShell module.

Configure automatic assignment

Automatic assignment is the default assignment type for new personal desktop host pools created in your Azure Virtual Desktop environment. Automatically assigning users doesn't require a specific session host.

To automatically assign users, first assign them to the personal desktop host pool so that they can see the desktop in their feed. When an assigned user launches the desktop in their feed, they will claim an available session host if they have not already connected to the host pool, which completes the assignment process.

To configure a host pool to automatically assign users to VMs, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -
PersonalDesktopAssignmentType Automatic
```

To assign a user to the personal desktop host pool, run the following PowerShell cmdlet:

```
New-AzRoleAssignment -SignInName <userupn> -RoleDefinitionName "Desktop Virtualization User" -ResourceName
<appgroupname> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Configure direct assignment

Unlike automatic assignment, when you use direct assignment, you must assign the user to both the personal desktop host pool and a specific session host before they can connect to their personal desktop. If the user is only assigned to a host pool without a session host assignment, they won't be able to access resources.

To configure a host pool to require direct assignment of users to session hosts, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -
PersonalDesktopAssignmentType Direct
```

To assign a user to the personal desktop host pool, run the following PowerShell cmdlet:

```
New-AzRoleAssignment -SignInName <userupn> -RoleDefinitionName "Desktop Virtualization User" -ResourceName
<appgroupname> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

To assign a user to a specific session host, run the following PowerShell cmdlet:

```
Update-AzWvdSessionHost -HostPoolName <hostpoolname> -Name <sessionhostname> -ResourceGroupName
<resourcegroupname> -AssignedUser <userupn>
```

To directly assign a user to a session host in the Azure portal:

1. Sign in to the Azure portal at https://portal.azure.com.
2. Enter **Azure Virtual Desktop** into the search bar.
3. Under **Services**, select **Azure Virtual Desktop**.
4. At the Azure Virtual Desktop page, go the menu on the left side of the window and select **Host pools**.
5. Select the name of the host pool you want to update.
6. Next, go to the menu on the left side of the window and select **Application groups**.
7. Select the name of the desktop app group you want to edit, then select **Assignments** in the menu on the left side of the window.
8. Select **+ Add**, then select the users or user groups you want to publish this desktop app group to.
9. Select **Assign VM** in the Information bar to assign a session host to a user.
10. Select the session host you want to assign to the user, then select **Assign**.
11. Select the user you want to assign the session host to from the list of available users.
12. When you're done, select **Select**.

Next steps

Now that you've configured the personal desktop assignment type, you can sign in to a Azure Virtual Desktop client to test it as part of a user session. These next two How-tos will tell you how to connect to a session using the client of your choice:

- Connect with the Windows Desktop client
- Connect with the web client
- Connect with the Android client
- Connect with the iOS client
- Connect with the macOS client

Apply Windows license to session host virtual machines

12/6/2021 • 2 minutes to read • Edit Online

Customers who are properly licensed to run Azure Virtual Desktop workloads are eligible to apply a Windows license to their session host virtual machines and run them without paying for another license. For more information, see Azure Virtual Desktop pricing.

Ways to use your Azure Virtual Desktop license

Azure Virtual Desktop licensing allows you to apply a license to any Windows or Windows Server virtual machine that is registered as a session host in a host pool and receives user connections. This license does not apply to virtual machines that are running as file share servers, domain controllers, and so on.

There are a few ways to use the Azure Virtual Desktop license:

- You can create a host pool and its session host virtual machines using the Azure Marketplace offering. Virtual machines created this way automatically have the license applied.
- You can create a host pool and its session host virtual machines using the GitHub Azure Resource Manager template. Virtual machines created this way automatically have the license applied.
- You can apply a license to an existing session host virtual machine. To do this, first follow the instructions in Create a host pool with PowerShell to create a host pool and associated VMs, then return to this article to learn how to apply the license.

Apply a Windows license to a session host VM

Make sure you have installed and configured the latest Azure PowerShell. Run the following PowerShell cmdlet to apply the Windows license:

```
$vm = Get-AzVM -ResourceGroup <resourceGroupName> -Name <vmName>
$vm.LicenseType = "Windows_Client"
Update-AzVM -ResourceGroupName <resourceGroupName> -VM $vm
```

Verify your session host VM is utilizing the licensing benefit

After deploying your VM, run this cmdlet to verify the license type:

```
Get-AzVM -ResourceGroupName <resourceGroupName> -Name <vmName>
```

A session host VM with the applied Windows license will show you something like this:

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             : Windows_Client
```

VMs without the applied Windows license will show you something like this:

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             :
```

Run the following cmdlet to see a list of all session host VMs that have the Windows license applied in your Azure subscription:

```
$vms = Get-AzVM
$vms | Where-Object {$_.LicenseType -like "Windows_Client"} | Select-Object ResourceGroupName, Name,
LicenseType
```

Requirements for deploying Windows Server Remote Desktop Services

If you deploy Windows Server as Azure Virtual Desktop hosts in your deployment, a Remote Desktop Services license server must be accessible from those virtual machines. The Remote Desktop Services license server can be located on-premises or in Azure. For more information, see Activate the Remote Desktop Services license server.

Create a golden image in Azure

12/6/2021 • 4 minutes to read • Edit Online

This article will walk you through how to use the Azure portal to create a custom image to use for your Azure Virtual Desktop session hosts. This custom image, which we'll call a "golden image," contains all apps and configuration settings you want to apply to your deployment. There are other approaches to customizing your session hosts, such as using device management tools like Microsoft Endpoint Manager or automating your image build using tools like Azure Image Builder with Azure DevOps. Which strategy works best depends on the complexity and size of your planned Azure Virtual Desktop environment and your current application deployment processes.

Create an image from an Azure VM

When creating a new VM for your golden image, make sure to choose an OS that's in the list of supported virtual machine OS images. We recommend using a Windows 10 multi-session (with or without Microsoft 365) or Windows Server image for pooled host pools. We recommend using Windows 10 Enterprise images for personal host pools. You can use either Generation 1 or Generation 2 VMs; Gen 2 VMs support features that aren't supported for Gen 1 machines. Learn more about Generation 1 and Generation 2 VMs at Support for generation 2 VMs on Azure.

Take your first snapshot

First, create the base VM for your chosen image. After you've deployed the image, take a snapshot of the disk of your image VM. Snapshots are save states that will let you roll back any changes if you run into problems while building the image. Since you'll be taking many snapshots throughout the build process, make sure to give the snapshot a name you can easily identify.

Customize your VM

Sign in to the VM and start customizing it with apps, updates, and other things you'll need for your image. If the VM needs to be domain-joined during customization, remove it from the domain before running sysprep. If you need to install many apps, we recommend you take multiple snapshots to revert your VM if a problem happens. Make sure you've done the following things before taking the final snapshot:

- Install the latest Windows updates.
- Complete any necessary cleanup, such as cleaning up temporary files, defragmenting disks, and removing unnecessary user profiles.

> **NOTE**
> If your machine will include an antivirus app, it may cause issues when you start sysprep. To avoid this, disable all antivirus programs before running sysprep.

Take the final snapshot

When you are done installing your applications to the image VM, take a final snapshot of the disk. If sysprep or capture fails, you will be able to create a new base VM with your applications already installed from this snapshot.

Run sysprep

Some optional things you can do before running Sysprep:

- Reboot once

- Clean up temp files in system storage
- Optimize drivers (defrag)
- Remove any user profiles Generalize the VM by running sysprep.

Capture the VM

After you've completed sysprep and shut down your machine in the Azure portal, open the **VM** tab and select the **Capture** button to save the image for later use. When you capture a VM, you can either add the image to a shared image gallery or capture it as a managed image. The Shared Image Gallery lets you add features and use existing images in other deployments. Images from a Shared Image Gallery are highly-available, ensure easy versioning, and you can deploy them at scale. However, if you have a simpler deployment, you may want to use a standalone managed image instead.

> **IMPORTANT**
> We recommend using Shared Image Gallery images for production environments because of their enhanced capabilities, such as replication and image versioning. When you create a capture, you'll need to delete the VM afterwards, as you'll no longer be able to use it after the capture process is finished. Don't try to capture the same VM twice, even if there's an issue with the capture. Instead, create a new VM from your latest snapshot, then run sysprep again. Once you've finished the capture process, you can use your image to create your session hosts. To find the image, open the **Host pool** tab, choose **Gallery**, then select all images. Next, select **My items** and look for your managed images under **My images**. Your image definitions should appear under the shared items section.

Other recommendations

Here are some extra things you should keep in mind when creating a golden image:

- Don't capture a VM that already exists in your host pools. The image will conflict with the existing VM's configuration, and the new VM won't work.
- Make sure to remove the VM from the domain before running sysprep.
- Delete the base VM once you've captured the image from it.
- After you've captured your image, don't use the same VM you captured again. Instead, create a new base VM from the last snapshot you created. You'll need to periodically update and patch this new VM on a regular basis.
- Don't create a new base VM from an existing custom image.

Next steps

If you want to add a language pack to your image, see Language packs.

Prepare and customize a VHD image for Azure Virtual Desktop

12/6/2021 • 6 minutes to read • Edit Online

This article tells you how to prepare a master virtual hard disk (VHD) image for upload to Azure, including how to create virtual machines (VMs) and install software on them. These instructions are for a Azure Virtual Desktop-specific configuration that can be used with your organization's existing processes.

> **IMPORTANT**
>
> We recommend you use an image from the Azure Image Gallery. However, if you do need to use a customized image, make sure you don't already have the Azure Virtual Desktop Agent installed on your VM. Using a customized image with the Azure Virtual Desktop Agent can cause problems with the image, such as blocking registration and preventing user session connections.

Create a VM

Windows 10 Enterprise multi-session is available in the Azure Image Gallery. There are two options for customizing this image.

The first option is to provision a virtual machine (VM) in Azure by following the instructions in Create a VM from a managed image, and then skip ahead to Software preparation and installation.

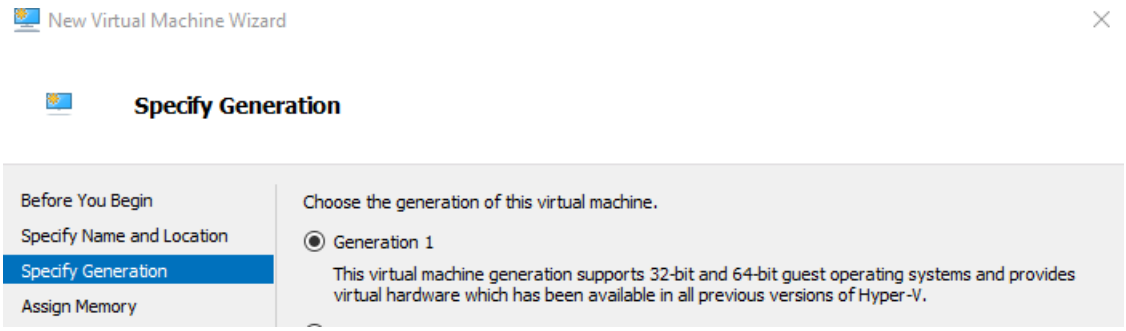
The second option is to create the image locally by downloading the image, provisioning a Hyper-V VM, and customizing it to suit your needs, which we cover in the following section.

Local image creation

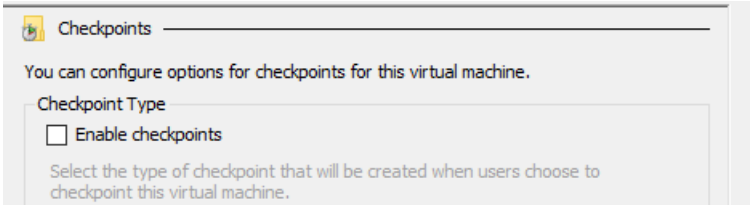
Once you've downloaded the image to a local location, open **Hyper-V Manager** to create a VM with the VHD you copied. The following instructions are a simple version, but you can find more detailed instructions in Create a virtual machine in Hyper-V.

To create a VM with the copied VHD:

1. Open the **New Virtual Machine Wizard**.
2. On the Specify Generation page, select **Generation 1**.



3. Under Checkpoint Type, disable checkpoints by unchecking the check box.

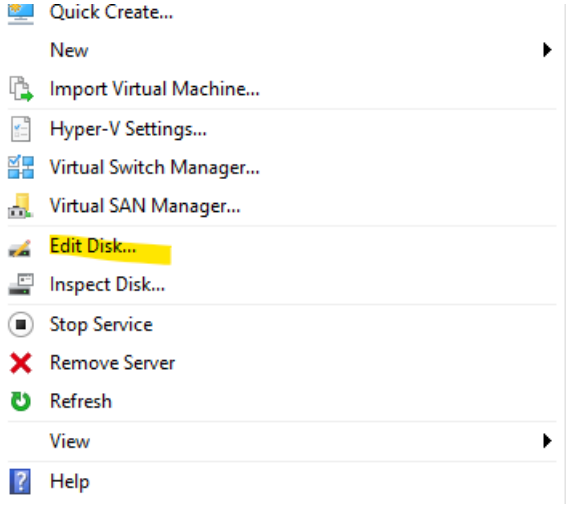


You can also run the following cmdlet in PowerShell to disable checkpoints.

```
Set-VM -Name <VMNAME> -CheckpointType Disabled
```

Fixed disk

If you create a VM from an existing VHD, it creates a dynamic disk by default. It can be changed to a fixed disk by selecting **Edit Disk...** as shown in the following image. For more detailed instructions, see Prepare a Windows VHD or VHDX to upload to Azure.



You can also run the following PowerShell cmdlet to change the disk to a fixed disk.

```
Convert-VHD –Path c:\test\MY-VM.vhdx –DestinationPath c:\test\MY-NEW-VM.vhd -VHDType Fixed
```

Software preparation and installation

This section covers how to prepare and install FSLogix and Windows Defender, as well as some basic configuration options for apps and your image's registry.

If you're installing Microsoft 365 Apps for enterprise and OneDrive on your VM, go to Install Office on a master VHD image and follow the instructions there to install the apps. After you're done, return to this article.

If your users need to access certain LOB applications, we recommend you install them after completing this section's instructions.

Set up user profile container (FSLogix)

To include the FSLogix container as part of the image, follow the instructions in Create a profile container for a host pool using a file share. You can test the functionality of the FSLogix container with this quickstart.

Configure Windows Defender

If Windows Defender is configured in the VM, make sure it's configured to not scan the entire contents of VHD and VHDX files during attachment.

This configuration only removes scanning of VHD and VHDX files during attachment, but won't affect real-time scanning.

For more detailed instructions for how to configure Windows Defender on Windows Server, see Configure Windows Defender Antivirus exclusions on Windows Server.

To learn more about how to configure Windows Defender to exclude certain files from scanning, see Configure and validate exclusions based on file extension and folder location.

Disable Automatic Updates

To disable Automatic Updates via local Group Policy:

1. Open **Local Group Policy Editor\Administrative Templates\Windows Components\Windows Update**.
2. Right-click **Configure Automatic Update** and set it to **Disabled**.

You can also run the following command on a command prompt to disable Automatic Updates.

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" /v NoAutoUpdate /t REG_DWORD /d 1 /f
```

Specify Start layout for Windows 10 PCs (optional)

Run this command to specify a Start layout for Windows 10 PCs.

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" /v SpecialRoamingOverrideAllowed /t
REG_DWORD /d 1 /f
```

Set up time zone redirection

Time zone redirection can be enforced on Group Policy level since all VMs in a host pool are part of the same security group.

To redirect time zones:

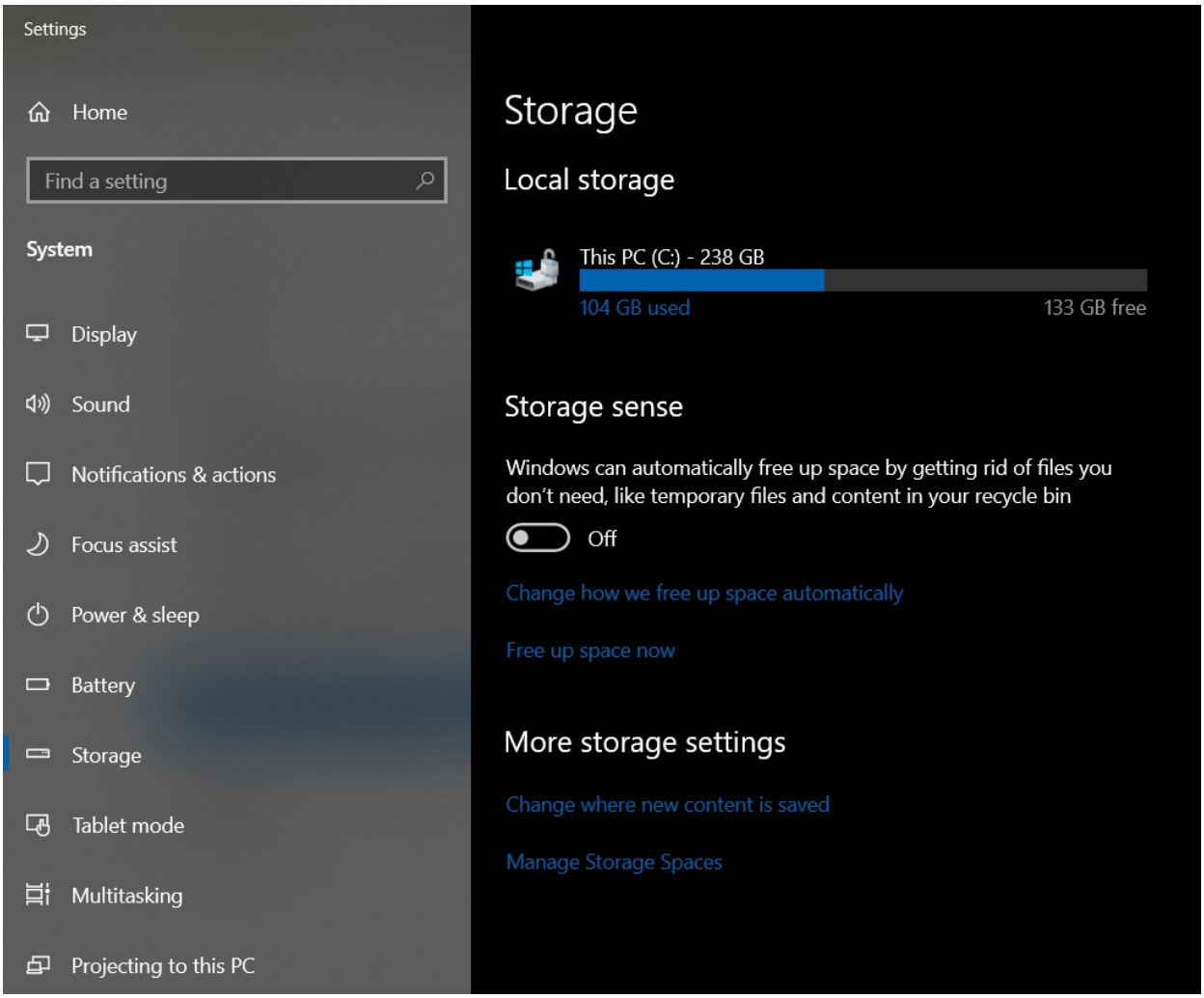
1. On the Active Directory server, open the **Group Policy Management Console**.
2. Expand your domain and Group Policy Objects.
3. Right-click the **Group Policy Object** that you created for the group policy settings and select **Edit**.
4. In the **Group Policy Management Editor**, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Device and Resource Redirection**.
5. Enable the **Allow time zone redirection** setting.

You can also run this command on the master image to redirect time zones:

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" /v fEnableTimeZoneRedirection /t
REG_DWORD /d 1 /f
```

Disable Storage Sense

For Azure Virtual Desktop session host that use Windows 10 Enterprise or Windows 10 Enterprise multi-session, we recommend disabling Storage Sense. You can disable Storage Sense in the Settings menu under **Storage**, as shown in the following screenshot:



You can also change the setting with the registry by running the following command:

```
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\StorageSense\Parameters\StoragePolicy" /v 01 /t
REG_DWORD /d 0 /f
```

Include additional language support

This article doesn't cover how to configure language and regional support. For more information, see the following articles:

- Add languages to Windows images
- Features on demand
- Language and region features on demand (FOD)

Other applications and registry configuration

This section covers application and operating system configuration. All configuration in this section is done through registry entries that can be executed by command-line and regedit tools.

> **NOTE**
> You can implement best practices in configuration with either Group Policy Objects (GPOs) or registry imports. The administrator can choose either option based on their organization's requirements.

For feedback hub collection of telemetry data on Windows 10 Enterprise multi-session, run this command:

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\DataCollection" /v AllowTelemetry /t REG_DWORD /d 3 /f
```

Run the following command to fix Watson crashes:

```
remove CorporateWerServer* from Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error
Reporting
```

Enter the following commands into the registry editor to fix 5k resolution support. You must run the commands before you can enable the side-by-side stack.

```
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v MaxMonitors /t
REG_DWORD /d 4 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v MaxXResolution /t
REG_DWORD /d 5120 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v MaxYResolution /t
REG_DWORD /d 2880 /f

reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\rdp-sxs" /v MaxMonitors /t
REG_DWORD /d 4 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\rdp-sxs" /v MaxXResolution /t
REG_DWORD /d 5120 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\rdp-sxs" /v MaxYResolution /t
REG_DWORD /d 2880 /f
```

Prepare the image for upload to Azure

After you've finished configuration and installed all applications, follow the instructions in Prepare a Windows VHD or VHDX to upload to Azure to prepare the image.

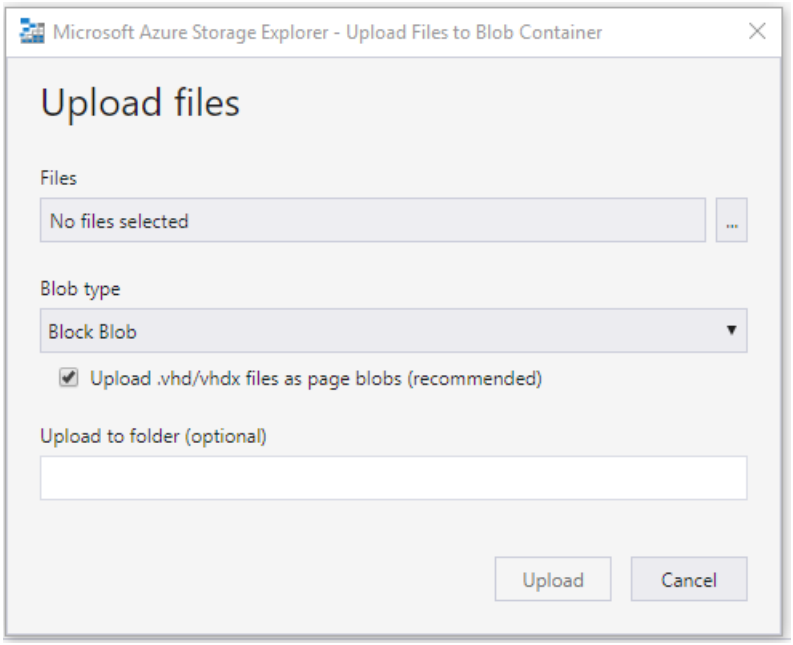
After preparing the image for upload, make sure the VM remains in the off or deallocated state.

Upload master image to a storage account in Azure

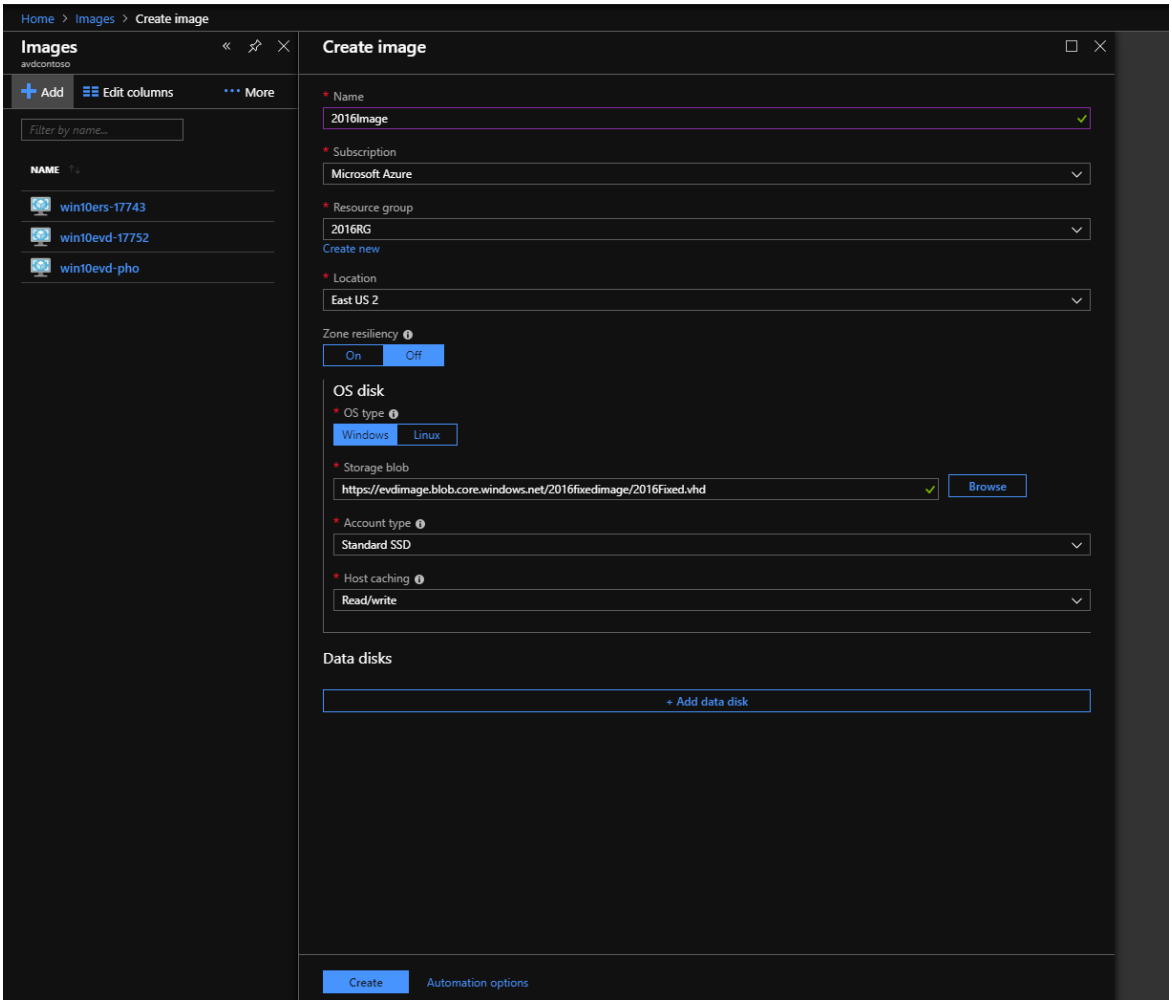
This section only applies when the master image was created locally.

The following instructions will tell you how to upload your master image into an Azure storage account. If you don't already have an Azure storage account, follow the instructions in this article to create one.

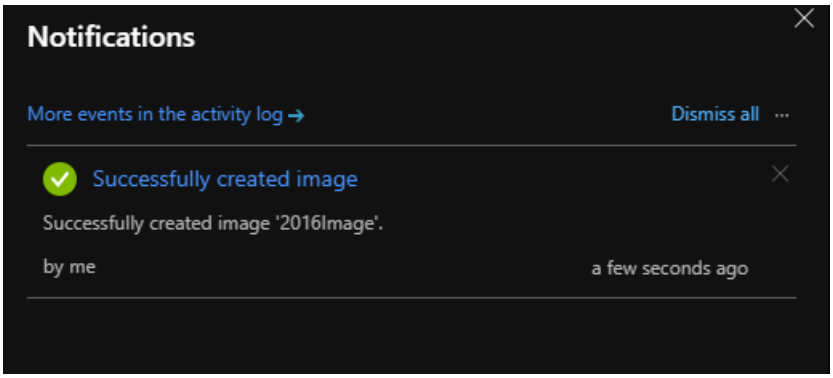
1. Convert the VM image (VHD) to Fixed if you haven't already. If you don't convert the image to Fixed, you can't successfully create the image.
2. Upload the VHD to a blob container in your storage account. You can upload quickly with the Storage Explorer tool. To learn more about the Storage Explorer tool, see this article.



3. Next, go to the Azure portal in your browser and search for "Images." Your search should lead you to the **Create image** page, as shown in the following screenshot:



4. Once you've created the image, you should see a notification like the one in the following screenshot:



Next steps

Now that you have an image, you can create or update host pools. To learn more about how to create and update host pools, see the following articles:

- Create a host pool with an Azure Resource Manager template
- Tutorial: Create a host pool with Azure Marketplace
- Create a host pool with PowerShell
- Create a profile container for a host pool using a file share
- Configure the Azure Virtual Desktop load-balancing method

If you encountered a connectivity problem after preparing or customizing your VHD image, check out the troubleshooting guide for help.

Install Office on a master VHD image

12/6/2021 • 4 minutes to read • Edit Online

This article tells you how to install Microsoft 365 Apps for enterprise, OneDrive, and other common applications on a master virtual hard disk (VHD) image for upload to Azure. If your users need to access certain line of business (LOB) applications, we recommend you install them after completing the instructions in this article.

This article assumes you've already created a virtual machine (VM). If not, see Prepare and customize a master VHD image

This article also assumes you have elevated access on the VM, whether it's provisioned in Azure or Hyper-V Manager. If not, see Elevate access to manage all Azure subscription and management groups.

> **NOTE**
> These instructions are for a Azure Virtual Desktop-specific configuration that can be used with your organization's existing processes.

Install Office in shared computer activation mode

Shared computer activation lets you to deploy Microsoft 365 Apps for enterprise to a computer in your organization that is accessed by multiple users. For more information about shared computer activation, see Overview of shared computer activation for Microsoft 365 Apps.

Use the Office Deployment Tool to install Office. Windows 10 Enterprise multi-session only supports the following versions of Office:

- Microsoft 365 Apps for enterprise
- Microsoft 365 Apps for business that comes with a Microsoft 365 Business Premium subscription

The Office Deployment Tool requires a configuration XML file. To customize the following sample, see the Configuration Options for the Office Deployment Tool.

This sample configuration XML we've provided will do the following things:

- Install Office from the Monthly Enterprise Channel and deliver updates from the Monthly Enterprise Channel.
- Use the x64 architecture.
- Disable automatic updates.
- Remove any existing installations of Office and migrate their settings.
- Enable shared computer activation.

> **NOTE**
> Visio's stencil search feature may not work as expected in Azure Virtual Desktop.

Here's what this sample configuration XML won't do:

- Install Skype for Business
- Install OneDrive in per-user mode. To learn more, see Install OneDrive in per-machine mode.

> **NOTE**
>
> Shared Computer Activation can be set up through Group Policy Objects (GPOs) or registry settings. The GPO is located at **Computer Configuration\Policies\Administrative Templates\Microsoft Office 2016 (Machine)\Licensing Settings**

The Office Deployment Tool contains setup.exe. To install Office, run the following command in a command line:

```
Setup.exe /configure configuration.xml
```

Sample configuration.xml

The following XML sample will install the Monthly Enterprise Channel release.

```
<Configuration>
  <Add OfficeClientEdition="64" Channel="MonthlyEnterprise">
    <Product ID="O365ProPlusRetail">
      <Language ID="en-US" />
      <Language ID="MatchOS" />
      <ExcludeApp ID="Groove" />
      <ExcludeApp ID="Lync" />
      <ExcludeApp ID="OneDrive" />
      <ExcludeApp ID="Teams" />
    </Product>
  </Add>
  <RemoveMSI/>
  <Updates Enabled="FALSE"/>
  <Display Level="None" AcceptEULA="TRUE" />
  <Logging Level="Standard" Path="%temp%\WVDOfficeInstall" />
  <Property Name="FORCEAPPSHUTDOWN" Value="TRUE"/>
  <Property Name="SharedComputerLicensing" Value="1"/>
</Configuration>
```

> **NOTE**
>
> The Office team recommends using 64-bit install for the **OfficeClientEdition** parameter.

After installing Office, you can update the default Office behavior. Run the following commands individually or in a batch file to update the behavior.

```
rem Mount the default user registry hive
reg load HKU\TempDefault C:\Users\Default\NTUSER.DAT
rem Must be executed with default registry hive mounted.
reg add HKU\TempDefault\SOFTWARE\Policies\Microsoft\office\16.0\common /v InsiderSlabBehavior /t REG_DWORD
/d 2 /f
rem Set Outlook's Cached Exchange Mode behavior
rem Must be executed with default registry hive mounted.
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v enable /t REG_DWORD
/d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v syncwindowsetting
/t REG_DWORD /d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v
CalendarSyncWindowSetting /t REG_DWORD /d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v
CalendarSyncWindowSettingMonths  /t REG_DWORD /d 1 /f
rem Unmount the default user registry hive
reg unload HKU\TempDefault

rem Set the Office Update UI behavior.
reg add HKLM\SOFTWARE\Policies\Microsoft\office\16.0\common\officeupdate /v hideupdatenotifications /t
REG_DWORD /d 1 /f
reg add HKLM\SOFTWARE\Policies\Microsoft\office\16.0\common\officeupdate /v hideenabledisableupdates /t
REG_DWORD /d 1 /f
```

Install OneDrive in per-machine mode

OneDrive is normally installed per-user. In this environment, it should be installed per-machine.

Here's how to install OneDrive in per-machine mode:

1. First, create a location to stage the OneDrive installer. A local disk folder or [\\unc] (file://unc) location is fine.
2. Download OneDriveSetup.exe to your staged location with this link: https://aka.ms/OneDriveWVD-Installer
3. If you installed office with OneDrive by omitting **<ExcludeApp ID="OneDrive" />**, uninstall any existing OneDrive per-user installations from an elevated command prompt by running the following command:

   ```
   "[staged location]\OneDriveSetup.exe" /uninstall
   ```

4. Run this command from an elevated command prompt to set the **AllUsersInstall** registry value:

   ```
   REG ADD "HKLM\Software\Microsoft\OneDrive" /v "AllUsersInstall" /t REG_DWORD /d 1 /reg:64
   ```

5. Run this command to install OneDrive in per-machine mode:

   ```
   Run "[staged location]\OneDriveSetup.exe" /allusers
   ```

6. Run this command to configure OneDrive to start at sign in for all users:

   ```
   REG ADD "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v OneDrive /t REG_SZ /d "C:\Program
   Files (x86)\Microsoft OneDrive\OneDrive.exe /background" /f
   ```

7. Enable **Silently configure user account** by running the following command.

```
REG ADD "HKLM\SOFTWARE\Policies\Microsoft\OneDrive" /v "SilentAccountConfig" /t REG_DWORD /d 1 /f
```

8. Redirect and move Windows known folders to OneDrive by running the following command.

```
REG ADD "HKLM\SOFTWARE\Policies\Microsoft\OneDrive" /v "KFMSilentOptIn" /t REG_SZ /d "<your-AzureAdTenantId>" /f
```

Microsoft Teams and Skype for Business

Azure Virtual Desktop doesn't support Skype for Business.

For help with installing Microsoft Teams, see Use Microsoft Teams on Azure Virtual desktop.

Next steps

Now that you've added Office to the image, you can continue to customize your master VHD image. See Prepare and customize a master VHD image.

Scale session hosts using Azure Automation

12/6/2021 • 16 minutes to read • Edit Online

You can reduce your total Azure Virtual Desktop deployment cost by scaling your virtual machines (VMs). This means shutting down and deallocating session host VMs during off-peak usage hours, then turning them back on and reallocating them during peak hours.

In this article, you'll learn about the scaling tool built with the Azure Automation account and Azure Logic App that automatically scales session host VMs in your Azure Virtual Desktop environment. To learn how to use the scaling tool, skip ahead to Prerequisites.

How the scaling tool works

The scaling tool provides a low-cost automation option for customers who want to optimize their session host VM costs.

You can use the scaling tool to:

- Schedule VMs to start and stop based on Peak and Off-Peak business hours.
- Scale out VMs based on number of sessions per CPU core.
- Scale in VMs during Off-Peak hours, leaving the minimum number of session host VMs running.

The scaling tool uses a combination of an Azure Automation account, a PowerShell runbook, a webhook, and the Azure Logic App to function. When the tool runs, Azure Logic App calls a webhook to start the Azure Automation runbook. The runbook then creates a job.

During peak usage time, the job checks the current number of sessions and the VM capacity of the current running session host for each host pool. It uses this information to calculate if the running session host VMs can support existing sessions based on the *SessionThresholdPerCPU* parameter defined for the **CreateOrUpdateAzLogicApp.ps1** file. If the session host VMs can't support existing sessions, the job starts additional session host VMs in the host pool.

> **NOTE**
>
> *SessionThresholdPerCPU* doesn't restrict the number of sessions on the VM. This parameter only determines when new VMs need to be started to load-balance the connections. To restrict the number of sessions, you need to follow the instructions Update-AzWvdHostPool to configure the *MaxSessionLimit* parameter accordingly.

During the off-peak usage time, the job determines how many session host VMs should be shut down based on the *MinimumNumberOfRDSH* parameter. If you set the *LimitSecondsToForceLogOffUser* parameter to a non-zero positive value, the job will set the session host VMs to drain mode to prevent new sessions from connecting to the hosts. The job will then notify any currently signed in users to save their work, wait the configured amount of time, and then force the users to sign out. Once all user sessions on the session host VM have been signed out, the job will shut down the VM. After the VM shuts down, the job will reset its session host drain mode.

> **NOTE**
> If you manually set the session host VM to drain mode, the job won't manage the session host VM. If the session host VM is running and set to drain mode, it will be treated as unavailable, which will make the job start additional VMs to handle the load. We recommend you tag any Azure VMs before you manually set them to drain mode. You can name the tag with the *MaintenanceTagName* parameter when you create Azure Logic App Scheduler later. Tags will help you distinguish these VMs from the ones the scaling tool manages. Setting the maintenance tag also prevents the scaling tool from making changes to the VM until you remove the tag.

If you set the *LimitSecondsToForceLogOffUser* parameter to zero, the job allows the session configuration setting in specified group policies to handle signing off user sessions. To see these group policies, go to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Session Time Limits**. If there are any active sessions on a session host VM, the job will leave the session host VM running. If there aren't any active sessions, the job will shut down the session host VM.

During any time, the job also takes host pool's *MaxSessionLimit* into account to determine if the current number of sessions is more than 90% of the maximum capacity. If it is, the job will start additional session host VMs.

The job runs periodically based on a set recurrence interval. You can change this interval based on the size of your Azure Virtual Desktop environment, but remember that starting and shutting down VMs can take some time, so remember to account for the delay. We recommend setting the recurrence interval to every 15 minutes.

However, the tool also has the following limitations:

- This solution applies only to pooled multi-session session host VMs.
- This solution manages VMs in any region, but can only be used in the same subscription as your Azure Automation account and Azure Logic App.
- The maximum runtime of a job in the runbook is 3 hours. If starting or stopping the VMs in the host pool takes longer than that, the job will fail. For more details, see Shared resources.
- At least one VM or session host needs to be turned on for the scaling algorithm to work properly.
- The scaling tool doesn't support scaling based on CPU or memory.
- Scaling only works with existing hosts in the host pool. The scaling tool doesn't support scaling new session hosts.

> **NOTE**
> The scaling tool controls the load balancing mode of the host pool it's currently scaling. The tool uses breadth-first load balancing mode for both peak and off-peak hours.

Prerequisites

Before you start setting up the scaling tool, make sure you have the following things ready:

- An Azure Virtual Desktop host pool
- Session host pool VMs configured and registered with the Azure Virtual Desktop service
- A user with Contributor access on Azure subscription

The machine you use to deploy the tool must have:

- Windows PowerShell 5.1 or later
- The Microsoft Az PowerShell module

If you have everything ready, then let's get started.

Create or update an Azure Automation account

> **NOTE**
> If you already have an Azure Automation account with a runbook running an older version of the scaling script, all you need to do is follow the instructions below to make sure it's updated.

First, you'll need an Azure Automation account to run the PowerShell runbook. The process this section describes is valid even if you have an existing Azure Automation account that you want to use to set up the PowerShell runbook. Here's how to set it up:

1. Open Windows PowerShell.
2. Run the following cmdlet to sign in to your Azure account.

```
Login-AzAccount
```

> **NOTE**
> Your account must have contributor rights on the Azure subscription where you want to deploy the scaling tool.

3. Run the following cmdlet to download the script for creating the Azure Automation account:

```
New-Item -ItemType Directory -Path "C:\Temp" -Force
Set-Location -Path "C:\Temp"
$Uri = "https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/wvd-scaling-
script/CreateOrUpdateAzAutoAccount.ps1"
# Download the script
Invoke-WebRequest -Uri $Uri -OutFile ".\CreateOrUpdateAzAutoAccount.ps1"
```

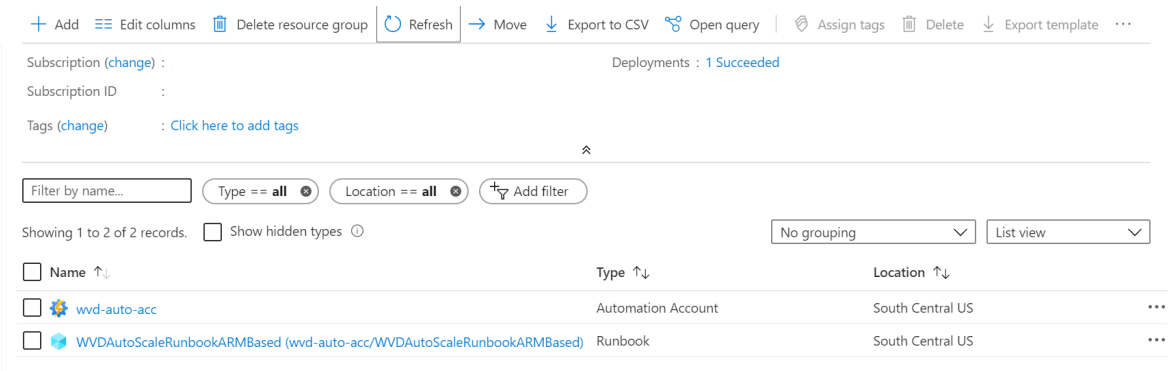
4. Run the following cmdlet to execute the script and create the Azure Automation account. You can either fill in values for the parameters or comment them to use their defaults.

```
$Params = @{
     "AADTenantId"           = "<Azure_Active_Directory_tenant_ID>"   # Optional. If not specified,
it will use the current Azure context
     "SubscriptionId"        = "<Azure_subscription_ID>"              # Optional. If not specified,
it will use the current Azure context
     "UseARMAPI"             = $true
     "ResourceGroupName"     = "<Resource_group_name>"                # Optional. Default:
"WVDAutoScaleResourceGroup"
     "AutomationAccountName" = "<Automation_account_name>"            # Optional. Default:
"WVDAutoScaleAutomationAccount"
     "Location"              = "<Azure_region_for_deployment>"
     "WorkspaceName"         = "<Log_analytics_workspace_name>"       # Optional. If specified, Log
Analytics will be used to configure the custom log table that the runbook PowerShell script can send
logs to
}

.\CreateOrUpdateAzAutoAccount.ps1 @Params
```

5. The cmdlet's output will include a webhook URI. Make sure to keep a record of the URI because you'll use it as a parameter when you set up the execution schedule for the Azure Logic App.
6. If you specified the parameter **WorkspaceName** for Log Analytics, the cmdlet's output will also include the Log Analytics Workspace ID and its Primary Key. Make sure to remember URI because you'll need to

use it again later as a parameter when you set up the execution schedule for the Azure Logic App.

7. After you've set up your Azure Automation account, sign in to your Azure subscription and check to make sure your Azure Automation account and the relevant runbook have appeared in your specified resource group, as shown in the following image:



   To check if your webhook is where it should be, select the name of your runbook. Next, go to your runbook's Resources section and select **Webhooks**.

Create an Azure Automation Run As account

Now that you have an Azure Automation account, you'll also need to create an Azure Automation Run As account if you don't have one already. This account will let the tool access your Azure resources.

An Azure Automation Run As account provides authentication for managing resources in Azure with Azure cmdlets. When you create a Run As account, it creates a new service principal user in Azure Active Directory and assigns the Contributor role to the service principal user at the subscription level. An Azure Run As account is a great way to authenticate securely with certificates and a service principal name without needing to store a username and password in a credential object. To learn more about Run As account authentication, see Limit Run As account permissions.

Any user who's a member of the Subscription Admins role and coadministrator of the subscription can create a Run As account.

To create a Run As account in your Azure Automation account:

1. In the Azure portal, select **All services**. In the list of resources, enter and select **Automation accounts**.
2. On the **Automation accounts** page, select the name of your Azure Automation account.
3. In the pane on the left side of the window, select **Run As accounts** under the **Account Settings** section.
4. Select **Azure Run As account**. When the **Add Azure Run As account** pane appears, review the overview information, and then select **Create** to start the account creation process.
5. Wait a few minutes for Azure to create the Run As account. You can track the creation progress in the menu under Notifications.
6. When the process finishes, it will create an asset named **AzureRunAsConnection** in the specified Azure Automation account. Select **Azure Run As account**. The connection asset holds the application ID, tenant ID, subscription ID, and certificate thumbprint. You can also find the same information on the **Connections** page. To go to this page, in the pane on the left side of the window, select **Connections** under the **Shared Resources** section and click on the connection asset named **AzureRunAsConnection**.

Create the Azure Logic App and execution schedule

Finally, you'll need to create the Azure Logic App and set up an execution schedule for your new scaling tool. First, download and import the Desktop Virtualization PowerShell module to use in your PowerShell session if you haven't already.

1. Open Windows PowerShell.
2. Run the following cmdlet to sign in to your Azure account.

```
Login-AzAccount
```

3. Run the following cmdlet to download the script for creating the Azure Logic App.

```
New-Item -ItemType Directory -Path "C:\Temp" -Force
Set-Location -Path "C:\Temp"
$Uri = "https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/wvd-scaling-
script/CreateOrUpdateAzLogicApp.ps1"
# Download the script
Invoke-WebRequest -Uri $Uri -OutFile ".\CreateOrUpdateAzLogicApp.ps1"
```

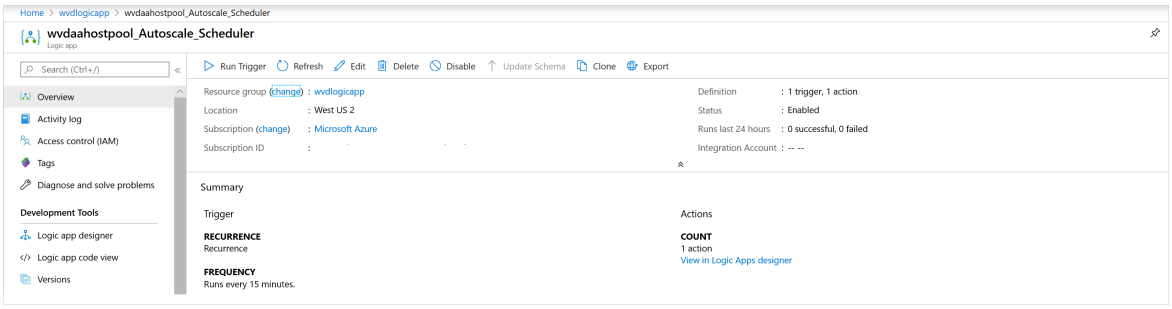
4. Run the following PowerShell script to create the Azure Logic App and execution schedule for your host pool

> **NOTE**
> You'll need to run this script for each host pool you want to autoscale, but you need only one Azure Automation account.

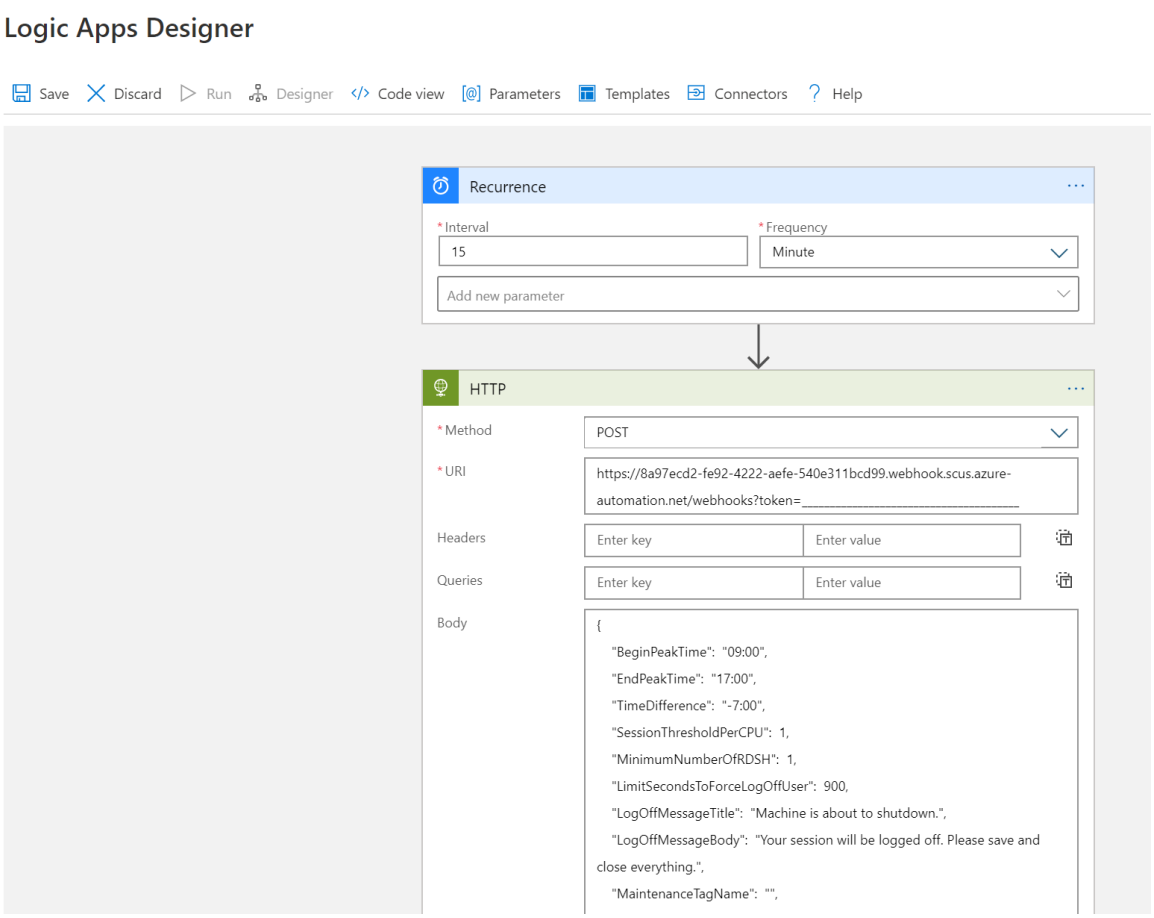
```
$AADTenantId = (Get-AzContext).Tenant.Id

$AzSubscription = Get-AzSubscription | Out-GridView -OutputMode:Single -Title "Select your Azure
Subscription"
Select-AzSubscription -Subscription $AzSubscription.Id

$ResourceGroup = Get-AzResourceGroup | Out-GridView -OutputMode:Single -Title "Select the resource
group for the new Azure Logic App"

$WVDHostPool = Get-AzResource -ResourceType "Microsoft.DesktopVirtualization/hostpools" | Out-
GridView -OutputMode:Single -Title "Select the host pool you'd like to scale"

$LogAnalyticsWorkspaceId = Read-Host -Prompt "If you want to use Log Analytics, enter the Log
Analytics Workspace ID returned by when you created the Azure Automation account, otherwise leave it
blank"
$LogAnalyticsPrimaryKey = Read-Host -Prompt "If you want to use Log Analytics, enter the Log
Analytics Primary Key returned by when you created the Azure Automation account, otherwise leave it
blank"
$RecurrenceInterval = Read-Host -Prompt "Enter how often you'd like the job to run in minutes, e.g.
'15'"
$BeginPeakTime = Read-Host -Prompt "Enter the start time for peak hours in local time, e.g. 9:00"
$EndPeakTime = Read-Host -Prompt "Enter the end time for peak hours in local time, e.g. 18:00"
$TimeDifference = Read-Host -Prompt "Enter the time difference between local time and UTC in hours,
e.g. +5:30"
$SessionThresholdPerCPU = Read-Host -Prompt "Enter the maximum number of sessions per CPU that will
be used as a threshold to determine when new session host VMs need to be started during peak hours"
$MinimumNumberOfRDSH = Read-Host -Prompt "Enter the minimum number of session host VMs to keep
running during off-peak hours"
$MaintenanceTagName = Read-Host -Prompt "Enter the name of the Tag associated with VMs you don't want
to be managed by this scaling tool"
$LimitSecondsToForceLogOffUser = Read-Host -Prompt "Enter the number of seconds to wait before
automatically signing out users. If set to 0, any session host VM that has user sessions, will be
left untouched"
$LogOffMessageTitle = Read-Host -Prompt "Enter the title of the message sent to the user before they
```

```
are forced to sign out"
$LogOffMessageBody = Read-Host -Prompt "Enter the body of the message sent to the user before they
are forced to sign out"

$AutoAccount = Get-AzAutomationAccount | Out-GridView -OutputMode:Single -Title "Select the Azure
Automation account"
$AutoAccountConnection = Get-AzAutomationConnection -ResourceGroupName $AutoAccount.ResourceGroupName
-AutomationAccountName $AutoAccount.AutomationAccountName | Out-GridView -OutputMode:Single -Title
"Select the Azure RunAs connection asset"

$WebhookURIAutoVar = Get-AzAutomationVariable -Name 'WebhookURIARMBased' -ResourceGroupName
$AutoAccount.ResourceGroupName -AutomationAccountName $AutoAccount.AutomationAccountName

$Params = @{
     "AADTenantId"                  = $AADTenantId                              # Optional. If not
specified, it will use the current Azure context
     "SubscriptionID"               = $AzSubscription.Id                        # Optional. If not
specified, it will use the current Azure context
     "ResourceGroupName"            = $ResourceGroup.ResourceGroupName          # Optional. Default:
"WVDAutoScaleResourceGroup"
     "Location"                     = $ResourceGroup.Location                   # Optional. Default:
"West US2"
     "UseARMAPI"                    = $true
     "HostPoolName"                 = $WVDHostPool.Name
     "HostPoolResourceGroupName"    = $WVDHostPool.ResourceGroupName            # Optional. Default:
same as ResourceGroupName param value
     "LogAnalyticsWorkspaceId"      = $LogAnalyticsWorkspaceId                  # Optional. If not
specified, script will not log to the Log Analytics
     "LogAnalyticsPrimaryKey"       = $LogAnalyticsPrimaryKey                   # Optional. If not
specified, script will not log to the Log Analytics
     "ConnectionAssetName"          = $AutoAccountConnection.Name               # Optional. Default:
"AzureRunAsConnection"
     "RecurrenceInterval"           = $RecurrenceInterval                       # Optional. Default:
15
     "BeginPeakTime"                = $BeginPeakTime                            # Optional. Default:
"09:00"
     "EndPeakTime"                  = $EndPeakTime                              # Optional. Default:
"17:00"
     "TimeDifference"               = $TimeDifference                           # Optional. Default:
"-7:00"
     "SessionThresholdPerCPU"       = $SessionThresholdPerCPU                   # Optional. Default:
1
     "MinimumNumberOfRDSH"          = $MinimumNumberOfRDSH                      # Optional. Default:
1
     "MaintenanceTagName"           = $MaintenanceTagName                       # Optional.
     "LimitSecondsToForceLogOffUser" = $LimitSecondsToForceLogOffUser           # Optional. Default:
1
     "LogOffMessageTitle"           = $LogOffMessageTitle                       # Optional. Default:
"Machine is about to shutdown."
     "LogOffMessageBody"            = $LogOffMessageBody                        # Optional. Default:
"Your session will be logged off. Please save and close everything."
     "WebhookURI"                   = $WebhookURIAutoVar.Value
}

.\CreateOrUpdateAzLogicApp.ps1 @Params
```

After you run the script, the Azure Logic App should appear in a resource group, as shown in the following image.



To make changes to the execution schedule, such as changing the recurrence interval or time zone, go to the Azure Logic App autoscale scheduler and select **Edit** to go to the Azure Logic App Designer.



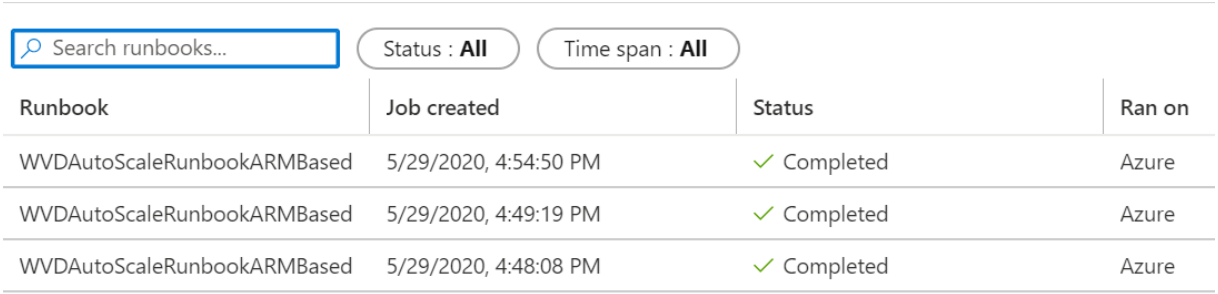
Manage your scaling tool

Now that you've created your scaling tool, you can access its output. This section describes a few features you might find helpful.

View job status

You can view a summarized status of all runbook jobs or view a more in-depth status of a specific runbook job in the Azure portal.

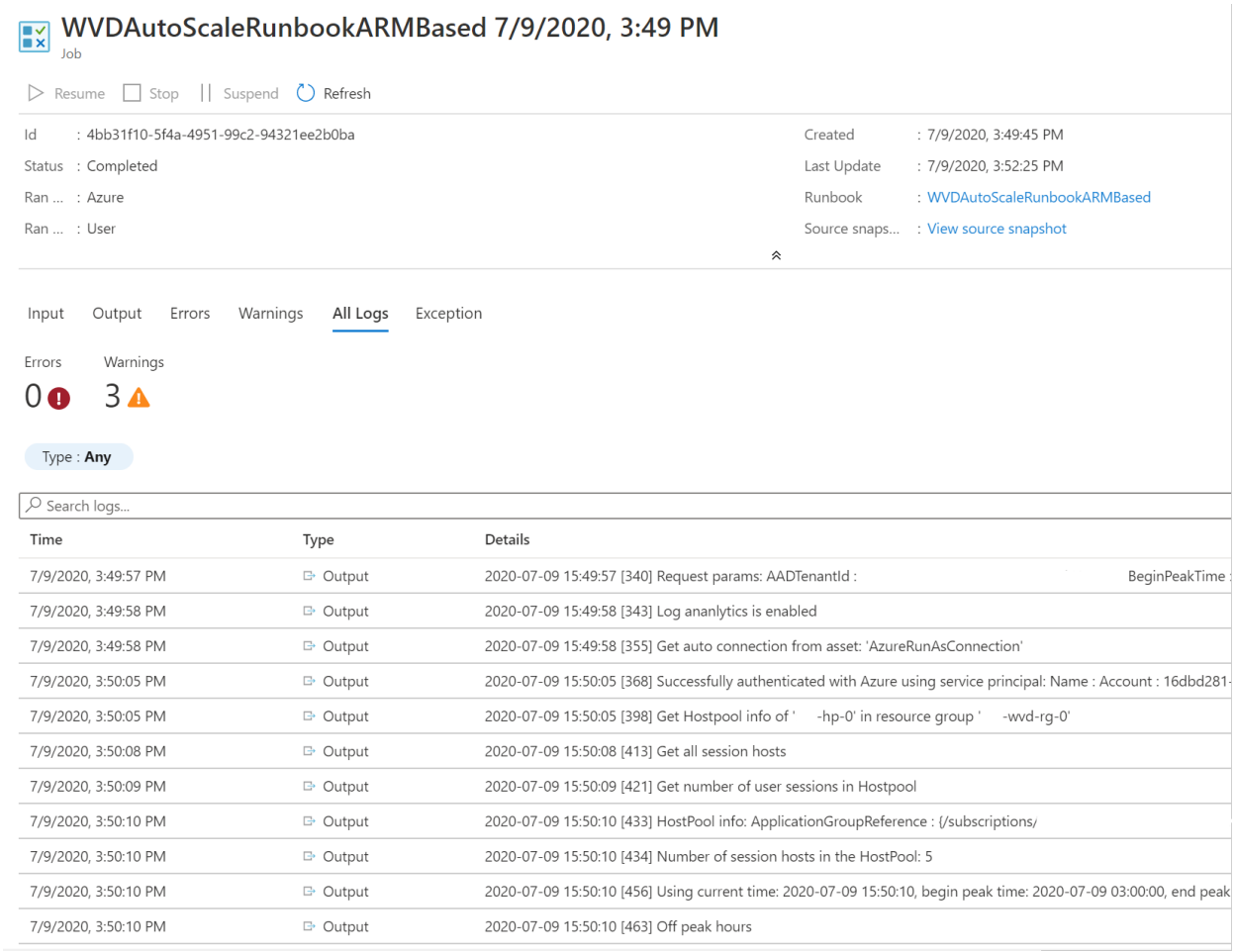
On the right of your selected Azure Automation account, under "Job Statistics," you can view a list of summaries of all runbook jobs. Opening the **Jobs** page on the left side of the window shows current job statuses, start times, and completion times.



View logs and scaling tool output

You can view the logs of scale-out and scale-in operations by opening your runbook and selecting the job.

Navigate to the runbook in your resource group hosting the Azure Automation account and select **Overview**. On the overview page, select a job under **Recent Jobs** to view its scaling tool output, as shown in the following image.



Check the runbook script version number

You can check which version of the runbook script you're using by opening the runbook file in your Azure Automation account and selecting **View**. A script for the runbook will appear on the right side of the screen. In the script, you'll see the version number in the format `v#.#.#` under the `SYNOPSIS` section. You can find the latest version number here. If you don't see a version number in your runbook script, that means you're running an earlier version of the script and you should update it right away. If you need to update your runbook script, follow the instructions in Create or update an Azure Automation account.

Reporting issues

When you report an issue, you'll need to provide the following information to help us troubleshoot:

- A complete log from the **All Logs** tab in the job that caused the issue. To learn how to get the log, follow the instructions in View logs and scaling tool output. If there's any sensitive or private information in the

log, you can remove it before submitting the issue to us.

- The version of the runbook script you're using. To find out how to get the version number, see Check the runbook script version number
- The version number of each of the following PowerShell modules installed in your Azure Automation account. To find these modules, open Azure Automation account, select **Modules** under the **Shared Resources** section in the pane on the left side of the window, and then search for the module's name.
  - Az.Accounts
  - Az.Compute
  - Az.Resources
  - Az.Automation
  - OMSIngestionAPI
  - Az.DesktopVirtualization
- The expiration date for your Run As account. To find this, open your Azure Automation account, then select **Run As accounts** under **Account Settings** in the pane on the left side of the window. The expiration date should be under **Azure Run As account**.

Log Analytics

If you decided to use Log Analytics, you can view all the log data in a custom log named **WVDTenantScale_CL** under **Custom Logs** in the **Logs** view of your Log Analytics Workspace. We've listed some sample queries you might find helpful.

- To see all logs for a host pool, enter the following query

```
WVDTenantScale_CL
| where hostpoolName_s == "<host_pool_name>"
| project TimeStampUTC = TimeGenerated, TimeStampLocal = TimeStamp_s, HostPool = hostpoolName_s,
LineNumAndMessage = logmessage_s, AADTenantId = TenantId
```

- To view the total number of currently running session host VMs and active user sessions in your host pool, enter the following query

```
WVDTenantScale_CL
| where logmessage_s contains "Number of running session hosts:"
     or logmessage_s contains "Number of user sessions:"
     or logmessage_s contains "Number of user sessions per Core:"
| where hostpoolName_s == "<host_pool_name>"
| project TimeStampUTC = TimeGenerated, TimeStampLocal = TimeStamp_s, HostPool = hostpoolName_s,
LineNumAndMessage = logmessage_s, AADTenantId = TenantId
```

- To view the status of all session host VMs in a host pool, enter the following query

```
WVDTenantScale_CL
| where logmessage_s contains "Session host:"
| where hostpoolName_s == "<host_pool_name>"
| project TimeStampUTC = TimeGenerated, TimeStampLocal = TimeStamp_s, HostPool = hostpoolName_s,
LineNumAndMessage = logmessage_s, AADTenantId = TenantId
```

- To view any errors and warnings, enter the following query

```
WVDTenantScale_CL
| where logmessage_s contains "ERROR:" or logmessage_s contains "WARN:"
| project TimeStampUTC = TimeGenerated, TimeStampLocal = TimeStamp_s, HostPool = hostpoolName_s,
LineNumAndMessage = logmessage_s, AADTenantId = TenantId
```

Report issues

Issue reports for the scaling tool are currently being handled by Microsoft Support. When you make an issue report, make sure to follow the instructions in Reporting issues. If you have feedback about the tool or want to request new features, open a GitHub issue labeled "4-WVD-scaling-tool" on the RDS GitHub page.

Autoscale (preview) for Azure Virtual Desktop host pools

12/6/2021 • 9 minutes to read • Edit Online

IMPORTANT

The autoscale feature is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

The autoscale feature (preview) lets you scale your Azure Virtual Desktop deployment's virtual machines (VMs) up or down to optimize deployment costs. Based on your needs, you can make a scaling plan based on:

- Time of day
- Specific days of the week
- Session limits per session host

NOTE

- Azure Virtual Desktop (classic) doesn't support the autoscale feature.
- Autoscale doesn't support Azure Virtual Desktop for Azure Stack HCI
- Autsoscale doesn't support scaling of ephemeral disks.

For best results, we recommend using autoscale with VMs you deployed with Azure Virtual Desktop Azure Resource Manager templates or first-party tools from Microsoft.

IMPORTANT

The preview version of this feature currently has the following limitations:

- You can only use autoscale in the Azure public cloud.
- You can only configure autoscale with the Azure portal.
- You can only deploy the scaling plan to US and European regions.

Requirements

Before you create your first scaling plan, make sure you follow these guidelines:

- You can currently only configure autoscale with pooled existing host pools.
- All host pools you autoscale must have a configured MaxSessionLimit parameter. Don't use the default value. You can configure this value in the host pool settings in the Azure portal or run the New-AZWvdHostPool or Update-AZWvdHostPool cmdlets in PowerShell.
- You must grant Azure Virtual Desktop access to manage power on your VM Compute resources.

Create a Custom RBAC role

To start creating a scaling plan, you'll first need to create a custom Role-based Access Control (RBAC) role in your subscription. This role will allow Windows Virtual Desktop to power manage all VMs in your subscription. It will also let the service apply actions on both host pools and VMs when there are no active user sessions.

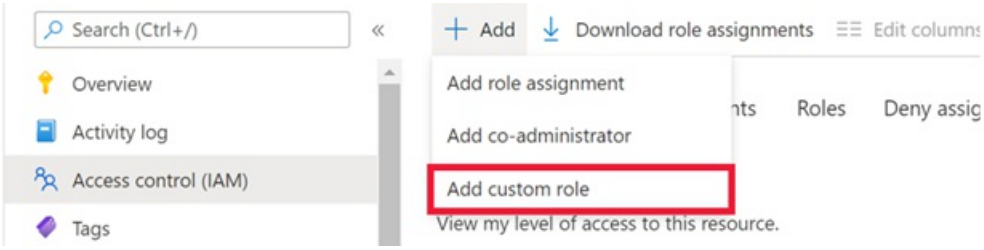
To create the custom role, follow the instructions in Azure custom roles while using the following JSON template. This template already includes any permissions you need. For more detailed instructions, see Assign custom roles with the Azure portal.

```
{
"properties": {
"roleName": "Autoscale",
"description": "Friendly description.",
"assignableScopes": [
"/subscriptions/<SubscriptionID>"
],
 "permissions": [
  {
  "actions": [
    "Microsoft.Insights/eventtypes/values/read",
    "Microsoft.Compute/virtualMachines/deallocate/action",
    "Microsoft.Compute/virtualMachines/restart/action",
    "Microsoft.Compute/virtualMachines/powerOff/action",
    "Microsoft.Compute/virtualMachines/start/action",
    "Microsoft.Compute/virtualMachines/read",
    "Microsoft.DesktopVirtualization/hostpools/read",
    "Microsoft.DesktopVirtualization/hostpools/write",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/read",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/write",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/delete",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/read",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/sendMessage/action",
    "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/read"
],
  "notActions": [],
  "dataActions": [],
  "notDataActions": []
  }
 ]
}
}
```

Assign custom roles with the Azure portal

To create and assign the custom role to your subscription with the Azure portal:

1. Open the Azure portal and go to **Subscriptions**.
2. Select the **+** button in the top left-hand corner of the screen, then select **Add custom role** from the drop-down menu, as shown in the following screenshot.



3. Next, name the custom role and add a description. We recommend you name the role "Autoscale."
4. On the **Permissions** tab, add the following permissions to the subscription you're assigning the role to:

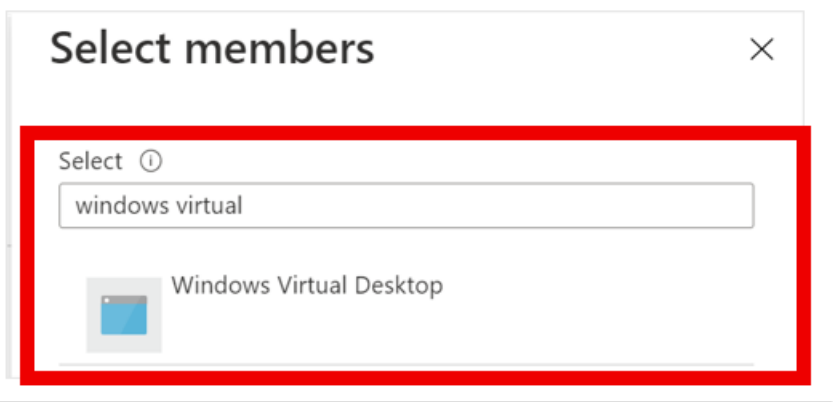
```
"Microsoft.Insights/eventtypes/values/read"
   "Microsoft.Compute/virtualMachines/deallocate/action"
   "Microsoft.Compute/virtualMachines/restart/action"
   "Microsoft.Compute/virtualMachines/powerOff/action"
   "Microsoft.Compute/virtualMachines/start/action"
   "Microsoft.Compute/virtualMachines/read"
   "Microsoft.DesktopVirtualization/hostpools/read"
   "Microsoft.DesktopVirtualization/hostpools/write"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/read"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/write"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/delete"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/read"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/sendMessage/action"
   "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/read"
```

5. When you're finished, select **Ok**.

After that, you'll need to assign the role to grant access to Azure Virtual Desktop.

To assign the custom role to grant access:

1. In the **Access control (IAM) tab**, select **Add role assignments**.
2. Select the role you just created and continue to the next screen.
3. Select **+Select members**. In the search bar, enter and select **Windows Virtual Desktop**, as shown in the following screenshot. When you have a Azure Virtual Desktop (classic) deployment and an Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects, you will see two apps with the same name. Select them both.



4. Select **Review + assign** to complete the assignment.

How creating a scaling plan works

Before you create your plan, keep the following things in mind:

- You can assign one scaling plan to one or more host pools of the same host pool type. The scaling plan's schedule will also be applied across all assigned host pools.
- You can only associate one scaling plan per host pool. If you assign a single scaling plan to multiple host pools, those host pools can't be assigned to another scaling plan.
- A scaling plan can only operate in its configured time zone.
- A scaling plan can have one or multiple schedules. For example, different schedules during weekdays versus the weekend.
- Make sure you understand usage patterns before defining your schedule. You'll need to schedule around

the following times of day:

  - Ramp-up: the start of the day, when usage picks up.
  - Peak hours: the time of day when usage is highest.
  - Ramp-down: when usage tapers off. This is usually when you shut down your VMs to save costs.
  - Off-peak hours: the time with the lowest possible usage. You can define the maximum number of VMs that can be active during this time.
- The scaling plan will take effect as soon as you enable it.

Also, keep these limitations in mind:

- Don't use autoscale in combination with other scaling Microsoft or third-party scaling tools. Ensure that you disable those for the host pools you apply the scaling plans.
- Autoscale overwrites drain mode, so make sure to use exclusion tags when updating VMs in host pools.
- Autoscale ignores existing load-balancing algorithms in your host pool settings, and instead applies load balancing based on your schedule configuration.

Create a scaling plan

To create a scaling plan:

1. Sign in to the Azure portal at https://portal.azure.com.
2. Go to **Azure Virtual Desktop** > **Scaling Plans**, then select **Create**.
3. In the **Basics** tab, look under **Project details** and select the name of the subscription you will assign the scaling plan to.
4. If you want to make a new resource group, select **Create new**. If you want to use an existing resource group, select its name from the drop-down menu.
5. Enter a name for the scaling plan into the **Name** field.
6. Optionally, you can also add a "friendly" name that will be displayed to your users and a description for your plan.
7. For **Region**, select a region for your scaling plan. The metadata for the object will be stored in the geography associated with the region. To learn more about regions, see Data locations.
8. For **Time zone**, select the time zone you'll use with your plan.
9. In **Exclusion tags**, enter tags for VMs you don't want to include in scaling operations. For example, you might want to use this functionality for maintenance. When you have set VMs on Drain mode use the tag so autoscale doesn't override drain mode.
10. Select **Next**, which should take you to the **Schedules** tab.

Configure a schedule

Schedules let you define when autoscale activates ramp-up and ramp-down modes throughout the day. In each phase of the schedule, autoscale only turns off VMs when a session host has no sessions active. The default values you'll see when you try to create a schedule are the suggested values for weekdays, but you can change them as needed.

To create or change a schedule:

1. In the **Schedules** tab, select **Add schedule**.

2. Enter a name for your schedule into the **Schedule name** field.
3. In the **Repeat on** field, select which days your schedule will repeat on.
4. In the **Ramp up** tab, fill out the following fields:
   - For **Start time**, select a time from the drop-down menu to start preparing VMs for peak business hours.
   - For **Load balancing algorithm**, we recommend selecting **breadth-first algorithm**. Breadth-first load balancing will distribute users across existing VMs to keep access times fast.

     > **NOTE**
     > The load balancing preference you select here will override the one you selected for your original host pool settings.

   - For **Minimum percentage of hosts**, enter the percentage of session hosts you want to always remain on in this phase. If the percentage you enter isn't a whole number, it's rounded up to the nearest whole number. For example, in a host pool of 7 session hosts, if the minimum percentage of hosts is **10%** for the ramp-up hours, one VM will always stay on during ramp-up hours and the autoscale feature won't turn off this VM.
   - For **Capacity threshold**, enter the percentage of available host pool capacity that will trigger a scaling action to take place. For example, if 2 session hosts in the host pool with a max session limit of 20 are turned on, the available host pool capacity is 40. If you set the capacity threshold to **75%** and the session hosts have more than 30 user sessions, the autoscale feature will turn on a third session host. This will then change the available host pool capacity from 40 to 60.
5. In the **Peak hours** tab, fill out the following fields:
   - For **Start time**, enter a start time for when your usage rate is highest during the day. Make sure the time is in the same time zone you specified for your scaling plan. This time is also the end time for the ramp-up phase.
   - For **Load balancing**, you can select either breadth-first or depth-first load balancing. Breadth-first load balancing distributes new user sessions across all available sessions in the host pool. Depth-first load balancing distributes new sessions to any available session host with the highest number of connections that hasn't reached its session limit yet. For more information about load-balancing types, see Configure the Azure Virtual Desktop load-balancing method.

     > **NOTE**
     > You can't change the capacity threshold here. Instead, the setting you entered in **Ramp-up** will carry over to this setting.

   - For **Ramp-down**, you'll enter values into similar fields to **Ramp-up**, but this time it will be for when your host pool usage drops off. This will include the following fields:
     - Start time
     - Load-balancing algorithm
     - Minimum percentage of hosts (%)
     - Capacity threshold (%)
     - Force logoff users

> **IMPORTANT**
> If you've enabled the autoscale feature to force users to sign out during ramp-down, the feature will choose the session host with the lowest number of user sessions to shut down. The autoscale feature will put the session host in drain mode, send all active user sessions a notification telling them they'll be signed out, and then sign out all users after the specified wait time is over. After the autoscale feature signs out all user sessions, it then deallocates the VM. If you haven't enabled forced sign out during ramp-down, session hosts with no active or disconnected sessions will be deallocated.

- Likewise, **Off-peak hours** works the same way as **Peak hours**:
  - Start time, which is also the end of the ramp-down period.
  - Load-balancing algorithm. We recommend choosing **depth-first** to gradually reduce the number of session hosts based on sessions on each VM.
  - Just like peak hours, you can't configure the capacity threshold here. Instead, the value you entered in **Ramp-down** will carry over.

Assign host pools

Now that you've set up your scaling plan, it's time to assign the plan to your host pools. Select the check box next to each host pool you want to include. If you don't want to enable autoscale, unselect all check boxes. You can always return to this setting later and change it.

> **NOTE**
> When you create or update a scaling plan that's already assigned to host pools, its changes will immediately be applied.

Add tags

After that, you'll need to enter tags. Tags are name and value pairs that categorize resources for consolidated billing. You can apply the same tag to multiple resources and resource groups. To learn more about tagging resources, see Use tags to organize your Azure resources.

> **NOTE**
> If you change resource settings on other tabs after creating tags, your tags will be automatically updated.

Once you're done, go to the **Review + create** tab and select **Create** to deploy your host pool.

Next steps

Now that you've created your scaling plan, here are some things you can do:

- Assign your scaling plan to new and existing host pools
- Enable diagnostics for your scaling plan

Enable scaling plans for existing and new host pools (preview)

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

The autoscale feature is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

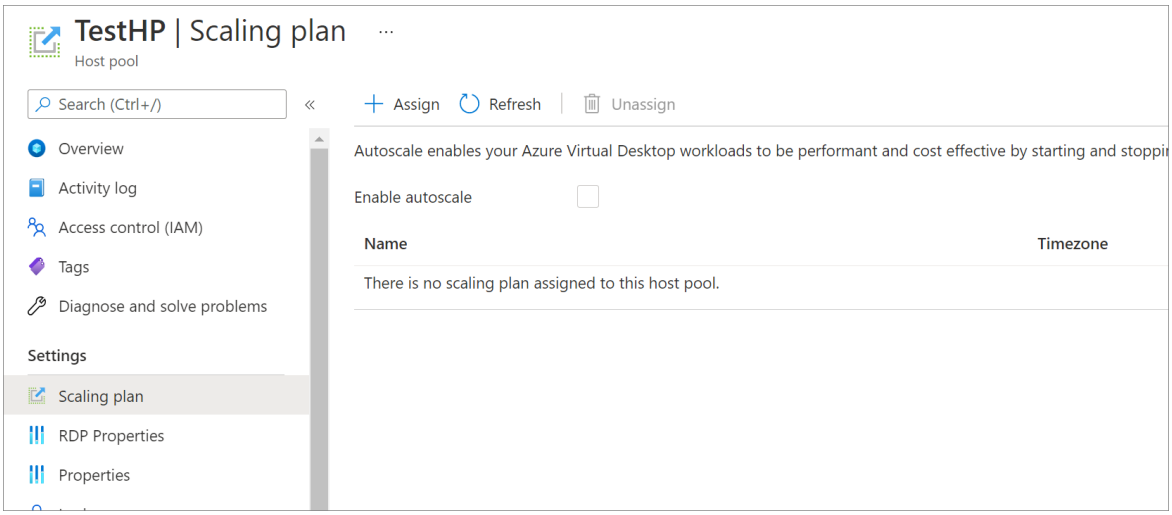
You can enable scaling plans for any existing host pools in your deployment. When you apply your scaling plan to the host pool, the plan will also apply to all session hosts within that host pool. Scaling also automatically applies to any new session hosts you create in your assigned host pool.

If you disable a scaling plan, all assigned resources will remain in the scaling state they were in at the time you disabled it.

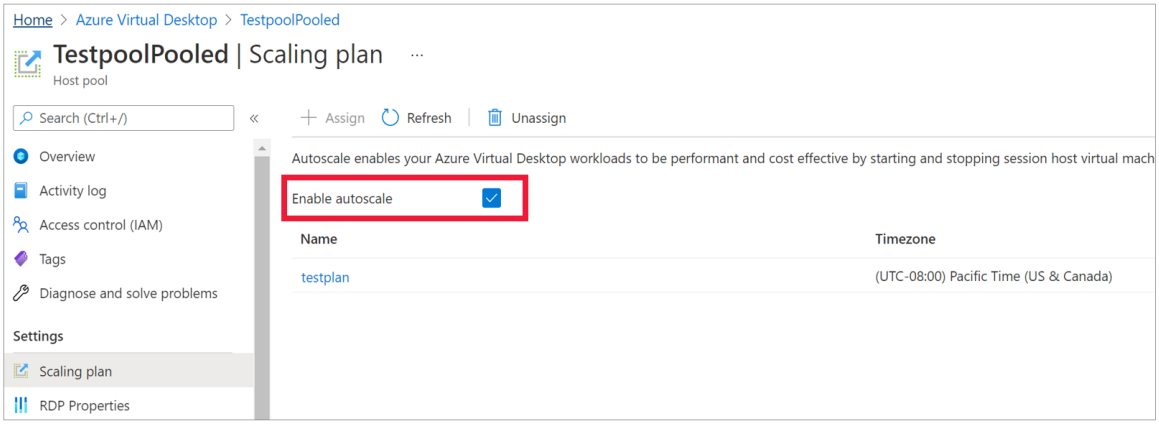
Assign a scaling plan to an existing host pool

To assign a scaling plan to an existing host pool:

1. Open the Azure portal.
2. Go to **Windows Virtual Desktop**.
3. Select **Host pools**, then go to **Scaling plan** and select **New**.
4. Select **Scaling plan**, then select **+ Assign** to assign a scaling plan to an unassigned host pool, as shown in the following screenshot.



If you've enabled the scaling plan during deployment, then you'll also have the option to disable the plan for the selected host pool in the **Scaling plan** menu by unselecting the **Enable scaling plan** checkbox, as shown in the following screenshot.



Edit an existing scaling plan

To edit an existing scaling plan:

1. Sign in to the Azure portal at https://portal.azure.com.
2. Go to **Azure Virtual Desktop**.
3. Select **Scaling plan**, then select the name of the scaling plan you want to edit. The settings window should open.
4. To edit the plan's display name, description, time zone, or exclusion tags, go to the **Properties** tab.
5. To assign host pools or edit schedules, go to the **Manage** tab.

Next steps

- Review how to create a scaling plan at Autoscale (preview) for Azure Virtual Desktop session hosts.
- Learn how to troubleshoot your scaling plan at Enable diagnostics for your scaling plan.

Set up diagnostics for the autoscale feature (preview)

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

The autoscale feature is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Diagnostics lets you monitor potential issues and fix them before they interfere with your autoscale (preview) scaling plan.

Currently, you can either send diagnostic logs for the autoscale feature to an Azure Storage account or consume logs with the Events hub. If you're using an Azure Storage account, make sure it's in the same region as your scaling plan. Learn more about diagnostic settings at Create diagnostic settings. For more information about resource log data ingestion time, see Log data ingestion time in Azure Monitor.

Enable diagnostics for scaling plans

To enable diagnostics for your scaling plan:

1. Sign in to the Azure portal at https://portal.azure.com.
2. Select **Scaling plans**, then select the scaling plan you'd like the report to track.
3. Go to **Diagnostic Settings** and select **Add diagnostic setting**.
4. Enter a name for the diagnostic setting.
5. Next, select **Autoscale** and choose either **storage account** or **event hub** depending on where you want to send the report.
6. Select **Save**.

Set log location in Azure Storage

After you've configured your diagnostic settings, you can find the logs by following these instructions:

1. In the Azure portal, go to the storage group you sent the diagnostic logs to.
2. Select **Containers**. A folder called **insight-logs-autoscaling** should open.
3. Select the **insight-logs-autoscaling folder** and open the log you want to review. Open folders within that folder until you see the JSON file, then select all items in that folder, right-click, and download them to your local computer.
4. Finally, open the JSON file in the text editor of your choice.

View diagnostic logs

Now that you've opened the JSON file, let's do a quick overview of what each piece of the report means:

- The **CorrelationID** is the ID that you need to show when you create a support case.

- **OperationName** is the type of operation running while the issue happened.
- **ResultType** is the result of the operation. This item can show you where issues are if you notice any incomplete results.
- **Message** is the error message that provides information on the incomplete operation. This message can include links to important troubleshooting documentation, so review it carefully.

The following JSON file is an example of what you'll see when you open a report:


```
{
    "host_Ring": "R0",
    "Level": 4,
    "ActivityId": "c1111111-1111-1111-b111-11111cd1ba1b1",
    "time": "2021-08-31T16:00:46.5246835Z",
    "resourceId": "/SUBSCRIPTIONS/AD11111A-1C21-1CF1-A7DE-
CB1111E1D111/RESOURCEGROUPS/TEST/PROVIDERS/MICROSOFT.DESKTOPVIRTUALIZATION/SCALINGPLANS/TESTPLAN",
    "operationName": "HostPoolLoadBalancerTypeUpdated",
    "category": "Autoscale",
    "resultType": "Succeeded",
    "level": "Informational",
    "correlationId": "35ec619b-b5d8-5b5f-9242-824aa4d2b878",
    "properties": {
        "Message": "Host pool's load balancing algorithm updated",
        "HostPoolArmPath": "/subscriptions/AD11111A-1C21-1CF1-A7DE-
CB1111E1D111/resourcegroups/test/providers/microsoft.desktopvirtualization/hostpools/testHostPool ",
        "PreviousLoadBalancerType": "BreadthFirst",
        "NewLoadBalancerType": "DepthFirst"
    }
}. L
```


Next steps

- Review how to create a scaling plan at Autoscale for Azure Virtual Desktop session hosts.
- Assign your scaling plan to new or existing host pools.

Customize the feed for Azure Virtual Desktop users

12/6/2021 • 2 minutes to read • Edit Online

> **IMPORTANT**
> This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

You can customize the feed so the RemoteApp and remote desktop resources appear in a recognizable way for your users.

Prerequisites

This article assumes you've already downloaded and installed the Azure Virtual Desktop PowerShell module. If you haven't, follow the instructions in Set up the PowerShell module.

Customize the display name for a RemoteApp

You can change the display name for a published RemoteApp by setting the friendly name. By default, the friendly name is the same as the name of the RemoteApp program.

To retrieve a list of published RemoteApps for an app group, run the following PowerShell cmdlet:

```
Get-AzWvdApplication -ResourceGroupName <resourcegroupname> -ApplicationGroupName <appgroupname>
```

To assign a friendly name to a RemoteApp, run the following cmdlet with the required parameters:

```
Update-AzWvdApplication -ResourceGroupName <resourcegroupname> -ApplicationGroupName <appgroupname> -Name
<applicationname> -FriendlyName <newfriendlyname>
```

For example, let's say you retrieved the current applications with the following example cmdlet:

```
Get-AzWvdApplication -ResourceGroupName 0301RG -ApplicationGroupName 0301RAG | format-list
```

The output would look like this:

```
CommandLineArgument :
CommandLineSetting  : DoNotAllow
Description         :
FilePath            : C:\Program Files\Windows NT\Accessories\wordpad.exe
FriendlyName        : Microsoft Word
IconContent         : {0, 0, 1, 0…}
IconHash            : --iom0PS6XLu-EMMlHWVW3F7LLsNt63Zz2K10RE0_64
IconIndex           : 0
IconPath            : C:\Program Files\Windows NT\Accessories\wordpad.exe
Id                  :
/subscriptions/<subid>/resourcegroups/0301RG/providers/Microsoft.DesktopVirtualization/applicationgroups/030
1RAG/applications/Microsoft Word
Name                : 0301RAG/Microsoft Word
ShowInPortal        : False
Type                : Microsoft.DesktopVirtualization/applicationgroups/applications
```

To update the friendly name, run this cmdlet:

```
Update-AzWvdApplication -GroupName 0301RAG -Name "Microsoft Word" -FriendlyName "WordUpdate" -
ResourceGroupName 0301RG -IconIndex 0 -IconPath "C:\Program Files\Windows NT\Accessories\wordpad.exe" -
ShowInPortal:$true -CommandLineSetting DoNotallow -FilePath "C:\Program Files\Windows
NT\Accessories\wordpad.exe"
```

To confirm you've successfully updated the friendly name, run this cmdlet:

```
Get-AzWvdApplication -ResourceGroupName 0301RG -ApplicationGroupName 0301RAG | format-list FriendlyName
```

The cmdlet should give you the following output:

```
FriendlyName        : WordUpdate
```

Customize the display name for a Remote Desktop

You can change the display name for a published remote desktop by setting a friendly name. If you manually created a host pool and desktop app group through PowerShell, the default friendly name is "Session Desktop." If you created a host pool and desktop app group through the GitHub Azure Resource Manager template or the Azure Marketplace offering, the default friendly name is the same as the host pool name.

To retrieve the remote desktop resource, run the following PowerShell cmdlet:

```
Get-AzWvdDesktop -ResourceGroupName <resourcegroupname> -ApplicationGroupName <appgroupname> -Name
<applicationname>
```

To assign a friendly name to the remote desktop resource, run the following PowerShell cmdlet:

```
Update-AzWvdDesktop -ResourceGroupName <resourcegroupname> -ApplicationGroupName <appgroupname> -Name
<applicationname> -FriendlyName <newfriendlyname>
```

Customize a display name in Azure portal

You can change the display name for a published remote desktop by setting a friendly name using the Azure portal.

1. Sign in to the Azure portal at https://portal.azure.com.

2. Search for **Azure Virtual Desktop**.
3. Under Services, select **Azure Virtual Desktop**.
4. On the Azure Virtual Desktop page, select **Application groups** on the left side of the screen, then select the name of the app group you want to edit. (For example, if you want to edit the display name of the desktop app group, select the app group named **Desktop**.)
5. Select **Applications** in the menu on the left side of the screen.
6. Select the application you want to update, then enter a new **Display name**.
7. Select **Save**. The application you edited should now display the updated name.

Next steps

Now that you've customized the feed for users, you can sign in to a Azure Virtual Desktop client to test it out. To do so, continue to the Connect to Azure Virtual Desktop How-tos:

- Connect with Windows 10 or Windows 7
- Connect with the web client
- Connect with the Android client
- Connect with the iOS client
- Connect with the macOS client

Use Log Analytics for the diagnostics feature

12/6/2021 • 6 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Azure Virtual Desktop uses Azure Monitor for monitoring and alerts like many other Azure services. This lets admins identify issues through a single interface. The service creates activity logs for both user and administrative actions. Each activity log falls under the following categories:

- Management Activities:
  - Track whether attempts to change Azure Virtual Desktop objects using APIs or PowerShell are successful. For example, can someone successfully create a host pool using PowerShell?
- Feed:
  - Can users successfully subscribe to workspaces?
  - Do users see all resources published in the Remote Desktop client?
- Connections:
  - When users initiate and complete connections to the service.
- Host registration:
  - Was the session host successfully registered with the service upon connecting?
- Errors:
  - Are users encountering any issues with specific activities? This feature can generate a table that tracks activity data for you as long as the information is joined with the activities.
- Checkpoints:
  - Specific steps in the lifetime of an activity that were reached. For example, during a session, a user was load balanced to a particular host, then the user was signed on during a connection, and so on.

Connections that don't reach Azure Virtual Desktop won't show up in diagnostics results because the diagnostics role service itself is part of Azure Virtual Desktop. Azure Virtual Desktop connection issues can happen when the user is experiencing network connectivity issues.

Azure Monitor lets you analyze Azure Virtual Desktop data and review virtual machine (VM) performance counters, all within the same tool. This article will tell you more about how to enable diagnostics for your Azure Virtual Desktop environment.

NOTE

To learn how to monitor your VMs in Azure, see Monitoring Azure virtual machines with Azure Monitor. Also, make sure to review the performance counter thresholds for a better understanding of your user experience on the session host.

Before you get started

Before you can use Log Analytics, you'll need to create a workspace. To do that, follow the instructions in one of the following two articles:

- If you prefer using Azure portal, see Create a Log Analytics workspace in Azure portal.

- If you prefer PowerShell, see Create a Log Analytics workspace with PowerShell.

After you've created your workspace, follow the instructions in Connect Windows computers to Azure Monitor to get the following information:

- The workspace ID
- The primary key of your workspace

You'll need this information later in the setup process.

Make sure to review permission management for Azure Monitor to enable data access for those who monitor and maintain your Azure Virtual Desktop environment. For more information, see Get started with roles, permissions, and security with Azure Monitor.

Push diagnostics data to your workspace

You can push diagnostics data from your Azure Virtual Desktop objects into the Log Analytics for your workspace. You can set up this feature right away when you first create your objects.

To set up Log Analytics for a new object:

1. Sign in to the Azure portal and go to **Azure Virtual Desktop**.
2. Navigate to the object (such as a host pool, app group, or workspace) that you want to capture logs and events for.
3. Select **Diagnostic settings** in the menu on the left side of the screen.
4. Select **Add diagnostic setting** in the menu that appears on the right side of the screen.

   The options shown in the Diagnostic Settings page will vary depending on what kind of object you're editing.

   For example, when you're enabling diagnostics for an app group, you'll see options to configure checkpoints, errors, and management. For workspaces, these categories configure a feed to track when users subscribe to the list of apps. To learn more about diagnostic settings see Create diagnostic setting to collect resource logs and metrics in Azure.

   > **IMPORTANT**
   > Remember to enable diagnostics for each Azure Resource Manager object that you want to monitor. Data will be available for activities after diagnostics has been enabled. It might take a few hours after first set-up.

5. Enter a name for your settings configuration, then select **Send to Log Analytics**. The name you use shouldn't have spaces and should conform to Azure naming conventions. As part of the logs, you can select all the options that you want added to your Log Analytics, such as Checkpoint, Error, Management, and so on.
6. Select **Save**.

> **NOTE**
> Log Analytics gives you the option to stream data to Event Hubs or archive it in a storage account. To learn more about this feature, see Stream Azure monitoring data to an event hub and Archive Azure resource logs to storage account.

How to access Log Analytics

You can access Log Analytics workspaces on the Azure portal or Azure Monitor.

Access Log Analytics on a Log Analytics workspace

1. Sign in to the Azure portal.
2. Search for **Log Analytics workspace**.
3. Under Services, select **Log Analytics workspaces**.
4. From the list, select the workspace you configured for your Azure Virtual Desktop object.
5. Once in your workspace, select **Logs**. You can filter out your menu list with the **Search** function.

Access Log Analytics on Azure Monitor

1. Sign into the Azure portal
2. Search for and select **Monitor**.
3. Select **Logs**.
4. Follow the instructions in the logging page to set the scope of your query.
5. You are ready to query diagnostics. All diagnostics tables have a "WVD" prefix.

> **NOTE**
> For more detailed information about the tables stored in Azure Monitor Logs, see the Azure Monitor data refence. All tables related to Azure Virtual Desktop are labeled "WVD."

Cadence for sending diagnostic events

Diagnostic events are sent to Log Analytics when completed.

Log Analytics only reports in these intermediate states for connection activities:

- Started: when a user selects and connects to an app or desktop in the Remote Desktop client.
- Connected: when the user successfully connects to the VM where the app or desktop is hosted.
- Completed: when the user or server disconnects the session the activity took place in.

Example queries

Access example queries through the Azure Monitor Log Analytics UI:

1. Go to your Log Analytics workspace, and then select **Logs**. The example query UI is shown automatically.
2. Change the filter to **Category**.
3. Select **Azure Virtual Desktop** to review available queries.
4. Select **Run** to run the selected query.

Learn more about the sample query interface in Saved queries in Azure Monitor Log Analytics.

The following query list lets you review connection information or issues for a single user. You can run these queries in the Log Analytics query editor. For each query, replace `userupn` with the UPN of the user you want to look up.

To find all connections for a single user:

```
WVDConnections
|where UserName == "userupn"
|take 100
|sort by TimeGenerated asc, CorrelationId
```

To find the number of times a user connected per day:

```
WVDConnections
|where UserName == "userupn"
|take 100
|sort by TimeGenerated asc, CorrelationId
|summarize dcount(CorrelationId) by bin(TimeGenerated, 1d)
```

To find session duration by user:

```
let Events = WVDConnections | where UserName == "userupn" ;
Events
| where State == "Connected"
| project CorrelationId , UserName, ResourceAlias , StartTime=TimeGenerated
| join (Events
| where State == "Completed"
| project EndTime=TimeGenerated, CorrelationId)
on CorrelationId
| project Duration = EndTime - StartTime, ResourceAlias
| sort by Duration asc
```

To find errors for a specific user:

```
WVDErrors
| where UserName == "userupn"
|take 100
```

To find out whether a specific error occurred for other users:

```
WVDErrors
| where CodeSymbolic =="ErrorSymbolicCode"
| summarize count(UserName) by CodeSymbolic
```

NOTE

- When a user opens Full Desktop, their app usage in the session isn't tracked as checkpoints in the WVDCheckpoints table.
- The ResourcesAlias column in the WVDConnections table shows whether a user has connected to a full desktop or a published app. The column only shows the first app they open during the connection. Any published apps the user opens are tracked in WVDCheckpoints.
- The WVDErrors table shows you management errors, host registration issues, and other issues that happen while the user subscribes to a list of apps or desktops.
- WVDErrors helps you to identify issues that can be resolved by admin tasks. The value on ServiceError always says “false” for those types of issues. If ServiceError = “true”, you'll need to escalate the issue to Microsoft. Ensure you provide the CorrelationID for the errors you escalate.

Next steps

To review common error scenarios that the diagnostics feature can identify for you, see Identify and diagnose

issues.

Publish built-in apps in Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

This article will tell you how to publish apps in your Azure Virtual Desktop environment.

Publish built-in apps

To publish a built-in app:

1. Connect to one of the virtual machines in your host pool.
2. Get the **PackageFamilyName** of the app you want to publish by following the instructions in this article.
3. Finally, run the following cmdlet with `<PackageFamilyName>` replaced by the **PackageFamilyName** you found in the previous step:

```
New-AzWvdApplication -Name <applicationname> -ResourceGroupName <resourcegroupname> -
ApplicationGroupName <appgroupname> -FilePath "shell:appsFolder\<PackageFamilyName>!App" -
CommandLineSetting <Allow|Require|DoNotAllow> -IconIndex 0 -IconPath <iconpath> -ShowInPortal:$true
```

NOTE

Azure Virtual Desktop only supports publishing apps with install locations that begin with `C:\Program Files\WindowsApps`.

Update app icons

After you publish an app, it will have the default Windows app icon instead of its regular icon picture. To change the icon to its regular icon, put the image of the icon you want on a network share. Supported image formats are PNG, BMP, GIF, JPG, JPEG, and ICO.

Publish Microsoft Edge

The process you use to publish Microsoft Edge is a little different from the publishing process for other apps. To publish Microsoft Edge with the default homepage, run this cmdlet:

```
New-AzWvdApplication -Name -ResourceGroupName -ApplicationGroupName -FilePath
"shell:Appsfolder\Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge" -CommandLineSetting
<Allow|Require|DoNotAllow> -iconPath
"C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\microsoftedge.exe" -iconIndex 0 -
ShowInPortal:$true
```

Next steps

- Learn about how to configure feeds to organize how apps are displayed for users at Customize feed for Azure Virtual Desktop users.
- Learn about the MSIX app attach feature at Set up MSIX app attach.

Set up MSIX app attach with the Azure portal

12/6/2021 • 7 minutes to read • Edit Online

This article will walk you through how to set up MSIX app attach in a Azure Virtual Desktop environment.

Requirements

Here's what you need to configure MSIX app attach:

- A functioning Azure Virtual Desktop deployment. To learn how to deploy Azure Virtual Desktop (classic), see Create a tenant in Azure Virtual Desktop. To learn how to deploy Azure Virtual Desktop with Azure Resource Manager integration, see Create a host pool with the Azure portal.
- An Azure Virtual Desktop host pool with at least one active session host.
- The MSIX packaging tool.
- An MSIX-packaged application expanded into an MSIX image that's uploaded into a file share.
- A file share in your Azure Virtual Desktop deployment where the MSIX package will be stored.
- The file share where you uploaded the MSIX image must also be accessible to all virtual machines (VMs) in the host pool. Users will need read-only permissions to access the image.
- If the certificate isn't publicly trusted, follow the instructions in Install certificates.

Turn off automatic updates for MSIX app attach applications

Before you get started, you must disable automatic updates for MSIX app attach applications. To disable automatic updates, you'll need to run the following commands in an elevated command prompt:

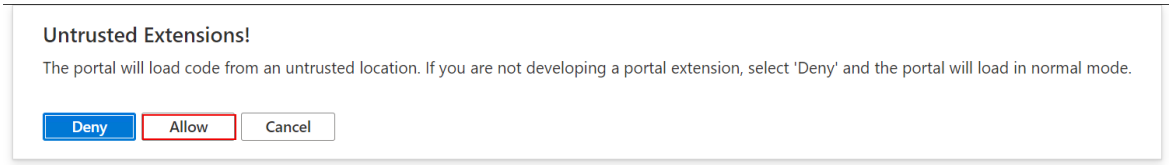
```
rem Disable Store auto update:

reg add HKLM\Software\Policies\Microsoft\WindowsStore /v AutoDownload /t REG_DWORD /d 0 /f
Schtasks /Change /Tn "\Microsoft\Windows\WindowsUpdate\Automatic app update" /Disable
Schtasks /Change /Tn "\Microsoft\Windows\WindowsUpdate\Scheduled Start" /Disable

rem Disable Content Delivery auto download apps that they want to promote to users:

reg add HKCU\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager /v PreInstalledAppsEnabled /t
REG_DWORD /d 0 /f

reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager\Debug /v
ContentDeliveryAllowedOverride /t REG_DWORD /d 0x2 /f
```

Configure the MSIX app attach management interface

Next, you'll need to download and configure the the MSIX app attach management interface for the Azure portal.

To set up the management interface:

1. Open the Azure portal.
2. If you get a prompt asking if you consider the extension trustworthy, select **Allow**.



Add an MSIX image to the host pool

Next you'll need to add the MSIX image to your host pool.

To add the MSIX image:

1. Open the Azure portal.
2. Enter **Azure Virtual Desktop** into the search bar, then select the service name.
3. Select the host pool where you plan to put the MSIX apps.
4. Select **MSIX packages** to open the data grid with all **MSIX packages** currently added to the host pool.
5. Select **+ Add** to open the **Add MSIX package** tab.
6. In the **Add MSIX package** tab, enter the following values:
   - For **MSIX image path**, enter a valid UNC path pointing to the MSIX image on the file share. (For example, `\\storageaccount.file.core.windows.net\msixshare\appfolder\MSIXimage.vhd`.) When you're done, select **Add** to interrogate the MSIX container to check if the path is valid.
   - For **MSIX package**, select the relevant MSIX package name from the drop-down menu. This menu will only be populated if you've entered a valid image path in **MSIX image path**.
   - For **Package applications**, make sure the list contains all MSIX applications you want to be available to users in your MSIX package.
   - Optionally, enter a **Display name** if you want your package to have a more user-friendly in your user deployments.
   - Make sure the **Version** has the correct version number.
   - Select the **Registration type** you want to use. Which one you use depends on your needs:
   - **On-demand registration** postpones the full registration of the MSIX application until the user starts the application. This is the registration type we recommend you use.
   - **Log on blocking** only registers while the user is signing in. We don't recommend this type because it can lead to longer sign-in times for users.
7. For **State**, select your preferred state.
   - The **Active** status lets users interact with the package.
   - The **Inactive** status causes Azure Virtual Desktop to ignore the package and not deliver it to users.
8. When you're done, select **Add**.

Publish MSIX apps to an app group

Next, you'll need to publish the apps into the package. You'll need to do this for both desktop and remote app application groups.

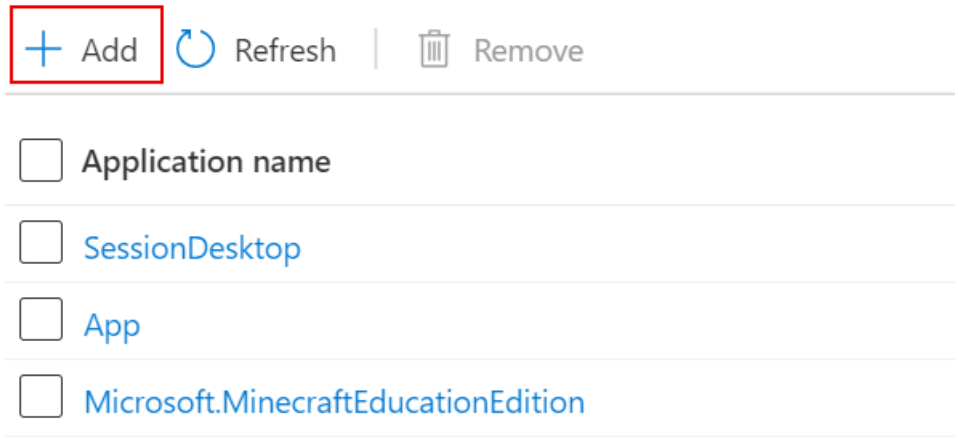
If you already have an MSIX image, skip ahead to Publish MSIX apps to an app group. If you want to test legacy applications, follow the instructions in Create an MSIX package from a desktop installer on a VM to convert the legacy application to an MSIX package.

To publish the apps:

1. In the Azure Virtual Desktop resource provider, select the **Application groups** tab.
2. Select the application group you want to publish the apps to.

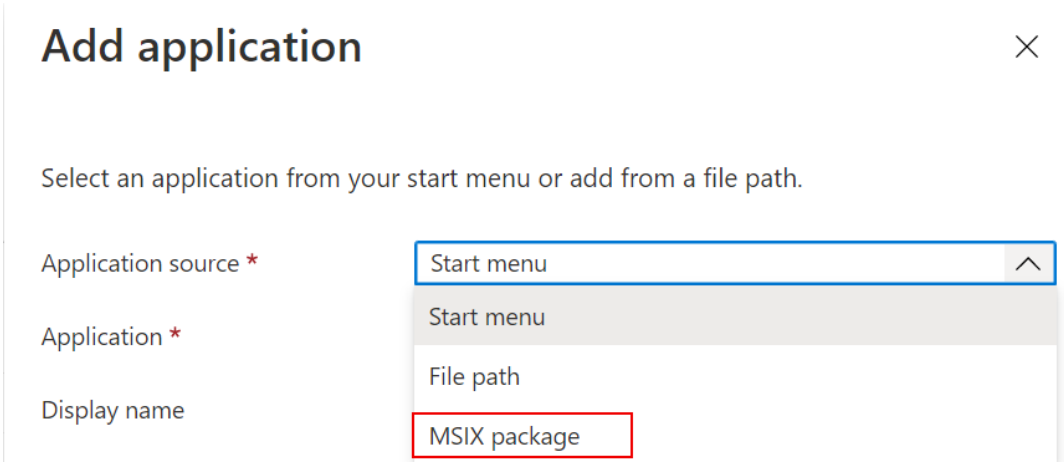
   > **NOTE**
   > MSIX applications can be delivered with MSIX app attach to both remote app and desktop app groups

3. Once you're in the app group, select the **Applications** tab. The **Applications** grid will display all existing apps within the app group.
4. Select **+ Add** to open the **Add application** tab.



5. For **Application source**, choose the source for your application.

   - If you're using a Desktop app group, choose **MSIX package**.



   - If you're using a remote app group, choose one of the following options:
     - Start menu
     - App path
     - MSIX package
   - For **Application name**, enter a descriptive name for the application.

   You can also configure the following optional features:

   - For **Display name**, enter a new name for the package that your users will see.

- For **Description**, enter a short description of the app package.
- If you're using a remote app group, you can also configure these options:
  - **Icon path**
  - **Icon index**
  - **Show in web feed**

6. When you're done, select **Save**.

> **NOTE**
> When a user is assigned to remote app group and desktop app group from the same host pool the desktop app group will be displayed in the feed.

Assign a user to an app group

After assigning MSIX apps to an app group, you'll need to grant users access to them. You can assign access by adding users or user groups to an app group with published MSIX applications. Follow the instructions in Manage app groups with the Azure portal to assign your users to an app group.

> **NOTE**
> MSIX app attach remote apps may disappear from the feed when you test remote apps during public preview. The apps don't appear because the host pool you're using in the evaluation environment is being served by an RD Broker in the production environment. Because the RD Broker in the production environment doesn't register the presence of the MSIX app attach remote apps, the apps won't appear in the feed.

Change MSIX package state

Next, you'll need to change the MSIX package state to either **Active** or **Inactive**, depending on what you want to do with the package. Active packages are packages your users can interact with once they're published. Inactive packages are ignored by Azure Virtual Desktop, so your users can't interact with the apps inside.

Change state with the Applications list

To change the package state with the Applications list:

1. Go to your host pool and select **MSIX packages**. You should see a list of all existing MSIX packages within the host pool.
2. Select the MSIX packages whose states you need to change, then select **Change state**.

Change state with update package

To change the package state with an update package:

1. Go to your host pool and select **MSIX packages**. You should see a list of all existing MSIX packages within the host pool.
2. Select the name of the package whose state you want to change from the MSIX package list. This will open the **Update package** tab.
3. Toggle the **State** switch to either **Inactive** or **Active**, then select **Save**.

Change MSIX package registration type

To change the package's registration type:

1. Select **MSIX packages**. You should see a list of all existing MSIX packages within the host pool.
2. Select **Package name in** the **MSIX packages grid** this will open the blade to update the package.
3. Toggle the **Registration type** via the **On-demand/Log on blocking** button as desired and select **Save**.

Remove an MSIX package

To remove an MSIX package from your host pool:

1. Select **MSIX packages**. You should see a list of all existing MSIX packages within the host pool.
2. Select the ellipsis on the right side the name of the package you want to delete, then select **Remove**.

Remove MSIX apps

To remove individual MSIX apps from your package:

1. Go to the host pool and select **Application groups**.
2. Select the application group you want to remove MSIX apps from.
3. Open the **Applications** tab.
4. Select the app you want to remove, then select **Remove**.

Next steps

Ask our community questions about this feature at the Azure Virtual Desktop TechCommunity.

You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.

Here are some other articles you might find helpful:

- MSIX app attach glossary
- MSIX app attach FAQ

Set up MSIX app attach using PowerShell

12/6/2021 • 5 minutes to read • Edit Online

In addition to the Azure portal, you can also set up MSIX app attach manually with PowerShell. This article will walk you through how to use PowerShell to set up MSIX app attach.

Requirements

Here's what you need to configure MSIX app attach:

- A functioning Azure Virtual Desktop deployment. To learn how to deploy Azure Virtual Desktop (classic), see Create a tenant in Azure Virtual Desktop. To learn how to deploy Azure Virtual Desktop with Azure Resource Manager integration, see Create a host pool with the Azure portal.
- A Azure Virtual Desktop host pool with at least one active session host.
- A Desktop remote app group.
- The MSIX packaging tool.
- An MSIX-packaged application expanded into an MSIX image that's uploaded into a file share.
- A file share in your Azure Virtual Desktop deployment where the MSIX package will be stored.
- The file share where you uploaded the MSIX image must also be accessible to all virtual machines (VMs) in the host pool. Users will need read-only permissions to access the image.
- Download and install PowerShell Core.
- Download the public preview Azure PowerShell module and expand it to a local folder.
- Install the Azure module by running the following cmdlet:

```
Install-Module -Name Az -Force
```

Sign in to Azure and import the module

Once you've got all the requirements ready, open PowerShell core in an elevated command prompt and run this cmdlet:

```
Connect-AzAccount
```

After you run it, authenticate your account using your credentials. In this case, you might be asked for a device URL or a token.

Import the Az.WindowsVirtualDesktop module

You'll need the Az.DesktopVirtualization module to follow the instructions in this article.

> **NOTE**
> For the public preview, we will provide the module as separate ZIP files that you must manually import.

Before you start, you can run the following cmdlet to see if the Az.DesktopVirtualization module is already installed on your session or VM:

```
Get-Module | Where-Object { $_.Name -Like "desktopvirtualization" }
```

If you wan to uninstall an existing copy of the module and start over, run this cmdlet:

```
Uninstall-Module Az.DesktopVirtualization
```

If the module is blocked on your VM, run this cmdlet to unblock it:

```
Unblock-File "<path>\Az.DesktopVirtualization.psm1"
```

With that cleanup out of the way, it's time to import the module.

1. Run the following cmdlet, then press the **R** key when prompted to agree to run the custom code.

   ```
   Import-Module -Name "<path>\Az.DesktopVirtualization.psm1" -Verbose
   ```

2. Once you've run the import cmdlet, check to see if it has the cmdlets for MSIX by running the following cmdlet:

   ```
   Get-Command -Module Az.DesktopVirtualization | Where-Object { $_.Name -match "MSIX" }
   ```

   If the cmdlets are there, the output should look like this:

   ```
   CommandType     Name                                               Version    Source

   -----------     ----                                               -------    ------

   Function        Expand-AzWvdMsixImage                              0.0
   Az.DesktopVirtualization

   Function        Get-AzWvdMsixPackage                               0.0
   Az.DesktopVirtualization

   Function        New-AzWvdMsixPackage                               0.0
   Az.DesktopVirtualization

   Function        Remove-AzWvdMsixPackage                            0.0
   Az.DesktopVirtualization

   Function        Update-AzWvdMsixPackage                            0.0
   Az.DesktopVirtualization
   ```

   If you don't see this output, close all PowerShell and PowerShell Core sessions and try again.

Set up helper variables

Once you've imported the module, you'll need to set up the helper variables. The following examples will show you how to do each one.

To get your subscription ID:

```
Get-AzContext -ListAvailable | fl
```

To select the context of an Azure tenant and subscription with a name:

```
$obj = Select-AzContext -Name "<Name>"
```

To set the subscription variable:

```
$subId = $obj.Subscription.Id
```

To set the workspace name:

```
$ws = "<WorksSpaceName>"
```

To set the host pool name:

```
$hp = "<HostPoolName>"
```

To set up the resource group where the session host VMs are configured:

```
$rg = "<ResourceGroupName>"
```

And finally, to confirm you've correctly set all the variables:

```
Get-AzWvdWorkspace -Name $ws -ResourceGroupName $rg -SubscriptionId $subID
```

Add an MSIX package to a host pool

Once you've set everything up, it's time to add the MSIX package to a host pool. To do that, you'll first need to get UNC path to the MSIX image.

Using the UNC path, run this cmdlet to expand the MSIX image:

```
$obj = Expand-AzWvdMsixImage -HostPoolName $hp -ResourceGroupName $rg -SubscriptionId $subID -Uri <UNCPath>
```

Run this cmdlet to add the MSIX package to your desired host pool:

```
New-AzWvdMsixPackage -HostPoolName $hp -ResourceGroupName $rg -SubscriptionId $subId -PackageAlias
$obj.PackageAlias -DisplayName <DisplayName> -ImagePath <UNCPath> -IsActive:$true
```

Once you're done, confirm the package was created with this cmdlet:

```
Get-AzWvdMsixPackage -HostPoolName $hp -ResourceGroupName $rg -SubscriptionId $subId | Where-Object
{$_.PackageFamilyName -eq $obj.PackageFamilyName}
```

Remove an MSIX package from a host pool

To remove a package from a host pool:

Get a list of all packages associated with a host pool with this cmdlet, then find the name of the package you want to remove in the output:

```
Get-AzWvdMsixPackage -HostPoolName $hp -ResourceGroupName $rg -SubscriptionId $subId
```

Alternatively, you can also get a particular package based on its display name with this cmdlet:

```
Get-AzWvdMsixPackage -HostPoolName $hp -ResourceGroupName $rg -SubscriptionId $subId | Where-Object {
$_.Name -like "Power" }
```

To remove the package, run this cmdlet:

```
Remove-AzWvdMsixPackage -FullName $obj.PackageFullName -HostPoolName $hp -ResourceGroupName $rg
```

Publish MSIX apps to an app group

You can only follow the instructions in this section if you've finished following the instructions in the previous sections. If you have a host pool with an active session host, at least one Desktop app group, and have added an MSIX package to the host pool, you're ready to go.

To publish an app from the MSIX package to an app group, you'll need to find its name, then use that name in the publishing cmdlet.

To publish an app:

Run this cmdlet to list all available app groups:

```
Get-AzWvdApplicationGroup -ResourceGroupName $rg -SubscriptionId $subId
```

When you've found the name of the app group you want to publish apps to, use its name in this cmdlet:

```
$grName = "<AppGroupName>"
```

Finally, you'll need to publish the app.

- To publish MSIX application to a desktop app group, run this cmdlet:

  ```
  New-AzWvdApplication -ResourceGroupName $rg -SubscriptionId $subId -Name PowerBi -ApplicationType
  MsixApplication -ApplicationGroupName $grName -MsixPackageFamilyName $obj.PackageFamilyName -
  CommandLineSetting 0
  ```

- To publish the app to a remote app group, run this cmdlet instead:

  ```
  New-AzWvdApplication -ResourceGroupName $rg -SubscriptionId $subId -Name PowerBi -ApplicationType
  MsixApplication -ApplicationGroupName $grName -MsixPackageFamilyName $obj.PackageFamilyName -
  CommandLineSetting 0 -MsixPackageApplicationId $obj.PackageApplication.AppId
  ```

> **NOTE**
> If a user is assigned to both a remote app group and a desktop app group in the same host pool, when the user connects to their remote desktop, they will see MSIX apps from both groups.

Next steps

Ask our community questions about this feature at the Azure Virtual Desktop TechCommunity.

You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.

Here are some other articles you might find helpful:

- MSIX app attach glossary
- MSIX app attach FAQ

Create PowerShell scripts for MSIX app attach

12/6/2021 • 6 minutes to read • Edit Online

This topic will walk you through how to set up PowerShell scripts for MSIX app attach.

Install certificates

You must install certificates on all session hosts in the host pool that will host the apps from your MSIX app attach packages.

If your app uses a certificate that isn't public-trusted or was self-signed, here's how to install it:

1. Right-click the package and select **Properties**.
2. In the window that appears, select the **Digital signatures** tab. There should be only one item in the list on the tab. Select that item to highlight the item, then select **Details**.
3. When the digital signature details window appears, select the **General** tab, then select **View Certificate**, then select **Install certificate**.
4. When the installer opens, select **local machine** as your storage location, then select **Next**.
5. If the installer asks you if you want to allow the app to make changes to your device, select **Yes**.
6. Select **Place all certificates in the following store**, then select **Browse**.
7. When the select certificate store window appears, select **Trusted people**, then select **OK**.
8. Select **Next** and **Finish**.

Enable Microsoft Hyper-V

Microsoft Hyper-V must be enabled because the `Mount-VHD` command is needed to stage and `Dismount-VHD` is needed to destage.

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

> **NOTE**
> This change will require that you restart the virtual machine.

Prepare PowerShell scripts for MSIX app attach

MSIX app attach has four distinct phases that must be performed in the following order:

1. Stage
2. Register
3. Deregister
4. Destage

Each phase creates a PowerShell script. Sample scripts for each phase are available here.

Stage PowerShell script

Before you update the PowerShell scripts, make sure you have the volume GUID of the volume in the VHD. To get the volume GUID:

1. Open the network share where the VHD is located inside the VM where you'll run the script.
2. Right-click the VHD and select **Mount**. This will mount the VHD to a drive letter.
3. After you mount the VHD, the **File Explorer** window will open. Capture the parent folder and update the **$parentFolder** variable

   > **NOTE**
   > If you don't see a parent folder, that means the MSIX wasn't expanded properly. Redo the previous section and try again.

4. Open the parent folder. If correctly expanded, you'll see a folder with the same name as the package. Update the **$packageName** variable to match the name of this folder.

   For example, `VSCodeUserSetup-x64-1.38.1_1.38.1.0_x64__8wekyb3d8bbwe`.

5. Open a command prompt and enter **mountvol**. This command will display a list of volumes and their GUIDs. Copy the GUID of the volume where the drive letter matches the drive you mounted your VHD to in step 2.

   For example, in this example output for the mountvol command, if you mounted your VHD to Drive C, you'll want to copy the value above `C:\`:

   ```
   Possible values for VolumeName along with current mount points are:

   \\?\Volume{a12b3456-0000-0000-0000-10000000000}\
   *** NO MOUNT POINTS ***

   \\?\Volume{c78d9012-0000-0000-0000-20000000000}\
       E:\

   \\?\Volume{d34e5678-0000-0000-0000-30000000000}\
       C:\
   ```

6. Update the **$volumeGuid** variable with the volume GUID you just copied.
7. Open an Admin PowerShell prompt and update the following PowerShell script with the variables that apply to your environment.

```
#MSIX app attach staging sample

#region variables
$vhdSrc="<path to vhd>"
$packageName = "<package name>"
$parentFolder = "<package parent folder>"
$parentFolder = "\" + $parentFolder + "\"
$volumeGuid = "<vol guid>"
$msixJunction = "C:\temp\AppAttach\"
#endregion

#region mountvhd
try
{
      Mount-Diskimage -ImagePath $vhdSrc -NoDriveLetter -Access ReadOnly
      Write-Host ("Mounting of " + $vhdSrc + " was completed!") -BackgroundColor Green
}
catch
{
      Write-Host ("Mounting of " + $vhdSrc + " has failed!") -BackgroundColor Red
}
#endregion

#region makelink
$msixDest = "\\?\Volume{" + $volumeGuid + "}\"
if (!(Test-Path $msixJunction))
{
     md $msixJunction
}

$msixJunction = $msixJunction + $packageName
cmd.exe /c mklink /j $msixJunction $msixDest
#endregion

#region stage
[Windows.Management.Deployment.PackageManager,Windows.Management.Deployment,ContentType=WindowsRuntim
e] | Out-Null
Add-Type -AssemblyName System.Runtime.WindowsRuntime
$asTask = ([System.WindowsRuntimeSystemExtensions].GetMethods() | Where { $_.ToString() -eq
'System.Threading.Tasks.Task`1[TResult] AsTask[TResult,TProgress]
(Windows.Foundation.IAsyncOperationWithProgress`2[TResult,TProgress])'})[0]
$asTaskAsyncOperation = $asTask.MakeGenericMethod([Windows.Management.Deployment.DeploymentResult],
[Windows.Management.Deployment.DeploymentProgress])
$packageManager = [Windows.Management.Deployment.PackageManager]::new()
$path = $msixJunction + $parentFolder + $packageName
$path = ([System.Uri]$path).AbsoluteUri
$asyncOperation = $packageManager.StagePackageAsync($path, $null, "StageInPlace")
$task = $asTaskAsyncOperation.Invoke($null, @($asyncOperation))
$task
#endregion
```

Register PowerShell script

To run the register script, run the following PowerShell cmdlets with the placeholder values replaced with values that apply to your environment.

```
#MSIX app attach registration sample

#region variables
$packageName = "<package name>"
$path = "C:\Program Files\WindowsApps\" + $packageName + "\AppxManifest.xml"
#endregion

#region register
Add-AppxPackage -Path $path -DisableDevelopmentMode -Register
#endregion
```

Deregister PowerShell script

For this script, replace the placeholder for **$packageName** with the name of the package you're testing.

```
#MSIX app attach deregistration sample

#region variables
$packageName = "<package name>"
#endregion

#region deregister
Remove-AppxPackage -PreserveRoamableApplicationData $packageName
#endregion
```

Destage PowerShell script

For this script, replace the placeholder for **$packageName** with the name of the package you're testing. In a production deployment it would be best to run this on Shutdown.

```
#MSIX app attach de staging sample

$vhdSrc="<path to vhd>"

#region variables
$packageName = "<package name>"
$msixJunction = "C:\temp\AppAttach"
#endregion

#region deregister
Remove-AppxPackage -AllUsers -Package $packageName
Remove-Item "$msixJunction\$packageName" -Recurse -Force -Verbose
#endregion

#region Detach VHD
Dismount-DiskImage -ImagePath $vhdSrc -Confirm:$false
#endregion
```

> **NOTE**
> You can shut down the device even while the **$volumeGuid** point remains after executing the destage script.

Set up simulation scripts for the MSIX app attach agent

After you create the scripts, users can manually run them or set them up to run automatically as startup, logon, logoff, and shutdown scripts. To learn more about these types of scripts, see Using startup, shutdown, logon, and logoff scripts in Group Policy.

Each of these automatic scripts runs one phase of the app attach scripts:

- The startup script runs the stage script.
- The logon script runs the register script.
- The logoff script runs the deregister script.
- The shutdown script runs the destage script.

> **NOTE**
> You can run the task scheduler with the stage script. To run the script, set the task trigger to **When the computer starts**, then enable **Run with highest privileges**.

Use packages offline

If you're using packages from the Microsoft Store for Business or the Microsoft Store for Education within your network or on devices that aren't connected to the internet, you need to get the package licenses from the Microsoft Store and install them on your device to successfully run the app. If your device is online and can connect to the Microsoft Store for Business, the required licenses should download automatically, but if you're offline, you'll need to set up the licenses manually.

To install the license files, you'll need to use a PowerShell script that calls the MDM_EnterpriseModernAppManagement_StoreLicenses02_01 class in the WMI Bridge Provider.

Here's how to set up the licenses for offline use:

1. Download the app package, licenses, and required frameworks from the Microsoft Store for Business. You need both the encoded and unencoded license files. Detailed download instructions can be found here.
2. Update the following variables in the script for step 3:
   a. `$contentID` is the ContentID value from the Unencoded license file (.xml). You can open the license file in a text editor of your choice.
   b. `$licenseBlob` is the entire string for the license blob in the Encoded license file (.bin). You can open the encoded license file in a text editor of your choice.
3. Run the following script from an Admin PowerShell prompt. A good place to perform license installation is at the end of the staging script that also needs to be run from an Admin prompt.

```
$namespaceName = "root\cimv2\mdm\dmmap"
$className = "MDM_EnterpriseModernAppManagement_StoreLicenses02_01"
$methodName = "AddLicenseMethod"
$parentID = "./Vendor/MSFT/EnterpriseModernAppManagement/AppLicenses/StoreLicenses"

#TODO - Update $contentID with the ContentID value from the unencoded license file (.xml)
$contentID = "{'ContentID'_in_unencoded_license_file}"

#TODO - Update $licenseBlob with the entire String in the encoded license file (.bin)
$licenseBlob = "{Entire_String_in_encoded_license_file}"

$session = New-CimSession

#The final string passed into the AddLicenseMethod should be of the form <License Content="encoded license
blob" />
$licenseString = '<License Content='+ '"' + $licenseBlob +'"' + ' />'

$params = New-Object Microsoft.Management.Infrastructure.CimMethodParametersCollection
$param = [Microsoft.Management.Infrastructure.CimMethodParameter]::Create("param",$licenseString ,"String",
"In")
$params.Add($param)


try
{
   $instance = New-CimInstance -Namespace $namespaceName -ClassName $className -Property
@{ParentID=$parentID;InstanceID=$contentID}
   $session.InvokeMethod($namespaceName, $instance, $methodName, $params)

}
catch [Exception]
{
     write-host $_ | out-string
}
```

Next steps

This feature isn't currently supported, but you can ask questions to the community at the Azure Virtual Desktop TechCommunity.

You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.

Prepare an MSIX image for Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

MSIX app attach is an application layering solution that allows you to dynamically attach apps from an MSIX package to a user session. The MSIX package system separates apps from the operating system, making it easier to build images for virtual machines. MSIX packages also give you greater control over which apps your users can access in their virtual machines. You can even separate apps from the master image and give them to users later.

Instructions on how to convert a desktop installer (such as MSI, EXE, ClickOnce, App-V, or Script) to MSIX are available in Create an MSIX package from any desktop installer (MSI, EXE, ClickOnce, or App-V).

Create a VHD or VHDX package for MSIX

MSIX packages need to be in a VHD or VHDX format to work properly. This means that, to get started, you'll need to create a VHD or VHDX package.

> **NOTE**
> If you haven't already, make sure you enable Hyper-V by following the instructions in Install Hyper-V on Windows 10.

To create a VHD or VHDX package for MSIX:

1. First, open PowerShell.
2. Next, run the following cmdlet to create a VHD:

```
New-VHD -SizeBytes <size>MB -Path c:\temp\<name>.vhd -Dynamic -Confirm:$false
```

> **NOTE**
> Make sure the VHD is large enough to hold the expanded MSIX package.

3. Run the following cmdlet to mount the VHD you just created:

```
$vhdObject = Mount-VHD c:\temp\<name>.vhd -Passthru
```

4. Next, run this cmdlet to initialize the mounted VHD:

```
$disk = Initialize-Disk -Passthru -Number $vhdObject.Number
```

5. After that, run this cmdlet to create a new partition for the initialized VHD:

```
$partition = New-Partition -AssignDriveLetter -UseMaximumSize -DiskNumber $disk.Number
```

6. Run this cmdlet to format the partition:

```
Format-Volume -FileSystem NTFS -Confirm:$false -DriveLetter $partition.DriveLetter -Force
```

7. Finally, create a parent folder on the mounted VHD. This step is required because the MSIX package must have a parent folder to work properly. It doesn't matter what you name the parent folder, so long as the parent folder exists.

Expand MSIX

After that, you'll need to expand the MSIX image by "unpacking" its files into the VHD.

To expand the MSIX image:

1. Download the msixmgr tool and save the .zip folder to a folder within a session host VM.
2. Unzip the msixmgr tool .zip folder.
3. Put the source MSIX package into the same folder where you unzipped the msixmgr tool.
4. Open a command prompt as Administrator and navigate to the folder where you downloaded and unzipped the msixmgr tool.
5. Run the following cmdlet to unpack the MSIX into the VHD you created in the previous section.

   ```
   msixmgr.exe -Unpack -packagePath <package>.msix -destination "f:\<name of folder you created earlier>" -applyacls
   ```

   The following message should appear after you're done unpacking:

   ```
   Successfully unpacked and applied ACLs for package: <package name>.msix
   ```

   NOTE

   If you're using packages from the Microsoft Store for Business or Education on your network or on devices not connected to the internet, you'll need to download and install package licenses from the Microsoft Store to run the apps. To get the licenses, see Use packages offline.

6. Go the mounted VHD and open the app folder to make sure the package contents are there.
7. Unmount the VHD.

Upload MSIX image to share

After you've created the MSIX package, you'll need to upload the resulting VHD, VHDX, or CIM file to a share where your users' virtual machines can access it.

Next steps

Ask our community questions about this feature at the Azure Virtual Desktop TechCommunity.

You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.

Here are some other articles you might find helpful:

- MSIX app attach glossary
- MSIX app attach FAQ

Set up a file share for MSIX app attach

12/6/2021 • 4 minutes to read • Edit Online

All MSIX images must be stored on a network share that can be accessed by users in a host pool with read-only permissions.

MSIX app attach doesn't have any dependencies on the type of storage fabric the file share uses. The considerations for the MSIX app attach share are same as those for an FSLogix share. To learn more about storage requirements, see Storage options for FSLogix profile containers in Azure Virtual Desktop.

Performance requirements

MSIX app attach image size limits for your system depend on the storage type you're using to store the VHD or VHDx files, as well as the size limitations of the VHD, VHDX or CIM files and the file system.

The following table gives an example of how many resources a single 1 GB MSIX image with one MSIX app inside of it requires for each VM:

RESOURCE	REQUIREMENTS
Steady state IOPs	1 IOPs
Machine boot sign in	10 IOPs
Latency	400 ms

Requirements can vary widely depending how many MSIX-packaged applications are stored in the MSIX image. For larger MSIX images, you'll need to allocate more bandwidth.

Storage recommendations

Azure offers multiple storage options that can be used for MISX app attach. We recommend using Azure Files or Azure NetApp Files as those options offer the best value between cost and management overhead. The article Storage options for FSLogix profile containers in Azure Virtual Desktop compares the different managed storage solutions Azure offers in the context of Azure Virtual Desktop.

Optimize MSIX app attach performance

Here are some other things we recommend you do to optimize MSIX app attach performance:

- The storage solution you use for MSIX app attach should be in the same datacenter location as the session hosts.
- To avoid performance bottlenecks, exclude the following VHD, VHDX, and CIM files from antivirus scans:
  - `<MSIXAppAttachFileShare\>\*.VHD`
  - `<MSIXAppAttachFileShare\>\*.VHDX`
  - `\\storageaccount.file.core.windows.net\share*.VHD`
  - `\\storageaccount.file.core.windows.net\share*.VHDX`
  - `<MSIXAppAttachFileShare>.CIM`
  - `\\storageaccount.file.core.windows.net\share**.CIM`
- Separate the storage fabric for MSIX app attach from FSLogix profile containers.

- All VM system accounts and user accounts must have read-only permissions to access the file share.
- Any disaster recovery plans for Azure Virtual Desktop must include replicating the MSIX app attach file share in your secondary failover location. To learn more about disaster recovery, see Set up a business continuity and disaster recovery plan.

How to set up the file share

The setup process for MSIX app attach file share is largely the same as the setup process for FSLogix profile file shares. However, you'll need to assign users different permissions. MSIX app attach requires read-only permissions to access the file share.

If you're storing your MSIX applications in Azure Files, then for your session hosts, you'll need to assign all session host VMs both storage account role-based access control (RBAC) and file share New Technology File System (NTFS) permissions on the share.

AZURE OBJECT	REQUIRED ROLE	ROLE FUNCTION
Session host (VM computer objects)	Storage File Data SMB Share Contributor	Read and Execute, Read, List folder contents
Admins on File Share	Storage File Data SMB Share Elevated Contributor	Full control
Users on File Share	Storage File Data SMB Share Contributor	Read and Execute, Read, List folder contents

To assign session host VMs permissions for the storage account and file share:

1. Create an Active Directory Domain Services (AD DS) security group.
2. Add the computer accounts for all session host VMs as members of the group.
3. Sync the AD DS group to Azure Active Directory (Azure AD).
4. Create a storage account.
5. Create a file share under the storage account by following the instructions in Create an Azure file share.
6. Join the storage account to AD DS by following the instructions in Part one: enable AD DS authentication for your Azure file shares.
7. Assign the synced AD DS group to Azure AD, and assign the storage account the Storage File Data SMB Share Contributor role.
8. Mount the file share to any session host by following the instructions in Part two: assign share-level permissions to an identity.
9. Grant NTFS permissions on the file share to the AD DS group.
10. Set up NTFS permissions for the user accounts. You'll need an operating unit (OU) sourced from the AD DS that the accounts in the VM belong to.

Once you've assigned the identity to your storage, follow the instructions in the articles in Next steps to grant other required permissions to the identity you've assigned to the VMs.

You'll also need to make sure your session host VMs have New Technology File System (NTFS) permissions. You must have an operational unit container that's sourced from Active Directory Domain Services (AD DS), and your users must be members of that operational unit to use these permissions.

Next steps

Here are the other things you'll need to do after you've set up the file share:

- Learn how to set up Azure Active Directory Domain Services (AD DS) at Create a profile container with Azure Files and AD DS.
- Learn how to set up Azure Files and Azure AD DS at Create a profile container with Azure Files and Azure AD DS.
- Learn how to set up Azure NetApp Files for MSIX app attach at Create a profile container with Azure NetApp Files and AD DS.
- Learn how to use a virtual machine-based file share at Create a profile container for a host pool using a file share.

Once you're finished, here are some other resources you might find helpful:

- Ask our community questions about this feature at the Azure Virtual Desktop TechCommunity.
- You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.
- MSIX app attach glossary
- MSIX app attach FAQ

Using the MSIXMGR tool

12/6/2021 • 2 minutes to read • Edit Online

The MSIXMGR tool is for expanding MSIX-packaged applications into MSIX images. The tool takes an MSIX-packaged application (.MSIX) and expands it into a VHD, VHDx, or CIM file. The resulting MSIX image is stored in the Azure Storage account that your Azure Virtual Desktop deployment uses.This article will show you how to use the MSIXMGR tool.

> **NOTE**
> To guarantee compatibility, make sure the CIMs storing your MSIX images are generated on the OS version you're running in your Azure Virtual Desktop host pools. MSIXMGR can create CIM files, but you can only use those files with a host pool running Windows 10 20H2.

Requirements

Before you can follow the instructions in this article, you'll need to do the following things:

- Download the MSIXMGR tool
- Get an MSIX-packaged application (.MSIX file)
- Get administrative permissions on the machine where you'll create the MSIX image

Create an MSIX image

Expansion is the process of taking an MSIX packaged application (.MSIX) and unzipping it into a MSIX image (.VHD(x) or .CIM file).

To expand an MSIX file:

1. Download the MSIXMGR tool if you haven't already.
2. Unzip MSIXMGR.zip into a local folder.
3. Open a command prompt in elevated mode.
4. Find the local folder from step 2.
5. Run the following command in the command prompt to create an MSIX image.

   ```
   msixmgr.exe -Unpack -packagePath <path to package> -destination <output folder> [-applyacls] [-
   create] [-vhdSize <size in MB>] [-filetype <CIM | VHD | VHDX>] [-rootDirectory <rootDirectory>]
   ```

   Remember to replace the placeholder values with the relevant values. For example:

   ```
   msixmgr.exe -Unpack -packagePath
   "C:\Users\%username%\Desktop\packageName_3.51.1.0_x64__81q6ced8g4aa0.msix" -destination
   "c:\temp\packageName.vhdx" -applyacls -create -vhdSize 200 -filetype "vhdx" -rootDirectory apps
   ```

6. Now that you've created the image, go to the destination folder and make sure you successfully created the MSIX image (.VHDX).

Create an MSIX image in a CIM file

You can also use the command in step 5 to create CIM and VHDX files by replacing the file type and destination path.

For example, here's how you'd use that command to make a CIM file:

```
msixmgr.exe -Unpack -packagePath "C:\Users\ssa\Desktop\packageName_3.51.1.0_x64__81q6ced8g4aa0.msix" -
destination "c:\temp\packageName.cim" -applyacls -create -vhdSize 200 -filetype "cim" -rootDirectory apps
```

Here's how you'd use that command to make a VHDX:

```
msixmgr.exe -Unpack -packagePath "C:\Users\ssa\Desktop\packageName_3.51.1.0_x64__81q6ced8g4aa0.msix" -
destination "c:\temp\packageName.vhdx" -applyacls -create -vhdSize 200 -filetype "vhdx" -rootDirectory apps
```

Next steps

Learn more about MSIX app attach at What is MSIX app attach?

To learn how to set up app attach, check out these articles:

- Set up MSIX app attach with the Azure portal
- Set up MSIX app attach using PowerShell
- Create PowerShell scripts for MSIX app attach
- Prepare an MSIX image for Azure Virtual Desktop
- Set up a file share for MSIX app attach

If you have questions about MSIX app attach, see our App attach FAQ and App attach glossary.

Use Microsoft Teams on Azure Virtual desktop

12/6/2021 • 7 minutes to read • Edit Online

IMPORTANT

Media optimization for Teams is supported for Microsoft 365 Government (GCC) and GCC-High environments. Media optimization for Teams is not supported for Microsoft 365 DoD.

NOTE

Media optimization for Microsoft Teams is only available for the Windows Desktop client on Windows 10 machines. Media optimizations require Windows Desktop client version 1.2.1026.0 or later.

Microsoft Teams on Azure Virtual Desktop supports chat and collaboration. With media optimizations, it also supports calling and meeting functionality. To learn more about how to use Microsoft Teams in Virtual Desktop Infrastructure (VDI) environments, see Teams for Virtualized Desktop Infrastructure.

With media optimization for Microsoft Teams, the Windows Desktop client handles audio and video locally for Teams calls and meetings. You can still use Microsoft Teams on Azure Virtual Desktop with other clients without optimized calling and meetings. Teams chat and collaboration features are supported on all platforms. To redirect local devices in your remote session, check out Customize Remote Desktop Protocol properties for a host pool.

Prerequisites

Before you can use Microsoft Teams on Azure Virtual Desktop, you'll need to do these things:

- Prepare your network for Microsoft Teams.
- Install the Windows Desktop client on a Windows 10 or Windows 10 IoT Enterprise device that meets the Microsoft Teams hardware requirements for Teams on a Windows PC.
- Connect to a Windows 10 Multi-session or Windows 10 Enterprise virtual machine (VM).

Install the Teams desktop app

This section will show you how to install the Teams desktop app on your Windows 10 Multi-session or Windows 10 Enterprise VM image. To learn more, check out Install or update the Teams desktop app on VDI.

Prepare your image for Teams

To enable media optimization for Teams, set the following registry key on the host:

1. From the start menu, run **RegEdit** as an administrator. Navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Teams**. Create the Teams key if it doesn't already exist.

2. Create the following value for the Teams key:

NAME	TYPE	DATA/VALUE
IsWVDEnvironment	DWORD	1

Install the Teams WebSocket Service

Install the latest version of the Remote Desktop WebRTC Redirector Service on your VM image. If you encounter an installation error, install the latest Microsoft Visual C++ Redistributable and try again.

Latest WebSocket Service versions

The following table lists the latest versions of the WebSocket Service:

VERSION	RELEASE DATE
1.4.2111.18001	12/02/2021
1.1.2110.16001	10/15/2021
1.0.2106.14001	07/29/2021
1.0.2006.11001	07/28/2020
0.11.0	05/29/2020

Updates for version 1.4.2111.18001

- Fixed a mute notification problem.
- Multiple z-ordering fixes in Teams on Azure Virtual Desktop and Teams on Microsoft 365.
- Removed timeout that prevented the WebRTC redirector service from starting when the user connects.
- Fixed setup problems that prevented side-by-side installation from working.

Updates for version 1.1.2110.16001

- Fixed an issue that caused the screen to turn black while screen sharing. If you've been experiencing this issue, confirm that this update will resolve it by resizing the Teams window. If screen sharing starts working again after resizing, the update will resolve this issue.
- You can now control the meeting, ringtone, and notification volume from the host VM. You can only use this feature with version 1.2.2459 or later of the Windows Desktop client.
- The installer will now make sure that Teams is closed before installing updates.
- Fixed an issue that prevented users from returning to full screen mode after leaving the call window.

Updates for version 1.0.2106.14001

Increased the connection reliability between the WebRTC redirector service and the WebRTC client plugin.

Updates for version 1.0.2006.11001

- Fixed an issue where minimizing the Teams app during a call or meeting caused incoming video to drop.
- Added support for selecting one monitor to share in multi-monitor desktop sessions.

Install Microsoft Teams

You can deploy the Teams desktop app using a per-machine or per-user installation. To install Microsoft Teams in your Azure Virtual Desktop environment:

1. Download the Teams MSI package that matches your environment. We recommend using the 64-bit installer on a 64-bit operating system.

   > **IMPORTANT**
   > The latest update of the Teams Desktop client version 1.3.00.21759 fixed an issue where Teams showed UTC time zone in chat, channels, and calendar. The new version of the client will show the remote session time zone.

2. Run one of the following commands to install the MSI to the host VM:

- Per-user installation

  ```
  msiexec /i <path_to_msi> /l*v <install_logfile_name>
  ```

  This process is the default installation, which installs Teams to the **%AppData%** user folder. Teams won't work properly with per-user installation on a non-persistent setup.

- Per-machine installation

  ```
  msiexec /i <path_to_msi> /l*v <install_logfile_name> ALLUSER=1
  ```

  This installs Teams to the Program Files (x86) folder on a 32-bit operating system and to the Program Files folder on a 64-bit operating system. At this point, the golden image setup is complete. Installing Teams per-machine is required for non-persistent setups.

  There are two flags that may be set when installing teams, **ALLUSER=1** and **ALLUSERS=1**. It is important to understand the difference between these parameters. The **ALLUSER=1** parameter is used only in VDI environments to specify a per-machine installation. The **ALLUSERS=1** parameter can be used in non-VDI and VDI environments. When you set this parameter, **Teams Machine-Wide Installer** appears in Program and Features in Control Panel as well as Apps & features in Windows Settings. All users with admin credentials on the machine can uninstall Teams.

  > **NOTE**
  > Users and admins can't disable automatic launch for Teams during sign-in at this time.

3. To uninstall the MSI from the host VM, run this command:

   ```
   msiexec /passive /x <msi_name> /l*v <uninstall_logfile_name>
   ```

   This uninstalls Teams from the Program Files (x86) folder or Program Files folder, depending on the operating system environment.

   > **NOTE**
   > When you install Teams with the MSI setting ALLUSER=1, automatic updates will be disabled. We recommend you make sure to update Teams at least once a month. To learn more about deploying the Teams desktop app, check out Deploy the Teams desktop app to the VM.

Verify media optimizations loaded

After installing the WebSocket Service and the Teams desktop app, follow these steps to verify that Teams media optimizations loaded:

1. Quit and restart the Teams application.
2. Select your user profile image, then select **About**.
3. Select **Version**.

   If media optimizations loaded, the banner will show you **Azure Virtual Desktop Media optimized**. If the banner shows you **Azure Virtual Desktop Media not connected**, quit the Teams app and try again.

4. Select your user profile image, then select **Settings**.

If media optimizations loaded, the audio devices and cameras available locally will be enumerated in the device menu. If the menu shows **Remote audio**, quit the Teams app and try again. If the devices still don't appear in the menu, check the Privacy settings on your local PC. Ensure the under **Settings** > **Privacy** > **App permissions - Microphone** the setting **"Allow apps to access your microphone"** is toggled **On**. Disconnect from the remote session, then reconnect and check the audio and video devices again. To join calls and meetings with video, you must also grant permission for apps to access your camera.

If optimizations do not load, uninstall then reinstall Teams and check again.

Known issues and limitations

Using Teams in a virtualized environment is different from using Teams in a non-virtualized environment. For more information about the limitations of Teams in virtualized environments, check out Teams for Virtualized Desktop Infrastructure.

Client deployment, installation, and setup

- With per-machine installation, Teams on VDI isn't automatically updated the same way non-VDI Teams clients are. To update the client, you'll need to update the VM image by installing a new MSI.
- Media optimization for Teams is only supported for the Windows Desktop client on machines running Windows 10.
- Use of explicit HTTP proxies defined on the client endpoint device is not supported.

Calls and meetings

- The Teams desktop client in Azure Virtual Desktop environments doesn't support creating live events, but you can join live events. For now, we recommend you create live events from the Teams web client in your remote session instead.
- Calls or meetings don't currently support application sharing. Desktop sessions support desktop sharing.
- Give control and take control aren't currently supported.
- Teams on Azure Virtual Desktop only supports one incoming video input at a time. This means that whenever someone tries to share their screen, their screen will appear instead of the meeting leader's screen.
- Due to WebRTC limitations, incoming and outgoing video stream resolution is limited to 720p.
- The Teams app doesn't support HID buttons or LED controls with other devices.
- New Meeting Experience (NME) is not currently supported in VDI environments.

For Teams known issues that aren't related to virtualized environments, see Support Teams in your organization.

Collect Teams logs

If you encounter issues with the Teams desktop app in your Azure Virtual Desktop environment, collect client logs under **%appdata%\Microsoft\Teams\logs.txt** on the host VM.

If you encounter issues with calls and meetings, collect Teams Web client logs with the key combination **Ctrl** + **Alt** + **Shift** + **1**. Logs will be written to **%userprofile%\Downloads\MSTeams Diagnostics Log DATE_TIME.txt** on the host VM.

Contact Microsoft Teams support

To contact Microsoft Teams support, go to the Microsoft 365 admin center.

Customize Remote Desktop Protocol properties for a host pool

Customizing a host pool's Remote Desktop Protocol (RDP) properties, such as multi-monitor experience or

enabling microphone and audio redirection, lets you deliver an optimal experience for your users based on their needs.

Enabling device redirections is not required when using Teams with media optimization. If you are using Teams without media optimization, set the following RDP properties to enable microphone and camera redirection:

- `audiocapturemode:i:1` enables audio capture from the local device and redirects audio applications in the remote session.
- `audiomode:i:0` plays audio on the local computer.
- `camerastoredirect:s:*` redirects all cameras.

To learn more, check out Customize Remote Desktop Protocol properties for a host pool.

Enable Azure multifactor authentication for Azure Virtual Desktop

12/6/2021 • 3 minutes to read • Edit Online

IMPORTANT

If you're visiting this page from the Azure Virtual Desktop (classic) documentation, make sure to return to the Azure Virtual Desktop (classic) documentation once you're finished.

The Windows client for Azure Virtual Desktop is an excellent option for integrating Azure Virtual Desktop with your local machine. However, when you configure your Azure Virtual Desktop account into the Windows Client, there are certain measures you'll need to take to keep yourself and your users safe.

When you first sign in, the client asks for your username, password, and Azure multifactor authentication. After that, the next time you sign in, the client will remember your token from your Azure Active Directory (AD) Enterprise Application. When you select **Remember me** on the prompt for credentials for the session host, your users can sign in after restarting the client without needing to reenter their credentials.

While remembering credentials is convenient, it can also make deployments on Enterprise scenarios or personal devices less secure. To protect your users, you can make sure the client keeps asking for Azure multifactor authentication credentials more frequently. This article will show you how to configure the Conditional Access policy for Azure Virtual Desktop to enable this setting.

Prerequisites

Here's what you'll need to get started:

- Assign users a license that includes Azure Active Directory Premium P1 or P2.
- An Azure Active Directory group with your users assigned as group members.
- Enable Azure multifactor authentication for all your users. For more information about how to do that, see How to require two-step verification for a user.

NOTE

The following setting also applies to the Azure Virtual Desktop web client.

Create a Conditional Access policy

Here's how to create a Conditional Access policy that requires multifactor authentication when connecting to Azure Virtual Desktop:

1. Sign in to the **Azure portal** as a global administrator, security administrator, or Conditional Access administrator.
2. Browse to **Azure Active Directory** > **Security** > **Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.

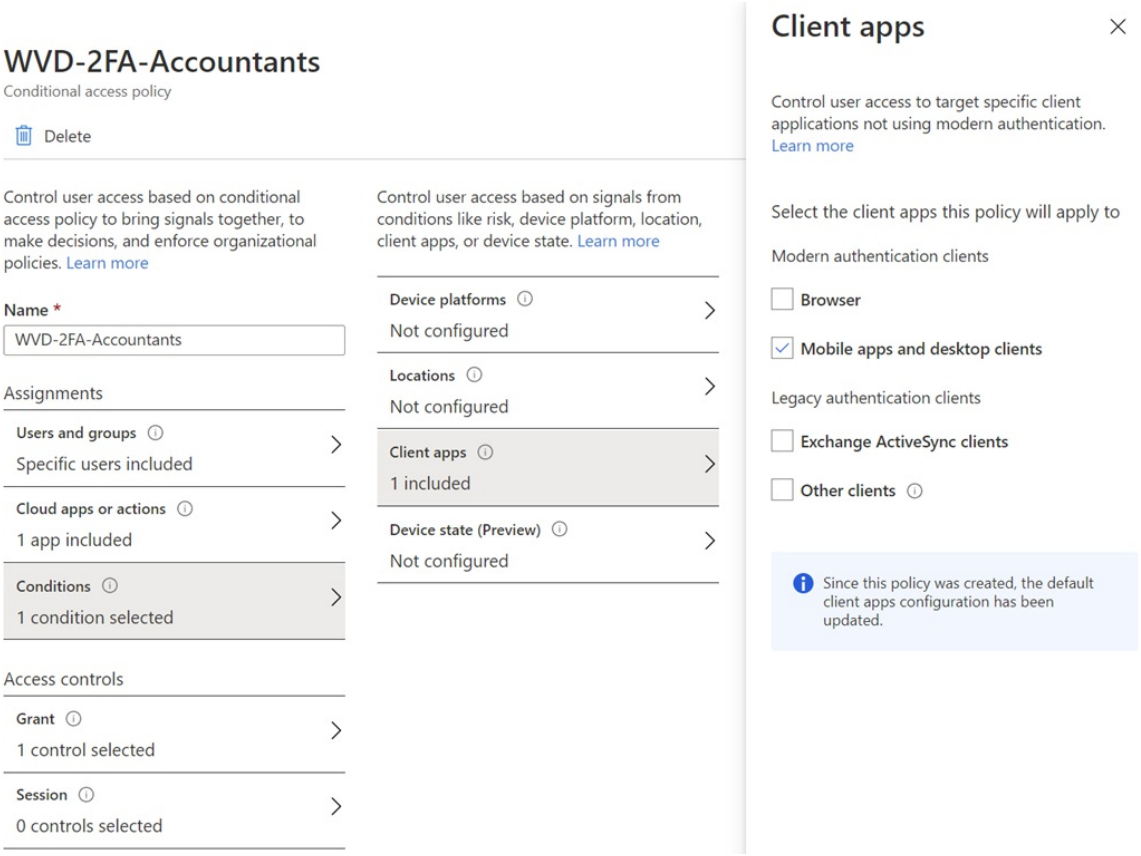
5. Under **Assignments**, select **Users and groups**.
6. Under **Include**, select **Select users and groups** > **Users and groups** > Choose the group you created in the prerequisites stage.
7. Select **Done**.
8. Under **Cloud apps or actions** > **Include**, select **Select apps**.
9. Select one of the following apps based on which version of Azure Virtual Desktop you're using.
   - If you're using Azure Virtual Desktop (classic), choose these apps:
     - **Azure Virtual Desktop** (App ID 5a0aa725-4958-4b0c-80a9-34562e23f3b7)
     - **Azure Virtual Desktop Client** (App ID fa4345a4-a730-4230-84a8-7d9651b86739), which will let you set policies on the web client

     After that, skip ahead to step 11.
   - If you're using Azure Virtual Desktop, choose this app instead:
     - **Azure Virtual Desktop** (App ID 9cdead84-a844-4324-93f2-b2e6bb768d07)

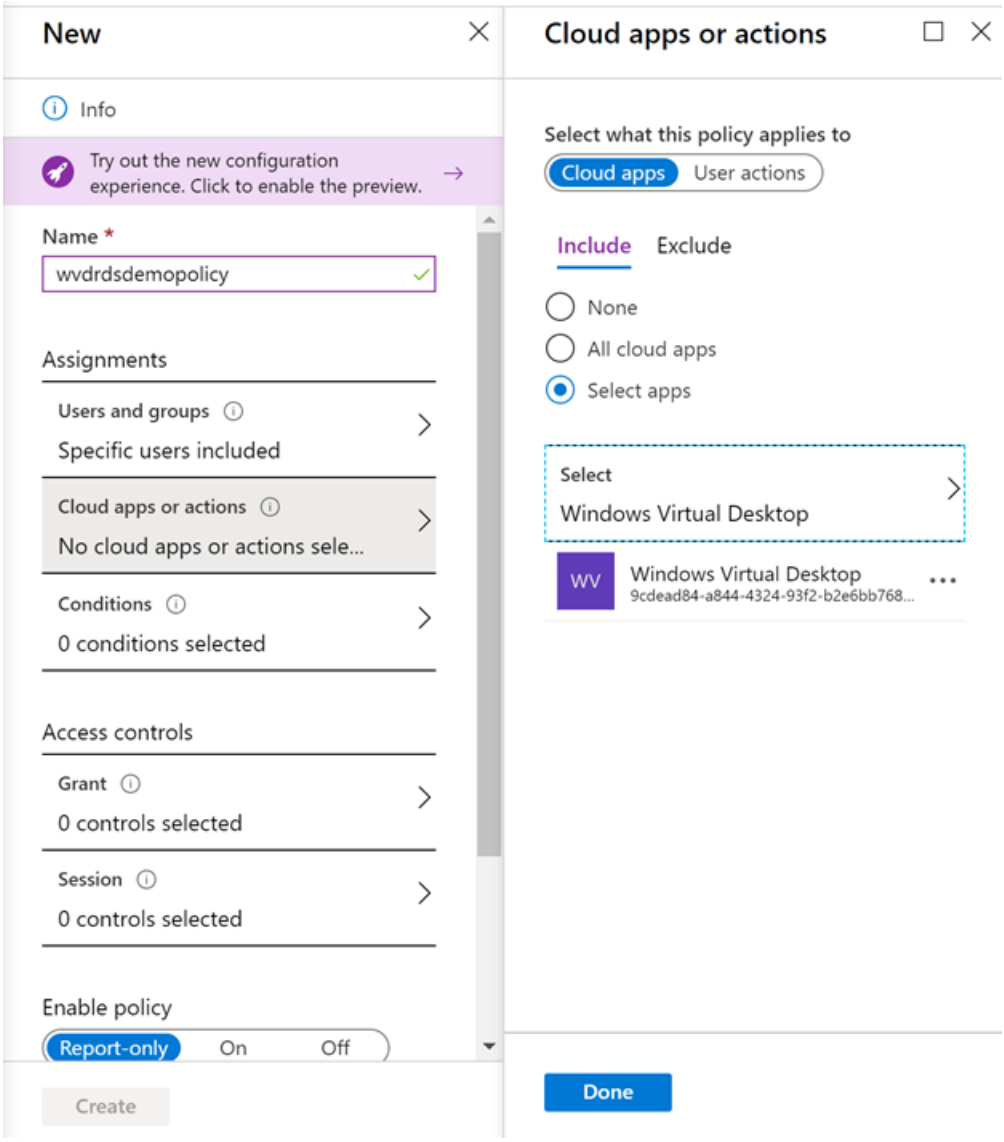
     After that, go to step 10.

   > **IMPORTANT**
   > Don't select the app called Azure Virtual Desktop Azure Resource Manager Provider (50e95039-b200-4007-bc97-8d5790743a63). This app is only used for retrieving the user feed and shouldn't have multifactor authentication.
   >
   > If you're using Azure Virtual Desktop (classic), if the Conditional Access policy blocks all access and only excludes Azure Virtual Desktop app IDs, you can fix this by adding the app ID 9cdead84-a844-4324-93f2-b2e6bb768d07 to the policy. Not adding this app ID will block feed discovery of Azure Virtual Desktop (classic) resources.

10. Go to **Conditions** > **Client apps**. In **Configure**, select **Yes**, and then select where to apply the policy:
    - Select **Browser** if you want the policy to apply to the web client.
    - Select **Mobile apps and desktop clients** if you want to apply the policy to other clients.
    - Select both check boxes if you want to apply the policy to all clients.



11. Once you've selected your app, choose **Select**, and then select **Done**.



> **NOTE**
>
> To find the App ID of the app you want to select, go to **Enterprise Applications** and select **Microsoft Applications** from the application type drop-down menu.

12. Under **Access controls** > **Grant**, select **Grant access**, **Require multi-factor authentication**, and then **Select**.
13. Under **Access controls** > **Session**, select **Sign-in frequency**, set the value to the time you want between prompts, and then select **Select**. For example, setting the value to **1** and the unit to **Hours**, will require multifactor authentication if a connection is launched an hour after the last one.
14. Confirm your settings and set **Enable policy** to **On**.
15. Select **Create** to enable your policy.

> **NOTE**
>
> When you use the web client to sign in to Azure Virtual Desktop through your browser, the log will list the client app ID as a85cf173-4192-42f8-81fa-777a763e6e2c (Azure Virtual Desktop client). This is because the client app is internally linked to the server app ID where the conditional access policy was set.

Next steps

- Learn more about Conditional Access policies
- Learn more about user sign in frequency

Configure AD FS single sign-on for Azure Virtual Desktop

12/6/2021 • 12 minutes to read • Edit Online

This article will walk you through the process of configuring Active Directory Federation Service (AD FS) single sign-on (SSO) for Azure Virtual Desktop.

> **NOTE**
> Azure Virtual Desktop (Classic) doesn't support this feature.

Requirements

Before configuring AD FS single sign-on, you must have the following setup running in your environment:

- You must deploy the **Active Directory Certificate Services (CA)** role. All servers running the role must be domain-joined, have the latest Windows updates installed, and be configured as enterprise certificate authorities.
- You must deploy the **Active Directory Federation Services (AD FS)** role. All servers running this role must be domain-joined, have the latest Windows updates installed, and be running Windows Server 2016 or later. See our federation tutorial to get started setting up this role.
- We recommend setting up the **Web Application Proxy** role to secure your environment's connection to the AD FS servers. All servers running this role must have the latest Windows updates installed, and be running Windows Server 2016 or later. See this Web Application Proxy guide to get started setting up this role.
- You must deploy **Azure AD Connect** to sync users to Azure AD. Azure AD Connect must be configured in federation mode.
- Set up your PowerShell environment for Azure Virtual Desktop on the AD FS server.
- When using Windows 10 20H1 or 20H2 to connect to Azure Virtual Desktop, you must install the **2021-04 Cumulative Update for Windows 10 (KB5001330)** or later for single sign-on to function properly.

> **NOTE**
> This solution is not supported with Azure AD Domain Services. You must use an Active Directory Domain Controller.

Supported clients

The following Azure Virtual Desktop clients support this feature:

- Windows Desktop client
- Web client

Configure the certificate authority to issue certificates

You must properly create the following certificate templates so that AD FS can use SSO:

- First, you'll need to create the **Exchange Enrollment Agent (Offline Request)** certificate template. AD FS uses the Exchange Enrollment Agent certificate template to request certificates on the user's behalf.
- You'll also need to create the **Smartcard Logon** certificate template, which AD FS will use to create the sign

in certificate.

After you create these certificate templates, you'll need to enable the templates on the certificate authority so AD FS can request them.

> **NOTE**
>
> This solution generates new short term certificates for every user logon which can fill up the Certificate Authority database over time if you have a lot of users. You can avoid this by setting up a CA for non-persistent certificate processing.

Create the enrollment agent certificate template

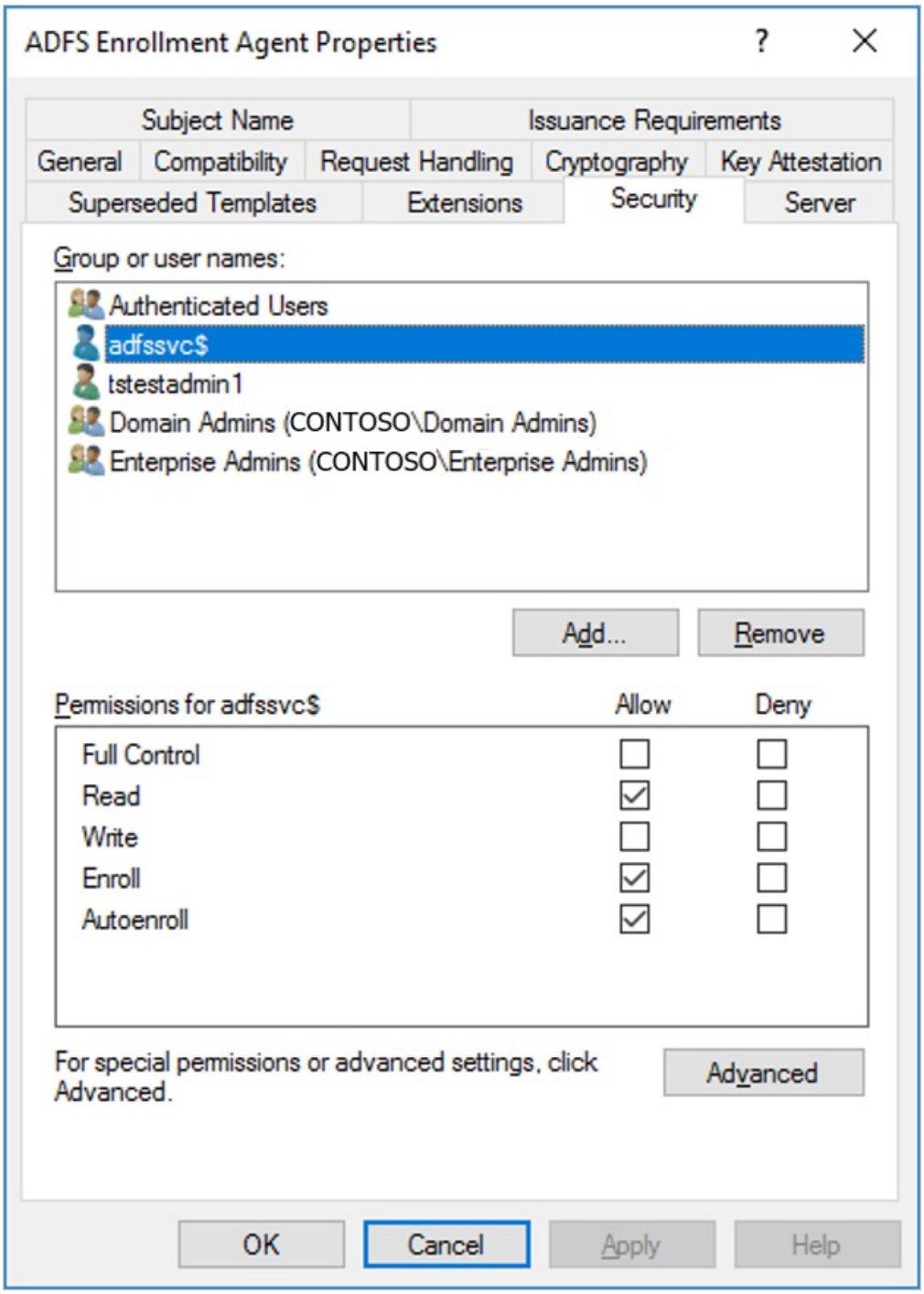
Depending on your environment, you may already have configured an enrollment agent certificate template for other purposes like Windows Hello for Business, Logon certificates or VPN certificates. If so, you will need to modify it to support SSO. If not, you can create a new template.

To determine if you are already using an enrollment agent certificate template, run the following PowerShell command on the AD FS server and see if a value is returned. If it's empty, create a new enrollment agent certificate template. Otherwise, remember the name and update the existing enrollment agent certificate template.

```
Import-Module adfs
(Get-AdfsCertificateAuthority).EnrollmentAgentCertificateTemplateName
```

To create a new enrollment agent certificate template:

1. On the certificate authority, run **mmc.exe** from the Start menu to launch the **Microsoft Management Console**.
2. Select **File...** > **Add/Remote Snap-in...** > **Certificate Templates** > **Add** > > **OK** to view the list of certificate templates.
3. Expand the **Certificate Templates**, right-click **Exchange Enrollment Agent (Offline Request)** and select **Duplicate Template**.
4. Select the **General** tab, then enter "ADFS Enrollment Agent" into the **Template display name** field. This will automatically set the template name to "ADFSEnrollmentAgent".
5. Select the **Security** tab, then select **Add...**.
6. Next, select **Object Types...**, then **Service Accounts**, and then **OK**.
7. Enter the service account name for AD FS and select **OK**.
   - In an isolated AD FS setup, the service account will be named "adfssvc$"
   - If you set up AD FS using Azure AD Connect, the service account will be named "aadcsvc$"
8. After the service account is added and is visible in the **Security** tab, select it in the **Group or user names** pane, select **Allow** for both "Enroll" and "Autoenroll" in the **Permissions for the AD FS service account** pane, then select **OK** to save.



To update an existing enrollment agent certificate template:

1. On the certificate authority, run **mmc.exe** from the Start menu to launch the **Microsoft Management Console**.
2. Select **File...** > **Add/Remote Snap-in...** > **Certificate Templates** > **Add** > > **OK** to view the list of certificate templates.
3. Expand the **Certificate Templates**, double-click the template that corresponds to the one configured on the AD FS server. On the **General** tab, the template name should match the name you found above.
4. Select the **Security** tab, then select **Add...**.
5. Next, select **Object Types...**, then **Service Accounts**, and then **OK**.
6. Enter the service account name for AD FS and select **OK**.
   - In an isolated AD FS setup, the service account will be named "adfssvc$"
   - If you set up AD FS using Azure AD Connect, the service account will be named "aadcsvc$"
7. After the service account is added and is visible in the **Security** tab, select it in the **Group or user names** pane, select **Allow** for both "Enroll" and "Autoenroll" in the **Permissions for the AD FS service account**

pane, then select **OK** to save.

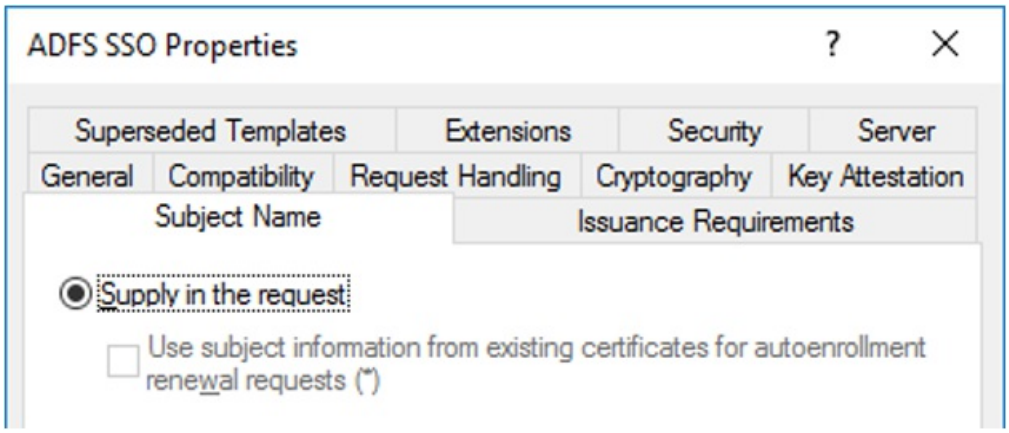
Create the Smartcard Logon certificate template

To create the Smartcard Logon certificate template:

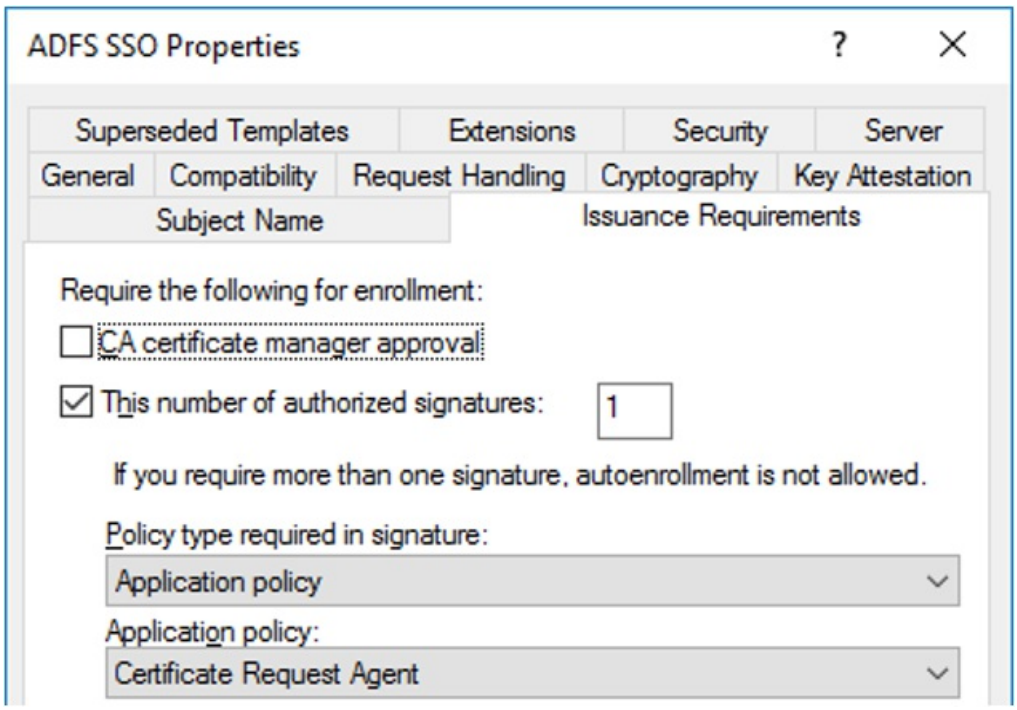
1. On the certificate authority, run **mmc.exe** from the Start menu to launch the **Microsoft Management Console**.
2. Select **File...** > **Add/Remote Snap-in...** > **Certificate Templates** > **Add** > **OK** to view the list of certificate templates.
3. Expand the **Certificate Templates**, right-click **Smartcard Logon** and select **Duplicate Template**.
4. Select the **General** tab, then enter "ADFS SSO" into the **Template display name** field. This will automatically set the template name to "ADFSSSO".

   > **NOTE**
   > Since this certificate is requested on-demand, we recommend shortening the validity period to 8 hours and the renewal period to 1 hour.

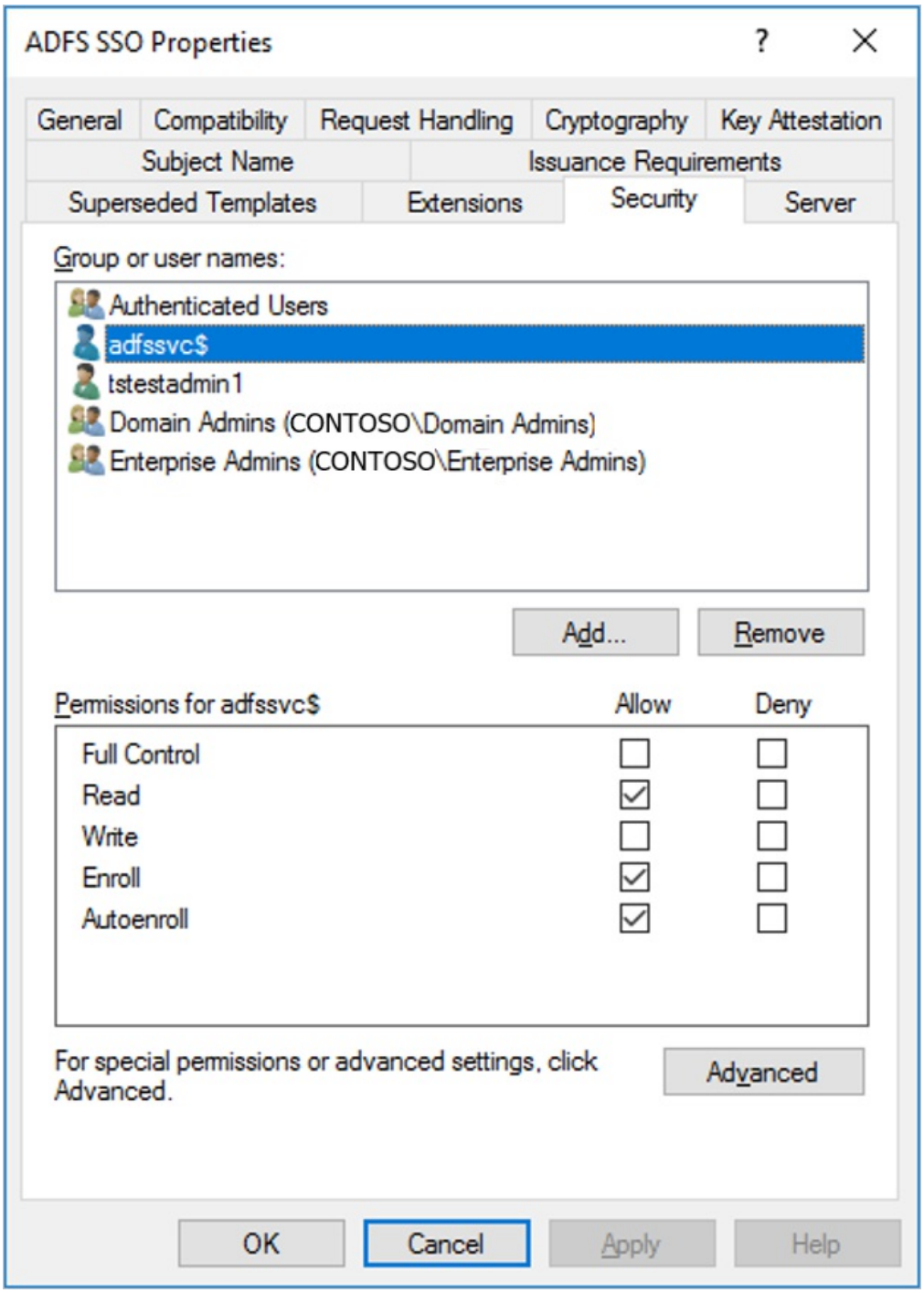
5. Select the **Subject name** tab and then select **Supply in the request**. When you see a warning message, select **OK**.



6. Select the **Issuance Requirements** tab.
7. Select **This number of authorized signatures** and enter the value of **1**.



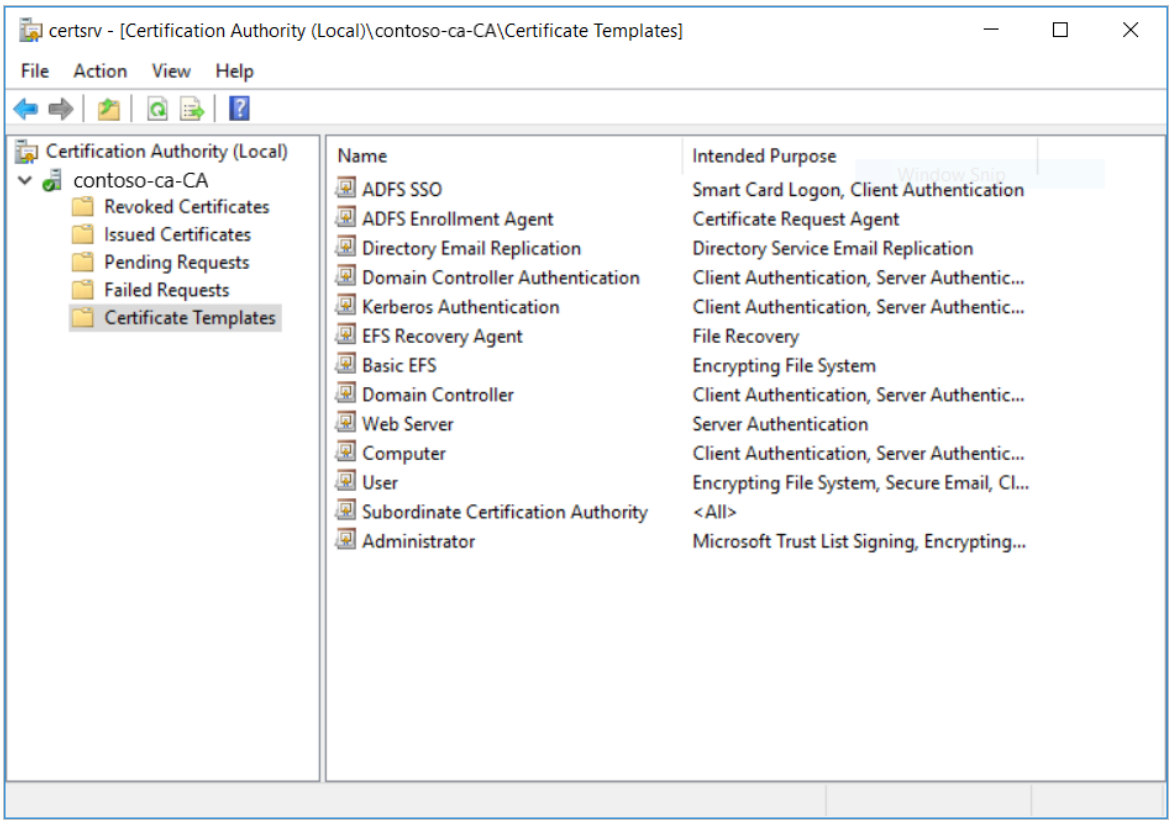
8. For **Application policy**, select **Certificate Request Agent**.
9. Select the **Security** tab, then select **Add...**.
10. Select **Object Types...**, **Service Accounts**, and **OK**.
11. Enter the service account name for AD FS just like you did in the Create the enrollment agent certificate template section.
    - In an isolated AD FS setup, the service account will be named "adfssvc$"
    - If you set up AD FS using Azure AD Connect, the service account will be named "aadcsvc$"
12. After the service account is added and is visible in the **Security** tab, select it in the **Group or user names** pane, select **Allow** for both "Enroll" and "Autoenroll", then select **OK** to save.



Enable the new certificate templates:

To enable the new certificate templates:

1. On the certificate authority, run **mmc.exe** from the Start menu to launch the **Microsoft Management Console**.
2. Select **File...** > **Add/Remove Snap-in...** > **Certification Authority** > **Add** > > **Finish** > and **OK** to view the Certification Authority.
3. Expand the Certification Authority on the left-hand pane and open **Certificate Templates**.
4. Right-click in the middle pane that shows the list of certificate templates, select **New**, then select **Certificate Template to Issue**.
5. Select both **ADFS Enrollment Agent** and **ADFS SSO**, then select **OK**. You should see both templates in the middle pane.



> **NOTE**
> If you already have an enrollment agent certificate template configured, you only need to add the ADFS SSO template.

Configure the AD FS Servers

You must configure the Active Directory Federation Services (AD FS) servers to use the new certificate templates and set the relying-party trust to support SSO.

The relying-party trust between your AD FS server and the Azure Virtual Desktop service allows single sign-on certificate requests to be forwarded correctly to your domain environment.

When configuring AD FS single sign-on you must choose shared key or certificate:

- If you have a single AD FS server, you can choose shared key or certificate.
- If you have multiple AD FS servers, it's required to choose certificate.

The shared key or certificate used to generate the token to sign in to Windows must be stored securely in Azure Key Vault. You can store the secret in an existing Key Vault or deploy a new one. In either case, you must ensure to set the right access policy so the Azure Virtual Desktop service can access it.

When using a certificate, you can use any general purpose certificate and there is no requirement on the subject name or Subject Alternative Name (SAN). While not required, it's recommended to create a certificate issued by a valid Certificate Authority. This certificate can be created directly in Azure Key Vault and needs to have an exportable private key. The public key can be exported and used to configure the AD FS server using the script below. Note that this certificate is different from the AD FS SSL certificate that must have a proper subject name and valid Certificate Authority.

The PowerShell script **ConfigureWVDSSO.ps1** available in the PowerShell Gallery will configure your AD FS server for the relying-party trust and install the certificate if needed.

This script only has one required parameter, *ADFSAuthority*, which is the URL that resolves to your AD FS and

uses "/adfs" as its suffix. For example, `https://adfs.contoso.com/adfs`.

1. On the AD FS VMs, run the following PowerShell cmdlet to configure AD FS to use the certificate templates from the previous section:

```
Set-AdfsCertificateAuthority -EnrollmentAgentCertificateTemplate "ADFSEnrollmentAgent" -
LogonCertificateTemplate "ADFSSSO" -EnrollmentAgent
```

> **NOTE**
> If you already have an EnrollmentAgentCertificateTemplate configured, ensure you use the existing template name instead of ADFSEnrollmentAgent.

2. Run the ConfigureWVDSSO.ps1 script.

> **NOTE**
> You need the `$config` variable values to complete the next part of the instructions, so don't close the PowerShell window you used to complete the previous instructions. You can either keep using the same PowerShell window or leave it open while launching a new PowerShell session.

- If you're using a shared key in the Key Vault, run the following PowerShell cmdlet on the AD FS server with ADFSServiceUrl replaced with the full URL to reach your AD FS service:

```
Install-Script ConfigureWVDSSO
$config = ConfigureWVDSSO.ps1 -ADFSAuthority "<ADFSServiceUrl>" [-WvdWebAppAppIDUri "<WVD Web
App URI>"] [-RdWebURL "<RDWeb URL>"]
```

> **NOTE**
> You need the WvdWebAppAppIDUri and RdWebURL properties to configure an environment in a sovereign cloud like Azure Government. In the Azure Commercial Cloud, these properties are automatically set to `https://www.wvd.microsoft.com` and `https://rdweb.wvd.microsoft.com` respectively.

- If you're using a certificate in the Key Vault, run the following PowerShell cmdlet on the AD FS server with ADFSServiceUrl replaced with the full URL to reach your AD FS service:

```
Install-Script ConfigureWVDSSO
$config = ConfigureWVDSSO.ps1 -ADFSAuthority "<ADFSServiceUrl>" -UseCert -CertPath "<Path to
the pfx file>" -CertPassword <Password to the pfx file> [-WvdWebAppAppIDUri "<WVD Web App
URI>"] [-RdWebURL "<RDWeb URL>"]
```

> **NOTE**
> You need the WvdWebAppAppIDUri and RdWebURL properties to configure an environment in a sovereign cloud like Azure Government. In the Azure Commercial Cloud, these properties are automatically set to `https://www.wvd.microsoft.com` and `https://rdweb.wvd.microsoft.com` respectively.

3. Set the access policy on the Azure Key Vault by running the following PowerShell cmdlet:

```
Set-AzKeyVaultAccessPolicy -VaultName "<Key Vault Name>" -ServicePrincipalName 9cdead84-a844-4324-
93f2-b2e6bb768d07 -PermissionsToSecrets get -PermissionsToKeys sign
```

4. Store the shared key or certificate in Azure Key Vault with a Tag containing a coma separated list of subscription IDs allowed to use the secret.

   - If you're using a shared key in the Key Vault, run the following PowerShell cmdlet to store the shared key and set the tag:

     ```
     $hp = Get-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group
     Name>"
     $secret = Set-AzKeyVaultSecret -VaultName "<Key Vault Name>" -Name "adfsssosecret" -
     SecretValue (ConvertTo-SecureString -String $config.SSOClientSecret  -AsPlainText -Force) -Tag
     @{ 'AllowedWVDSubscriptions' = $hp.Id.Split('/')[2]}
     ```

   - If your certificate is already in the Key Vault, run the following PowerShell cmdlet to set the tag:

     ```
     $hp = Get-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group
     Name>"
     $secret = Update-AzKeyVaultCertificate -VaultName "<Key Vault Name>" -Name "<Certificate
     Name>" -Tag @{ 'AllowedWVDSubscriptions' = $hp.Id.Split('/')[2]} -PassThru
     ```

   - If you have a local certificate, run the following PowerShell cmdlet to import the certificate in the Key Vault and set the tag:

     ```
     $hp = Get-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group
     Name>"
     $secret = Import-AzKeyVaultCertificate -VaultName "<Key Vault Name>" -Name "adfsssosecret" -
     Tag @{ 'AllowedWVDSubscriptions' = $hp.Id.Split('/')[2]} -FilePath "<Path to pfx>" -Password
     (ConvertTo-SecureString -String "<pfx password>"  -AsPlainText -Force)
     ```

> **NOTE**
> You can optionally configure how often users are prompted for credentials by changing the AD FS single sign-on settings. By default, users will be prompted every 8 hours on unregistered devices.

Configure your Azure Virtual Desktop host pool

It's time to configure the AD FS SSO parameters on your Azure Virtual Desktop host pool. To do this, set up your PowerShell environment for Azure Virtual Desktop if you haven't already and connect to your account.

After that, update the SSO information for your host pool by running one of the following two cmdlets in the same PowerShell window on the AD FS VM:

- If you're using a shared key in the Key Vault, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group Name>" -
SsoadfsAuthority "<ADFSServiceUrl>" -SsoClientId "<WVD Web App URI>" -SsoSecretType
SharedKeyInKeyVault -SsoClientSecretKeyVaultPath $secret.Id
```

> **NOTE**
> You need to set the SsoClientId property to match the Azure cloud you're deploying SSO in. In the Azure Commercial Cloud, this property should be set to `https://www.wvd.microsoft.com`. However, the required setting for this property will be different for other clouds, like the Azure Government cloud.

- If you're using a certificate in the Key Vault, run the following PowerShell cmdlet:

```
Update-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group Name>" -
SsoadfsAuthority "<ADFSServiceUrl>" -SsoClientId "<WVD Web App URI>" -SsoSecretType
CertificateInKeyVault -SsoClientSecretKeyVaultPath $secret.Id
```

> **NOTE**
> You need to set the SsoClientId property to match the Azure cloud you're deploying SSO in. In the Azure Commercial Cloud, this property should be set to `https://www.wvd.microsoft.com`. However, the required setting for this property will be different for other clouds, like the Azure Government cloud.

Configure additional host pools

When you need to configure additional host pools, you can retrieve the settings you used to configure an existing host pool to setup the new one.

To retrieve the settings from your existing host pool, open a PowerShell window and run this cmdlet:

```
Get-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group Name>" | fl *
```

You can follow the steps to Configure your Azure Virtual Desktop host pool using the same *SsoClientId*, *SsoClientSecretKeyVaultPath*, *SsoSecretType*, and *SsoadfsAuthority* values.

Removing SSO

To disable SSO on the host pool, run the following cmdlet:

```
Update-AzWvdHostPool -Name "<Host Pool Name>" -ResourceGroupName "<Host Pool Resource Group Name>" -
SsoadfsAuthority ''
```

If you also want to disable SSO on your AD FS server, run this cmdlet:

```
Install-Script UnConfigureWVDSSO
UnConfigureWVDSSO.ps1 -WvdWebAppAppIDUri "<WVD Web App URI>" -WvdClientAppApplicationID "a85cf173-4192-42f8-
81fa-777a763e6e2c"
```

> **NOTE**
> The WvdWebAppAppIDUri property needs to match the Azure cloud you are deploying in. In the Azure Commercial Cloud, this property is `https://www.wvd.microsoft.com`. It will be different for other clouds like the Azure Government cloud.

Next steps

Now that you've configured single sign-on, you can sign in to a supported Azure Virtual Desktop client to test it as part of a user session. If you want to learn how to connect to a session using your new credentials, check out these articles:

- Connect with the Windows Desktop client
- Connect with the web client

Configure Microsoft Endpoint Configuration Manager

12/6/2021 • 2 minutes to read • Edit Online

This article explains how to configure Microsoft Endpoint Configuration Manager to automatically apply updates to a Azure Virtual Desktop host running Windows 10 Enterprise multi-session.

Prerequisites

To configure this setting, you'll need the following things:

- Make sure you've installed the Microsoft Endpoint Configuration Manager Agent on your virtual machines.
- Make sure your version of Microsoft Endpoint Configuration Manager is at least on branch level 1906. For best results, use branch level 1910 or higher.

Receiving updates for Windows 10 and 11 Enterprise multi-session

You can update Windows 10 Enterprise multi-session with the corresponding Windows 10 client updates. For example, you can update Windows 10 Enterprise multi-session, version 21H2 by installing the Windows 10, version 21H2 client updates.

> **NOTE**
> Currently, you can't update Windows 10 Enterprise multi-session version 21H2 and Windows 11 Enterprise multi-session with their corresponding Windows client updates.

Create a query-based collection

To create a collection of Windows 10 Enterprise multi-session virtual machines, a query-based collection can be used to identify the specific operating system SKU.

To create a collection:

1. Select **Assets and Compliance**.
2. Go to **Overview** > **Device Collections** and right-click **Device collections** and select **Create Device Collection** from the drop-down menu.
3. In the **General** tab of the menu that opens, enter a name that describes your collection in the **Name** field. In the **Comment** field, you can give additional information describing what the collection is. In **Limiting Collection**, define which machines you're including in the collection query.
4. In the **Membership Rules** tab, add a rule for your query by selecting **Add Rule**, then selecting **Query Rule**.
5. In **Query Rule Properties**, enter a name for your rule, then define the parameters of the rule by selecting **Edit Query Statement**.
6. Select **Show Query Statement**.
7. In the statement, enter the following string:

```
select
SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.ResourceType,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,
SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client
from SMS_R_System inner join SMS_G_System_OPERATING_SYSTEM on
SMS_G_System_OPERATING_SYSTEM.ResourceId = SMS_R_System.ResourceId where
SMS_G_System_OPERATING_SYSTEM.OperatingSystemSKU = 175
```

8. Select **OK** to create the collection.
9. To check if you successfully created the collection, go to **Assets and Compliance** > **Overview** > **Device Collections**.

Multimedia redirection for Azure Virtual Desktop (preview)

12/6/2021 • 7 minutes to read • Edit Online

IMPORTANT

Multimedia redirection for Azure Virtual Desktop is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

NOTE

Azure Virtual Desktop doesn't currently support multimedia redirection on Azure Virtual Desktop for Microsoft 365 Government (GCC), GCC-High environments, and Microsoft 365 DoD.

Multimedia redirection on Azure Virtual Desktop is only available for the Windows Desktop client on Windows 10 machines. Multimedia redirection requires the Windows Desktop client, version 1.2.2222 or later.

Multimedia redirection (MMR) gives you smooth video playback while watching videos in your Azure Virtual Desktop browser. Multimedia redirection remotes the media element from the browser to the local machine for faster processing and rendering. Both Microsoft Edge and Google Chrome support the multimedia redirection feature. However, the public preview version of multimedia redirection for Azure Virtual Desktop has restricted playback on YouTube. To test YouTube within your organization's deployment, you'll need to enable an extension.

Requirements

Before you can use Multimedia Redirection on Azure Virtual Desktop, you'll need to do these things:

1. Install the Windows Desktop client on a Windows 10 or Windows 10 IoT Enterprise device that meets the hardware requirements for Teams on a Windows PC. Installing version 1.2.2222 or later of the client will also install the multimedia redirection plugin (MsMmrDVCPlugin.dll) on the client device. To learn more about updates and new versions, see What's new in the Windows Desktop client.

2. Create a host pool for your users.

3. Configure the client machine to let your users access the Insiders program. To configure the client for the Insider group, set the following registry information:

   - **Key**: HKLM\Software\Microsoft\MSRDC\Policies
   - **Type**: REG_SZ
   - **Name**: ReleaseRing
   - **Data**: insider

   To learn more about the Insiders program, see Windows Desktop client for admins.

4. Use the MSI installer (MsMmrHostMri) to install the multimedia redirection extensions for your internet browser on your Azure VM. Multimedia redirection for Azure Virtual Desktop currently only supports Microsoft Edge and Google Chrome.

Managing group policies for the multimedia redirection browser

extension

Using the multimedia redirection MSI will install the browser extensions. However, as this service is still in public preview, user experience may vary. For more information about known issues, see Known issues.

In some cases, you can change the group policy to manage the browser extensions and improve user experience. For example:

- You can install the extension without user interaction.
- You can restrict which websites use multimedia redirection.
- You can pin the extension icon in Google Chrome by default. The extension icon is already pinned by default in Microsoft Edge, so you'll only need to change this setting in Chrome.

Configure Microsoft Edge group policies for multimedia redirection

To configure the group policies, you'll need to edit the Microsoft Edge Administrative Template. You should see the extension configuration options under **Administrative Templates Microsoft Edge Extensions** > **Configure extension management settings**.

The following code is an example of a Microsoft Edge group policy that makes the browser install the multimedia redirection extension and only lets multimedia redirection load on YouTube:

```
{ "joeclbldhdmoijbaagobkhlpfjglcihd": { "installation_mode": "force_installed", "runtime_allowed_hosts": [
"*://*.youtube.com" ], "runtime_blocked_hosts": [ "*://*" ], "update_url":
"https://edge.microsoft.com/extensionwebstorebase/v1/crx" } }
```

To learn more about group policy configuration, see Microsoft Edge group policy.

Configure Google Chrome group policies for multimedia redirection

To configure the Google Chrome group policies, you'll need to edit the Google Chrome Administrative Template. You should see the extension configuration options under **Administrative Templates** > **Google** > **Google Chrome Extensions** > **Extension management settings**.

The following example is much like the code example in Configure Microsoft Edge group policies for multimedia redirection. This policy will force the multimedia redirection extension to install with the icon pinned in the top-right menu, and will only allow multimedia redirection to load on YouTube.

```
{ "lfmemoeeciijgkjkgbgikoonlkabmlno": { "installation_mode": "force_installed", "runtime_allowed_hosts": [
"*://*.youtube.com" ], "runtime_blocked_hosts": [ "*://*" ], "toolbar_pin": "force_pinned", "update_url":
"https://clients2.google.com/service/update2/crx" } }
```

Additional information on configuring Google Chrome group policy.

Run the multimedia redirection extension manually on a browser

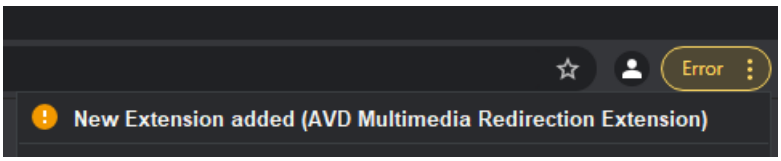
MMR uses remote apps and the session desktop for Microsoft Edge and Google Chrome browsers. Once you've fulfilled the requirements, open your supported browser. If you didn't install the browsers or extension with a group policy, users will need to manually run the extension. This section will tell you how to manually run the extension in one of the currently supported browsers.

Microsoft Edge

To run the extension on Microsoft Edge manually, look for the yellow exclamation mark on the overflow menu. You should see a prompt to enable the Azure Virtual Desktop Multimedia Redirection extension. Select **Enable extension**.

Google Chrome

To run the extension on Google Chrome manually, look for the notification message that says the new extension was installed, as shown in the following screenshot.



Select the notification to allow your users to enable the extension. Users should also pin the extension so that they can see from the icon if multimedia redirection is connected.

The multimedia redirection status icon

To quickly tell if multimedia redirection is active in your browser, we've added the following icon states:

ICON STATE	DEFINITION
	The default icon appearance with no status applied.
	The red square with an "X" inside of it means that the client couldn't connect to multimedia redirection.
	The green square with a check mark inside of it means that the client successfully connected to multimedia redirection.

Selecting the icon will display a pop-up menu that has a checkbox you can select to enable or disable multimedia redirection on all websites. It also lists the version numbers for each component of the service.

Support during public preview

Microsoft Support is not handling issues for multimedia redirection during public preview.

If you run into any issues, you can tell us in the feedback hub on both the client and VM host.

To send us feedback:

1. Open the **feedback hub** on both the client and server.
2. Select **Report a problem**.
3. Use the same title on both issue reports, but specify where you're submitting the report from by putting either "[Client]" or "[Host]" at the beginning.

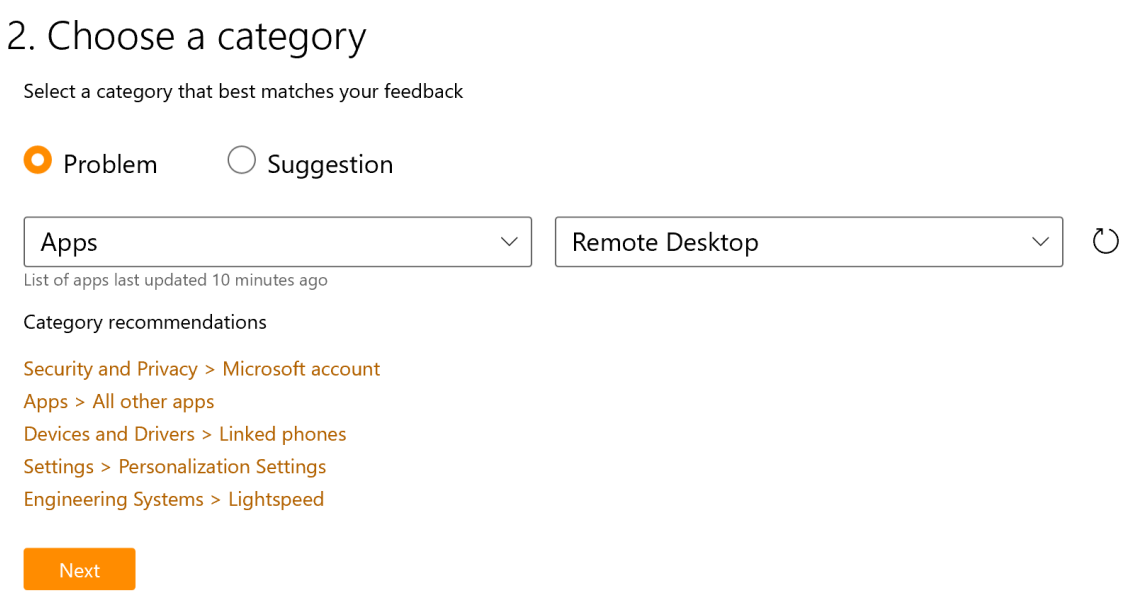
   For example, if you're submitting an issue from the client, you'd format it like this:

   > [Client] Title of your report

   If you're submitting an issue from the host, you'd format it like this:

   > [Host] Title of your report

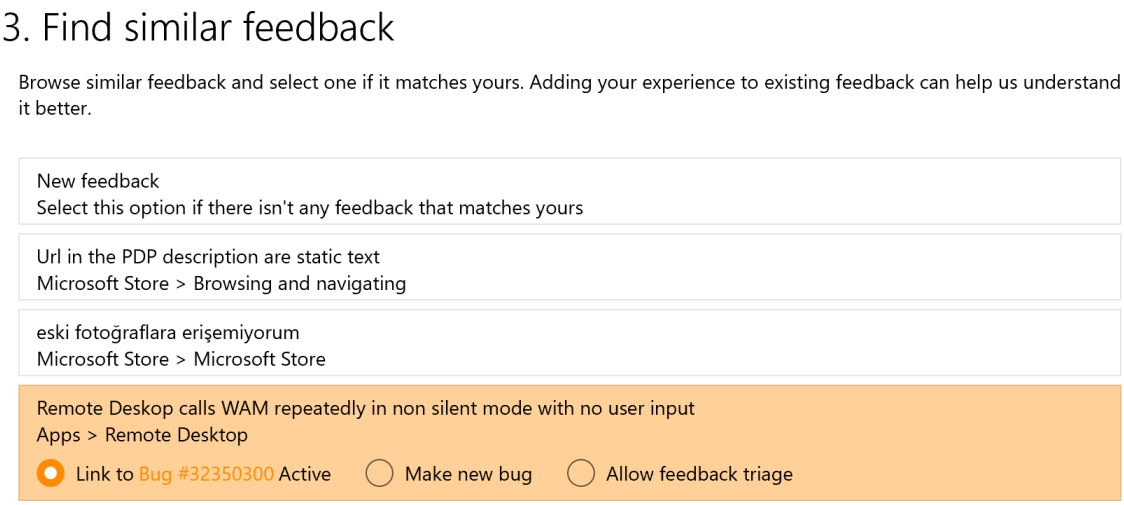
4. In the **Explain in more detail** field, describe the issue you're experiencing. We recommend also including the URL of the video you were watching when the issue happened.
5. Once you're done, select **Next**.
6. Select the **Problem** bubble, then select **Apps** and **Remote Desktop** from the two drop-down menus, as shown in the following screenshot.



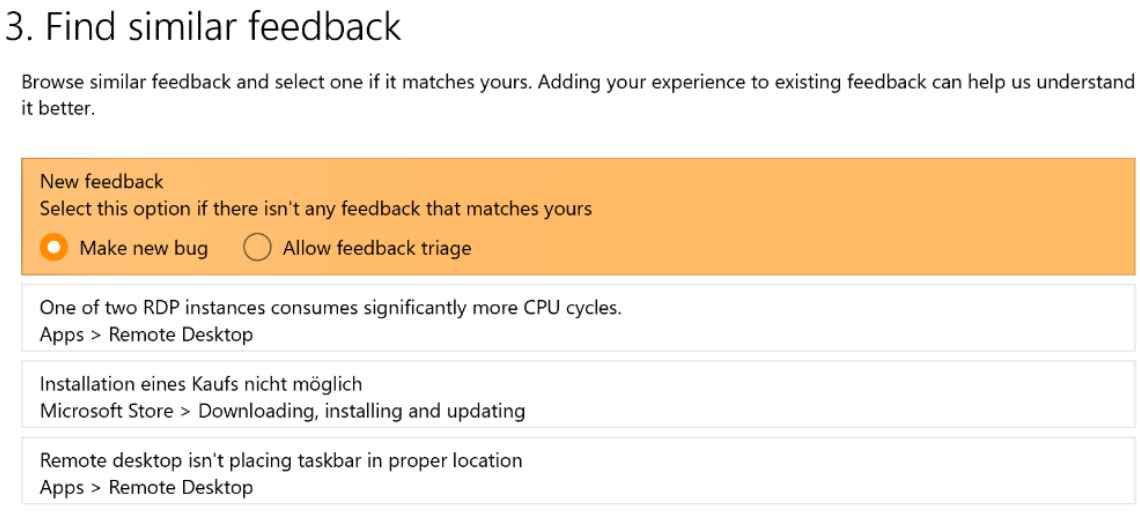
7. Select **Next**.

8. Check to see if there's a similar issue in the list to the one you plan to submit.

   - If a bubble appears that links to an active bug, make sure the bug's description matches the issue you're reporting. If it does, select the bubble, then select **Link to bug**, as shown in the following screenshot.



   - If you don't see a similar issue, select **Make new bug**.



9. Select **Next**.

10. In the **Add more details** window, select **Include data about Remote Desktop (Default)**, then answer all questions with as much detail as possible.

    If you'd like to add a video recording of the issue, select **Include data about Remote Desktop (Default)**, then select the **Start recording** button. While recording, open Remote Desktop and do the process that led to the issue happening. When you're done, return to the browser, then test the video to make sure it recorded properly.

    Once you're done, agree to send the attached files and diagnostics to Microsoft, then select **Submit**.

Known issues and limitations

The following issues are ones we're already aware of, so you won't need to report them:

- Multimedia redirection only works on the Windows Desktop client, not the web client.
- Multimedia redirection doesn't currently support protected content, so videos from Pluralsight and Netflix won't work.
- During public preview, multimedia redirection will be disabled on all sites except YouTube. However, if you have the extension, you can enable multimedia redirection for all websites. We added the extension so organizations can test the feature on their company websites.
- There's a small chance that the MSI installer won't be able to install the extension during internal testing. If you run into this issue, you'll need to install the multimedia redirection extension from the Microsoft Edge Store or Google Chrome Store.
    - Multimedia redirection browser extension (Microsoft Edge)
    - Multimedia browser extension (Google Chrome)
- Installing the extension on host machines with the MSI installer will either prompt users to accept the extension the first time they open the browser or display a warning or error message. If users deny this prompt, it can cause the extension to not load. To avoid this issue, install the extensions by editing the group policy.
- When you resize the video window, the window's size will adjust faster than the video itself. You'll also see this issue when minimizing and maximizing the window.

Next steps

If you're interested in video streaming on other parts of Azure Virtual Desktop, check out Teams for Azure Virtual Desktop.

Add language packs to a Windows 10 multi-session image

12/6/2021 • 8 minutes to read • Edit Online

Azure Virtual Desktop is a service that your users can deploy anytime, anywhere. That's why it's important that your users be able to customize which language their Windows 10 Enterprise multi-session image displays.

There are two ways you can accommodate the language needs of your users:

- Build dedicated host pools with a customized image for each language.
- Have users with different language and localization requirements in the same host pool, but customize their images to ensure they can select whichever language they need.

The latter method is a lot more efficient and cost-effective. However, it's up to you to decide which method best suits your needs. This article will show you how to customize languages for your images.

Prerequisites

You need the following things to customize your Windows 10 Enterprise multi-session images to add multiple languages:

- An Azure virtual machine (VM) with Windows 10 Enterprise multi-session, version 1903 or later
- The Language ISO, Feature on Demand (FOD) Disk 1, and Inbox Apps ISO of the OS version the image uses. You can download them here:
    - Language ISO:
        - Windows 10, version 1903 or 1909 Language Pack ISO
        - Windows 10, version 2004, 20H2 or 21H1 Language Pack ISO
    - FOD Disk 1 ISO:
        - Windows 10, version 1903 or 1909 FOD Disk 1 ISO
        - Windows 10, version 2004, 20H2 or 21H1 FOD Disk 1 ISO
    - Inbox Apps ISO:
        - Windows 10, version 1903 or 1909 Inbox Apps ISO
        - Windows 10, version 2004 Inbox Apps ISO
        - Windows 10, version 20H2 Inbox Apps ISO
        - Windows 10, version 21H1 Inbox Apps ISO
    - If you use Local Experience Pack (LXP) ISO files to localize your images, you will also need to download the appropriate LXP ISO for the best language experience
        - If you're using Windows 10, version 1903 or 1909:
            - Windows 10, version 1903 or 1909 LXP ISO
        - If you're using Windows 10, version 2004, 20H2, or 21H1, use the information in Adding languages in Windows 10: Known issues to figure out which of the following LXP ISOs is right for you:
            - Windows 10, version 2004, 20H2, or 21H1 **10C** LXP ISO
            - Windows 10, version 2004, 20H2, or 21H1 **11C** LXP ISO

    - Windows 10, version 2004, 20H2, or 21H1 **1C** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **2C** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **4B** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **5C** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **7C** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **9C** LXP ISO
    - Windows 10, version 2004, 20H2, or 21H1 **10C** LXP ISO
- An Azure Files Share or a file share on a Windows File Server Virtual Machine

> **NOTE**
> The file share (repository) must be accessible from the Azure VM you plan to use to create the custom image.

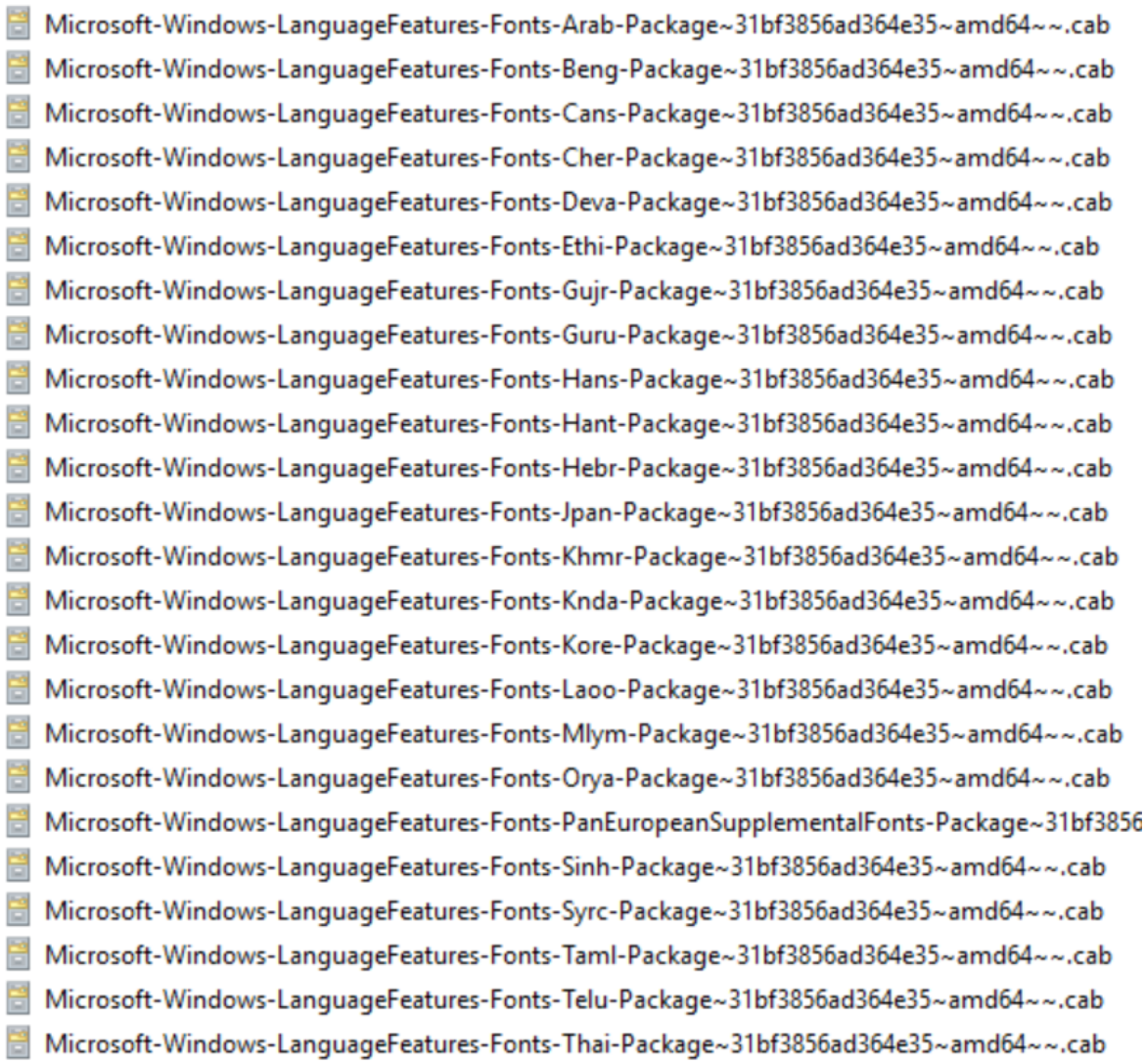
Create a content repository for language packages and features on demand

To create the content repository for language packages and FODs and a repository for the Inbox Apps packages:

1. On an Azure VM, download the Windows 10 Multi-Language ISO, FODs, and Inbox Apps for Windows 10 Enterprise multi-session, version 1903/1909, and 2004 images from the links in Prerequisites.
2. Open and mount the ISO files on the VM.
3. Go to the language pack ISO and copy the content from the **LocalExperiencePacks** and **x64\langpacks** folders, then paste the content into the file share.
4. Go to the **FOD ISO file**, copy all of its content, then paste it into the file share.
5. Go to the **amd64fre** folder on the Inbox Apps ISO and copy the content in the repository for the inbox apps that you've prepared.

   > **NOTE**
   > If you're working with limited storage, only copy the files for the languages you know your users need. You can tell the files apart by looking at the language codes in their file names. For example, the French file has the code "fr-FR" in its name. For a complete list of language codes for all available languages, see Available language packs for Windows.

   > **IMPORTANT**
   > Some languages require additional fonts included in satellite packages that follow different naming conventions. For example, Japanese font file names include “Jpan."



6. Set the permissions on the language content repository share so that you have read access from the VM you'll use to build the custom image.

Create a custom Windows 10 Enterprise multi-session image manually

To create a custom Windows 10 Enterprise multi-session image manually:

1. Deploy an Azure VM, then go to the Azure Gallery and select the current version of Windows 10 Enterprise multi-session you're using.
2. After you've deployed the VM, connect to it using RDP as a local admin.
3. Make sure your VM has all the latest Windows Updates. Download the updates and restart the VM, if necessary.
4. Connect to the language package, FOD, and Inbox Apps file share repository and mount it to a letter drive (for example, drive E).

Create a custom Windows 10 Enterprise multi-session image automatically

If you'd rather install languages through an automated process, you can set up a script in PowerShell. You can use the following script sample to install the Spanish (Spain), French (France), and Chinese (PRC) language packs and satellite packages for Windows 10 Enterprise multi-session, version 2004. The script integrates the language interface pack and all necessary satellite packages into the image. However, you can also modify this script to install other languages. Just make sure to run the script from an elevated PowerShell session, or else it won't work.

```
#########################################################
## Add Languages to running Windows Image for Capture##
#########################################################

##Disable Language Pack Cleanup##
Disable-ScheduledTask -TaskPath "\Microsoft\Windows\AppxDeploymentClient\" -TaskName "Pre-staged app
cleanup"

##Set Language Pack Content Stores##
[string]$LIPContent = "E:"

##Spanish##
Add-AppProvisionedPackage -Online -PackagePath $LIPContent\es-es\LanguageExperiencePack.es-es.Neutral.appx -
LicensePath $LIPContent\es-es\License.xml
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Client-Language-Pack_x64_es-es.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Basic-es-es-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Handwriting-es-es-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-OCR-es-es-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Speech-es-es-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-TextToSpeech-es-es-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-NetFx3-OnDemand-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-InternetExplorer-Optional-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-MSPaint-FoD-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Notepad-FoD-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-PowerShell-ISE-FOD-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Printing-WFS-FoD-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-StepsRecorder-
Package~31bf3856ad364e35~amd64~es-es~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-WordPad-FoD-
Package~31bf3856ad364e35~amd64~es-es~.cab
$LanguageList = Get-WinUserLanguageList
$LanguageList.Add("es-es")
Set-WinUserLanguageList $LanguageList -force

##French##
Add-AppProvisionedPackage -Online -PackagePath $LIPContent\fr-fr\LanguageExperiencePack.fr-fr.Neutral.appx -
LicensePath $LIPContent\fr-fr\License.xml
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Client-Language-Pack_x64_fr-fr.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Basic-fr-fr-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Handwriting-fr-fr-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-OCR-fr-fr-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Speech-fr-fr-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-TextToSpeech-fr-fr-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-NetFx3-OnDemand-
Package~31bf3856ad364e35~amd64~fr-fr~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-InternetExplorer-Optional-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-MSPaint-FoD-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Notepad-FoD-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-PowerShell-ISE-FOD-
```

```
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Printing-WFS-FoD-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-StepsRecorder-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-WordPad-FoD-
Package~31bf3856ad364e35~amd64~fr-FR~.cab
$LanguageList = Get-WinUserLanguageList
$LanguageList.Add("fr-fr")
Set-WinUserLanguageList $LanguageList -force

##Chinese(PRC)##
Add-AppProvisionedPackage -Online -PackagePath $LIPContent\zh-cn\LanguageExperiencePack.zh-cn.Neutral.appx -
LicensePath $LIPContent\zh-cn\License.xml
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Client-Language-Pack_x64_zh-cn.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Basic-zh-cn-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Fonts-Hans-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Handwriting-zh-cn-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-OCR-zh-cn-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-Speech-zh-cn-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-LanguageFeatures-TextToSpeech-zh-cn-
Package~31bf3856ad364e35~amd64~~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-NetFx3-OnDemand-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-InternetExplorer-Optional-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-MSPaint-FoD-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Notepad-FoD-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-PowerShell-ISE-FOD-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-Printing-WFS-FoD-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-StepsRecorder-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
Add-WindowsPackage -Online -PackagePath $LIPContent\Microsoft-Windows-WordPad-FoD-
Package~31bf3856ad364e35~amd64~zh-cn~.cab
$LanguageList = Get-WinUserLanguageList
$LanguageList.Add("zh-cn")
Set-WinUserLanguageList $LanguageList -force
```

The script might take a while depending on the number of languages you need to install.

Once the script is finished running, check to make sure the language packs installed correctly by going to **Start** > **Settings** > **Time & Language** > **Language**. If the language files are there, you're all set.

After adding additional languages to the Windows image, the inbox apps are also required to be updated to support the added languages. This can be done by refreshing the pre-installed apps with the content from the inbox apps ISO. To perform this refresh in an environment where the VM doesn't have internet access, you can use the following PowerShell script template to automate the process and update only installed versions of inbox apps.

```
########################################
## Update Inbox Apps for Multi Language##
########################################
##Set Inbox App Package Content Stores##
[string] $AppsContent = "F:\"

##Update installed Inbox Store Apps##
foreach ($App in (Get-AppxProvisionedPackage -Online)) {
 $AppPath = $AppsContent + $App.DisplayName + '_' + $App.PublisherId
 Write-Host "Handling $AppPath"
 $licFile = Get-Item $AppPath*.xml
 if ($licFile.Count) {
  $lic = $true
  $licFilePath = $licFile.FullName
 } else {
  $lic = $false
 }
 $appxFile = Get-Item $AppPath*.appx*
 if ($appxFile.Count) {
  $appxFilePath = $appxFile.FullName
  if ($lic) {
   Add-AppxProvisionedPackage -Online -PackagePath $appxFilePath -LicensePath $licFilePath
  } else {
   Add-AppxProvisionedPackage -Online -PackagePath $appxFilePath -skiplicense
  }
 }
}
```

IMPORTANT

The inbox apps included in the ISO aren't the latest versions of the pre-installed Windows apps. To get the latest version of all apps, you need to update the apps using the Windows Store App and perform an manual search for updates after you've installed the additional languages.

When you're done, make sure to disconnect the share.

Finish customizing your image

After you've installed the language packs, you can install any other software you want to add to your customized image.

Once you're finished customizing your image, you'll need to run the system preparation tool (sysprep).

To run sysprep:

1. Open an elevated command prompt and run the following command to generalize the image:

   ```
   C:\Windows\System32\Sysprep\sysprep.exe /oobe /generalize /shutdown
   ```

2. Stop the VM, then capture it in a managed image by following the instructions in Create a managed image of a generalized VM in Azure.

3. You can now use the customized image to deploy a Azure Virtual Desktop host pool. To learn how to deploy a host pool, see Tutorial: Create a host pool with the Azure portal.

Enable languages in Windows settings app

Finally, after you deploy the host pool, you'll need to add the language to each user's language list so they can

select their preferred language in the Settings menu.

To ensure your users can select the languages you installed, sign in as the user, then run the following PowerShell cmdlet to add the installed language packs to the Languages menu. You can also set up this script as an automated task or logon script that activates when the user signs in to their session.

```
$LanguageList = Get-WinUserLanguageList
$LanguageList.Add("es-es")
$LanguageList.Add("fr-fr")
$LanguageList.Add("zh-cn")
Set-WinUserLanguageList $LanguageList -force
```

After a user changes their language settings, they'll need to sign out of their Azure Virtual Desktop session and sign in again for the changes to take effect.

Next steps

If you're curious about known issues for language packs, see Adding language packs in Windows 10, version 1803 and later versions: Known issues.

If you have any other questions about Windows 10 Enterprise multi-session, check out our FAQ.

Add languages to a Windows 11 Enterprise image

12/6/2021 • 5 minutes to read • Edit Online

It's important to make sure users within your organization from all over the world can use your Azure Virtual Desktop deployment. That's why you can customize the Windows 11 Enterprise image you use for your virtual machines (VMs) to have different language packs. Starting with Windows 11, non-administrator user accounts can now add both the display language and its corresponding language features. This feature means you won't need to pre-install language packs for users in a personal host pool. For pooled host pools, we still recommend you add the languages you plan to add to a custom image. You can use the instructions in this article for both single-session and multi-session versions of Windows 11 Enterprise.

When your organization includes users with multiple different languages, you have two options:

- Create one dedicated host pool with a customized image per language.
- Have multiple users with different languages in the same host pool.

The second option is more efficient in terms of resources and cost, but requires a few extra steps. Fortunately, this article will help walk you through how to build an image that can accommodate users of all languages and localization needs.

Requirements

Before you can add languages to a Windows 11 Enterprise VM, you'll need to have the following things ready:

- An Azure VM with Windows 11 Enterprise installed
- A Language and Optional Features (LoF) ISO. You can download the ISO at Windows 11 Language and Optional Features LoF ISO
- An Azure Files share or a file share on a Windows File Server VM

> **NOTE**
> The file share repository must be accessible from the Azure VM that you're going to use to create the custom image.

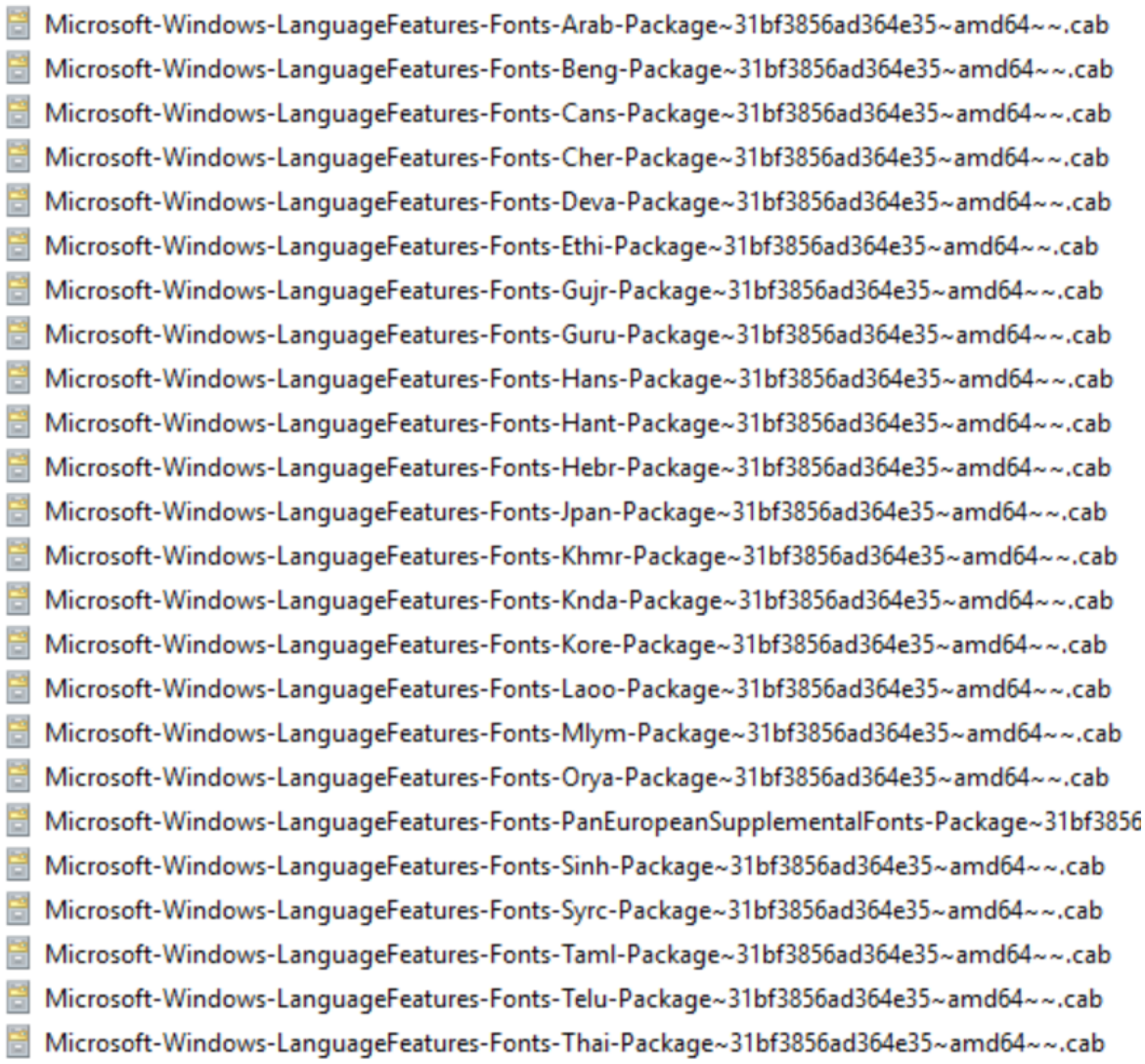
Create a content repository for language packages and features on demand

To create the content repository you'll use to add languages and features to your VM:

1. Open the VM you want to add languages to in Azure.
2. Open and mount the ISO file you downloaded in Requirements on the VM.
3. Create a folder on the file share.
4. Copy all content from the **LanguagesAndOptionalFeatures** folder in the ISO to the folder you created.

> **NOTE**
> If you're working with limited storage, you can use the mounted "Languages and Optional Features" ISO as a repository. To learn how to create a repository, see Build a custom FOD and language pack repository.

> **IMPORTANT**
>
> Some languages require additional fonts included in satellite packages that follow different naming conventions. For example, Japanese font file names include "Jpan."



5. Set the permissions on the language content repository share so that you have read access from the VM you'll use to build the custom image.

Create a custom Windows 11 Enterprise image manually

You can create a custom image by following these steps:

1. Deploy an Azure VM, then go to the Azure Gallery and select the current version of Windows 11 Enterprise you're using.
2. After you've deployed the VM, connect to it using RDP as a local admin.
3. Connect to the file share repository you created in Create a content repository for language packages and features on demand and mount it to a letter drive (for example, drive E).
4. Run the following PowerShell script from an elevated PowerShell session to install language packs and satellite packages on Windows 11 Enterprise:

```
#########################################################
## Add Languages to running Windows Image for Capture##
#########################################################
```

```
##Disable Language Pack Cleanup##
Disable-ScheduledTask -TaskPath "\Microsoft\Windows\AppxDeploymentClient\" -TaskName "Pre-staged app
cleanup"
Disable-ScheduledTask -TaskPath "\Microsoft\Windows\MUI\" -TaskName "LPRemove"
Disable-ScheduledTask -TaskPath "\Microsoft\Windows\LanguageComponentsInstaller" -TaskName
"Uninstallation"
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Control Panel\International" /v
"BlockCleanupOfUnusedPreinstalledLangPacks" /t REG_DWORD /d 1 /f

##Set Language Pack Content Stores##
$LIPContent = "E:"

##Set Path of CSV File##
$CSVFile = "Windows-10-1809-FOD-to-LP-Mapping-Table.csv"
$filePath = (Get-Location).Path + "/$CSVFile"

##Import Necesarry CSV File##
$FODList = Import-Csv -Path $filePath -Delimiter ";"

##Set Language (Target)##
$targetLanguage = "es-es"

$sourceLanguage = (($FODList | Where-Object {$_.'Target Lang' -eq $targetLanguage}) | Where-Object
{$_.'Source Lang' -ne $targetLanguage} | Select-Object -Property 'Source Lang' -Unique).'Source Lang'
if(!($sourceLanguage)){
    $sourceLanguage = $targetLanguage
}

$langGroup = (($FODList | Where-Object {$_.'Target Lang' -eq $targetLanguage}) | Where-Object
{$_.'Lang Group:' -ne ""} | Select-Object -Property 'Lang Group:' -Unique).'Lang Group:'

##List of additional features to be installed##
$additionalFODList = @(
    "$LIPContent\Microsoft-Windows-NetFx3-OnDemand-Package~31bf3856ad364e35~amd64~~.cab",
    "$LIPContent\Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~$sourceLanguage~.cab",
    "$LIPContent\Microsoft-Windows-SnippingTool-FoD-
Package~31bf3856ad364e35~amd64~$sourceLanguage~.cab",
    "$LIPContent\Microsoft-Windows-Lip-Language_x64_$sourceLanguage.cab" ##only if applicable##
)

$additionalCapabilityList = @(
 "Language.Basic~~~$sourceLanguage~0.0.1.0",
 "Language.Handwriting~~~$sourceLanguage~0.0.1.0",
 "Language.OCR~~~$sourceLanguage~0.0.1.0",
 "Language.Speech~~~$sourceLanguage~0.0.1.0",
 "Language.TextToSpeech~~~$sourceLanguage~0.0.1.0"
 )

 ##Install all FODs or fonts from the CSV file###
 Dism /Online /Add-Package /PackagePath:$LIPContent\Microsoft-Windows-Client-Language-
Pack_x64_$sourceLanguage.cab
 Dism /Online /Add-Package /PackagePath:$LIPContent\Microsoft-Windows-Lip-Language-
Pack_x64_$sourceLanguage.cab
 foreach($capability in $additionalCapabilityList){
    Dism /Online /Add-Capability /CapabilityName:$capability /Source:$LIPContent
 }

 foreach($feature in $additionalFODList){
 Dism /Online /Add-Package /PackagePath:$feature
 }

 if($langGroup){
 Dism /Online /Add-Capability /CapabilityName:Language.Fonts.$langGroup~~~und-$langGroup~0.0.1.0
 }

 ##Add installed language to language list##
 $LanguageList = Get-WinUserLanguageList
 $LanguageList.Add("$targetlanguage")
 Set-WinUserLanguageList $LanguageList -force
```

> **NOTE**
>
> This example script uses the Spanish (es-es) language code. To automatically install the appropriate files for a different language change the *$targetLanguage* parameter to the correct language code. For a list of language codes, see Available language packs for Windows.

The script might take a while to finish depending on the number of languages you need to install. You can also install additional languages after initial setup by running the script again with a different *$targetLanguage* parameter.

5. To automatically select the appropriate installation files, download and save the Available Windows 10 1809 Languages and Features on Demand table as a CSV file, then save it in the same folder as your PowerShell script.

6. Once the script is finished running, check to make sure the language packs installed correctly by going to **Start** > **Settings** > **Time & Language** > **Language**. If the language files are there, you're all set.

7. Finally, if the VM is connected to the Internet while installing languages, you'll need to run a cleanup process to remove any unnecessary language experience packs. To clean up the files, run these commands:

```
##Cleanup to prepare sysprep##
Remove-AppxPackage -Package Microsoft.LanguageExperiencePackes-ES_22000.8.13.0_neutral__8wekyb3d8bbwe

Remove-AppxPackage -Package Microsoft.OneDriveSync_22000.8.13.0_neutral__8wekyb3d8bbwe
```

To clean up different language packs, replace "es-ES" with a different language code.

8. Once you're done with cleanup, disconnect the share.

Finish customizing your image

After you've installed the language packs, you can install any other software you want to add to your customized image.

Once you're finished customizing your image, you'll need to run the system preparation tool (sysprep).

To run sysprep:

1. Open an elevated command prompt and run the following command to generalize the image:

```
C:\Windows\System32\Sysprep\sysprep.exe /oobe /generalize /shutdown
```

2. If you run into any issues, check the **SetupErr.log** file in your C drive at **Windows** > **System32** > **Sysprep** > **Panther**. After that, follow the instructions in Sysprep fails with Microsoft Store apps to troubleshoot your setup.

3. If setup is successful, stop the VM, then capture it in a managed image by following the instructions in Create a managed image of a generalized VM in Azure.

4. You can now use the customized image to deploy an Azure Virtual Desktop host pool. To learn how to deploy a host pool, see Tutorial: Create a host pool with the Azure portal.

> **NOTE**
>
> When a user changes their display language, they'll need to sign out of their Azure Virtual Desktop session, then sign back in. They must sign out from the Start menu.

Next steps

Learn how to install language packages for Windows 10 multi-session VMs at Add language packs to a Windows 10 multi-session image.

For a list of known issues, see Adding languages in Windows 10: Known issues.

Use Azure Advisor with Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

Azure Advisor can help users resolve common issues on their own without having to file support cases. The recommendations reduce the need to submit help requests, saving you time and costs.

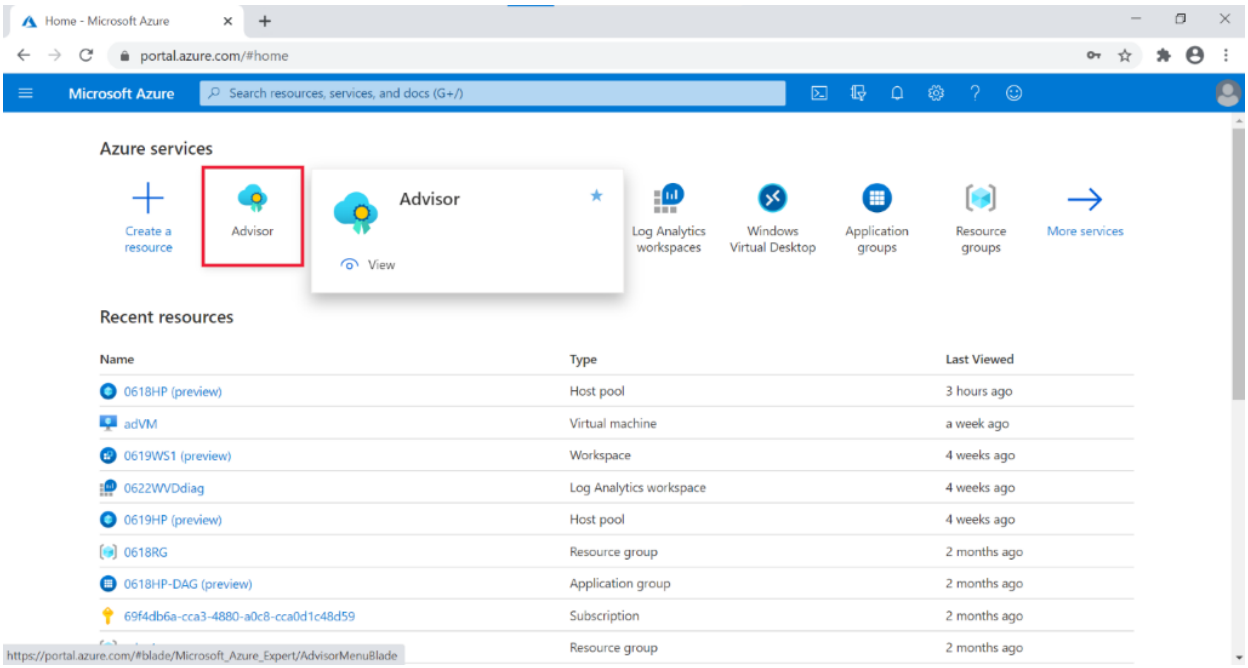
This article will tell you how to set up Azure Advisor in your Azure Virtual Desktop deployment to help your users.

What is Azure Advisor?

Azure Advisor analyzes your configurations and telemetry to offer personalized recommendations to solve common problems. With these recommendations, you can optimize your Azure resources for reliability, security, operational excellence, performance, and cost. Learn more at the Azure Advisor website.

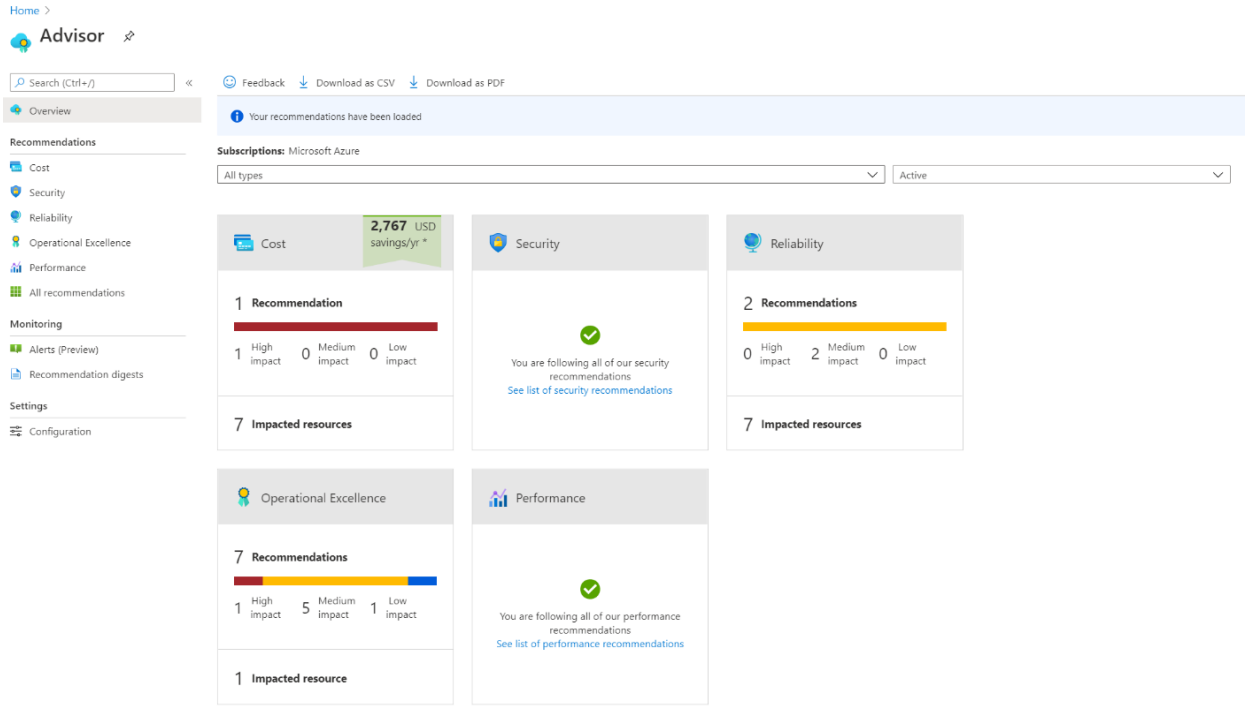
How to start using Azure Advisor

All you need to get started is an Azure account on the Azure portal. First, open the Azure portal at https://portal.azure.com/#home, then select **Advisor** under **Azure Services**, as shown in the following image. You can also enter "Azure Advisor" into the search bar in the Azure portal.

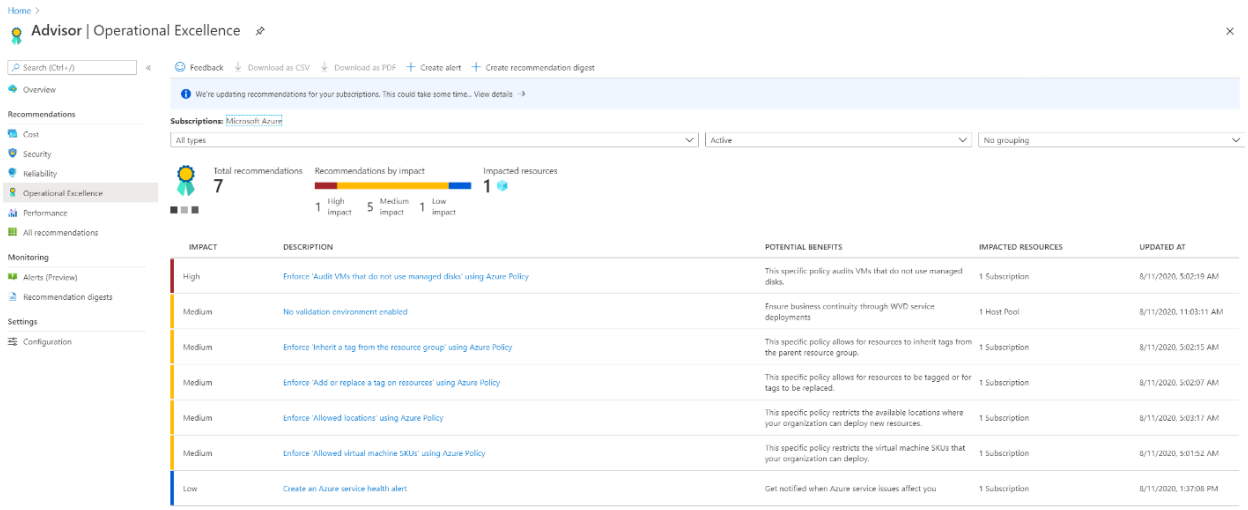


When you open Azure Advisor, you'll see five categories:

- Cost
- Security
- Reliability
- Operational Excellence
- Performance



When you select a category, you'll go to its active recommendations page. On this page, you can view which recommendations Azure Advisor has for you, as shown in the following image.



Additional tips for Azure Advisor

- Make sure to check your recommendations frequently, at least more than once a week. Azure Advisor updates its active recommendations multiple times per day. Checking for new recommendations can prevent larger issues by helping you spot and solve smaller ones.
- Always try to solve the issues with the highest priority level in Azure Advisor. High priority issues are marked with red. Leaving high-priority recommendations unresolved can lead to problems down the line.
- If a recommendation seems less important, you can dismiss it or postpone it. To dismiss or postpone a recommendation, go to the **Action** column and change the item's state.
- Don't dismiss recommendations until you know why they're appearing and are sure it won't have a negative impact on you or your users. Always select **Learn more** to see what the issue is. If you resolve an issue by following the instructions in Azure Advisor, it will automatically disappear from the list. You're better off resolving issues than postponing them repeatedly.
- Whenever you come across an issue in Azure Virtual Desktop, always check Azure Advisor first. Azure Advisor will give you directions for how to solve the problem, or at least point you towards a resource

that can help.

Next steps

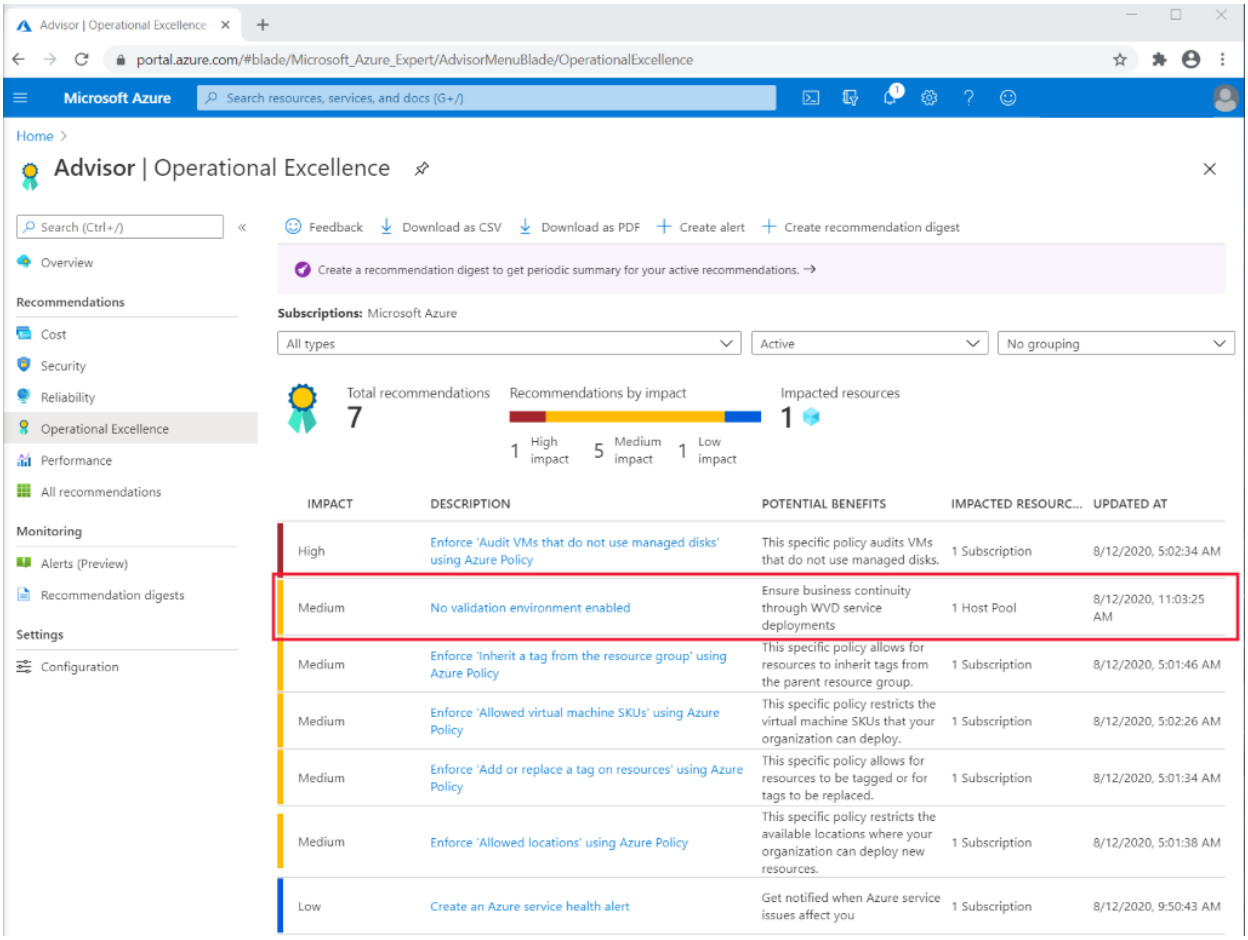
To learn how to resolve recommendations, see How to resolve Azure Advisor recommendations.

How to resolve Azure Advisor recommendations

12/6/2021 • 2 minutes to read • Edit Online

This article describes how you can resolve recommendations that appear in Azure Advisor for Azure Virtual Desktop.

"No validation environment enabled"



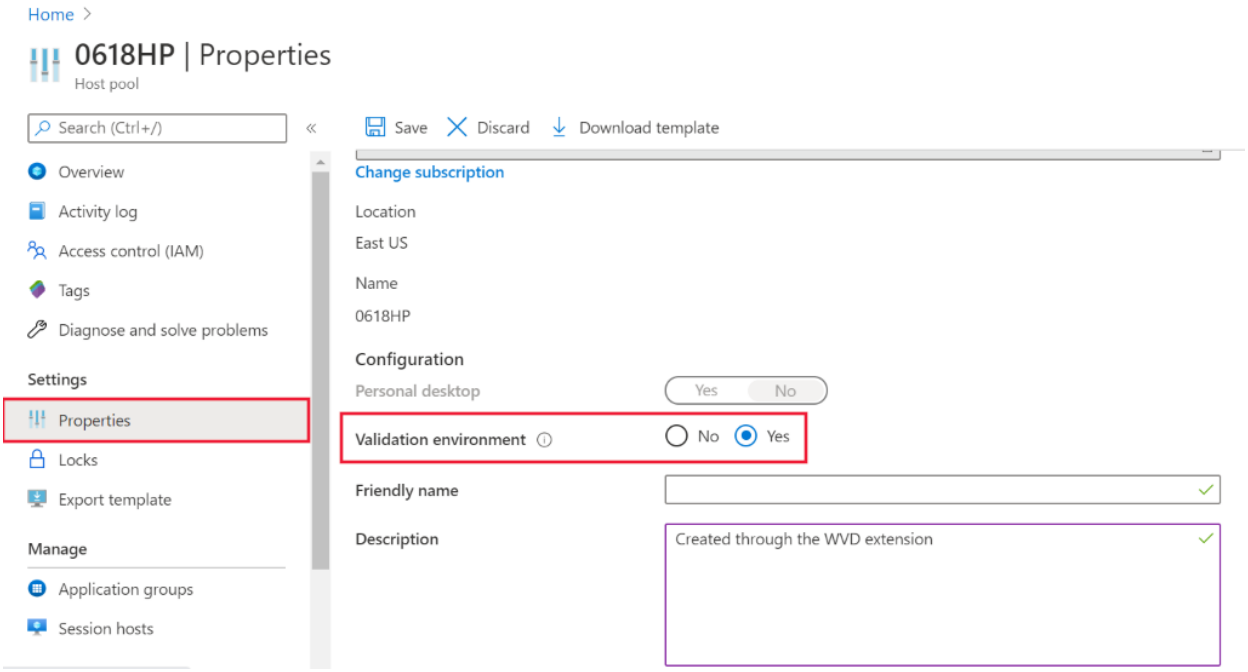
This recommendation appears under Operational Excellence. The recommendation should also show you a warning message like this:

"You don't have a validation environment enabled in this subscription. When you made your host pools, you selected **No** for "Validation environment" in the Properties tab. To ensure business continuity through Azure Virtual Desktop service deployments, make sure you have at least one host pool with a validation environment where you can test for potential issues."

You can make this warning message go away by enabling a validation environment in one of your host pools.

To enable a validation environment:

1. Go to your Azure portal home page and select the host pool you want to change.
2. Next, select the host pool you want to change from a production environment to a validation environment.
3. In your host pool, select **Properties** on the left column. Next, scroll down until you see "Validation environment." Select **Yes**, then select **Apply**.



These changes won't make the warning go away immediately, but it should disappear eventually. Azure Advisor updates twice a day. Until then, you can postpone or dismiss the recommendation manually. We recommend you let the recommendation go away on its own. That way, Azure Advisor can let you know if it comes across any problems as the settings change.

"Not enough production (non-validation) environments enabled"

This recommendation appears under Operational Excellence.

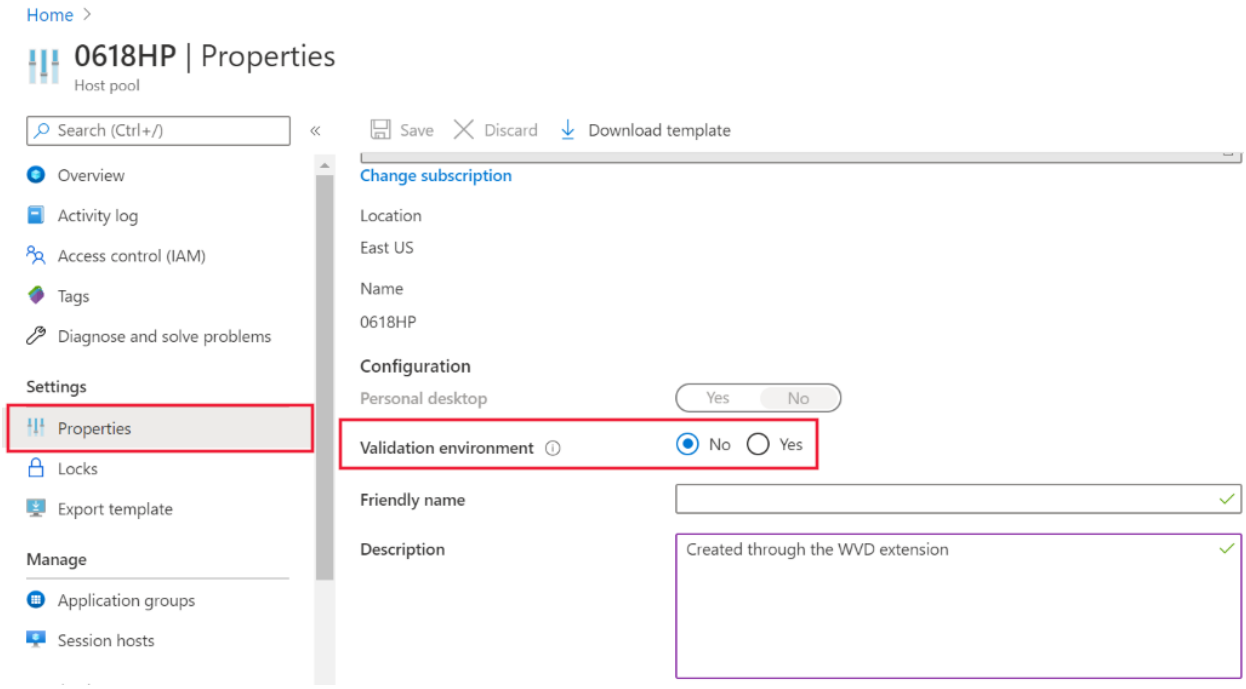
For this recommendation, the warning message appears for one of these reasons:

- You have too many host pools in your validation environment.
- You don't have any production host pools.

We recommend users have fewer than half of their host pools in a validation environment.

To resolve this warning:

1. Go to your Azure portal home page.
2. Select the host pools you want either want to change from validation to production.
3. In your host pool, select the **Properties** tab in the column on the right side of the screen. Next, scroll down until you see "Validation environment." Select **No**, then select **Apply**.



These changes won't make the warning go away immediately, but it should disappear eventually. Azure Advisor updates twice a day. Until then, you can postpone or dismiss the recommendation manually. We recommend you let the recommendation go away on its own. That way, Azure Advisor can let you know if it comes across any problems as the settings change.

"Not enough links are unblocked to successfully implement your VM"

This recommendation appears under Operational Excellence.

You need to unblock specific URLs to make sure that your virtual machine (VM) functions properly. You can see the list at Safe URL list. If the URLs aren't unblocked, then your VM won't work properly.

To solve this recommendation, make sure you unblock all the URLs on the Safe URL list. You can use Service Tag or FQDN tags to unblock URLs, too.

Next steps

If you're looking for more in-depth guides about how to resolve common issues, check out Troubleshooting overview, feedback, and support for Azure Virtual Desktop.

Use Azure Monitor for Azure Virtual Desktop to monitor your deployment

12/6/2021 • 9 minutes to read • Edit Online

Azure Monitor for Azure Virtual Desktop is a dashboard built on Azure Monitor Workbooks that helps IT professionals understand their Azure Virtual Desktop environments. This topic will walk you through how to set up Azure Monitor for Azure Virtual Desktop to monitor your Azure Virtual Desktop environments.

Requirements

Before you start using Azure Monitor for Azure Virtual Desktop, you'll need to set up the following things:

- All Azure Virtual Desktop environments you monitor must be based on the latest release of Azure Virtual Desktop that's compatible with Azure Resource Manager.
- At least one configured Log Analytics Workspace. Use a designated Log Analytics workspace for your Azure Virtual Desktop session hosts to ensure that performance counters and events are only collected from session hosts in your Azure Virtual Desktop deployment.
- Enable data collection for the following things in your Log Analytics workspace:
  - Diagnostics from your Azure Virtual Desktop environment
  - Recommended performance counters from your Azure Virtual Desktop session hosts
  - Recommended Windows Event Logs from your Azure Virtual Desktop session hosts

The data setup process described in this article is the only one you'll need to monitor Azure Virtual Desktop. You can disable all other items sending data to your Log Analytics workspace to save costs.

Anyone monitoring Azure Monitor for Azure Virtual Desktop for your environment will also need the following read-access permissions:

- Read-access to the Azure subscriptions that hold your Azure Virtual Desktop resources
- Read-access to the subscription's resource groups that hold your Azure Virtual Desktop session hosts
- Read access to the Log Analytics workspace or workspaces

> **NOTE**
> Read access only lets admins view data. They'll need different permissions to manage resources in the Azure Virtual Desktop portal.

Open Azure Monitor for Azure Virtual Desktop

You can open Azure Monitor for Azure Virtual Desktop with one of the following methods:

- Go to aka.ms/azmonwvdi.
- Search for and select **Azure Virtual Desktop** from the Azure portal, then select **Insights**.
- Search for and select **Azure Monitor** from the Azure portal. Select **Insights Hub** under **Insights**, then select **Azure Virtual Desktop**. Once you have the page open, enter the **Subscription**, **Resource group**, **Host pool**, and **Time range** of the environment you want to monitor.

> **NOTE**
> Azure Virtual Desktop currently only supports monitoring one subscription, resource group, and host pool at a time. If you can't find the environment you want to monitor, see our troubleshooting documentation for known feature requests and issues.

Log Analytics settings

To start using Azure Monitor for Azure Virtual Desktop, you'll need at least one Log Analytics workspace. Use a designated Log Analytics workspace for your Azure Virtual Desktop session hosts to ensure that performance counters and events are only collected form session hosts in your Azure Virtual Desktop deployment. If you already have a workspace set up, skip ahead to Set up using the configuration workbook. To set one up, see Create a Log Analytics workspace in the Azure portal.

> **NOTE**
> Standard data storage charges for Log Analytics will apply. To start, we recommend you choose the pay-as-you-go model and adjust as you scale your deployment and take in more data. To learn more, see Azure Monitor pricing.

Set up using the configuration workbook

If it's your first time opening Azure Monitor for Azure Virtual Desktop, you'll need set up Azure Monitor for your Azure Virtual Desktop environment. To configure your resources:

1. Open Azure Monitor for Azure Virtual Desktop in the Azure portal at aka.ms/azmonwvdi, then select **configuration workbook**.
2. Select an environment to configure under **Subscription**, **Resource Group**, and **Host Pool**.

The configuration workbook sets up your monitoring environment and lets you check the configuration after you've finished the setup process. It's important to check your configuration if items in the dashboard aren't displaying correctly, or when the product group publishes updates that require new settings.

Resource diagnostic settings

To collect information on your Azure Virtual Desktop infrastructure, you'll need to enable several diagnostic settings on your Azure Virtual Desktop host pools and workspaces (this is your Azure Virtual Desktop workspace, not your Log Analytics workspace). To learn more about host pools, workspaces, and other Azure Virtual Desktop resource objects, see our environment guide.

You can learn more about Azure Virtual Desktop diagnostics and the supported diagnostic tables at Send Azure Virtual Desktop diagnostics to Log Analytics.

To set your resource diagnostic settings in the configuration workbook:

1. Select the **Resource diagnostic settings** tab in the configuration workbook.
2. Select **Log Analytics workspace** to send Azure Virtual Desktop diagnostics.

Host pool diagnostic settings

To set up host pool diagnostics using the resource diagnostic settings section in the configuration workbook:

1. Under **Host pool**, check to see whether Azure Virtual Desktop diagnostics are enabled. If they aren't, an error message will appear that says "No existing diagnostic configuration was found for the selected host pool." You'll need to enable the following supported diagnostic tables:
    - Checkpoint
    - Error

- Management
- Connection
- HostRegistration
- AgentHealthStatus

> **NOTE**
> If you don't see the error message, you don't need to do steps 2 through 4.

2. Select **Configure host pool**.
3. Select **Deploy**.
4. Refresh the configuration workbook.

Workspace diagnostic settings

To set up workspace diagnostics using the resource diagnostic settings section in the configuration workbook:

1. Under **Workspace**, check to see whether Azure Virtual Desktop diagnostics are enabled for the Azure Virtual Desktop workspace. If they aren't, an error message will appear that says "No existing diagnostic configuration was found for the selected workspace." You'll need to enable the following supported diagnostics tables:

   - Checkpoint
   - Error
   - Management
   - Feed

   > **NOTE**
   > If you don't see the error message, you don't need to do steps 2-4.

2. Select **Configure workspace**.
3. Select **Deploy**.
4. Refresh the configuration workbook.

Session host data settings

To collect information on your Azure Virtual Desktop session hosts, you'll need to install the Log Analytics agent on all session hosts in the host pool, make sure the session hosts are sending to a Log Analytics workspace, and configure your Log Analytics agent settings to collect performance data and Windows Event Logs.

The Log Analytics workspace you send session host data to doesn't have to be the same one you send diagnostic data to. If you have Azure session hosts outside of your Azure Virtual Desktop environment, we recommend having a designated Log Analytics workspace for the Azure Virtual Desktop session hosts.

To set the Log Analytics workspace where you want to collect session host data:

1. Select the **Session host data settings** tab in the configuration workbook.
2. Select the **Log Analytics workspace** you want to send session host data to.

Session hosts

You'll need to install the Log Analytics agent on all session hosts in the host pool and send data from those hosts to your selected Log Analytics workspace. If Log Analytics isn't configured for all the session hosts in the host pool, you'll see a **Session hosts** section at the top of **Session host data settings** with the message "Some

hosts in the host pool are not sending data to the selected Log Analytics workspace."

> **NOTE**
> If you don't see the **Session hosts** section or error message, all session hosts are set up correctly. Skip ahead to set up instructions for Workspace performance counters.

To set up your remaining session hosts using the configuration workbook:

1. Select **Add hosts to workspace**.
2. Refresh the configuration workbook.

> **NOTE**
> The host machine needs to be running to install the Log Analytics extension. If automatic deployment doesn't work, you can install the extension on a host manually instead. To learn how to install the extension manually, see Log Analytics virtual machine extension for Windows.

Workspace performance counters

You'll need to enable specific performance counters to collect performance information from your session hosts and send it to the Log Analytics workspace.

If you already have performance counters enabled and want to remove them, follow the instructions in Configuring performance counters. You can add and remove performance counters in the same location.

To set up performance counters using the configuration workbook:

1. Under **Workspace performance counters** in the configuration workbook, check **Configured counters** to see the counters you've already enabled to send to the Log Analytics workspace. Check **Missing counters** to make sure you've enabled all required counters.
2. If you have missing counters, select **Configure performance counters**.
3. Select **Apply Config**.
4. Refresh the configuration workbook.
5. Make sure all the required counters are enabled by checking the **Missing counters** list.

Configure Windows Event Logs

You'll also need to enable specific Windows Event Logs to collect errors, warnings, and information from the session hosts and send them to the Log Analytics workspace.

If you've already enabled Windows Event Logs and want to remove them, follow the instructions in Configuring Windows Event Logs. You can add and remove Windows Event Logs in the same location.

To set up Windows Event Logs using the configuration workbook:

1. Under **Windows Event Logs configuration**, check **Configured Event Logs** to see the Event Logs you've already enabled to send to the Log Analytics workspace. Check **Missing Event Logs** to make sure you've enabled all Windows Event Logs.
2. If you have missing Windows Event Logs, select **Configure Events**.
3. Select **Deploy**.
4. Refresh the configuration workbook.
5. Make sure all the required Windows Event Logs are enabled by checking the **Missing Event Logs** list.

> **NOTE**
> If automatic event deployment fails, select **Open agent configuration** in the configuration workbook to manually add any missing Windows Event Logs.

Optional: configure alerts

Azure Monitor for Azure Virtual Desktop allows you to monitor Azure Monitor alerts happening within your selected subscription in the context of your Azure Virtual Desktop data. Azure Monitor alerts are an optional feature on your Azure subscriptions, and you need to set them up separately from Azure Monitor for Azure Virtual Desktop. You can use the Azure Monitor alerts framework to set custom alerts on Azure Virtual Desktop events, diagnostics, and resources. To learn more about Azure Monitor alerts, see Azure Monitor Log Alerts.

Diagnostic and usage data

Microsoft automatically collects usage and performance data through your use of the Azure Monitor service. Microsoft uses this data to improve the quality, security, and integrity of the service.

To provide accurate and efficient troubleshooting capabilities, the collected data includes the portal session ID, Azure Active Directory user ID, and the name of the portal tab where the event occurred. Microsoft doesn't collect names, addresses, or other contact information.

For more information about data collection and usage, see the Microsoft Online Services Privacy Statement.

> **NOTE**
> To learn about viewing or deleting your personal data collected by the service, see Azure Data Subject Requests for the GDPR. For more information about GDPR, see the GDPR section of the Service Trust portal.

Next steps

Now that you've configured Azure Monitor for your Azure Virtual Desktop environment, here are some resources that might help you start monitoring your environment:

- Check out our glossary to learn more about terms and concepts related to Azure Monitor for Azure Virtual Desktop.
- To estimate, measure, and manage your data storage costs, see Estimate Azure Monitor costs.
- If you encounter a problem, check out our troubleshooting guide for help and known issues.
- To see what's new in each version update, see What's new in Azure Monitor for Azure Virtual Desktop.

Azure Virtual Desktop disaster recovery

12/6/2021 • 8 minutes to read • Edit Online

To keep your organization's data safe, you may need to adopt a business continuity and disaster recovery (BCDR) strategy. A sound BCDR strategy keeps your apps and workload up and running during planned and unplanned service or Azure outages.

Azure Virtual Desktop offers BCDR for the Azure Virtual Desktop service to preserve customer metadata during outages. When an outage occurs in a region, the service infrastructure components will fail over to the secondary location and continue functioning as normal. You can still access service-related metadata, and users can still connect to available hosts. End-user connections will stay online as long as the tenant environment or hosts remain accessible.

To make sure users can still connect during a region outage, you need to replicate their virtual machines (VMs) in a different location. During outages, the primary site fails over to the replicated VMs in the secondary location. Users can continue to access apps from the secondary location without interruption. On top of VM replication, you'll need to keep user identities accessible at the secondary location. If you're using profile containers, you'll also need to replicate them. Finally, make sure your business apps that rely on data in the primary location can fail over with the rest of the data.

To summarize, to keep your users connected during an outage, you'll need to do the following things in this order:

- Replicate the VMs in a secondary location.
- If you're using profile containers, set up data replication in the secondary location.
- Make sure user identities you set up in the primary location are available in the secondary location.
- Make sure any line-of-business applications relying on data in your primary location are failed over to the secondary location.

VM replication

First, you'll need to replicate your VMs to the secondary location. Your options for doing so depend on how your VMs are configured:

- You can configure all your VMs for both pooled and personal host pools with Azure Site Recovery. With this method, you'll only need to set up one host pool and its related app groups and workspaces.
- You can create a new host pool in the failover region while keeping all resources in your failover location turned off. For this method, you'd need to set up new app groups and workspaces in the failover region. You can then use an Azure Site Recovery plan to turn host pools on.
- You can create a host pool that's populated by VMs built in both the primary and failover regions while keeping the VMs in the failover region turned off. In this case, you only need to set up one host pool and its related app groups and workspaces. You can use an Azure Site Recovery plan to power on host pools with this method.

We recommend you use Azure Site Recovery to manage replicating VMs in other Azure locations, as described in Azure-to-Azure disaster recovery architecture. We especially recommend using Azure Site Recovery for personal host pools, because Azure Site Recovery supports both server-based and client-based SKUs.

If you use Azure Site Recovery, you won't need to register these VMs manually. The Azure Virtual Desktop agent in the secondary VM will automatically use the latest security token to connect to the service instance closest to it. The VM (session host) in the secondary location will automatically become part of the host pool. The end-user

will have to reconnect during the process, but apart from that, there are no other manual operations.

If there are existing user connections during the outage, before the admin can start failover to the secondary region, you need to end the user connections in the current region.

To disconnect users in Azure Virtual Desktop (classic), run this cmdlet:

```
Invoke-RdsUserSessionLogoff
```

To disconnect users in the Azure-integrated version of Azure Virtual Desktop, run this cmdlet:

```
Remove-AzWvdUserSession
```

Once you've signed out all users in the primary region, you can fail over the VMs in the primary region and let users connect to the VMs in the secondary region. For more information about how this process works, see Replicate Azure VMs to another Azure region.

Virtual network

Next, consider your network connectivity during the outage. You'll need to make sure you've set up a virtual network (VNET) in your secondary region. If your users need to access on-premises resources, you'll need to configure this VNET to access them. You can establish on-premises connections with a VPN, ExpressRoute, or virtual WAN.

We recommend you use Azure Site Recovery to set up the VNET in the failover region because it preserves your primary network's settings and doesn't need peering.

User identities

Next, ensure that the domain controller is available at the secondary location.

There are three ways to keep the domain controller available:

- Have Active Directory Domain Controller at secondary location
- Use an on-premises Active Directory Domain Controller
- Replicate Active Directory Domain Controller using Azure Site Recovery

User and app data

If you're using profile containers, the next step is to set up data replication in the secondary location. You have five options to store FSLogix profiles:

- Storage Spaces Direct (S2D)
- Network drives (VM with extra drives)
- Azure Files
- Azure NetApp Files
- Cloud Cache for replication

For more information, check out Storage options for FSLogix profile containers in Azure Virtual Desktop.

If you're setting up disaster recovery for profiles, these are your options:

- Set up Native Azure Replication (for example, Azure Files Standard storage account replication, Azure NetApp Files replication, or Azure Files Sync for file servers).

> **NOTE**
>
> NetApp replication is automatic after you first set it up. With Azure Site Recovery plans, you can add pre-scripts and post-scripts to fail over non-VM resources replicate Azure Storage resources.

- Set up FSLogix Cloud Cache for both app and user data.
- Set up disaster recovery for app data only to ensure access to business-critical data at all times. With this method, you can retrieve user data after the outage is over.

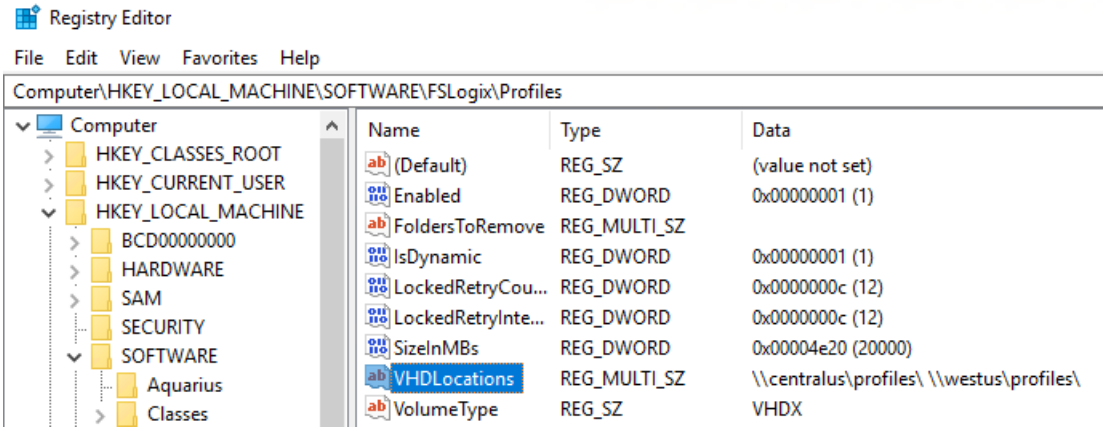
Let's take a look at how to configure FSLogix to set up disaster recovery for each option.

FSLogix configuration

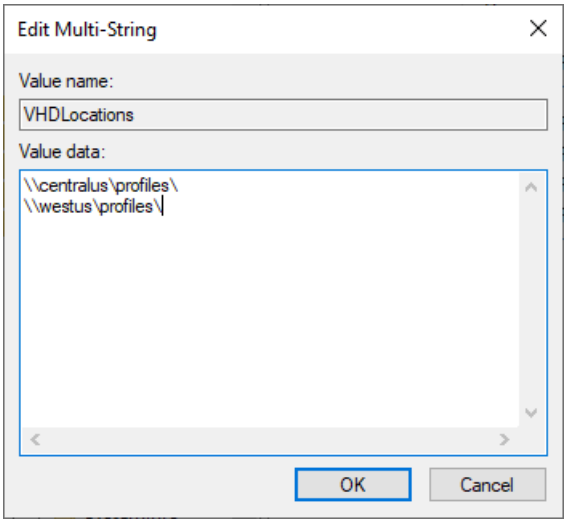
The FSLogix agent can support multiple profile locations if you configure the registry entries for FSLogix.

To configure the registry entries:

1. Open the **Registry Editor**.
2. Go to **Computer** > **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **FSLogix** > **Profiles**.



3. Right-click on **VHDLocations** and select **Edit Multi-String**.



4. In the **Value Data** field, enter the locations you want to use.
5. When you're done, select **OK**.

If the first location is unavailable, the FSLogix agent will automatically fail over to the second, and so on.

We recommend you configure the FSLogix agent with a path to the secondary location in the main region. Once the primary location shuts down, the FLogix agent will replicate as part of the VM Azure Site Recovery

replication. Once the replicated VMs are ready, the agent will automatically attempt to path to the secondary region.

For example, let's say your primary session host VMs are in the Central US region, but your profile container is in the Central US region for performance reasons.

In this case, you would configure the FSLogix agent with a path to the storage in Central US. You would configure the session host VMs to replicate in West US. Once the path to Central US fails, the agent will try to create a new path for storage in West US instead.

S2D

Since S2D handles replication across regions internally, you don't need to manually set up the secondary path.

Network drives (VM with extra drives)

If you replicate the network storage VMs using Azure Site Recovery like the session host VMs, then the recovery keeps the same path, which means you don't need to reconfigure FSlogix.

Azure Files

Azure Files supports cross-region asynchronous replication that you can specify when you create the storage account. If the asynchronous nature of Azure Files already covers your disaster recovery goals, then you don't need to do additional configuration.

If you need synchronous replication to minimize data loss, then we recommend you use FSLogix Cloud Cache instead.

> **NOTE**
> This section doesn't cover the failover authentication mechanism for Azure Files.

Azure NetApp Files

Learn more about Azure NetApp Files at Create replication peering for Azure NetApp Files.

App dependencies

Finally, make sure that any business apps that rely on data located in the primary region can fail over to the secondary location. Also, be sure to configure the settings the apps need to work in the new location. For example, if one of the apps is dependent on the SQL backend, make sure to replicate SQL in the secondary location. You should configure the app to use the secondary location as either part of the failover process or as its default configuration. You can model app dependencies on Azure Site Recovery plans. To learn more, see About recovery plans.

Disaster recovery testing

After you're done setting up disaster recovery, you'll want to test your plan to make sure it works.

Here are some suggestions for how to test your plan:

- If the test VMs have internet access, they will take over any existing session host for new connections, but all existing connections to the original session host will remain active. Make sure the admin running the test signs out all active users before testing the plan.
- You should only do full disaster recovery tests during a maintenance window to not disrupt your users. You can also use a host pool in the validation environment for the test.
- Make sure your test covers all business-critical apps.
- We recommend you only failover up to 100 VMs at a time. If you have more VMs than that, we recommend you fail them over in batches 10 minutes apart.

Next steps

If you have questions about how to keep your data secure in addition to planning for outages, check out our security guide.

Configure a Kerberos Key Distribution Center proxy

12/6/2021 • 2 minutes to read • Edit Online

Security-conscious customers, such as financial or government organizations, often sign in using Smartcards. Smartcards make deployments more secure by requiring multifactor authentication (MFA). However, for the RDP portion of a Azure Virtual Desktop session, Smartcards require a direct connection, or "line of sight," with an Active Directory (AD) domain controller for Kerberos authentication. Without this direct connection, users can't automatically sign in to the organization's network from remote connections. Users in a Azure Virtual Desktop deployment can use the KDC proxy service to proxy this authentication traffic and sign in remotely. The KDC proxy allows for authentication for the Remote Desktop Protocol of a Azure Virtual Desktop session, letting the user sign in securely. This makes working from home much easier, and allows for certain disaster recovery scenarios to run more smoothly.

However, setting up the KDC proxy typically involves assigning the Windows Server Gateway role in Windows Server 2016 or later. How do you use a Remote Desktop Services role to sign in to Azure Virtual Desktop? To answer that, let's take a quick look at the components.

There are two components to the Azure Virtual Desktop service that need to be authenticated:

- The feed in the Azure Virtual Desktop client that gives users a list of available desktops or applications they have access to. This authentication process happens in Azure Active Directory, which means this component isn't the focus of this article.
- The RDP session that results from a user selecting one of those available resources. This component uses Kerberos authentication and requires a KDC proxy for remote users.

This article will show you how to configure the feed in the Azure Virtual Desktop client in the Azure portal. If you want to learn how to configure the RD Gateway role, see Deploy the RD Gateway role.

Requirements

To configure a Azure Virtual Desktop session host with a KDC proxy, you'll need the following things:

- Access to the Azure portal and an Azure administrator account.
- The remote client machines must be running either Windows 10 or Windows 7 and have the Windows Desktop client installed. Currently, the web client is not supported.
- You must have a KDC proxy already installed on your machine. To learn how to do that, see Set up the RD Gateway role for Azure Virtual Desktop.
- The machine's OS must be Windows Server 2016 or later.

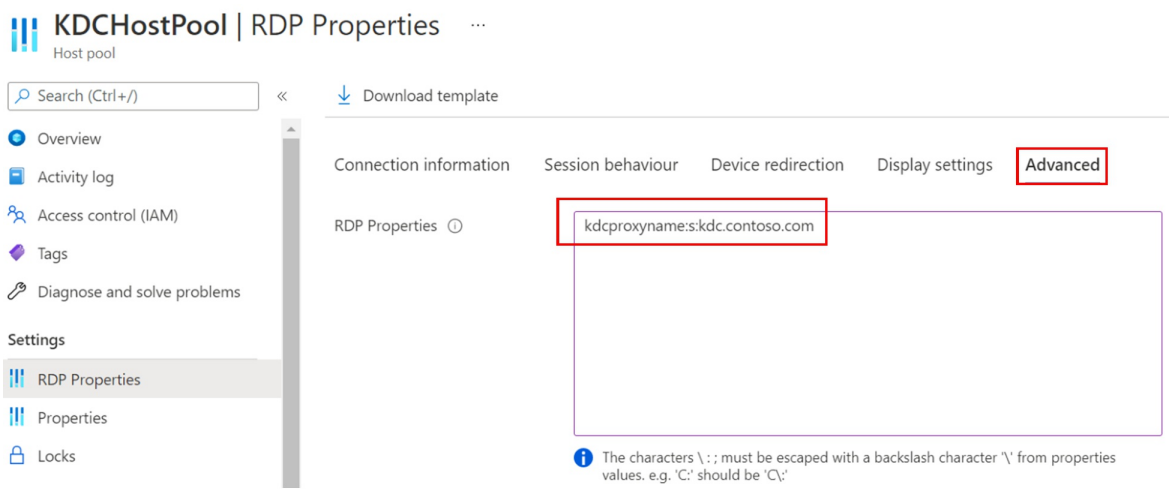
Once you've made sure you meet these requirements, you're ready to get started.

How to configure the KDC proxy

To configure the KDC proxy:

1. Sign in to the Azure portal as an administrator.
2. Go to the Azure Virtual Desktop page.
3. Select the host pool you want to enable the KDC proxy for, then select **RDP Properties**.
4. Select the **Advanced** tab, then enter a value in the following format without spaces:

```
kdcproxyname:s:<fqdn>
```



5. Select **Save**.
6. The selected host pool should now begin to issue RDP connection files that include the kdcproxyname value you entered in step 4.

Next steps

To learn how to manage the Remote Desktop Services side of the KDC proxy and assign the RD Gateway role, see Deploy the RD Gateway role.

If you're interested in scaling your KDC proxy servers, learn how to set up high availability for KDC proxy at Add high availability to the RD Web and Gateway web front.

Start Virtual Machine on Connect

12/6/2021 • 5 minutes to read • Edit Online

The Start Virtual Machine (VM) on Connect feature lets you save costs by allowing end users to turn on their VMs only when they need them. You can then turn off VMs when they're not needed.

> **NOTE**
> Azure Virtual Desktop (classic) doesn't support this feature.

Requirements and limitations

You can enable the start VM on Connect feature for personal or pooled host pools using PowerShell and the Azure portal.

The following Remote Desktop clients support the Start VM on Connect feature:

- The web client
- The Windows client (version 1.2.2061 or later)
- The Android client (version 10.0.10 or later)
- The macOS client (version 10.6.4 or later)
- The iOS client (version 10.2.5 or later)
- The Microsoft Store client (version 10.2.2005.0 or later)
- The thin clients listed in Thin client support

Create a custom role for Start VM on Connect

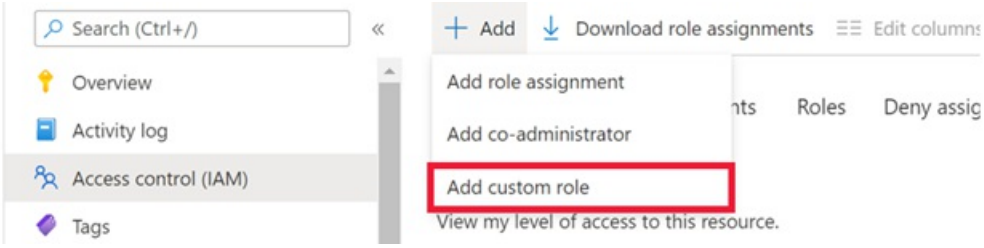
Before you can configure the Start VM on Connect feature, you'll need to assign your VM a custom RBAC (role-based access control) role. This role will let Azure Virtual Desktop manage the VMs in your subscription. You can also use this role to turn on VMs, check their status, and report diagnostic info. If you want to know more about what each role does, take a look at Azure custom roles.

> **NOTE**
> If your VMs and host pool are in different subscriptions, the RBAC role needs to be assigned to the subscription that the VMs are in.

Use the Azure portal

To use the Azure portal to assign a custom role for Start VM on Connect:

1. Open the Azure portal and go to **Subscriptions**.
2. Select the subscription that your VMs are in.
3. Go to **Access control (IAM)** and select **Add a custom role**.



4. Next, name the custom role and add a description. We recommend you name it "start VM on connect."
5. On the **Permissions** tab, add one of the two following sets of permissions to the subscription you're assigning the role to:

   - Microsoft.Compute/virtualMachines/start/action
   - Microsoft.Compute/virtualMachines/read
   - Microsoft.Compute/virtualMachines/instanceView/read

   You can also use these permissions instead:

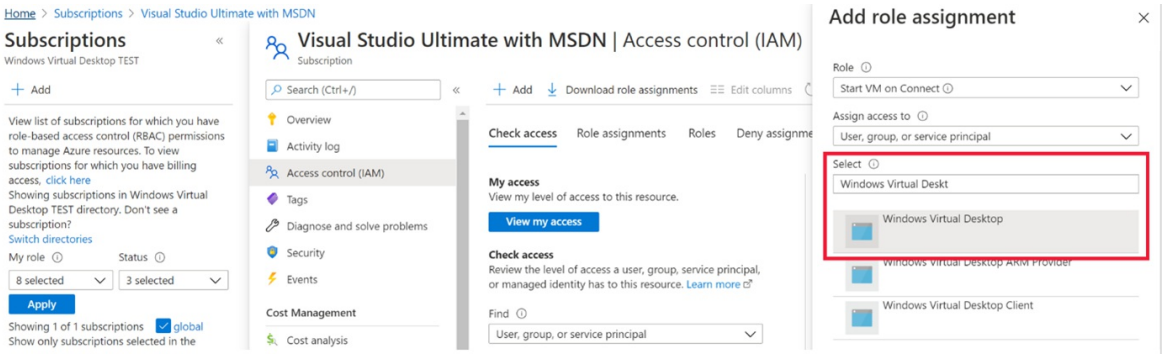
   - Microsoft.Compute/virtualMachines/start/action
   - Microsoft.Compute/virtualMachines/*/read
6. When you're finished, select **Ok**.

After that, you'll need to assign the role to grant access to Azure Virtual Desktop.

To assign the custom role:

1. In the **Access control (IAM) tab**, select **Add role assignments**.
2. Select the role you just created.
3. In the search bar, enter and select **Windows Virtual Desktop** (this will soon be updated to "Azure Virtual Desktop").

> **NOTE**
> You might see two apps if you have deployed Azure Virtual Desktop (classic). Assign the role to both apps you see.



Create a custom role with a JSON file template

If you're using a JSON file to create the custom role, the following example shows a basic template you can use. Make sure you replace the subscription ID value in *AssignableScopes* with the subscription ID you want to assign the role to.

```
{
  "Name": "Start VM on connect (Custom)",
  "IsCustom": true,
  "Description": "Start VM on connect with AVD (Custom)",
  "Actions": [
    "Microsoft.Compute/virtualMachines/start/action",
    "Microsoft.Compute/virtualMachines/*/read"
  ],
  "NotActions": [],
  "DataActions": [],
  "NotDataActions": [],
  "AssignableScopes": [
    "/subscriptions/00000000-0000-0000-0000-000000000000"
  ]
}
```

To use the JSON template, save the JSON file, add the relevant subscription information to *Assignable Scopes*, then run the following cmdlet in PowerShell:

```
New-AzRoleDefinition -InputFile "C:\temp\filename"
```

To learn more about creating custom roles, see Create or update Azure custom roles using Azure PowerShell.

Configure the Start VM on Connect feature

Now that you've assigned your subscription the role, it's time to configure the Start VM on Connect feature!

Deployment considerations

Start VM on Connect is a host pool setting. If you only want a select group of users to use this feature, make sure you only assign the required role to the users you want to add.

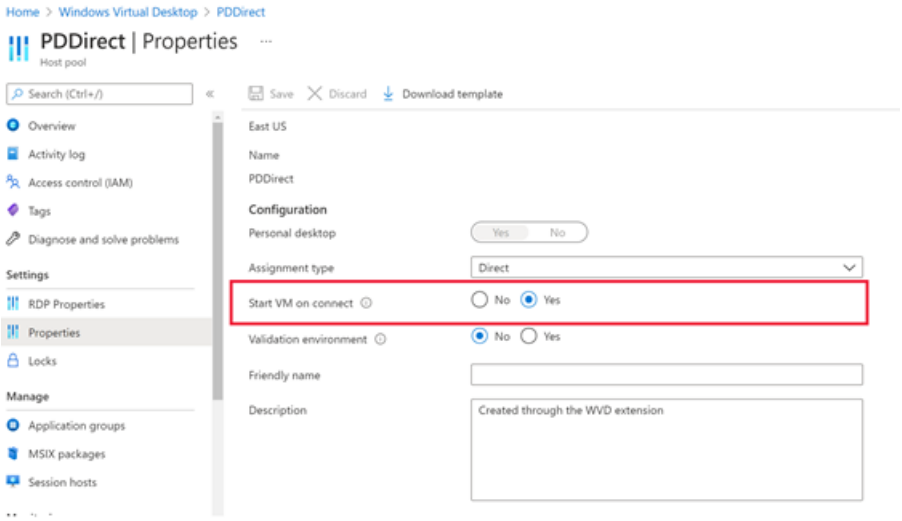
For personal desktops, the feature will only turn on an existing VM that the service has already assigned or will assign to a user. In a pooled host pool scenario, the service will only turn on a VM when none are turned on. The feature will only turn on additional VMs when the first VM reaches the session limit.

> **IMPORTANT**
> You can only configure this feature in existing host pools. This feature isn't available when you create a new host pool.

Use the Azure portal

To use the Azure portal to configure Start VM on Connect:

1. Open your browser and go to the Azure portal.
2. In the Azure portal, go to **Azure Virtual Desktop**.
3. Select **Host pools**, then go to the host pool where you want to enable the setting.
4. In the host pool, select **Properties**. Under **Start VM on connect**, select **Yes**, then select **Save** to instantly apply the setting.



Use PowerShell

To configure this setting with PowerShell, you need to make sure you have the names of the resource group and host pools you want to configure. You'll also need to install the Azure PowerShell module (version 2.1.0 or later).

To configure Start VM on Connect using PowerShell:

1. Open a PowerShell command window.
2. Run the following cmdlet to enable Start VM on Connect:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -
StartVMOnConnect:$true
```

3. Run the following cmdlet to disable Start VM on Connect:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -
StartVMOnConnect:$false
```

User experience

In typical sessions, the time it takes for a user to connect to a deallocated VM increases because the VM needs time to turn on again, much like turning on a physical computer. The Remote Desktop client has an indicator that lets the user know the PC is being powered on while they're connecting.

Troubleshooting

If the feature runs into any issues, we recommend you use the Azure Virtual Desktop diagnostics feature to check for problems. If you receive an error message, make sure to pay close attention to the message content and copy down the error name somewhere for reference.

You can also use Azure Monitor for Azure Virtual Desktop to get suggestions for how to resolve issues.

If the VM doesn't turn on, you'll need to check the health of the VM you tried to turn on before you do anything else.

Next steps

If you run into any issues that the troubleshooting documentation or the diagnostics feature couldn't solve, check out the Start VM on Connect FAQ.

Start VM on Connect FAQ

12/6/2021 • 2 minutes to read • Edit Online

This article covers frequently asked questions about the Start Virtual Machine (VM) on Connect feature for Azure Virtual Desktop host pools.

Are VMs automatically deallocated when a user stops using them?

No. You'll need to configure additional policies to sign users out of their sessions and run Azure automation scripts to deallocate VMs.

To configure the deallocation policy:

1. Connect remotely to the VM that you want to set the policy for.
2. Open the **Group Policy Editor**, then go to **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Session Time Limits**.
3. Find the policy that says **Set time limit for disconnected sessions**, then change its value to **Enabled**.
4. After you've enabled the policy, select **End a disconnected session**.

> **NOTE**
> Make sure to set the time limit for the "End a disconnected session" policy to a value greater than five minutes. A low time limit can cause users' sessions to end if their network loses connection for too long, resulting in lost work.

Signing users out won't deallocate their VMs. To learn how to deallocate VMs, see Start or stop VMs during off hours for personal host pools and Scale session hosts using Azure Automation for pooled host pools.

Can users turn off the VM from their clients?

Yes. Users can shut down the VM by using the Start menu within their session, just like they would with a physical machine. However, shutting down the VM won't deallocate the VM. To learn how to deallocate VMs, see Start or stop VMs during off hours for personal host pools and Scale session hosts using Azure Automation for pooled host pools.

Next steps

To learn how to configure Start VM on Connect, see Start virtual machine on connect.

If you have more general questions about Azure Virtual Desktop, check out our general FAQ.

Use drain mode to isolate session hosts and apply patches

12/6/2021 • 2 minutes to read • Edit Online

Drain mode isolates a session host when you want to apply patches and do maintenance without disrupting user sessions. When isolated, the session host won't accept new user sessions. Any new connections will be redirected to the next available session host. Existing connections in the session host will keep working until the user signs out or the administrator ends the session. When the session host is in drain mode, admins can also remotely connect to the server without going through the Azure Virtual Desktop service. You can apply this setting to both pooled and personal desktops.

Set drain mode using the Azure portal

To turn on drain mode in the Azure portal:

1. Open the Azure portal and go to the host pool you want to isolate.
2. In the navigation menu, select **Session hosts**.
3. Next, select the hosts you want to turn on drain mode for, then select **Turn drain mode on**.
4. To turn off drain mode, select the host pools that have drain mode turned on, then select **Turn drain mode off**.

Set drain mode using PowerShell

You can set drain mode in PowerShell with the *AllowNewSessions* parameter, which is part of the Update-AzWvdSessionhost command.

Run this cmdlet to enable drain mode:

```
Update-AzWvdSessionHost -ResourceGroupName <resourceGroupName> -HostPoolName <hostpoolname> -Name <hostname>
-AllowNewSession:$False
```

Run this cmdlet to disable drain mode:

```
Update-AzWvdSessionHost -ResourceGroupName <resourceGroupName> -HostPoolName <hostpoolname> -Name <hostname>
-AllowNewSession:$True
```

IMPORTANT

You'll need to run this command for every session host you're applying the setting to.

Next steps

If you want to learn more about the Azure portal for Azure Virtual Desktop, check out our tutorials. If you're already familiar with the basics, check out some of the other features you can use with the Azure portal, such as MSIX app attach and Azure Advisor.

If you're using the PowerShell method and want to see what else the module can do, check out Set up the

PowerShell module for Azure Virtual Desktop and our PowerShell reference.

Set up Azure Virtual Desktop for Azure Stack HCI (preview)

12/6/2021 • 7 minutes to read • Edit Online

> **IMPORTANT**
>
> Azure Virtual Desktop for Azure Stack HCI is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

With Azure Virtual Desktop for Azure Stack HCI (preview), you can use Azure Virtual Desktop session hosts in your on-premises Azure Stack HCI infrastructure. For more information, see Azure Virtual Desktop for Azure Stack HCI (preview).

Requirements

In order to use Azure Virtual Desktop for Azure Stack HCI, you'll need the following things:

- An Azure Stack HCI cluster registered with Azure.
- An Azure subscription for Azure Virtual Desktop session host pool creation with all required admin permissions.
- An on-premises Active Directory (AD) synced with Azure Active Directory.
- A stable connection to Azure from your on-premises network.
- Access from your on-premises network to all the required URLs listed in Azure Virtual Desktop's required URL list for virtual machines.

Configure Azure Virtual Desktop for Azure Stack HCI

To set up Azure Virtual Desktop for Azure Stack HCI:

1. Create a new host pool with no virtual machines by following the instructions in Begin the host pool setup process. At the end of that section, come back to this article and start on step 2.
2. Configure the newly created host pool to be a validation host pool by following the steps in Define your host pool as a validation host pool to enable the Validation environment property.
3. Follow the instructions in Workspace information to create a workspace for yourself.
4. Deploy a new virtual machine on your Azure Stack HCI infrastructure by following the instructions in Create a new VM. Deploy a VM with a supported OS and join it to a domain.

   > **NOTE**
   >
   > Install the Remote Desktop Session Host (RDSH) role if the VM is running a Windows Server OS.

5. Enable Azure to manage the new virtual machine through Azure Arc by installing the Connected Machine agent to it. Follow the directions in Connect hybrid machines with Azure Arc-enabled servers to install the

Windows agent to the virtual machine.

6. Add the virtual machine to the Azure Virtual Desktop host pool you created earlier by installing the Azure Virtual Desktop Agent. After that, follow the instructions in Register the VMs to the Azure Virtual Desktop host pool to register the VM to the Azure Virtual Desktop service.
7. Follow the directions in Create app groups and manage user assignments to create an app group for testing and assign user access to it.
8. Go to the web client and grant your users access to the new deployment.

Optional configurations

Now that you've set up Azure Virtual Desktop for Azure Stack HCI, here are a few extra things you can do depending on your deployment's needs.

Create a profile container using a file share on Azure Stack HCI

To create a profile container using a file share:

1. Deploy a file share on a single or clustered Windows Server VM deployment. The Windows Server VMs with file server role can also be colocated on the same cluster where the session host VMs are deployed.
2. Connect to the virtual machine with the credentials you provided when creating the virtual machine.
3. On the virtual machine, launch **Control Panel** and select **System**.
4. Select Computer name, select **Change settings**, and then select **Change...**.
5. Select **Domain**, then enter the Active Directory domain on the virtual network.
6. Authenticate with a domain account that has privileges to domain-join machines.
7. Follow the directions in Prepare the VM to act as a file share to prepare your VM for deployment.
8. Follow the directions in Configure the FSLogix profile container to configure your profile container for use.

Download supported OS images from Azure Marketplace

You can run any OS images that both Azure Virtual Desktop and Azure Stack HCI support on your deployment. To learn which OSes Azure Virtual Desktop supports, see Supported VM OS images.

You have two options to download an image:

- Deploy a VM with your preferred OS image, then follow the instructions in Download a Windows VHD from Azure.
- Download a Windows Virtual Hard Disk (VHD) from Azure without deploying a VM.

Downloading a Windows VHD without deploying a VM has several extra steps. To download a VHD from Azure without deploying a VM, you'll need to complete the instructions in the following sections in order.

Requirements to download a VHD without a VM

Before you begin, make sure you're connected to Azure and are running Azure Cloud Shell in either a command prompt or in the bash environment. You can also run CLI reference commands on the Azure command-line interface (CLI).

If you're using a local installation, run the az login command to sign into Azure.

After that, follow any other prompts you see to finish signing in. For additional sign-in options, see Sign in with the Azure CLI.

If this is your first time using Azure CLI, install any required extensions by following the instructions in Use extensions with the Azure CLI.

Finally, run the az version command to make sure your client is up to date. If it's out of date, run the az upgrade command to upgrade to the latest version.

Search Azure Marketplace for Azure Virtual Desktop images

You can find the image you're looking for by using the **Search** function in Azure Marketplace in the Azure portal. To find images specifically for Azure Virtual Desktop, you can run one of the following example queries.

If you're looking for Windows 10 multi-session, you can run a search with this criteria:

```
az vm image list --all --publisher "microsoftwindowsdesktop" --offer "windows-10" --sku "21h1-evd-g2"
```

This command should return the following URN:

```
MicrosoftWindowsDesktop:Windows-10:21h1-evd-g2:latest
```

If you're looking for Windows Server 2019 datacenter, you can run the following criteria in your Azure CLI:

```
az vm image list --all --publisher "microsoftwindowsserver" --offer "WindowsServer" --sku "2019-Datacenter-
gen2"
```

This command should return the following URN:

```
MicrosoftWindowsServer:windowsserver-gen2preview:2019-datacenter-gen2:latest
```

> **IMPORTANT**
> Make sure to only use generation 2 ("gen2") images. Azure Virtual Desktop for Azure Stack HCI doesn't support creating a VM with a first-generation ("gen1") image. Avoid SKUs with a "-g1" suffix.

Create a new Azure managed disk from the image

Next, you'll need to create an Azure managed disk from the image you downloaded from the Azure Marketplace.

To create an Azure managed disk:

1. Run the following commands in an Azure command-line prompt to set the parameters of your managed disk. Make sure to replace the items in brackets with the values relevant to your scenario.

```
$urn = <URN of the Marketplace image> #Example: “MicrosoftWindowsServer:WindowsServer:2019-
Datacenter:Latest”
$diskName = <disk name> #Name for new disk to be created
$diskRG = <resource group> #Resource group that contains the new disk
```

2. Run these commands to create the disk and generate a Serial Attached SCSI (SAS) access URL.

```
az disk create -g $diskRG -n $diskName --image-reference $urn
$sas = az disk grant-access --duration-in-seconds 36000 --access-level Read --name $diskName --resource-
group $diskRG
$diskAccessSAS = ($sas | ConvertFrom-Json)[0].accessSas
```

Export a VHD from the managed disk to Azure Stack HCI cluster

After that, you'll need to export the VHD you created from the managed disk to your Azure Stack HCI cluster, which will let you create new VMs. You can use the following method in a regular web browser or Storage Explorer.

To export the VHD:

1. Open a browser and go to the SAS URL of the managed disk you generated in Create a new Azure managed disk from the image. You can download the VHD image for the image you downloaded at the Azure Marketplace at this URL.
2. Download the VHD image. The downloading process may take several minutes, so be patient. Make sure the image has fully downloaded before going to the next section.

> **NOTE**
>
> If you're running azcopy, you may need to skip the md5check by running this command:
>
> ```
> azcopy copy “$sas" "destination_path_on_cluster" --check-md5 NoCheck
> ```

Clean up the managed disk

When you're done with your VHD, you'll need to free up space by deleting the managed disk.

To delete the managed disk you created, run these commands:

```
az disk revoke-access --name $diskName --resource-group $diskRG
az disk delete --name $diskName --resource-group $diskRG --yes
```

This command may take a few minutes to finish, so be patient.

> **NOTE**
>
> Optionally, you can also convert the download VHD to a dynamic VHDx by running this command:
>
> ```
> Convert-VHD -Path " destination_path_on_cluster\file_name.vhd" -DestinationPath "
> destination_path_on_cluster\file_name.vhdx" -VHDType Dynamic
> ```

Next steps

If you need to refresh your memory about the basics or pricing information, go to Azure Virtual Desktop for Azure Stack HCI.

If you have additional questions, check out our FAQ.

Microsoft Endpoint Manager and Intune for Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

We recommend using Microsoft Endpoint Manager to manage your Azure Virtual Desktop environment after deployment. Microsoft Endpoint Manager is a unified management platform that includes Microsoft Endpoint Configuration Manager and Microsoft Intune.

> **NOTE**
> Managing Azure Virtual Desktop session hosts using Microsoft Endpoint Manager is currently only supported in the Azure Public cloud.

Microsoft Endpoint Configuration Manager

Microsoft Endpoint Configuration Manager versions 1906 and later can manage your Azure Virtual Desktop devices. For more information, see Supported OS versions for clients and devices for Configuration Manager.

Microsoft Intune

Intune supports Windows 10 Enterprise virtual machines (VMs) for Azure Virtual Desktop. For more information about support, see Using Windows 10 Enterprise with Intune.

Intune support for Windows 10 Enterprise multi-session VMs on Azure Virtual Desktop is currently in public preview. To see what the public preview version currently supports, check out Using Windows 10 Enterprise multi-session with Intune.

Licensing

Microsoft Endpoint Configuration Manager and Microsoft Intune licenses are included with most Microsoft 365 subscriptions.

Learn more about licensing requirements at the following resources:

- Frequently asked questions for Configuration Manager branches and licensing
- Microsoft Intune licensing

Built-in roles for Azure Virtual Desktop

12/6/2021 • 3 minutes to read • Edit Online

Azure Virtual Desktop uses Azure role-based access controls (RBAC) to assign roles to users and admins. These roles give admins permission to carry out certain tasks. To learn more about built-in roles for Azure RBAC, see Azure built-in roles.

The standard built-in roles for Azure are Owner, Contributor, and Reader. However, Azure Virtual Desktop has additional roles that let you separate management roles for host pools, app groups, and workspaces. This separation lets you have more granular control over administrative tasks. These roles are named in compliance with Azure's standard roles and least-privilege methodology.

Azure Virtual Desktop doesn't have a specific Owner role. However, you can use a standard Owner role for the service objects.

Desktop Virtualization Contributor

The Desktop Virtualization Contributor role lets you manage all aspects of the deployment. However, it doesn't grant you access to compute resources. You'll also need the User Access Administrator role to publish app groups to users or user groups.

- Microsoft.DesktopVirtualization/*
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/*
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Reader

The Desktop Virtualization Reader role lets you view everything in the deployment but doesn't let you make any changes.

- Microsoft.DesktopVirtualization/*/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Host Pool Contributor

The Host Pool Contributor role lets you manage all aspects of host pools, including access to resources. You'll need an extra contributor role, Virtual Machine Contributor, to create virtual machines. You will need AppGroup and Workspace contributor roles to create host pool using the portal or you can use Desktop Virtualization Contributor role.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/hostpools/*

- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/*
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Host Pool Reader

The Host Pool Reader role lets you view everything in the host pool, but won't allow you to make any changes.

- Microsoft.DesktopVirtualization/hostpools/*/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Application Group Contributor

The Application Group Contributor role lets you manage all aspects of app groups. If you want to publish app groups to users or user groups, you'll need the User Access Administrator role.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/applicationgroups/*
- Microsoft.DesktopVirtualization/hostpools/read
- Microsoft.DesktopVirtualization/hostpools/sessionhosts/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/*
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Application Group Reader

The Application Group Reader role lets you view everything in the app group and will not allow you to make any changes.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/applicationgroups/*/read
- Microsoft.DesktopVirtualization/applicationgroups/read
- Microsoft.DesktopVirtualization/hostpools/read
- Microsoft.DesktopVirtualization/hostpools/sessionhosts/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Workspace Contributor

The Workspace Contributor role lets you manage all aspects of workspaces. To get information on applications added to the app groups, you'll also need to be assigned the Application Group Reader role.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/workspaces/*
- Microsoft.DesktopVirtualization/applicationgroups/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/*
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Workspace Reader

The Workspace Reader role lets you view everything in the workspace, but won't allow you to make any changes.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/workspaces/read
- Microsoft.DesktopVirtualization/applicationgroups/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization User Session Operator

The User Session Operator role lets you send messages, disconnect sessions, and use the "logoff" function to sign sessions out of the session host. However, this role doesn't let you perform session host management like removing session host, changing drain mode, and so on. This role can see assignments, but can't modify admins. We recommend you assign this role to specific host pools. If you give this permission at a resource group level, the admin will have read permission on all host pools under a resource group.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/hostpools/read
- Microsoft.DesktopVirtualization/hostpools/sessionhosts/read
- Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/*
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Desktop Virtualization Session Host Operator

The Session Host Operator role lets you view and remove session hosts, as well as change drain mode. They can't add session hosts using the Azure portal because they don't have write permission for host pool objects. If the registration token is valid (generated and not expired), you can use this role to add session hosts to the host pool outside of Azure portal if the admin has compute permissions through the Virtual Machine Contributor

role.

The following list describes which permissions this role can access:

- Microsoft.DesktopVirtualization/hostpools/read
- Microsoft.DesktopVirtualization/hostpools/sessionhosts/*
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Resources/deployments/read
- Microsoft.Authorization/*/read
- Microsoft.Insights/alertRules/*
- Microsoft.Support/*

Supported authentication methods

12/6/2021 • 3 minutes to read • Edit Online

In this article, we'll give you a brief overview of what kinds of authentication you can use in Azure Virtual Desktop.

Identities

Azure Virtual desktop supports different types of identities depending on which configuration you choose. This section explains which identities you can use for each configuration.

On-premise identity

Since users must be discoverable through Azure Active Directory (Azure AD) to access the Azure Virtual Desktop, user identities that exist only in Active Directory Domain Services (AD DS) are not supported. This includes standalone Active Directory deployments with Active Directory Federation Services (AD FS).

Hybrid identity

Azure Virtual Desktop supports hybrid identities through Azure AD, including those federated using AD FS. You can manage these user identities in AD DS and sync them to Azure AD using Azure AD Connect. You can also use Azure AD to manage these identities and sync them to Azure AD Directory Services (Azure AD DS).

When accessing Azure Virtual Desktop using hybrid identities, sometimes the User Principal Name (UPN) or Security Identifier (SID) for the user in Active Directory (AD) and Azure AD don't match. For example, the AD account user@contoso.local may correspond to user@contoso.com in Azure AD. Azure Virtual Desktop only supports this type of configuration if either the UPN or SID for both your AD and Azure AD accounts match. SID refers to the user object property "ObjectSID" in AD and "OnPremisesSecurityIdentifier" in Azure AD.

Cloud-only identity

Azure Virtual Desktop supports cloud-only identities when using Azure AD-joined VMs.

External identity

Azure Virtual Desktop currently doesn't support external identities.

Service authentication

To access Azure Virtual Desktop resources, you must first authenticate to the service by signing in to an Azure AD account. Authentication happens when subscribing to a workspace to retrieve your resources or every time you connect to apps or desktops. You can use third-party identity providers as long as they federate with Azure AD.

Multifactor authentication

Follow the instructions in Set up multifactor authentication in Azure Virtual Desktop to learn how to enable multifactor authentication (MFA) for your deployment. That article will also tell you how to configure how often your users are prompted to enter their credentials. When deploying Azure AD-joined VMs, follow the configuration guide in Enabling MFA for Azure AD-joined VMs.

Smart card authentication

To use a smart card to authenticate to Azure AD, you must first configure AD FS for user certificate authentication.

Session host authentication

If you haven't already enabled single sign-on or saved your credentials locally, you'll also need to authenticate to the session host. These are the sign-in methods for the session host that the Azure Virtual Desktop clients currently support:

- Windows Desktop client
  - Username and password
  - Smartcard
  - Windows Hello for Business certificate trust
  - Windows Hello for Business key trust with certificates
- Windows Store client
  - Username and password
- Web client
  - Username and password
- Android
  - Username and password
- iOS
  - Username and password
- macOS
  - Username and password

Azure Virtual Desktop supports both NT LAN Manager (NTLM) and Kerberos for session host authentication. Smart card and Windows Hello for Business can only use Kerberos to sign in. To use Kerberos, the client needs to get Kerberos security tickets from a Key Distribution Center (KDC) service running on a domain controller. To get tickets, the client needs a direct networking line-of-sight to the domain controller. You can get a line-of-sight by connecting directly within your corporate network, using a VPN connection or setting up a KDC Proxy server.

Single sign-on (SSO)

Azure Virtual Desktop supports SSO using Active Directory Federation Services (ADFS) for the Windows and web clients. SSO allows you to skip the session host authentication.

Otherwise, the only way to avoid being prompted for your credentials for the session host is to save them in the client. We recommend you only do this with secure devices to prevent other users from accessing your resources.

In-session authentication

Once you're connected to your remote app or desktop, you may be prompted for authentication inside the session. This section explains how to use credentials other than username and password in this scenario.

Smart cards

To use a smart card in your session, make sure you've installed the smart card drivers on the session host and enabled smart card redirection is enabled. Review the client comparison chart to make sure your client supports smart card redirection.

FIDO2 and Windows Hello for Business

Azure Virtual Desktop doesn't currently support in-session authentication with FIDO2 or Windows Hello for Business.

Next steps

- Curious about other ways to keep your deployment secure? Check out Security best practices.

- Having issues connecting to Azure AD-joined VMs? Troubleshoot connections to Azure AD-joined VMs.
- Want to use smart cards from outside your corporate network? Review how to setup a KDC Proxy server.

Azure Virtual Desktop environment

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Azure Virtual Desktop is a service that gives users easy and secure access to their virtualized desktops and RemoteApps. This topic will tell you a bit more about the general structure of the Azure Virtual Desktop environment.

Host pools

A host pool is a collection of Azure virtual machines that register to Azure Virtual Desktop as session hosts when you run the Azure Virtual Desktop agent. All session host virtual machines in a host pool should be sourced from the same image for a consistent user experience.

A host pool can be one of two types:

- Personal, where each session host is assigned to individual users.
- Pooled, where session hosts can accept connections from any user authorized to an app group within the host pool.

You can set additional properties on the host pool to change its load-balancing behavior, how many sessions each session host can take, and what the user can do to session hosts in the host pool while signed in to their Azure Virtual Desktop sessions. You control the resources published to users through app groups.

App groups

An app group is a logical grouping of applications installed on session hosts in the host pool. An app group can be one of two types:

- RemoteApp, where users access the RemoteApps you individually select and publish to the app group
- Desktop, where users access the full desktop

By default, a desktop app group (named "Desktop Application Group") is automatically created whenever you create a host pool. You can remove this app group at any time. However, you can't create another desktop app group in the host pool while a desktop app group exists. To publish RemoteApps, you must create a RemoteApp app group. You can create multiple RemoteApp app groups to accommodate different worker scenarios. Different RemoteApp app groups can also contain overlapping RemoteApps.

To publish resources to users, you must assign them to app groups. When assigning users to app groups, consider the following things:

- A user can be assigned to both a desktop app group and a RemoteApp app group in the same host pool. However, users can only launch one type of app group per session. Users can't launch both types of app groups at the same time in a single session.
- A user can be assigned to multiple app groups within the same host pool, and their feed will be an accumulation of both app groups.

Workspaces

A workspace is a logical grouping of application groups in Azure Virtual Desktop. Each Azure Virtual Desktop application group must be associated with a workspace for users to see the remote apps and desktops published to them.

End users

After you've assigned users to their app groups, they can connect to a Azure Virtual Desktop deployment with any of the Azure Virtual Desktop clients.

Next steps

Learn more about delegated access and how to assign roles to users at Delegated Access in Azure Virtual Desktop.

To learn how to set up your Azure Virtual Desktop host pool, see Create a host pool with the Azure portal.

To learn how to connect to Azure Virtual Desktop, see one of the following articles:

- Connect with Windows 10 or Windows 7
- Connect with a web browser
- Connect with the Android client
- Connect with the macOS client
- Connect with the iOS client

Get started with the Azure Virtual Desktop Agent

12/6/2021 • 4 minutes to read • Edit Online

In the Azure Virtual Desktop Service framework, there are three main components: the Remote Desktop client, the service, and the virtual machines. These virtual machines live in the customer subscription where the Azure Virtual Desktop agent and agent bootloader are installed. The agent acts as the intermediate communicator between the service and the virtual machines, enabling connectivity. Therefore, if you're experiencing any issues with the agent installation, update, or configuration, your virtual machines won't be able to connect to the service. The agent bootloader is the executable that loads the agent.

This article will give you a brief overview of the agent installation and update processes.

> **NOTE**
> This documentation is not for the FSLogix agent or the Remote Desktop Client agent.

Initial installation process

The Azure Virtual Desktop agent is initially installed in one of two ways. If you provision virtual machines (VMs) in the Azure portal and Azure Marketplace, the agent and agent bootloader are automatically installed. If you provision VMs using PowerShell, you must manually download the agent and agent bootloader .msi files when creating a Azure Virtual Desktop host pool with PowerShell. Once the agent is installed, it installs the Azure Virtual Desktop side-by-side stack and Geneva Monitoring agent. The side-by-side stack component is required for users to securely establish reverse server-to-client connections. The Geneva Monitoring agent monitors the health of the agent. All three of these components are essential for end-to-end user connectivity to function properly.

> **IMPORTANT**
> To successfully install the Azure Virtual Desktop agent, side-by-side stack, and Geneva Monitoring agent, you must unblock all the URLs listed in the Required URL list. Unblocking these URLs is required to use the Azure Virtual Desktop service.

Agent update process

The Azure Virtual Desktop service updates the agent whenever an update becomes available. Agent updates can include new functionality or fixes for previous issues. You must always have the latest stable version of the agent installed so your VMs don't lose connectivity or security. Once the initial version of the Azure Virtual Desktop agent is installed, the agent regularly queries the Azure Virtual Desktop service to determine if there's a newer version of the agent, stack, or monitoring component available. If a newer version of any of the components has already been deployed, the updated component is automatically installed by the flighting system.

New versions of the agent are deployed at regular intervals in five-day periods to all Azure subscriptions. These update periods are called "flights". It takes 24 hours for all VMs in a single broker region to receive the agent update in a flight. Because of this, when a flight happens, you may see VMs in your host pool receive the agent update at different times. Also, if the VMs are in different regions, they might update on different days in the five-day period. The flight will update all VM agents in all subscriptions by the end of the deployment period. The Azure Virtual Desktop flighting system enhances service reliability by ensuring the stability and quality of the agent update.

Other important things you should keep in mind:

- The agent update isn't connected to Azure Virtual Desktop infrastructure build updates. When the Azure Virtual Desktop infrastructure updates, that doesn't mean that the agent has updated along with it.
- Because VMs in your host pool may receive agent updates at different times, you'll need to be able to tell the difference between flighting issues and failed agent updates. If you go to the event logs for your VM at **Event Viewer** > **Windows Logs** > **Application** and see an event labeled "ID 3277," that means the Agent update didn't work. If you don't see that event, then the VM is in a different flight and will be updated later.
- When the Geneva Monitoring agent updates to the latest version, the old GenevaTask task is located and disabled before creating a new task for the new monitoring agent. The earlier version of the monitoring agent isn't deleted in case that the most recent version of the monitoring agent has a problem that requires reverting to the earlier version to fix. If the latest version has a problem, the old monitoring agent will be re-enabled to continue delivering monitoring data. All versions of the monitor that are earlier than the last one you installed before the update will be deleted from your VM.
- Your VM keeps three versions of the agent and of the side-by-side stack at a time. This allows for quick recovery if something goes wrong with the update. The earliest version of the agent or stack is removed from the VM whenever the agent or stack updates. If you delete these components prematurely and the agent or stack has a failure, the agent or stack won't be able to roll back to an earlier version, which will put your VM in an unavailable state.

The agent update normally lasts 2-3 minutes on a new VM and shouldn't cause your VM to lose connection or shut down. This update process applies to both Azure Virtual Desktop (classic) and the latest version of Azure Virtual Desktop with Azure Resource Manager.

Next steps

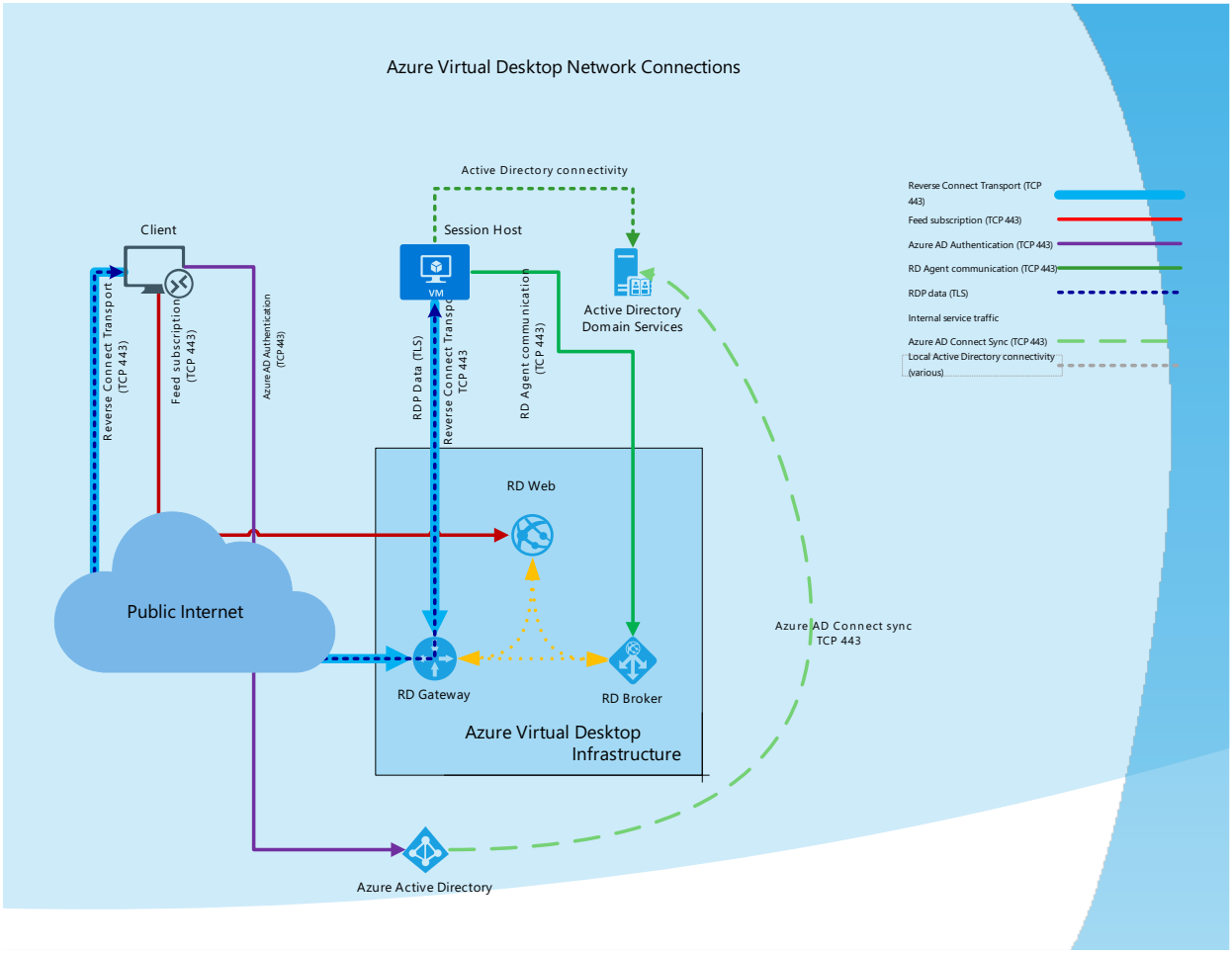
Now that you have a better understanding of the Azure Virtual Desktop agent, here are some resources that might help you:

- If you're experiencing agent or connectivity-related issues, check out the Azure Virtual Desktop Agent issues troubleshooting guide.

Understanding Azure Virtual Desktop network connectivity

12/6/2021 • 3 minutes to read • Edit Online

Azure Virtual Desktop provides the ability to host client sessions on the session hosts running on Azure. Microsoft manages portions of the services on the customer's behalf and provides secure endpoints for connecting clients and session hosts. The diagram below gives a high-level overview of the network connections used by Azure Virtual Desktop



Session connectivity

Azure Virtual Desktop uses Remote Desktop Protocol (RDP) to provide remote display and input capabilities over network connections. RDP was initially released with Windows NT 4.0 Terminal Server Edition and was continuously evolving with every Microsoft Windows and Windows Server release. From the beginning, RDP developed to be independent of its underlying transport stack, and today it supports multiple types of transport.

Reverse connect transport

Azure Virtual Desktop is using reverse connect transport for establishing the remote session and for carrying RDP traffic. Unlike the on-premises Remote Desktop Services deployments, reverse connect transport doesn't use a TCP listener to receive incoming RDP connections. Instead, it is using outbound connectivity to the Azure Virtual Desktop infrastructure over the HTTPS connection.

Session host communication channel

Upon startup of the Azure Virtual Desktop session host, the Remote Desktop Agent Loader service establishes the Azure Virtual Desktop broker's persistent communication channel. This communication channel is layered on top of a secure Transport Layer Security (TLS) connection and serves as a bus for service message exchange between session host and Azure Virtual Desktop infrastructure.

Client connection sequence

Client connection sequence described below:

1. Using supported Azure Virtual Desktop client user subscribes to the Azure Virtual Desktop Workspace
2. Azure Active Directory authenticates the user and returns the token used to enumerate resources available to a user
3. Client passes token to the Azure Virtual Desktop feed subscription service
4. Azure Virtual Desktop feed subscription service validates the token
5. Azure Virtual Desktop feed subscription service passes the list of available desktops and RemoteApps back to the client in the form of digitally signed connection configuration
6. Client stores the connection configuration for each available resource in a set of .rdp files
7. When a user selects the resource to connect, the client uses the associated .rdp file and establishes the secure TLS 1.2 connection to the closest Azure Virtual Desktop gateway instance and passes the connection information
8. Azure Virtual Desktop gateway validates the request and asks the Azure Virtual Desktop broker to orchestrate the connection
9. Azure Virtual Desktop broker identifies the session host and uses the previously established persistent communication channel to initialize the connection
10. Remote Desktop stack initiates the TLS 1.2 connection to the same Azure Virtual Desktop gateway instance as used by the client
11. After both client and session host connected to the gateway, the gateway starts relaying the raw data between both endpoints, this establishes the base reverse connect transport for the RDP
12. After the base transport is set, the client starts the RDP handshake

Connection security

TLS 1.2 is used for all connections initiated from the clients and session hosts to the Azure Virtual Desktop infrastructure components. Azure Virtual Desktop uses the same TLS 1.2 ciphers as Azure Front Door. It's important to make sure both client computers and session hosts can use these ciphers. For reverse connect transport, both client and session host connect to the Azure Virtual Desktop gateway. After establishing the TCP connection, the client or session host validates the Azure Virtual Desktop gateway's certificate. After establishing the base transport, RDP establishes a nested TLS connection between client and session host using the session host's certificates. By default, the certificate used for RDP encryption is self-generated by the OS during the deployment. If desired, customers may deploy centrally managed certificates issued by the enterprise certification authority. For more information about configuring certificates, see Windows Server documentation.

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see Understanding Remote Desktop Protocol (RDP) Bandwidth Requirements for Azure Virtual Desktop.
- To get started with Quality of Service (QoS) for Azure Virtual Desktop, see Implement Quality of Service (QoS) for Azure Virtual Desktop.

Azure Virtual Desktop RDP Shortpath for managed networks

12/6/2021 • 9 minutes to read • Edit Online

RDP Shortpath for managed networks is a feature of Azure Virtual Desktop that establishes a direct UDP-based transport between Remote Desktop Client and Session host. RDP uses this transport to deliver Remote Desktop and RemoteApp while offering better reliability and consistent latency.

Key benefits

- RDP Shortpath transport is based on top of highly efficient Universal Rate Control Protocol (URCP). URCP enhances UDP with active monitoring of the network conditions and provides fair and full link utilization. URCP operates at low delay and loss levels as needed by Remote Desktop. URCP achieves the best performance by dynamically learning network parameters and providing protocol with a rate control mechanism.
- RDP Shortpath establishes the direct connectivity between Remote Desktop client and Session Host. Direct connectivity reduces the dependency on the Azure Virtual Desktop gateways, improves the connection's reliability, and increases the bandwidth available for each user session.
- The removal of extra relay reduces the round-trip time, which improves user experience with latency-sensitive applications and input methods.
- RDP Shortpath brings support for configuring Quality of Service (QoS) priority for RDP connections through a Differentiated Services Code Point (DSCP) marks
- RDP Shortpath transport allows limiting outbound network traffic by specifying a throttle rate for each session.

Connection security

RDP Shortpath is extending RDP multi-transport capabilities. It doesn't replace reverse connect transport but complements it. All of the initial session brokering is managed through the Azure Virtual Desktop infrastructure.

Your deployment will only use the user-configured UDP port for incoming Shortpath traffic authenticated over reverse connect transport. The RDP Shortpath listener will ignore all connection attempts unless they match the reverse connect session.

RDP Shortpath uses a TLS connection between the client and the session host using the session host's certificates. By default, the certificate used for RDP encryption is self-generated by the OS during the deployment. If desired, customers may deploy centrally managed certificates issued by the enterprise certification authority. For more information about certificate configurations, see Windows Server documentation.

RDP Shortpath connection sequence

After establishing the reverse connect transport, the client and session host starts the RDP connection and negotiates the multi-transport capabilities.

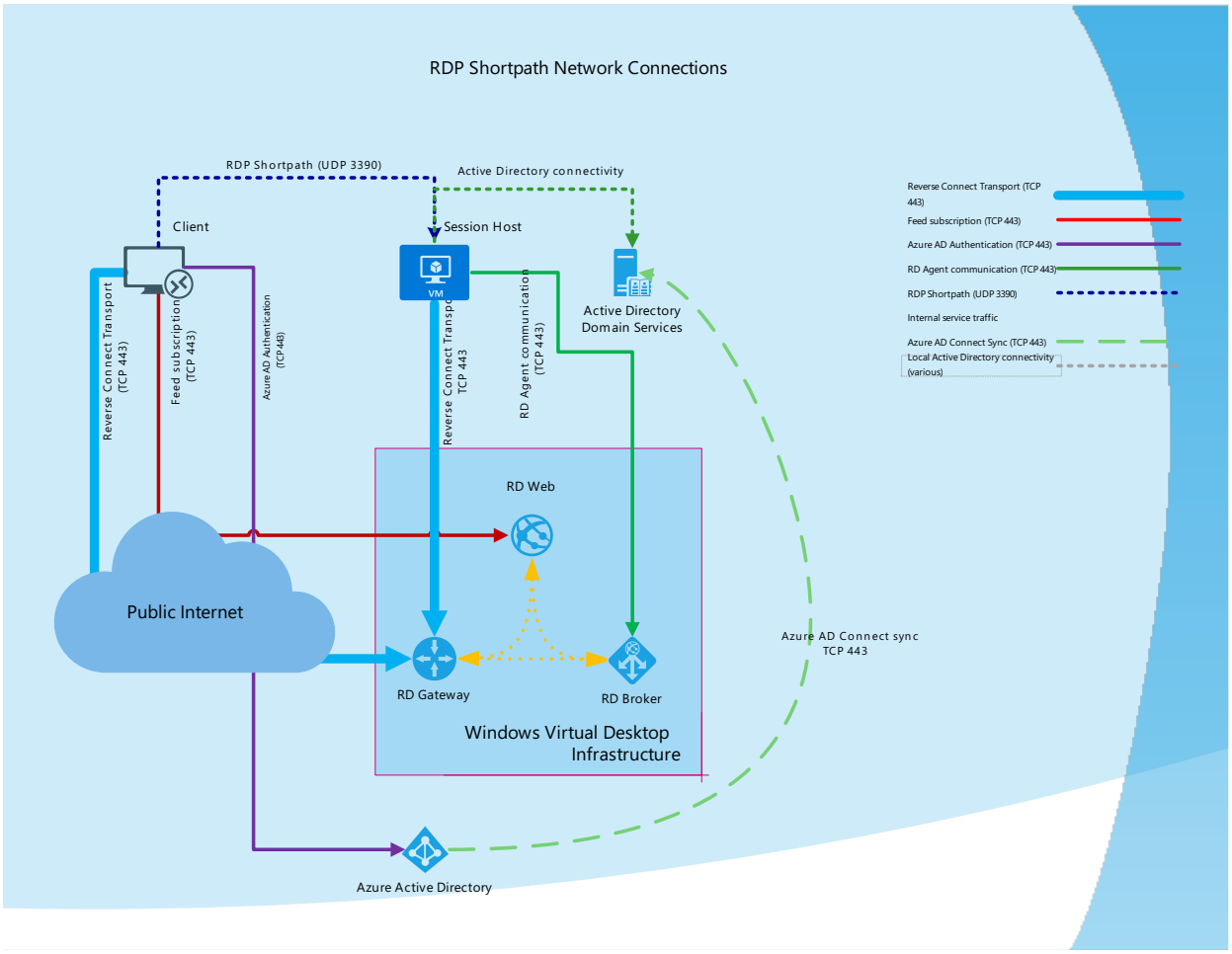
Here's how the session host negotiates multi-transport capabilities:

1. The session host sends the list of its private and public IPv4 and IPv6 addresses to the client.
2. The client starts the background thread to establish a parallel UDP-based transport directly to one of the

host's IP addresses.

3. While the client is probing the provided IP addresses, it continues the initial connection establishment over the reverse connect transport to ensure no delay in the user connection.
4. If the client has a direct line of sight, the client establishes a secure TLS connection with the session host.
5. After establishing the Shortpath transport, RDP moves all Dynamic Virtual Channels (DVCs), including remote graphics, input, and device redirection, to the new transport.
6. If a firewall or network topology prevents the client from establishing direct UDP connectivity, RDP continues with a reverse connect transport.

The diagram below gives a high-level overview of the RDP Shortpath network connection.



Requirements

To support RDP Shortpath, the Azure Virtual Desktop client needs a direct line of sight to the session host. You can get a direct line of sight by using one of these methods:

- Make sure the remote client machines must be running either Windows 10 or Windows 7 and have the Windows Desktop client installed. Currently, non-Windows clients aren't supported.
- Use ExpressRoute private peering
- Use a Site-to-Site virtual private network (VPN) (IPsec-based)
- Use a Point-to-Site VPN (IPsec-based)
- Use a public IP address assignment

If you're using other VPN types to connect to the Azure, we recommend using a User Datagram Protocol (UDP)-based VPN. While most Transmission Control Protocol (TCP)-based VPN solutions support nested UDP, they add inherited overhead of TCP congestion control, which slows down RDP performance.

Having a direct line of sight means that the client can connect directly to the session host without being blocked

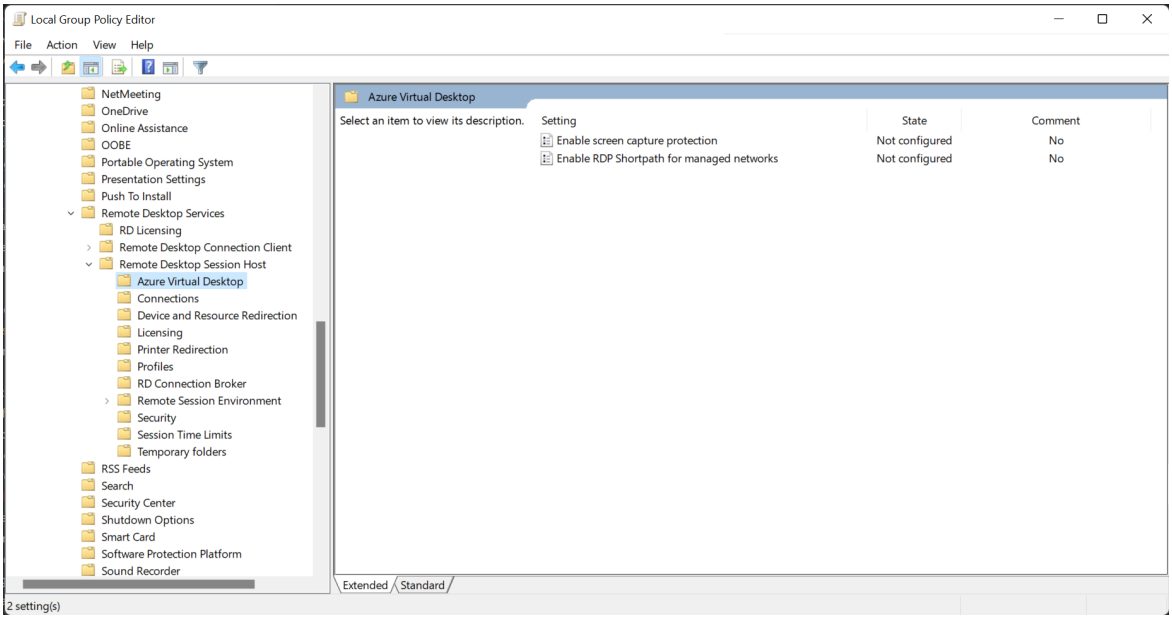
by firewalls.

Configure RDP Shortpath for managed networks

To enable RDP Shortpath for managed networks, you need to enable the RDP Shortpath listener on the session host. You can enable RDP Shortpath on any number of session hosts used in your environment. However, there's no requirement to enable RDP Shortpath on all hosts in your host pool.

To enable the RDP Shortpath listener:

1. First, install administrative templates that add rules and settings for Azure Virtual Desktop. Download the Azure Virtual Desktop policy templates file (AVDGPTemplate.cab) and extract the contents of the .cab file and .zip archive.
2. Copy the **terminalserver-avd.admx** file, then paste it into the **%windir%\PolicyDefinitions** folder.
3. Copy the **en-us\terminalserver-avd.adml** file, then paste it into the **%windir%\PolicyDefinitions\en-us** folder.
4. To confirm the files copied correctly, open the **Group Policy Editor** and go to **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Azure Virtual Desktop**.
5. You should see one or more Azure Virtual Desktop policies, as shown in the following screenshot



> **NOTE**
>
> You can also install administrative templates to the group policy Central Store in your Active Directory domain. For more information about Central Store for Group Policy Administrative Templates, see How to create and manage the Central Store for Group Policy Administrative Templates in Windows.

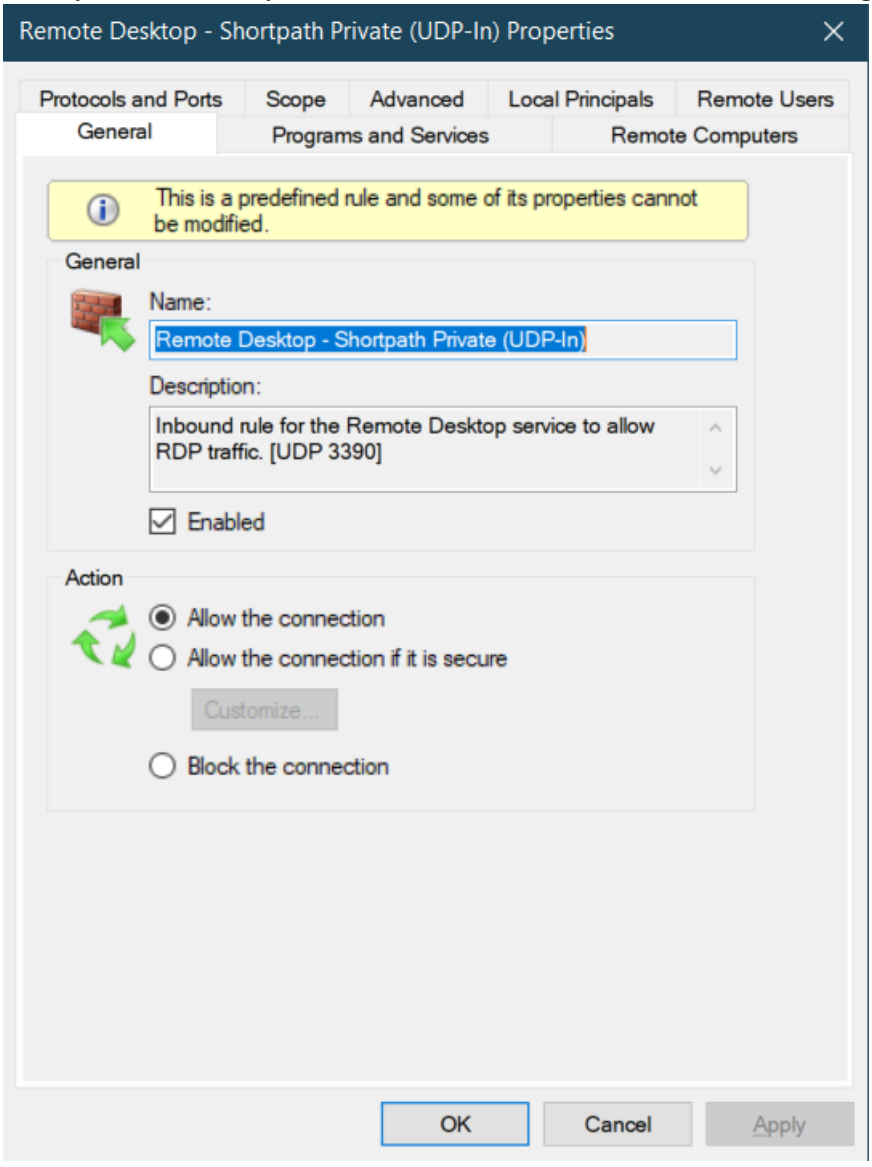
6. Open the **"Enable RDP Shortpath for managed networks"** policy and set it to **"Enabled"**. If you enable this policy setting, you can also configure the port number that the Azure Virtual Desktop session host will use to listen for incoming connections. The default port is 3390.
7. Restart your session host to apply the changes.

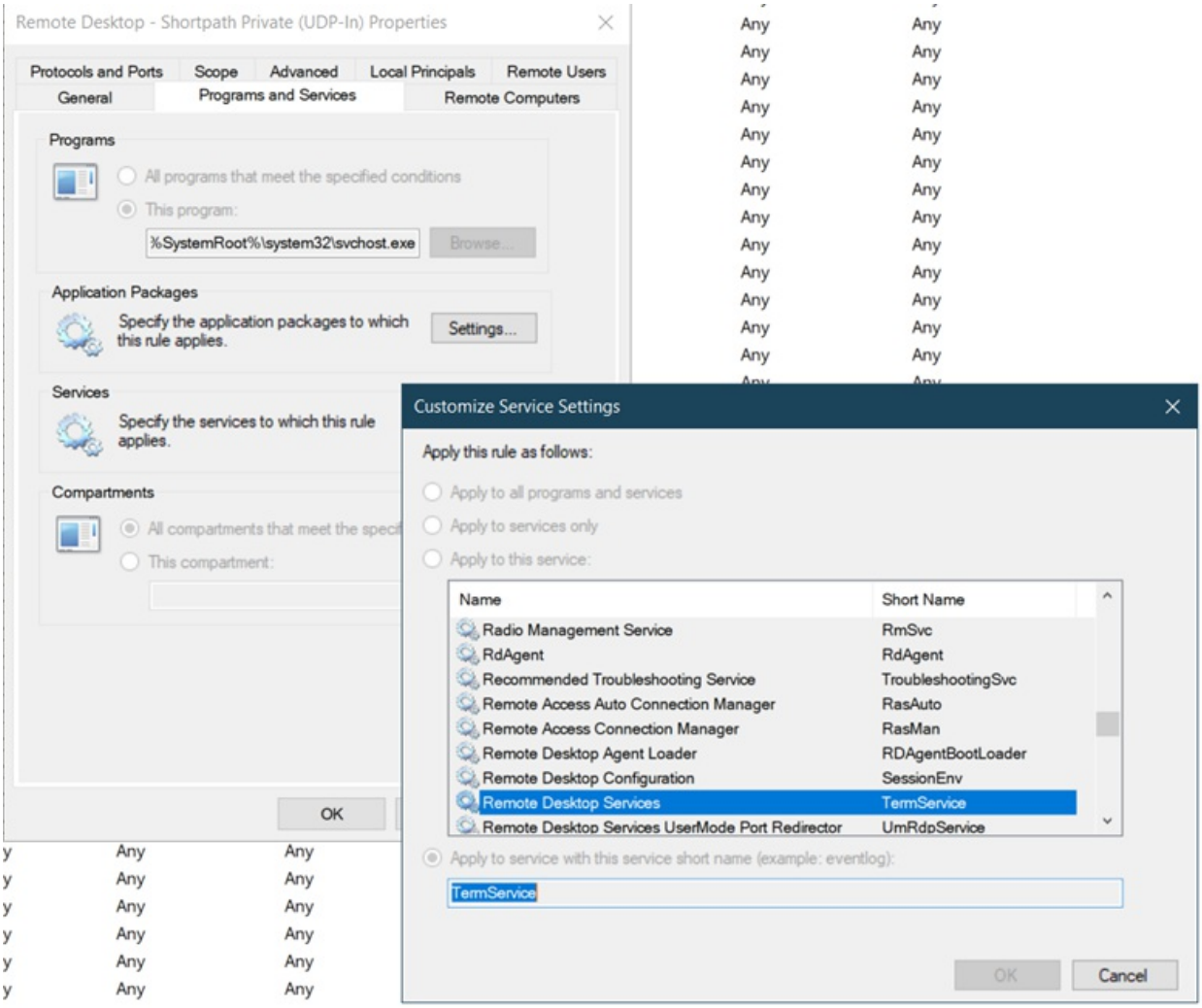
Configure Windows Defender Firewall with Advanced Security

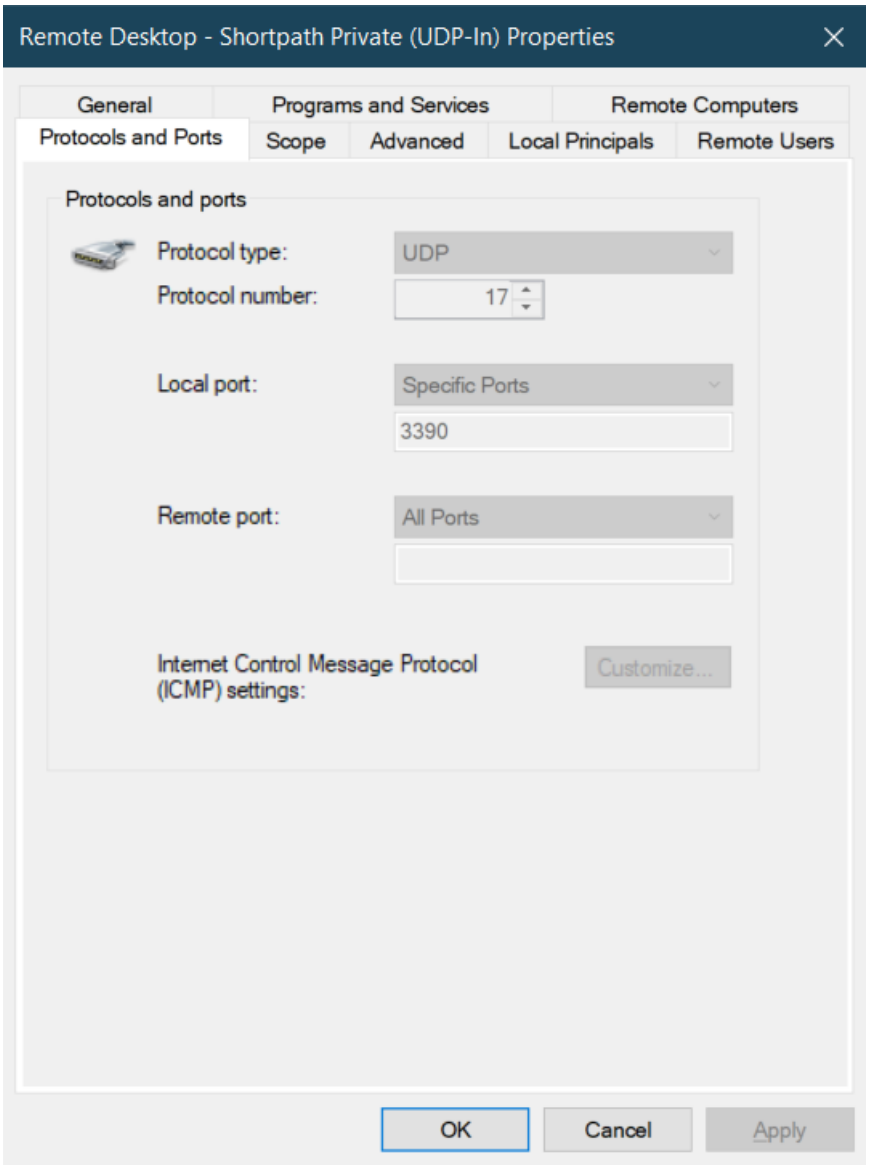
To allow inbound network traffic for RDP Shortpath, use the Windows Defender Firewall with Advanced Security node in the Group Policy Management MMC snap-in to create firewall rules.

1. Open the Group Policy Management Console to Windows Defender Firewall with Advanced Security.
2. In the navigation pane, select **Inbound Rules**.
3. Select **Action**, and then select **New rule**.
4. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**.
5. On the **Program** page, select **This program path**, and type "%SystemRoot%\system32\svchost.exe" then select **Next**.
6. On the **Protocol and Ports** page, select the UDP protocol type. In the **Local port**, select "Specific ports" and enter the configured UDP port. If you've left the default settings on, the port number will be 3390.
7. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**.
8. On the **Action** page, select **Allow the connection**, and then select **Next**.
9. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**.
10. On the **Name** page, enter a name and description for your rule, then select **Finish**.

When you're done, verify that the new rule matches the format in the following screenshot.




Remote Desktop - Shortpath Private (UDP-In) Properties
Protocols and Ports
Scope
Advanced
Local Principals
Remote Users
General
Programs and Services
Remote Computers
Programs
All programs that meet the specified conditions
This program:
%SystemRoot%\system32\svchost.exe
Browse...
Application Packages
Specify the application packages to which this rule applies.
Settings...
Services
Specify the services to which this rule applies.
Compartments
All compartments that meet the specif
This compartment:
OK
Customize Service Settings
Apply this rule as follows:
Apply to all programs and services
Apply to services only
Apply to this service:
Name
Short Name
Radio Management Service
RmSvc
RdAgent
RdAgent
Recommended Troubleshooting Service
TroubleshootingSvc
Remote Access Auto Connection Manager
RasAuto
Remote Access Connection Manager
RasMan
Remote Desktop Agent Loader
RDAgentBootLoader
Remote Desktop Configuration
SessionEnv
Remote Desktop Services
TermService
Remote Desktop Services UserMode Port Redirector
UmRdpService
Apply to service with this service short name (example: eventlog):
TermService
OK
Cancel
Any
Any



You can also use PowerShell to configure Windows Firewall:

```
New-NetFirewallRule -DisplayName 'Remote Desktop - Shortpath (UDP-In)'  -Action Allow -Description 'Inbound rule for the Remote Desktop service to allow RDP traffic. [UDP 3390]' -Group '@FirewallAPI.dll,-28752' -Name 'RemoteDesktop-UserMode-In-Shortpath-UDP'  -PolicyStore PersistentStore -Profile Domain, Private -Service TermService -Protocol udp -LocalPort 3390 -Program '%SystemRoot%\system32\svchost.exe' -Enabled:True
```

Using PowerShell to configure Windows Defender Firewall

You can also use PowerShell to configure the group policy by running the following cmdlet.

```
# Replace $domainName value with the name of your Active Directory domain
# Replace $policyName value with the name of existing Group Policy Object
$domainName = "contoso.com"
$policyName = "RDP Shortpath Policy"
$gpoSession = Open-NetGPO -PolicyStore "$domainName\$policyName"
New-NetFirewallRule -DisplayName 'Remote Desktop - Shortpath (UDP-In)'  -Action Allow -Description 'Inbound rule for the Remote Desktop service to allow RDP traffic. [UDP 3390]' -Group '@FirewallAPI.dll,-28752' -Name 'RemoteDesktop-UserMode-In-Shortpath-UDP' -Profile Domain, Private -Service TermService -Protocol udp -LocalPort 3390 -Program '%SystemRoot%\system32\svchost.exe' -Enabled:True -GPOSession $gpoSession
Save-NetGPO -GPOSession $gpoSession
```

Configuring Azure Network Security Group

To allow access to the RDP Shortpath listener across network security boundaries, you need to configure Azure

Network Security Group to allow inbound UDP port 3390. Follow the network security group documentation to create an inbound security rule allowing traffic with following parameters:

- **Source** - **Any** or the IP range where the clients are residing
- **Source port ranges** - *
- **Destination** - **Any**
- **Destination port ranges** - **3390**
- **Protocol** - **UDP**
- **Action** - **Allow**
- Optionally change the **Priority**. The priority affects the order in which rules are applied: the lower the numerical value, the earlier the rule is applied.
- **Name** - - **RDP Shortpath**

Disabling RDP Shortpath for a specific subnet

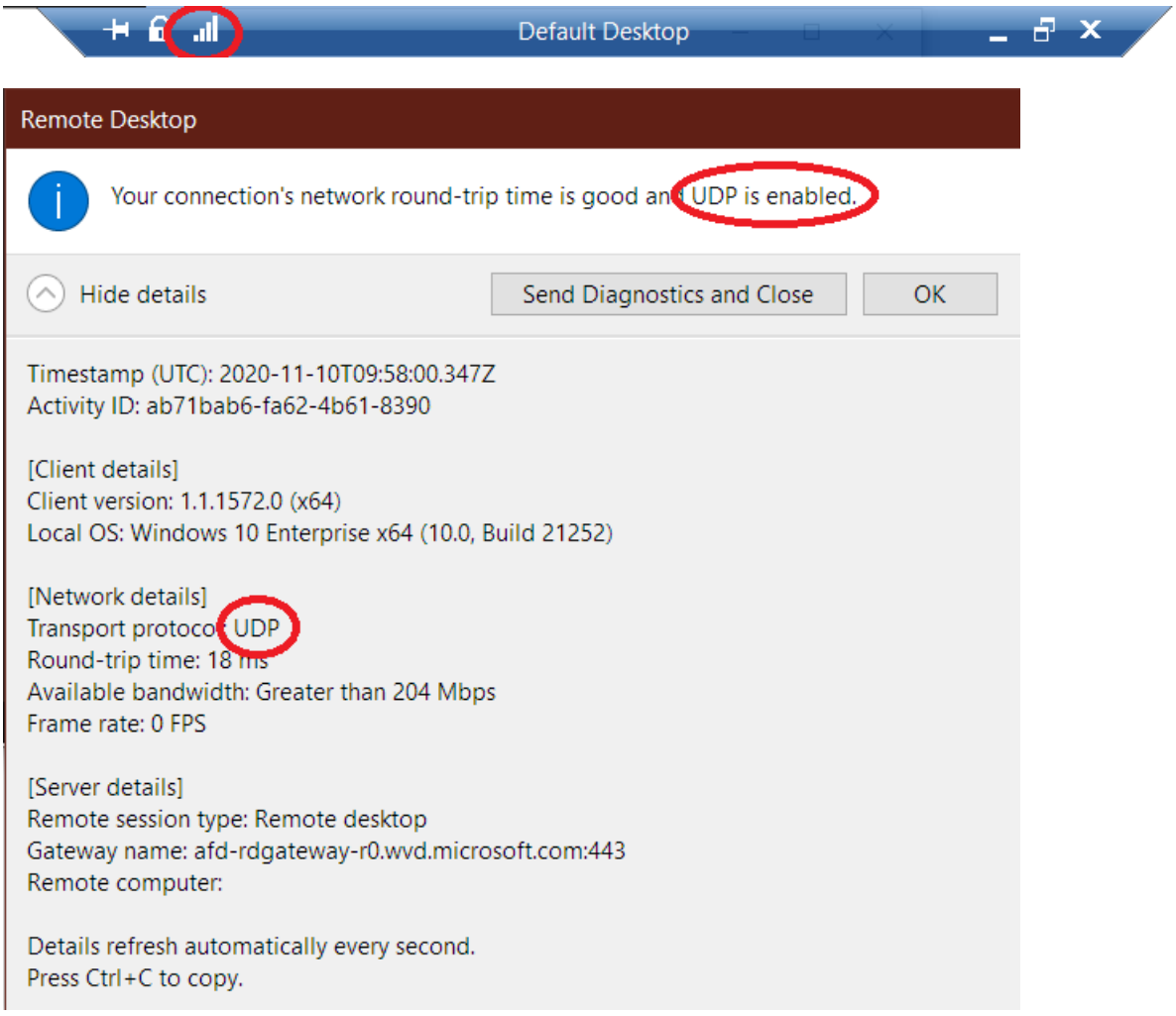
If you need to block specific subnets from using the RDP Shortpath transport, you can configure another network security group that specifies the correct Source IP ranges.

Verify your network connectivity

Next, you'll need to make sure your network is using RDP Shortpath. You can do this with either a "Connection Information" dialog or by using Log Analytics.

Connection Information dialog

Make sure connections are using RDP Shortpath, open the "Connection Information" dialog by going to the **Connection** tool bar on the top of the screen and selecting the antenna icon, as shown in the following screenshot.



Using event logs

To make sure your session is using RDP Shortpath transport:

1. Use the Azure Virtual Desktop client of your choice to connect to your VM desktop.
2. Open **Event Viewer**, then go to **Applications and Services Logs** > **Microsoft** > **Windows** > **RemoteDesktopServices-RdpCoreCDV** > **Microsoft-Windows-RemoteDesktopServices-RdpCoreCDV/Operational**.
3. If you can see Event ID 131, then your network is using RDP Shortpath transport.

Use Log Analytics

If you're using Azure Log Analytics, you can monitor connections by querying the WVDConnections table. A column named UdpUse indicates whether Azure Virtual Desktop RDP Stack is using UDP protocol on the current user connection. The possible values are:

- **0** - user connection isn't using RDP Shortpath
- **1** - THe user connection is using RDP Shortpath for managed networks.

The following query list lets you review connection information. You can run this query in the Log Analytics query editor. For each query, replace `userupn` with the UPN of the user you want to look up.

```
let Events = WVDConnections | where UserName == "userupn" ;
Events
| where State == "Connected"
| project CorrelationId , UserName, ResourceAlias , StartTime=TimeGenerated, UdpUse, SessionHostName,
SessionHostSxSStackVersion
| join (Events
| where State == "Completed"
| project EndTime=TimeGenerated, CorrelationId, UdpUse)
on CorrelationId
| project StartTime, Duration = EndTime - StartTime, ResourceAlias, UdpUse,  SessionHostName,
SessionHostSxSStackVersion
| sort by StartTime asc
```

Troubleshooting

Verify Shortpath listener

To verify that UDP listener is enabled, use the following PowerShell command on the session host:

```
Get-NetUDPEndpoint -OwningProcess ((Get-WmiObject win32_service -Filter "name = 'TermService'").ProcessId)
-LocalPort 3390
```

If enabled, you'll see the output like the following

```
LocalAddress                            LocalPort
------------                            ---------
::                                      3390
0.0.0.0                                 3390
```

If there's a conflict, you can identify the process that's blocking the port by running the following command:

```
Get-Process -id (Get-NetUDPEndpoint  -LocalPort 3390 -LocalAddress 0.0.0.0).OwningProcess
```

Disabling RDP Shortpath

In some cases, you may need to disable RDP Shortpath transport. You can disable RDP Shortpath by using the group policy.

Disabling RDP Shortpath on the client

To disable RDP Shortpath for a specific client, you can use the following Group Policy to disable the UDP support:

1. On the client, run **gpedit.msc**.
2. Go to **Computer Configuration** > **Administration Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Connection Client**.
3. Set the **"Turn Off UDP On Client"** setting to **Enabled**

Disable RDP Shortpath on the session host

To disable RDP Shortpath for a specific session host, you can use the following Group Policy to disable the UDP support:

1. On the Session Host Run **gpedit.msc**.
2. Go to **Computer Configuration > Administration Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections**.
3. Set the **"Select RDP Transport Protocols"** setting to **TCP Only**.

Next steps

- To learn about Azure Virtual Desktop network connectivity, see Understanding Azure Virtual Desktop network connectivity.
- To get started with Quality of Service (QoS) for Azure Virtual Desktop, see Implement Quality of Service (QoS) for Azure Virtual Desktop.

Implement Quality of Service (QoS) for Azure Virtual Desktop

12/6/2021 • 7 minutes to read • Edit Online

RDP Shortpath for managed networks provides a direct UDP-based transport between Remote Desktop Client and Session host. RDP Shortpath for managed networks enables configuration of Quality of Service (QoS) policies for the RDP data. QoS in Azure Virtual Desktop allows real-time RDP traffic that's sensitive to network delays to "cut in line" in front of traffic that's less sensitive. Example of such less sensitive traffic would be a downloading a new app, where an extra second to download isn't a large deal. QoS uses Windows Group Policy Objects to identify and mark all packets in real-time streams and help your network to give RDP traffic a dedicated portion of bandwidth.

If you support a large group of users experiencing any of the problems described in this article, you probably need to implement QoS. A small business with few users might not need QoS, but it should be helpful even there.

Without some form of QoS, you might see the following issues:

- Jitter – RDP packets arriving at different rates, which can result in visual and audio glitches
- Packet loss – packets dropped, which results in retransmission that requires additional time
- Delayed round-trip time (RTT) – RDP packets taking a long time to reach their destinations, which result in noticeable delays between input and reaction from the remote application.

The least complicated way to address these issues is to increase the data connections' size, both internally and out to the internet. Since that is often cost-prohibitive, QoS provides a way to manage the resources you have instead of adding bandwidth more effectively. To address quality issues, we recommend that you first use QoS, then add bandwidth only where necessary.

For QoS to be effective, you must apply consistent QoS settings throughout your organization. Any part of the path that fails to support your QoS priorities can degrade the quality RDP session.

Introduction to QoS queues

To provide QoS, network devices must have a way to classify traffic and must be able to distinguish RDP from other network traffic.

When network traffic enters a router, the traffic is placed into a queue. If a QoS policy isn't configured, there is only one queue, and all data is treated as first-in, first-out with the same priority. That means RDP traffic might get stuck behind traffic where a few extra milliseconds delay wouldn't be a problem.

When you implement QoS, you define multiple queues using one of several congestion management features, such as Cisco's priority queuing and Class-Based Weighted Fair Queueing (CBWFQ) and congestion avoidance features, such as weighted random early detection (WRED).

A simple analogy is that QoS creates virtual "carpool lanes" in your data network. So some types of data never or rarely encounter a delay. Once you create those lanes, you can adjust their relative size and much more effectively manage the connection bandwidth you have while still delivering business-grade experiences for your organization's users.

QoS implementation checklist

At a high level, do the following to implement QoS:

1. Make sure your network is ready
2. Make sure that RDP Shortpath for managed networks is enabled - QoS policies are not supported for reverse connect transport
3. Implement insertion of DSCP markers on session hosts

As you prepare to implement QoS, keep the following guidelines in mind:

- The shortest path to session host is best
- Any obstacles in between, such as proxies or packet inspection devices, aren't recommended

Make sure your network is ready

If you're considering a QoS implementation, you should already have determined your bandwidth requirements and other network requirements.

Traffic congestion across a network will significantly impact media quality. A lack of bandwidth leads to performance degradation and a poor user experience. As Azure Virtual Desktop adoption and usage grows, use Log Analytics to identify problems and then make adjustments using QoS and selective bandwidth additions.

VPN considerations

QoS only works as expected when implemented on all links between clients and session hosts. If you use QoS on an internal network and a user signs in from a remote location, you can only prioritize within your internal, managed network. Although remote locations can receive a managed connection by implementing a virtual private network (VPN), a VPN inherently adds packet overhead and creates delays in real-time traffic.

In a global organization with managed links that span continents, we strongly recommend QoS because bandwidth for those links is limited compared to the LAN.

Insert DSCP markers

You could implement QoS using a Group Policy Object (GPO) to direct session hosts to insert a DSCP marker in IP packet headers identifying it as a particular type of traffic. Routers and other network devices can be configured to recognize these markings and put the traffic in a separate, higher-priority queue.

You can compare DSCP markings to postage stamps that indicate to postal workers how urgent the delivery is and how best to sort it for speedy delivery. Once you've configured your network to give priority to RDP streams, lost packets and late packets should diminish significantly.

Once all network devices are using the same classifications, markings, and priorities, it's possible to reduce or eliminate delays, dropped packets, and jitter. From the RDP perspective, the essential configuration step is the classification and marking of packets. However, for end-to-end QoS to be successful, you also need to align the RDP configuration with the underlying network configuration carefully. The DSCP value tells a correspondingly configured network what priority to give a packet or stream.

We recommend using DSCP value 46 that maps to **Expedited Forwarding (EF)** DSCP class.

Implement QoS on session host using Group Policy

You can use policy-based Quality of Service (QoS) within Group Policy to set the predefined DSCP value.

To create a QoS policy for domain-joined session hosts, first, sign in to a computer on which Group Policy Management has been installed. Open Group Policy Management (select Start, point to Administrative Tools, and then select Group Policy Management), and then complete the following steps:

1. In Group Policy Management, locate the container where the new policy should be created. For example, if all your session hosts computers are located in an OU named **"session hosts"**, the new policy should

be created in the Session Hosts OU.

2. Right-click the appropriate container, and then select **Create a GPO in this domain, and Link it here**.
3. In the **New GPO** dialog box, type a name for the new Group Policy object in the **Name** box, and then select **OK**.
4. Right-click the newly created policy, and then select **Edit**.
5. In the Group Policy Management Editor, expand **Computer Configuration**, expand **Windows Settings**, right-click **Policy-based QoS**, and then select **Create new policy**.
6. In the **Policy-based QoS** dialog box, on the opening page, type a name for the new policy in the **Name** box. Select **Specify DSCP Value** and set the value to **46**. Leave **Specify Outbound Throttle Rate** unselected, and then select **Next**.
7. On the next page, select **Only applications with this executable name** and enter the name **svchost.exe**, and then select **Next**. This setting instructs the policy to only prioritize matching traffic from the Remote Desktop Service.
8. On the third page, make sure that both **Any source IP address** and **Any destination IP address** are selected, and then select **Next**. These two settings ensure that packets will be managed regardless of which computer (IP address) sent the packets and which computer (IP address) will receive the packets.
9. On page four, select **UDP** from the **Select the protocol this QoS policy applies to** drop-down list.
10. Under the heading **Specify the source port number**, select **From this source port or range**. In the accompanying text box, type **3390**. Select **Finish**.

The new policies you've created won't take effect until Group Policy has been refreshed on your session host computers. Although Group Policy periodically refreshes on its own, you can force an immediate refresh by following these steps:

1. On each session host for which you want to refresh Group Policy, open a Command Prompt as administrator (*Run as administrator*).
2. At the command prompt, enter

```
gpupdate /force
```

Implement QoS on session host using PowerShell

You can set QoS for RDP Shortpath for managed networks using the PowerShell cmdlet below:

```
New-NetQosPolicy -Name "RDP Shortpath for managed networks" -AppPathNameMatchCondition "svchost.exe" -
IPProtocolMatchCondition UDP -IPSrcPortStartMatchCondition 3390 -IPSrcPortEndMatchCondition 3390 -DSCPAction
46 -NetworkProfile All
```

Related articles

- Quality of Service (QoS) Policy

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see Understanding Remote Desktop Protocol (RDP) Bandwidth Requirements for Azure Virtual Desktop.
- To learn about Azure Virtual Desktop network connectivity, see Understanding Azure Virtual Desktop

network connectivity.

Required URL list

12/6/2021 • 4 minutes to read • Edit Online

In order to deploy and use Azure Virtual Desktop, you must unblock certain URLs so your virtual machines (VMs) can access them anytime. This article lists the required URLs you need to unblock in order for Azure Virtual Desktop to function properly.

> **IMPORTANT**
> Azure Virtual Desktop doesn't support deployments that block the URLs listed in this article.

Required URL Check tool

The Required URL Check tool will validate URLs and display whether the URLs the virtual machine needs to function are accessible. If not, then the tool will list the inaccessible URLs so you can unblock them, if needed.

It's important to keep the following things in mind:

- You can only use the Required URL Check tool for deployments in commercial clouds.
- The Required URL Check tool can't check URLs with wildcards so make sure you unblock those URLs first.

Requirements

You need the following things to use the Required URL Check tool:

- Your VM must have a .NET 4.6.2 framework
- RDAgent version 1.0.2944.400 or higher
- The WVDAgentUrlTool.exe file must be in the same folder as the WVDAgentUrlTool.config file

How to use the Required URL Check tool

To use the Required URL Check tool:

1. Open a command prompt as an administrator on your VM.
2. Run the following command to change the directory to the same folder as the current build agent (RDAgent_1.0.2944.1200 in this example):

   ```
   cd C:\Program Files\Microsoft RDInfra\RDAgent_1.0.2944.1200
   ```

3. Run the following command:

   ```
   WVDAgentUrlTool.exe
   ```

4. Once you run the file, you'll see a list of accessible and inaccessible URLs.

   For example, the following screenshot shows a scenario where you'd need to unblock two required non-wildcard URLs:

```
WVD
=====================================================

NOT Accessible URLs:
=====================================================
mrsglobalsteus2prod.blob.core.windows.net
gcs.prod.monitoring.core.windows.net

Accessible URLs:
=====================================================
rdbroker-g-us-r0.wvd.microsoft.com
rdbroker-g-us-r1.wvd.microsoft.com
rddiagnostics-g-us-r0.wvd.microsoft.com
rddiagnostics-g-us-r1.wvd.microsoft.com
rdweb-g-us-r0.wvd.microsoft.com
rdweb-g-us-r1.wvd.microsoft.com
rdgateway-g-us-r0.wvd.microsoft.com
rdgateway-g-us-r1.wvd.microsoft.com
catalogartifact.azureedge.net
production.diagnostics.monitoring.core.windows.net
wvdportalstorageblob.blob.core.windows.net
maupdateaccount.blob.core.windows.net
www.wvd.microsoft.com
rdbroker.wvd.microsoft.com
rddiagnostics.wvd.microsoft.com
rdweb.wvd.microsoft.com
rdgateway.wvd.microsoft.com
rdbroker-r0.wvd.microsoft.com
rdbroker-r1.wvd.microsoft.com
rddiagnostics-r0.wvd.microsoft.com
rddiagnostics-r1.wvd.microsoft.com
rdweb-r0.wvd.microsoft.com
rdweb-r1.wvd.microsoft.com
rdgateway-r0.wvd.microsoft.com
```

Here's what the output should look like once you've unblocked all required non-wildcard URLs:

```
WVD
==========================================================

NOT Accessible URLs:
==========================================================

Accessible URLs:
==========================================================
rdbroker-g-us-r0.wvd.microsoft.com
rdbroker-g-us-r1.wvd.microsoft.com
rddiagnostics-g-us-r0.wvd.microsoft.com
rddiagnostics-g-us-r1.wvd.microsoft.com
rdweb-g-us-r0.wvd.microsoft.com
rdweb-g-us-r1.wvd.microsoft.com
rdgateway-g-us-r0.wvd.microsoft.com
rdgateway-g-us-r1.wvd.microsoft.com
mrsglobalsteus2prod.blob.core.windows.net
gcs.prod.monitoring.core.windows.net
catalogartifact.azureedge.net
production.diagnostics.monitoring.core.windows.net
wvdportalstorageblob.blob.core.windows.net
maupdateaccount.blob.core.windows.net
www.wvd.microsoft.com
rdbroker.wvd.microsoft.com
rddiagnostics.wvd.microsoft.com
rdweb.wvd.microsoft.com
rdgateway.wvd.microsoft.com
rdbroker-r0.wvd.microsoft.com
rdbroker-r1.wvd.microsoft.com
rddiagnostics-r0.wvd.microsoft.com
rddiagnostics-r1.wvd.microsoft.com
rdweb-r0.wvd.microsoft.com
rdweb-r1.wvd.microsoft.com
rdgateway-r0.wvd.microsoft.com
```

Virtual machines

The Azure virtual machines you create for Azure Virtual Desktop must have access to the following URLs in the Azure commercial cloud:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*.wvd.microsoft.com	443	Service traffic	WindowsVirtualDesktop
gcs.prod.monitoring.core.windows.net	443	Agent traffic	AzureCloud
production.diagnostics.monitoring.core.windows.net	443	Agent traffic	AzureCloud
*xt.blob.core.windows.net	443	Agent traffic	AzureCloud
*eh.servicebus.windows.net	443	Agent traffic	AzureCloud

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*xt.table.core.windows.net	443	Agent traffic	AzureCloud
*xt.queue.core.windows.net	443	Agent traffic	AzureCloud
catalogartifact.azureedge.net	443	Azure Marketplace	AzureCloud
kms.core.windows.net	1688	Windows activation	Internet
mrsglobalsteus2prod.blob.core.windows.net	443	Agent and SXS stack updates	AzureCloud
wvdportalstorageblob.blob.core.windows.net	443	Azure portal support	AzureCloud
169.254.169.254	80	Azure Instance Metadata service endpoint	N/A
168.63.129.16	80	Session host health monitoring	N/A

IMPORTANT

Azure Virtual Desktop now supports the FQDN tag. For more information, see Use Azure Firewall to protect Azure Virtual Desktop deployments.

We recommend you use FQDN tags or service tags instead of URLs to prevent service issues. The listed URLs and tags only correspond to Azure Virtual Desktop sites and resources. They don't include URLs for other services like Azure Active Directory.

The Azure virtual machines you create for Azure Virtual Desktop must have access to the following URLs in the Azure Government cloud:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*.wvd.azure.us	443	Service traffic	WindowsVirtualDesktop
gcs.monitoring.core.usgovcloudapi.net	443	Agent traffic	AzureCloud
monitoring.core.usgovcloudapi.net	443	Agent traffic	AzureCloud
fairfax.warmpath.usgovcloudapi.net	443	Agent traffic	AzureCloud
*xt.blob.core.usgovcloudapi.net	443	Agent traffic	AzureCloud
*.servicebus.usgovcloudapi.net	443	Agent traffic	AzureCloud

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*xt.table.core.usgovcloudapi.net	443	Agent traffic	AzureCloud
Kms.core.usgovcloudapi.net	1688	Windows activation	Internet
mrsglobalstugviffx.blob.core.usgovcloudapi.net	443	Agent and SXS stack updates	AzureCloud
wvdportalstorageblob.blob.core.usgovcloudapi.net	443	Azure portal support	AzureCloud
169.254.169.254	80	Azure Instance Metadata service endpoint	N/A
168.63.129.16	80	Session host health monitoring	N/A

The following table lists optional URLs that your Azure virtual machines can have access to:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	AZURE GOV
*.microsoftonline.com	443	Authentication to Microsoft Online Services	login.microsoftonline.us
*.events.data.microsoft.com	443	Telemetry Service	None
www.msftconnecttest.com	443	Detects if the OS is connected to the internet	None
*.prod.do.dsp.mp.microsoft.com	443	Windows Update	None
login.windows.net	443	Sign in to Microsoft Online Services, Microsoft 365	login.microsoftonline.us
*.sfx.ms	443	Updates for OneDrive client software	oneclient.sfx.ms
*.digicert.com	443	Certificate revocation check	None
*.azure-dns.com	443	Azure DNS resolution	None
*.azure-dns.net	443	Azure DNS resolution	None

NOTE

Azure Virtual Desktop currently doesn't have a list of IP address ranges that you can unblock to allow network traffic. We only support unblocking specific URLs at this time.

If you're using a Next Generation Firewall (NGFW), you'll need to use a dynamic list specifically made for Azure IPs to make sure you can connect.

For a list of safe Office-related URLs, including required Azure Active Directory-related URLs, see Office 365 URLs and IP address ranges.

You must use the wildcard character (*) for URLs involving service traffic. If you prefer to not use * for agent-related traffic, here's how to find the URLs without wildcards:

1. Register your virtual machines to the Azure Virtual Desktop host pool.
2. Open **Event viewer**, then go to **Windows logs** > **Application** > **WVD-Agent** and look for Event ID 3701.
3. Unblock the URLs that you find under Event ID 3701. The URLs under Event ID 3701 are region-specific. You'll need to repeat the unblocking process with the relevant URLs for each region you want to deploy your virtual machines in.

Remote Desktop clients

Any Remote Desktop clients you use must have access to the following URLs:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	CLIENT(S)	AZURE GOV
*.wvd.microsoft.com	443	Service traffic	All	*.wvd.microsoft.us
*.servicebus.windows.net	443	Troubleshooting data	All	*.servicebus.usgovcloudapi.net
go.microsoft.com	443	Microsoft FWLinks	All	None
aka.ms	443	Microsoft URL shortener	All	None
docs.microsoft.com	443	Documentation	All	None
privacy.microsoft.com	443	Privacy statement	All	None
query.prod.cms.rt.microsoft.com	443	Client updates	Windows Desktop	None

IMPORTANT

Opening these URLs is essential for a reliable client experience. Blocking access to these URLs is unsupported and will affect service functionality.

These URLs only correspond to client sites and resources. This list doesn't include URLs for other services like Azure Active Directory. Azure Active Directory URLs can be found under ID 56, 59 and 125 on the Office 365 URLs and IP address ranges.

Remote Desktop Protocol (RDP) bandwidth requirements

12/6/2021 • 8 minutes to read • Edit Online

Remote Desktop Protocol (RDP) is a sophisticated technology that uses various techniques to perfect the server's remote graphics' delivery to the client device. Depending on the use case, availability of computing resources, and network bandwidth, RDP dynamically adjusts various parameters to deliver the best user experience.

Remote Desktop Protocol multiplexes multiple Dynamic Virtual Channels (DVCs) into a single data channel sent over different network transports. There are separate DVCs for remote graphics, input, device redirection, printing, and more. Azure Virtual Desktop partners can also use their extensions that use DVC interfaces.

The amount of the data sent over RDP depends on the user activity. For example, a user may work with basic textual content for most of the session and consume minimal bandwidth, but then generate a printout of a 200-page document to the local printer. This print job will use a significant amount of network bandwidth.

When using a remote session, your network's available bandwidth dramatically impacts the quality of your experience. Different applications and display resolutions require different network configurations, so it's essential to make sure your network configuration meets your needs.

Estimating bandwidth utilization

RDP uses various compression algorithms for different types of data. The table below guides estimating of the data transfers:

TYPE OF DATA	DIRECTION	HOW TO ESTIMATE
Remote Graphics	Session host to client	See the detailed guidelines
Heartbeats	Both directions	~ 20 bytes every 5 seconds
Input	Client to session Host	Amount of data is based on the user activity, less than 100 bytes for most of the operations
File transfers	Both directions	File transfers are using bulk compression. Use .zip compression for approximation
Printing	Session host to client	Print job transfer depends on the driver and using bulk compression, use .zip compression for approximation

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

Estimating bandwidth used by remote graphics

It's tough to predict bandwidth use by the remote desktop. The user activities generate most of the remote desktop traffic. Every user is unique, and differences in their work patterns may significantly change network use.

The best way to understand bandwidth requirements is to monitor real user connections. Monitoring can be performed by the built-in performance counters or by the network equipment.

However, in many cases, you may estimate network utilization by understanding how Remote Desktop Protocol works and by analyzing your users' work patterns.

The remote protocol delivers the graphics generated by the remote server to display it on a local monitor. More specifically, it provides the desktop bitmap entirely composed on the server. While sending a desktop bitmap seems like a simple task at first approach, it requires a significant amount of resources. For example, a 1080p desktop image in its uncompressed form is about 8Mb in size. Displaying this image on the locally connected monitor with a modest screen refresh rate of 30 Hz requires bandwidth of about 237 MB/s.

To reduce the amount of data transferred over the network, RDP uses the combination of multiple techniques, including but not limited to

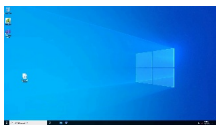
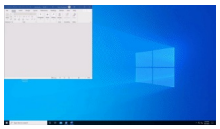
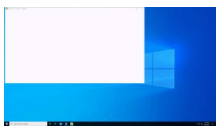
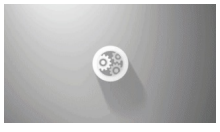
- Frame rate optimizations
- Screen content classification
- Content-specific codecs
- Progressive image encoding
- Client-side caching

To better understand remote graphics, consider the following:

- The richer the graphics, more bandwidth it will take
  - Text, window UI elements, and solid color areas are consuming less bandwidth than anything else.
  - Natural images are the most significant contributors to bandwidth use. But client-side caching helps with its reduction.
- Only changed parts of the screen are transmitted. If there are no visible updates on the screen, no updates are sent.
- Video playback and other high-frame-rate content are essentially an image slideshow. RDP dynamically uses appropriate video codecs to deliver them with the close to original frame rate. However, it's still graphics, and it's still the most significant contributor to bandwidth utilization.
- Idle time in remote desktop means no or minimal screen updates; so, network use is minimal during idle times.
- When remote desktop client window is minimized, no graphical updates are sent from the session host.

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. Different display resolutions require different available bandwidths.

The table below guides estimating of the data used by the different graphic scenarios. These numbers apply to a single monitor configuration with 1920x1080 resolution and with both default graphics mode and H.264/AVC 444 graphics mode.

SCENARIO	DEFAULT MODE	H.264/AVC 444 MODE	THUMBNAIL	DESCRIPTION OF THE SCENARIO
Idle	0.3 Kbps	0.3 Kbps		User is paused their work and there's no active screen updates
Microsoft Word	100-150 Kbps	200-300 Kbps		User is actively working with Microsoft Word, typing, pasting graphics and switching between documents
Microsoft Excel	150-200 Kbps	400-500 Kbps		User is actively working with Microsoft Excel, multiple cells with formulas and charts are updated simultaneously
Microsoft PowerPoint	4-4.5 Mbps	1.6-1.8 Mbps		User is actively working with Microsoft PowerPoint, typing, pasting. User also modifying rich graphics, and using slide transition effects
Web Browsing	6-6.5 Mbps	0.9-1 Mbps		User is actively working with a graphically rich website that contains multiple static and animated images. User scrolls the pages both horizontally and vertically
Image Gallery	3.3-3.6 Mbps	0.7-0.8 Mbps		User is actively working with the image gallery application. browsing, zooming, resizing and rotating images
Video playback	8.5-9.5 Mbps	2.5-2.8 Mbps		User is watching a 30 FPS video that consumes 1/2 of the screen
Fullscreen Video playback	7.5-8.5 Mbps	2.5-3.1 Mbps		User is watching a 30 FPS video that maximized to a fullscreen

Dynamic bandwidth allocation

Remote Desktop Protocol is a modern protocol designed to adjust to the changing network conditions dynamically. Instead of using the hard limits on bandwidth utilization, RDP uses continuous network detection that actively monitors available network bandwidth and packet round-trip time. Based on the findings, RDP dynamically selects the graphic encoding options and allocates bandwidth for device redirection and other virtual channels.
This technology allows RDP to use the full network pipe when available and rapidly back off when the network is needed for something else. RDP detects that and adjusts image quality, frame rate, or compression algorithms if other applications request the network.

Limit network bandwidth use with throttle rate

In most scenarios, there's no need to limit bandwidth utilization as limiting may affect user experience. Yet in the constrained networks you may want to limit network utilization. Another example is leased networks that are charged for the amount of traffic used.

In such cases, you could limit an RDP outbound network traffic by specifying a throttle rate in QoS Policy.

> **NOTE**
> Make sure that RDP Shortpath for managed networks is enabled - throttle rate-limiting are not supported for reverse connect transport.

Implement throttle rate limiting on session host using Group Policy

You can use policy-based Quality of Service (QoS) within Group Policy to set the predefined throttle rate.

To create a QoS policy for domain-joined session hosts, first, sign in to a computer on which Group Policy Management has been installed. Open Group Policy Management (select Start, point to Administrative Tools, and then select Group Policy Management), and then complete the following steps:

1. In Group Policy Management, locate the container where the new policy should be created. For example, if all your session hosts computers are located in an OU named **Session Hosts**, the new policy should be created in the Session Hosts OU.
2. Right-click the appropriate container, and then select **Create a GPO in this domain, and Link it here**.
3. In the **New GPO** dialog box, type a name for the new Group Policy object in the **Name** box, and then select **OK**.
4. Right-click the newly created policy, and then select **Edit**.
5. In the Group Policy Management Editor, expand **Computer Configuration**, expand **Windows Settings**, right-click **Policy-based QoS**, and then select **Create new policy**.
6. In the **Policy-based QoS** dialog box, on the opening page, type a name for the new policy in the **Name** box. Select **Specify Outbound Throttle Rate** and set the required value, and then select **Next**.
7. On the next page, select **Only applications with this executable name** and enter the name **svchost.exe**, and then select **Next**. This setting instructs the policy to only prioritize matching traffic from the Remote Desktop Service.
8. On the third page, make sure that both **Any source IP address** and **Any destination IP address** are selected. Select **Next**. These two settings ensure that packets will be managed regardless of which computer (IP address) sent the packets and which computer (IP address) will receive the packets.
9. On page four, select **UDP** from the **Select the protocol this QoS policy applies to** drop-down list.

10. Under the heading **Specify the source port number**, select **From this source port or range**. In the accompanying text box, type **3390**. Select **Finish**.

The new policies you've created won't take effect until Group Policy has been refreshed on your session host computers. Although Group Policy periodically refreshes on its own, you can force an immediate refresh by following these steps:

1. On each session host for which you want to refresh Group Policy, open a Command Prompt as administrator (*Run as administrator*).
2. At the command prompt, enter

```
gpupdate /force
```

Implement throttle rate limiting on session host using PowerShell

You can set throttle rate for RDP Shortpath for managed networks using the PowerShell cmdlet below:

```
New-NetQosPolicy -Name "RDP Shortpath for managed networks" -AppPathNameMatchCondition "svchost.exe" -
IPProtocolMatchCondition UDP -IPSrcPortStartMatchCondition 3390 -IPSrcPortEndMatchCondition 3390  -
ThrottleRateActionBitsPerSecond 10mb -NetworkProfile All
```

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see Understanding Remote Desktop Protocol (RDP) Bandwidth Requirements for Azure Virtual Desktop.
- To learn about Azure Virtual Desktop network connectivity, see Understanding Azure Virtual Desktop network connectivity.
- To get started with Quality of Service (QoS) for Azure Virtual Desktop, see Implement Quality of Service (QoS) for Azure Virtual Desktop.

Proxy server guidelines for Azure Virtual Desktop

12/6/2021 • 6 minutes to read • Edit Online

This article will show you how to use a proxy server with Azure Virtual Desktop. The recommendations in this article only apply to connections between Azure Virtual Desktop infrastructure, client, and session host agents. This article doesn't cover network connectivity for Office, Windows 10, FSLogix, or other Microsoft applications.

What are proxy servers?

We recommend bypassing proxies for Azure Virtual Desktop traffic. Proxies don't make Azure Virtual Desktop more secure because the traffic is already encrypted. To learn more about connection security, see Connection security.

Most proxy servers aren't designed for supporting long running WebSocket connections and may affect connection stability. Proxy server scalability also causes issues because Azure Virtual Desktop uses multiple long-term connections. If you do use proxy servers, they must be the right size to run these connections.

If the proxy server's geography is far from the host, then this distance will cause more latency in your user connections. More latency means slower connection time and worse user experience, especially in scenarios that need graphics, audio, or low-latency interactions with input devices. If you must use a proxy server, keep in mind that you need to place the server in the same geography as the Azure Virtual Desktop Agent and client.

If you configure your proxy server as the only path for Azure Virtual Desktop traffic to take, the Remote Desktop Protocol (RDP) data will be forced over Transmission Control Protocol (TCP) instead of User Datagram Protocol (UDP). This move lowers the visual quality and responsiveness of the remote connection.

In summary, we don't recommend using proxy servers on Azure Virtual Desktop because they cause performance-related issues from latency degradation and packet loss.

Bypassing a proxy server

If your organization's network and security policies require proxy servers for web traffic, you can configure your environment to bypass Azure Virtual Desktop connections while still routing the traffic through the proxy server. However, each organization's policies are unique, so some methods may work better for your deployment than others. Here are some configuration methods you can try to prevent performance and reliability loss in your environment:

- Azure service tags on the Azure firewall
- Proxy server bypass using Proxy Auto Configuration (.PAC) files
- Bypass list in the local proxy configuration
- Using proxy servers for per-user configuration
- Using RDP Shortpath for the RDP connection while keeping the service traffic over the proxy

Recommendations for using proxy servers

Some organizations require that all user traffic goes through a proxy server for tracking or packet inspection. This section describes how we recommend configuring your environment in these cases.

Use proxy servers in the same Azure geography

When you use a proxy server, it handles all communication with the Azure Virtual Desktop infrastructure and performs DNS resolution and Anycast routing to the nearest Azure Front Door. If your proxy servers are distant

or distributed across an Azure geography, your geographical resolution will be less accurate. Less accurate geographical resolution means connections will be routed to a more distant Azure Virtual Desktop cluster. To avoid this issue, only use proxy servers that are geographically close to your Azure Virtual Desktop cluster.

Use RDP Shortpath for managed networks for desktop connectivity

When you enable RDP Shortpath for managed networks, RDP data will bypass the proxy server, if possible. Bypassing the proxy server ensures optimal routing while using the UDP transport. Other Azure Virtual Desktop traffic, such as brokering, orchestration, and diagnostics will still go through the proxy server.

Don't use SSL termination on the proxy server

Secure Sockets Layer (SSL) termination replaces security certificates of the Azure Virtual Desktop components with certificates generated by proxy server. This proxy server feature enables packet inspection for HTTPS traffic on the proxy server. However, packet inspection also increases the service response time, making it take longer for users to sign in. For reverse-connect scenarios, RDP traffic packet inspection isn't necessary because reverse-connect RDP traffic is binary and uses extra levels of encryption.

If you configure your proxy server to use SSL inspection, remember that you can't revert your server to its original state after the SSL inspection makes changes. If something in your Azure Virtual Desktop environment stops working while you have SSL inspection enabled, you must disable SSL inspection and try again before you open a support case. SSL inspection can also cause the Azure Virtual Desktop agent to stop working because it interferes with trusted connections between the agent and the service.

Don't use proxy servers that need authentication

Azure Virtual Desktop components on the session host run in the context of their operating system, so they don't support proxy servers that require authentication. If the proxy server requires authentication, the connection will fail.

Plan for the proxy server network capacity

Proxy servers have capacity limits. Unlike regular HTTP traffic, RDP traffic has long running, chatty connections that are bi-directional and consume lots of bandwidth. Before you set up a proxy server, talk to your proxy server vendor about how much throughput your server has. Also make sure to ask them how many proxy sessions you can run at one time. After you deploy the proxy server, carefully monitor its resource use for bottlenecks in Azure Virtual Desktop traffic.

Proxy servers for Windows 7 session hosts

Session hosts running on Windows 7 don't support proxy server connections for reverse-connect RDP data. If the session host can't directly connect to the Azure Virtual Desktop gateways, the connection won't work.

Proxy servers and Teams optimization

Azure Virtual Desktop doesn't support proxy servers for Teams optimization.

Session host configuration recommendations

To configure a session host level proxy server, you need to enable a systemwide proxy. Remember that systemwide configuration affects all OS components and applications running on the session host. The following sections are recommendations for configuring systemwide proxies.

Use the Web Proxy Auto-Discovery (WPAD) protocol

The Azure Virtual Desktop agent automatically tries to locate a proxy server on the network using the Web Proxy Auto-Discovery (WPAD) protocol. During a location attempt, the agent searches the domain name server (DNS) for a file named **wpad.domainsuffix**. If the agent finds the file in the DNS, it makes an HTTP request for a file named **wpad.dat**. The response becomes the proxy configuration script that chooses the outbound proxy server.

To configure your network to use DNS resolution for WPAD, follow the instructions in Auto detect settings

Internet Explorer 11. Make sure the DNS server global query blocklist allows the WPAD resolution by following the directions in Set-DnsServerGlobalQueryBlockList.

Manually set a device-wide Internet Explorer proxy

You can set a device-wide proxy or Proxy Auto Configuration (.PAC) file that applies to all interactive, LocalSystem, and NetworkService users with the Network Proxy CSP.

You can also configure the proxy server for the local system account by running the following **bitsadmin** command, as shown in the following example:

```
bitsadmin /util /setieproxy LOCALSYSTEM AUTOSCRIPT http://server/proxy.pac
```

Client-side proxy support

The Azure Virtual Desktop client supports proxy servers configured with system settings or a Network Proxy CSP.

Support for clients running on Windows 7

Clients running on Windows 7 don't support proxy server connections for reverse-connect RDP data. If the client can't directly connect to the Azure Virtual Desktop gateways, the connection won't work.

Azure Virtual Desktop client support

The following table shows which Azure Virtual Desktop clients support proxy servers:

CLIENT NAME	PROXY SERVER SUPPORT
Windows Desktop	Yes
Web client	Yes
Android	No
iOS	Yes
macOS	Yes
Windows Store	Yes

For more information about proxy support on Linux based thin clients, see Thin client support.

Support limitations

There are many third-party services and applications that act as a proxy server. These third-party services include distributed next-gen firewalls, web security systems, and basic proxy servers. We can't guarantee that every configuration is compatible with Azure Virtual Desktop. Microsoft only provides limited support for connections established over a proxy server. If you're experiencing connectivity issues while using a proxy server, Microsoft support recommends you configure a proxy bypass and then try to reproduce the issue.

Next steps

For more information about keeping your Azure Virtual Desktop deployment secure, check out our security guide.

Determine user connection latency in Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

Azure Virtual Desktop is globally available. Administrators can create virtual machines (VMs) in any Azure region they want. Connection latency will vary depending on the location of the users and the virtual machines. Azure Virtual Desktop services will continuously roll out to new geographies to improve latency.

The Azure Virtual Desktop Experience Estimator tool can help you determine the best location to optimize the latency of your VMs. We recommend you use the tool every two to three months to make sure the optimal location hasn't changed as Azure Virtual Desktop rolls out to new areas.

Interpreting results from the Azure Virtual Desktop Experience Estimator tool

In Azure Virtual Desktop, latency up to 150 ms shouldn’t impact user experience that doesn't involve rendering or video. Latencies between 150 ms and 200 ms should be fine for text processing. Latency above 200 ms may impact user experience.

In addition, the Azure Virtual Desktop connection depends on the internet connection of the machine the user is using the service from. Users may lose connection or experience input delay in one of the following situations:

- The user doesn't have a stable local internet connection and the latency is over 200 ms.
- The network is saturated or rate-limited.

We recommend you choose VMs locations that are as close to your users as possible. For example, if the user is located in India but the VM is in the United States, there will be latency that will affect the overall user experience.

Azure Front Door

Azure Virtual Desktop uses Azure Front Door to redirect the user connection to the nearest Azure Virtual Desktop gateway based on the source IP address. Azure Virtual Desktop will always use the Azure Virtual Desktop gateway that the client chooses.

Next steps

- To check the best location for optimal latency, see the Azure Virtual Desktop Experience Estimator tool.
- For pricing plans, see Azure Virtual Desktop pricing.
- To get started with your Azure Virtual Desktop deployment, check out our tutorial.

Delegated access in Azure Virtual Desktop

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Azure Virtual Desktop has a delegated access model that lets you define the amount of access a particular user is allowed to have by assigning them a role. A role assignment has three components: security principal, role definition, and scope. The Azure Virtual Desktop delegated access model is based on the Azure RBAC model. To learn more about specific role assignments and their components, see the Azure role-based access control overview.

Azure Virtual Desktop delegated access supports the following values for each element of the role assignment:

- Security principal
  - Users
  - User groups
  - Service principals
- Role definition
  - Built-in roles
  - Custom roles
- Scope
  - Host pools
  - App groups
  - Workspaces

PowerShell cmdlets for role assignments

Before you start, make sure to follow the instructions in Set up the PowerShell module to set up the Azure Virtual Desktop PowerShell module if you haven't already.

Azure Virtual Desktop uses Azure role-based access control (Azure RBAC) while publishing app groups to users or user groups. The Desktop Virtualization User role is assigned to the user or user group and the scope is the app group. This role gives the user special data access on the app group.

Run the following cmdlet to add Azure Active Directory users to an app group:

```
New-AzRoleAssignment -SignInName <userupn> -RoleDefinitionName "Desktop Virtualization User" -ResourceName
<appgroupname> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Run the following cmdlet to add Azure Active Directory user group to an app group:

```
New-AzRoleAssignment -ObjectId <usergroupobjectid> -RoleDefinitionName "Desktop Virtualization User" -
ResourceName <appgroupname> -ResourceGroupName <resourcegroupname> -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Next steps

For a more complete list of PowerShell cmdlets each role can use, see the PowerShell reference.

For a complete list of roles supported in Azure RBAC, see Azure built-in roles.

For guidelines for how to set up a Azure Virtual Desktop environment, see Azure Virtual Desktop environment.

Host pool load-balancing algorithms

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Azure Virtual Desktop supports two load-balancing algorithms. Each algorithm determines which session host will host a user's session when they connect to a resource in a host pool.

The following load-balancing algorithms are available in Azure Virtual Desktop:

- Breadth-first load balancing allows you to evenly distribute user sessions across the session hosts in a host pool.
- Depth-first load balancing allows you to saturate a session host with user sessions in a host pool. Once the first session host reaches its session limit threshold, the load balancer directs any new user connections to the next session host in the host pool until it reaches its limit, and so on.

Each host pool can only configure one type of load-balancing specific to it. However, both load-balancing algorithms share the following behaviors no matter which host pool they're in:

- If a user already has an active or disconnected session in the host pool and signs in again, the load balancer will successfully redirect them to the session host with their existing session. This behavior applies even if that session host's AllowNewConnections property is set to False (drain mode is enabled).
- If a user doesn't already have a session in the host pool, then the load balancer won't consider session hosts whose AllowNewConnections property is set to False during load balancing.

Breadth-first load-balancing algorithm

The breadth-first load-balancing algorithm allows you to distribute user sessions across session hosts to optimize for session performance. This algorithm is ideal for organizations that want to provide the best experience for users connecting to their pooled virtual desktop environment.

The breadth-first algorithm first queries session hosts that allow new connections. The algorithm then selects a session host randomly from half the set of session hosts with the least number of sessions. For example, if there are nine machines with 11, 12, 13, 14, 15, 16, 17, 18, and 19 sessions, a new session you create won't automatically go to the first machine. Instead, it can go to any of the first five machines with the lowest number of sessions (11, 12, 13, 14, 15).

Depth-first load-balancing algorithm

The depth-first load-balancing algorithm allows you to saturate one session host at a time to optimize for scale down scenarios. This algorithm is ideal for cost-conscious organizations that want more granular control on the number of virtual machines they've allocated for a host pool.

The depth-first algorithm first queries session hosts that allow new connections and haven't gone over their maximum session limit. The algorithm then selects the session host with highest number of sessions. If there's a tie, the algorithm selects the first session host in the query.

IMPORTANT

The depth-first load balancing algorithm distributes sessions to session hosts based on the maximum session host limit. This parameter is required when you use the depth-first load balancing algorithm. For the best possible user experience, make sure to change the maximum session host limit parameter to a number that best suits your environment.

FSLogix profile containers and Azure files

12/6/2021 • 5 minutes to read • Edit Online

The Azure Virtual Desktop service recommends FSLogix profile containers as a user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Azure Virtual Desktop. It stores a complete user profile in a single container. At sign in, this container is dynamically attached to the computing environment using natively supported Virtual Hard Disk (VHD) and Hyper-V Virtual Hard disk (VHDX). The user profile is immediately available and appears in the system exactly like a native user profile. This article describes how FSLogix profile containers used with Azure Files function in Azure Virtual Desktop.

> **NOTE**
> If you're looking for comparison material about the different FSLogix Profile Container storage options on Azure, see Storage options for FSLogix profile containers.

User profiles

A user profile contains data elements about an individual, including configuration information like desktop settings, persistent network connections, and application settings. By default, Windows creates a local user profile that is tightly integrated with the operating system.

A remote user profile provides a partition between user data and the operating system. It allows the operating system to be replaced or changed without affecting the user data. In Remote Desktop Session Host (RDSH) and Virtual Desktop Infrastructures (VDI), the operating system may be replaced for the following reasons:

- An upgrade of the operating system
- A replacement of an existing Virtual Machine (VM)
- A user being part of a pooled (non-persistent) RDSH or VDI environment

Microsoft products operate with several technologies for remote user profiles, including these technologies:

- Roaming user profiles (RUP)
- User profile disks (UPD)
- Enterprise state roaming (ESR)

UPD and RUP are the most widely used technologies for user profiles in Remote Desktop Session Host (RDSH) and Virtual Hard Disk (VHD) environments.

Challenges with previous user profile technologies

Existing and legacy Microsoft solutions for user profiles came with various challenges. No previous solution handled all the user profile needs that come with an RDSH or VDI environment. For example, UPD cannot handle large OST files and RUP does not persist modern settings.

Functionality

The following table shows benefits and limitations of previous user profile technologies.

TECHNOLOGY	MODERN SETTINGS	WIN32 SETTINGS	OS SETTINGS	USER DATA	SUPPORTED ON SERVER SKU	BACK-END STORAGE ON AZURE	BACK-END STORAGE ON-PREMISES	VERSION SUPPORT	SUBSEQUENT SIGN IN TIME	NOTES
User Profile Disks (UPD)	Yes	Yes	Yes	Yes	Yes	No	Yes	Win 7+	Yes	
Roaming User Profile (RUP), maintenance mode	No	Yes	Yes	Yes	Yes	No	Yes	Win 7+	No	
Enterprise State Roaming (ESR)	Yes	No	Yes	No	See notes	Yes	No	Win 10	No	Functions on server SKU but no supporting user interface
User Experience Virtualization (UE-V)	Yes	Yes	Yes	No	Yes	No	Yes	Win 7+	No	

TECHNOLOGY	MODERN SETTINGS	WIN32 SETTINGS	OS SETTINGS	USER DATA	SUPPORTED ON SERVER SKU	BACK-END STORAGE ON AZURE	BACK-END STORAGE ON-PREMISES	VERSION SUPPORT	SUBSEQUENT SIGN IN TIME	NOTES
OneDrive cloud files	No	No	No	Yes	See notes	See notes	See Notes	Win 10 RS3	No	Not tested on server SKU. Back-end storage on Azure depends on sync client. Back-end storage on-prem needs a sync client.

Performance

UPD requires Storage Spaces Direct (S2D) to address performance requirements. UPD uses Server Message Block (SMB) protocol. It copies the profile to the VM in which the user is being logged.

Cost

While S2D clusters achieve the necessary performance, the cost is expensive for enterprise customers, but especially expensive for small and medium business (SMB) customers. For this solution, businesses pay for storage disks, along with the cost of the VMs that use the disks for a share.

Administrative overhead

S2D clusters require an operating system that is patched, updated, and maintained in a secure state. These processes and the complexity of setting up S2D disaster recovery make S2D feasible only for enterprises with a dedicated IT staff.

FSLogix profile containers

On November 19, 2018, Microsoft acquired FSLogix. FSLogix addresses many profile container challenges. Key among them are:

- **Performance:** The FSLogix profile containers are high performance and resolve performance issues that have historically blocked cached exchange mode.
- **OneDrive:** Without FSLogix profile containers, OneDrive for Business is not supported in non-persistent RDSH or VDI environments. The OneDrive VDI support page will tell you how they interact. For more information, see Use the sync client on virtual desktops.
- **Additional folders:** FSLogix provides the ability to extend user profiles to include additional folders.

Since the acquisition, Microsoft started replacing existing user profile solutions, like UPD, with FSLogix profile containers.

Azure Files integration with Azure Active Directory Domain Service

FSLogix profile containers' performance and features take advantage of the cloud. On August 7th, 2019, Microsoft Azure Files announced the general availability of Azure Files authentication with Azure Active Directory Domain Service (Azure AD DS). By addressing both cost and administrative overhead, Azure Files with Azure AD DS Authentication is a premium solution for user profiles in the Azure Virtual Desktop service.

Best practices for Azure Virtual Desktop

Azure Virtual Desktop offers full control over size, type, and count of VMs that are being used by customers. For more information, see What is Azure Virtual Desktop?.

To ensure your Azure Virtual Desktop environment follows best practices:

- Azure Files storage account must be in the same region as the session host VMs.
- Azure Files permissions should match permissions described in Requirements - Profile Containers.
- Each host pool VM must be built of the same type and size VM based on the same master image.
- Each host pool VM must be in the same resource group to aid management, scaling and updating.
- For optimal performance, the storage solution and the FSLogix profile container should be in the same data center location.
- The storage account containing the master image must be in the same region and subscription where the VMs are being provisioned.

Next steps

Use the following guides to set up a Azure Virtual Desktop environment.

- To start building out your desktop virtualization solution, see Create a tenant in Azure Virtual Desktop.
- To create a host pool within your Azure Virtual Desktop tenant, see Create a host pool with Azure Marketplace.
- To set up fully managed file shares in the cloud, see Set up Azure Files share.
- To configure FSLogix profile containers, see Create a profile container for a host pool using a file share.
- To assign users to a host pool, see Manage app groups for Azure Virtual Desktop.
- To access your Azure Virtual Desktop resources from a web browser, see Connect to Azure Virtual Desktop.

Storage options for FSLogix profile containers in Azure Virtual Desktop

12/6/2021 • 4 minutes to read • Edit Online

Azure offers multiple storage solutions that you can use to store your FSLogix profile container. This article compares storage solutions that Azure offers for Azure Virtual Desktop FSLogix user profile containers. We recommend storing FSLogix profile containers on Azure Files for most of our customers.

Azure Virtual Desktop offers FSLogix profile containers as the recommended user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Azure Virtual Desktop. At sign-in, this container is dynamically attached to the computing environment using a natively supported Virtual Hard Disk (VHD) and a Hyper-V Virtual Hard Disk (VHDX). The user profile is immediately available and appears in the system exactly like a native user profile.

The following tables compare the storage solutions Azure Storage offers for Azure Virtual Desktop FSLogix profile container user profiles.

Azure platform details

FEATURES	AZURE FILES	AZURE NETAPP FILES	STORAGE SPACES DIRECT
Use case	General purpose	Ultra performance or migration from NetApp on-premises	Cross-platform
Platform service	Yes, Azure-native solution	Yes, Azure-native solution	No, self-managed
Regional availability	All regions	Select regions	All regions
Redundancy	Locally redundant/zone-redundant/geo-redundant/geo-zone-redundant	Locally redundant	Locally redundant/zone-redundant/geo-redundant
Tiers and performance	Standard (Transaction optimized) Premium Up to max 100K IOPS per share with 10 GBps per share at about 3 ms latency	Standard Premium Ultra Up to 4.5GBps per volume at about 1 ms latency. For IOPS and performance details, see Azure NetApp Files performance considerations and the FAQ.	Standard HDD: up to 500 IOPS per-disk limits Standard SSD: up to 4k IOPS per-disk limits Premium SSD: up to 20k IOPS per-disk limits We recommend Premium disks for Storage Spaces Direct
Capacity	100 TiB per share, Up to 5 PiB per general purpose account	100 TiB per volume, up to 12.5 PiB per subscription	Maximum 32 TiB per disk
Required infrastructure	Minimum share size 1 GiB	Minimum capacity pool 4 TiB, min volume size 100 GiB	Two VMs on Azure IaaS (+ Cloud Witness) or at least three VMs without and costs for disks

FEATURES	AZURE FILES	AZURE NETAPP FILES	STORAGE SPACES DIRECT
Protocols	SMB 3.0/2.1, NFSv4.1 (preview), REST	NFSv3, NFSv4.1 (preview), SMB 3.x/2.x	NFSv3, NFSv4.1, SMB 3.1

Azure management details

FEATURES	AZURE FILES	AZURE NETAPP FILES	STORAGE SPACES DIRECT
Access	Cloud, on-premises and hybrid (Azure file sync)	Cloud, on-premises (via ExpressRoute)	Cloud, on-premises
Backup	Azure backup snapshot integration	Azure NetApp Files snapshots	Azure backup snapshot integration
Security and compliance	All Azure supported certificates	ISO completed	All Azure supported certificates
Azure Active Directory integration	Native Active Directory and Azure Active Directory Domain Services	Azure Active Directory Domain Services and Native Active Directory	Native Active Directory or Azure Active Directory Domain Services support only

Once you've chosen your storage method, check out Azure Virtual Desktop pricing for information about our pricing plans.

Azure Files tiers

Azure Files offers two different tiers of storage: premium and standard. These tiers let you tailor the performance and cost of your file shares to meet your scenario's requirements.

- Premium file shares are backed by solid-state drives (SSDs) and are deployed in the FileStorage storage account type. Premium file shares provide consistent high performance and low latency for input and output (IO) intensive workloads.
- Standard file shares are backed by hard disk drives (HDDs) and are deployed in the general purpose version 2 (GPv2) storage account type. Standard file shares provide reliable performance for IO workloads that are less sensitive to performance variability, such as general-purpose file shares and dev/test environments. Standard file shares are only available in a pay-as-you-go billing model.

The following table lists our recommendations for which performance tier to use based on your workload. These recommendations will help you select the performance tier that meets your performance targets, budget, and regional considerations. We've based these recommendations on the example scenarios from Remote Desktop workload types.

WORKLOAD TYPE	RECOMMENDED FILE TIER
Light (fewer than 200 users)	Standard file shares
Light (more than 200 users)	Premium file shares or standard with multiple file shares
Medium	Premium file shares
Heavy	Premium file shares

WORKLOAD TYPE	RECOMMENDED FILE TIER
Power	Premium file shares

For more information about Azure Files performance, see File share and file scale targets. For more information about pricing, see Azure Files pricing.

Next steps

To learn more about FSLogix profile containers, user profile disks, and other user profile technologies, see the table in FSLogix profile containers and Azure files.

If you're ready to create your own FSLogix profile containers, get started with one of these tutorials:

- Getting started with FSLogix profile containers on Azure Files in Azure Virtual Desktop
- Create an FSLogix profile container for a host pool using Azure NetApp files
- The instructions in Deploy a two-node Storage Spaces Direct scale-out file server for UPD storage in Azure also apply when you use an FSLogix profile container instead of a user profile disk

You can also start from the very beginning and set up your own Azure Virtual Desktop solution at Create a tenant in Azure Virtual Desktop.

What is MSIX app attach?

12/6/2021 • 2 minutes to read • Edit Online

MSIX is a new packaging format that offers many features aimed to improve packaging experience for all Windows apps. To learn more about MSIX, see the MSIX overview.

MSIX app attach is a way to deliver MSIX applications to both physical and virtual machines. However, MSIX app attach is different from regular MSIX because it's made especially for Azure Virtual Desktop. This article will describe what MSIX app attach is and what it can do for you.

Application delivery options in Azure Virtual Desktop

You can deliver apps in Azure Virtual Desktop through one of the following methods:

- Put apps in a master image
- Use tools like SCCM or Intune for central management
- Dynamic app provisioning (AppV, VMware AppVolumes, or Citrix AppLayering)
- Create custom tools or scripts using Microsoft and a third-party tool

What does MSIX app attach do?

In a Azure Virtual Desktop deployment, MSIX app attach can:

- Create separation between user data, the OS, and apps by using MSIX containers.
- Remove the need for repackaging when delivering applications dynamically.
- Reduce the time it takes for a user to sign in.
- Reduce infrastructure requirements and cost.

MSIX app attach must be applicable outside of VDI or SBC.

Traditional app layering compared to MSIX app attach

The following table compares key feature of MSIX app attach and app layering.

FEATURE	TRADITIONAL APP LAYERING	MSIX APP ATTACH
Format	Different app layering technologies require different proprietary formats.	Works with the native MSIX packaging format.
Repackaging overhead	Proprietary formats require sequencing and repackaging per update.	Apps published as MSIX don't require repackaging. However, if the MSIX package isn't available, repackaging overhead still applies.
Ecosystem	N/A (for example, vendors don't ship App-V)	MSIX is Microsoft's mainstream technology that key ISV partners and in-house apps like Office are adopting. You can use MSIX on both virtual desktops and physical Windows computers.

FEATURE	TRADITIONAL APP LAYERING	MSIX APP ATTACH
Infrastructure	Additional infrastructure required (servers, clients, and so on)	Storage only
Administration	Requires maintenance and update	Simplifies app updates
User experience	Impacts user sign-in time. Boundary exists between OS state, app state, and user data.	Delivered apps are indistinguishable from locally installed applications.

Next steps

If you want to learn more about MSIX app attach, check out our glossary and FAQ. Otherwise, get started with Set up app attach.

MSIX app attach glossary

12/6/2021 • 4 minutes to read • Edit Online

This article is a list of definitions for key terms and concepts related to MSIX app attach.

MSIX container

An MSIX container is where MSIX apps are run. To learn more, see MSIX containers.

MSIX application

An application stored in an .MSIX file.

MSIX package

An MSIX package is an MSIX file or application.

MSIX share

An MSIX share is a network share that holds expanded MSIX packages. MSIX shares must support SMB 3 or later. The shares must also be accessible to the Virtual Machines (VM) in the host pool system account. MSIX packages get staged from the MSIX share without having to move application files to the system drive.

MSIX image

An MSIX image is a VHD, VHDx, or CIM file that contains one or more MSIX packaged applications. Each application is delivered in the MSIX image using the MSIXMGR tool.

Repackage

Repackaging takes a non-MSIX application and converts it into MSIX using the MSIX Packaging Tool (MPT). For more information, see MSIX Packaging Tool overview.

Expand an MSIX package

Expanding an MSIX package is a multi-step process. Expansion takes the MSIX file and puts its content into a VHD(x) or CIM file.

To expand an MSIX package:

1. Get an MSIX package (MSIX file).
2. Rename the MSIX file to a .zip file.
3. Unzip the resulting .zip file in a folder.
4. Create a VHD that's the same size as the folder.
5. Mount the VHD.
6. Initialize a disk.
7. Create a partition.
8. Format the partition.
9. Copy the unzipped content into the VHD.
10. Use the MSIXMGR tool to apply ACLs on the content of the VHD.

11. Unmount the VHD(x) or CIM.

Upload an MSIX package

Uploading an MSIX package involves uploading the VHD(x) or CIM that contains an expanded MSIX package to the MSIX share.

In Azure Virtual Desktop, uploads happen once per MSIX share. Once you upload a package, all host pools in the same subscription can reference it.

Add an MSIX package

In Azure Virtual Desktop, adding an MSIX package links it to a host pool.

Publish an MSIX package

In Azure Virtual Desktop, a published MSIX package must be assigned to an Active Directory Domain Service (AD DS) or Azure Active Directory (Azure AD) user or user group.

Staging

Staging involves two things:

- Mounting the VHD(x) or CIM to the VM.
- Notifying the OS that the MSIX package is available for registration.

Registration

Registration makes a staged MSIX package available for your users. Registering is on a per-user basis. If you haven't explicitly registered an app for that specific user, they won't be able to run the app.

There are two types of registration: regular and delayed.

Regular registration

In regular registration, each application assigned to a user is fully registered. Registration happens during the time the user signs in to the session, which might impact the time it takes for them to start using Azure Virtual Desktop.

Delayed registration

In delayed registration, each application assigned to the user is only partially registered. Partial registration means that the Start menu tile and double-click file associations are registered. Registration happens while the user signs in to their session, so it has minimal impact on the time it takes to start using Azure Virtual Desktop. Registration completes only when the user runs the application in the MSIX package.

Delayed registration is currently the default configuration for MSIX app attach.

Deregistration

Deregistration removes a registered but non-running MSIX package for a user. Deregistration happens while the user signs out of their session. During deregistration, MSIX app attach pushes application data specific to the user to the local user profile.

Destage

Destaging notifies the OS that an MSIX package or application that currently isn't running and isn't staged for

any user can be unmounted. This removes all reference to it in the OS.

CIM

.CIM is a new file extension associated with Composite Image Files System (CimFS). Mounting and unmounting CIM files is faster that VHD files. CIM also consumes less CPU and memory than VHD.

A CIM file is a file with a .CIM extension that contains metadata and at least two additional files that contain actual data. The files within the CIM file don't have extensions. The following table is a list of example files you'd find inside a CIM:

FILE NAME	EXTENSION	SIZE
VSC	CIM	1 KB
objectid_b5742e0b-1b98-40b3-94a6-9cb96f497e56_0	NA	27 KB
objectid_b5742e0b-1b98-40b3-94a6-9cb96f497e56_1	NA	20 KB
objectid_b5742e0b-1b98-40b3-94a6-9cb96f497e56_2	NA	42 KB
region_b5742e0b-1b98-40b3-94a6-9cb96f497e56_0	NA	428 KB
region_b5742e0b-1b98-40b3-94a6-9cb96f497e56_1	NA	217 KB
region_b5742e0b-1b98-40b3-94a6-9cb96f497e56_2	NA	264,132 KB

The following table is a performance comparison between VHD and CimFS. These numbers were the result of a test run with five hundred 300 MB files in each format run on a DSv4 machine.

SPECS	VHD	CIMFS
Average mount time	356 ms	255 ms
Average unmount time	1615 ms	36 ms
Memory consumption	6% (of 8 GB)	2% (of 8 GB)
CPU (count spike)	Maxed out multiple times	No impact

Next steps

If you want to learn more about MSIX app attach, check out our overview and FAQ. Otherwise, get started with Set up app attach.

Azure Monitor for Azure Virtual Desktop glossary

12/6/2021 • 7 minutes to read • Edit Online

This article lists and briefly describes key terms and concepts related to Azure Monitor for Azure Virtual Desktop (preview).

Alerts

Any active Azure Monitor alerts that you've configured on the subscription and classified as severity 0 will appear in the Overview page. To learn how to set up alerts, see Azure Monitor Log Alerts.

Available sessions

Available sessions shows the number of available sessions in the host pool. The service calculates this number by multiplying the number of virtual machines (VMs) by the maximum number of sessions allowed per virtual machine, then subtracting the total sessions.

Connection success

This item shows connection health. "Connection success" means that the connection could reach the host, as confirmed by the stack on that virtual machine. A failed connection means that the connection couldn't reach the host.

Daily active users (DAU)

The total number of users that have started a session in the last 24 hours.

Daily alerts

The total number of alerts triggered each day.

Daily connections and reconnections

The total number of connections and reconnections started or completed within the last 24 hours.

Daily connected hours

The total number of hours spent connected to a session across users in the last 24 hours.

Diagnostics and errors

When an error or alert appears in Azure Monitor for Azure Virtual Desktop, it's categorized by three things:

- Activity type: this category is how the error is categorized by Azure Virtual Desktop diagnostics. The categories are management activities, feeds, connections, host registrations, errors, and checkpoints. Learn more about these categories at Use Log Analytics for the diagnostics feature.
- Kind: this category shows the error's location.
    - Errors marked as "service" or "ServiceError = TRUE" happened in the Azure Virtual Desktop service.
    - Errors marked as "deployment" or tagged "ServiceError = FALSE" happened outside of the Azure

Virtual Desktop service.
    - To learn more about the ServiceError tag, see Common error scenarios.
- Source: this category gives a more specific description of where the error happened.
    - Diagnostics: the service role responsible for monitoring and reporting service activity to let users observe and diagnose deployment issues.
    - RDBroker: the service role responsible for orchestrating deployment activities, maintaining the state of objects, validating authentication, and more.
    - RDGateway: the service role responsible for handling network connectivity between end-users and virtual machines.
    - RDStack: a software component that's installed on your VMs to allow them to communicate with the Azure Virtual Desktop service.
    - Client: software running on the end-user machine that provides the interface to the Azure Virtual Desktop service. It displays the list of published resources and hosts the Remote Desktop connection once you've made a selection.

Each diagnostics issue or error includes a message that explains what went wrong. To learn more about troubleshooting errors, see Identify and diagnose Azure Virtual Desktop issues.

Input delay

"Input delay" in Azure Monitor for Azure Virtual Desktop means the input delay per process performance counter for each session. In the host performance page at aka.ms/azmonwvdi, this performance counter is configured to send a report to the service once every 30 seconds. These 30-second intervals are called "samples," and the report the worst case in that window. The median and p95 values reflect the median and 95th percentile across all samples.

Under **Input delay by host**, you can select a session host row to filter all other visuals in the page to that host. You can also select a process name to filter the median input delay over time chart.

We put delays in the following categories:

- Good: below 150 milliseconds.
- Acceptable: 150-500 milliseconds.
- Poor: 500-2,000 milliseconds (below 2 seconds).
- Bad: over 2,000 milliseconds (2 seconds and up).

To learn more about how the input delay counter works, see User Input Delay performance counters.

Monthly active users (MAU)

The total number of users that have started a session in the last 28 days. If you store data for 30 days or less, you may see lower-than-expected MAU and Connection values during periods where you have fewer than 28 days of data available.

Performance counters

Performance counters show the performance of hardware components, operating systems, and applications.

The following table lists the recommended performance counters and time intervals that Azure Monitor uses for Azure Virtual Desktop:

PERFORMANCE COUNTER NAME	TIME INTERVAL
Logical Disk(C:)\Avg. Disk Queue Length	30 seconds
Logical Disk(C:)\Avg. Disk sec/Transfer	60 seconds
Logical Disk(C:)\Current Disk Queue Length	30 seconds
Memory(*)\Available Mbytes	30 seconds
Memory(*)\Page Faults/sec	30 seconds
Memory(*)\Pages/sec	30 seconds
Memory(*)\% Committed Bytes in Use	30 seconds
PhysicalDisk(*)\Avg. Disk Queue Length	30 seconds
PhysicalDisk(*)\Avg. Disk sec/Read	30 seconds
PhysicalDisk(*)\Avg. Disk sec/Transfer	30 seconds
PhysicalDisk(*)\Avg. Disk sec/Write	30 seconds
Processor Information(_Total)\% Processor Time	30 seconds
Terminal Services(*)\Active Sessions	60 seconds
Terminal Services(*)\Inactive Sessions	60 seconds
Terminal Services(*)\Total Sessions	60 seconds
*User Input Delay per Process(*)\Max Input Delay	30 seconds
*User Input Delay per Session(*)\Max Input Delay	30 seconds
RemoteFX Network(*)\Current TCP RTT	30 seconds
RemoteFX Network(*)\Current UDP Bandwidth	30 seconds

To learn more about how to read performance counters, see Configuring performance counters.

To learn more about input delay performance counters, see User Input Delay performance counters.

Potential connectivity issues

Potential connectivity issues shows the hosts, users, published resources, and clients with a high connection failure rate. Once you choose a "report by" filter, you can evaluate the issue's severity by checking the values in these columns:

- Attempts (number of connection attempts)
- Resources (number of published apps or desktops)
- Hosts (number of VMs)

- Clients

For example, if you select the **By user** filter, you can check to see each user's connection attempts in the **Attempts** column.

If you notice that a connection issue spans multiple hosts, users, resources, or clients, it's likely that the issue affects the whole system. If it doesn't, it's a smaller issue that lower priority.

You can also select entries to view additional information. You can view which hosts, resources, and client versions were involved with the issue. The display will also show any errors reported during the connection attempts.

Round-trip time (RTT)

Round-trip time (RTT) is an estimate of the connection's round-trip time between the end-user's location and the session host's Azure region. To see which locations have the best latency, look up your desired location in the Azure Virtual Desktop Experience Estimator tool.

Session history

The **Sessions** item shows the status of all sessions, connected and disconnected. **Idle sessions** only shows the disconnected sessions.

Severity 0 alerts

The most urgent items that you need to take care of right away. If you don't address these issues, they could cause your Azure Virtual Desktop deployment to stop working.

Time to connect

Time to connect is the time between when a user starts their session and when they're counted as being signed in to the service. Establishing new connections tends to take longer than reestablishing existing connections.

User report

The user report page lets you view a specific user's connection history and diagnostic information. Each user report shows usage patterns, user feedback, and any errors users have encountered during their sessions. Most smaller issues can be resolved with user feedback. If you need to dig deeper, you can also filter information about a specific connection ID or period of time.

Users per core

This is the number of users in each virtual machine core. Tracking the maximum number of users per core over time can help you identify whether the environment consistently runs at a high, low, or fluctuating number of users per core. Knowing how many users are active will help you efficiently resource and scale the environment.

Windows Event Logs

Windows Event Logs are data sources collected by Log Analytics agents on Windows virtual machines. You can collect events from standard logs like System and Application as well as custom logs created by applications you need to monitor.

The following table lists the required Windows Event Logs for Azure Monitor for Azure Virtual Desktop:

EVENT NAME	EVENT TYPE
Application	Error and Warning
Microsoft-Windows-TerminalServices-RemoteConnectionManager/Admin	Error, Warning, and Information
Microsoft-Windows-TerminalServices-LocalSessionManager/Operational	Error, Warning, and Information
System	Error and Warning
Microsoft-FSLogix-Apps/Operational	Error, Warning, and Information
Microsoft-FSLogix-Apps/Admin	Error, Warning, and Information

To learn more about Windows Event Logs, see Windows Event records properties.

Next steps

- To get started, see Use Azure Monitor for Azure Virtual Desktop to monitor your deployment.
- To estimate, measure, and manage your data storage costs, see Estimate Azure Monitor costs.
- If you encounter a problem, check out our troubleshooting guide for help and known issues.

You can also set up Azure Advisor to help you figure out how to resolve or prevent common issues. Learn more at Use Azure Advisor with Azure Virtual Desktop.

If you need help or have any questions, check out our community resources:

- Ask questions or make suggestions to the community at the Azure Virtual Desktop TechCommunity.
- To learn how to leave feedback, see Troubleshooting overview, feedback, and support for Azure Virtual Desktop.
- You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub

Data locations for Azure Virtual Desktop

12/6/2021 • 3 minutes to read • Edit Online

Azure Virtual Desktop is currently available for all geographical locations. Administrators can choose the location to store user data when they create the host pool virtual machines and associated services, such as file servers. Learn more about Azure geographies at Data residency in Azure.

> **NOTE**
> Microsoft doesn't control or limit the regions where you or your users can access your user and app-specific data.

> **IMPORTANT**
> Azure Virtual Desktop stores various types of information like host pool names, app group names, workspace names, and user principal names in a datacenter. While creating any of the service objects, the customer has to enter the location where the object needs to be created. The location of this object determines where the information for the object will be stored. The customer will choose an Azure region and the related information will be stored in the associated geography. For a list of all Azure regions and related geographies, visit https://azure.microsoft.com/global-infrastructure/geographies/.

This article describes which information the Azure Virtual Desktop service stores. To learn more about the customer data definitions, see How Microsoft categorizes data for online services.

Customer input

To set up the Azure Virtual Desktop service, the customer must create host pools and other service objects. During configuration, the customer must give information like the host pool name, application group name, and so on. This information is considered customer input. Customer input is stored in the geography associated with the region the object is created in. Azure Resource Manager paths to the objects are considered organizational information, so data residency doesn't apply to them. Data about Azure Resource Manager paths will be stored outside of the chosen geography.

Customer data

The service doesn't directly store any user or app-related information, but it does store customer data like application names and user principal names because they're part of the object setup process. This information is stored in the geography associated with the region the customer created the object in.

Diagnostic data

Azure Virtual Desktop gathers service-generated diagnostic data whenever the customer or user interacts with the service. This data is only used for troubleshooting, support, and checking the health of the service in aggregate form. For example, from the session host side, when a VM registers to the service, we generate information that includes the virtual machine (VM) name, which host pool the VM belongs to, and so on. This information is stored in the geography associated with the region the host pool is created in. Also, when a user connects to the service and launches a remote desktop, we generate diagnostic information that includes the user principal name, client location, client IP address, which host pool the user is connecting to, and so on. This information is sent to two different locations:

- The location closest to the user where the service infrastructure (client traces, user traces, diagnostic data) is

present.
- The location where the host pool is located.

Service-generated data

To keep Azure Virtual Desktop reliable and scalable, we aggregate traffic patterns and usage to check the health and performance of the infrastructure control plane. For example, to understand how to ramp up regional infrastructure capacity as service usage increases, we process service usage log data. We then review the logs for peak times and decide which data centers to add to meet this capacity. We aggregate this information from all locations where the service infrastructure is, then send it to the US region. The data sent to the US region includes scrubbed data, but not customer data.

We currently support storing the aforementioned data in the following locations:

- United States (US) (generally available)
- Europe (EU) (generally available)
- United Kingdom (UK) (generally available)
- Canada (CA) (generally available)

More geographies will be added as the service grows. The stored information is encrypted at rest, and geo-redundant mirrors are maintained within the geography. Customer data, such as app settings and user data, resides in the location the customer chooses and isn't managed by the service.

The outlined data is replicated within the Azure geography for disaster recovery purposes.

Screen capture protection

12/6/2021 • 2 minutes to read • Edit Online

The screen capture protection feature prevents sensitive information from being captured on the client endpoints. When you enable this feature, remote content will be automatically blocked or hidden in screenshots and screen shares. Also, the Remote Desktop client will hide content from malicious software that may be capturing the screen.

Prerequisites

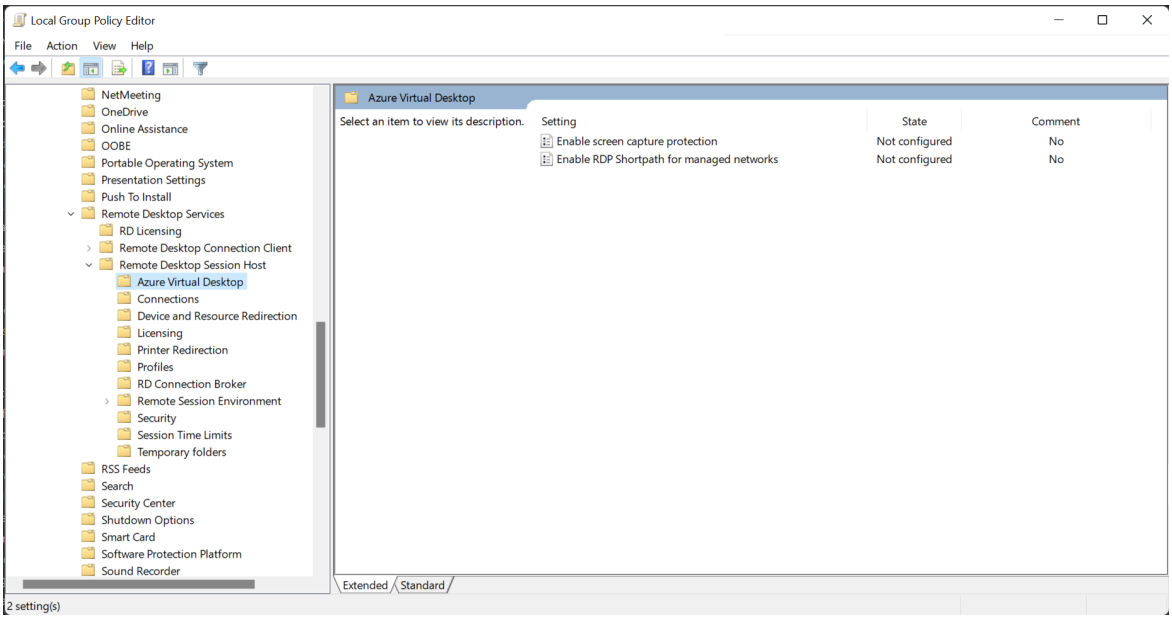
The screen capture protection feature is configured on the session host level and enforced on the client. Only clients that support this feature can connect to the remote session. Following clients currently support screen capture protection:

- Windows Desktop client supports screen capture protection for full desktops only.
- macOS client version 10.7.0 or later supports screen capture protection for both RemoteApp and full desktops.

Suppose the user attempts to use an unsupported client to connect to the protected session host. In that case, the connection will fail with error 0x1151.

Configure screen capture protection

1. To configure screen capture protection, you need to install administrative templates that add rules and settings for Azure Virtual Desktop.
2. Download the Azure Virtual Desktop policy templates file (AVDGPTemplate.cab) and extract the contents of the cab file and zip archive.
3. Copy the **terminalserver-avd.admx** file to **%windir%\PolicyDefinitions** folder.
4. Copy the **en-us\terminalserver-avd.adml** file to **%windir%\PolicyDefinitions\en-us** folder.
5. To confirm the files copied correctly, open the Group Policy Editor and navigate to **Computer Configuration** -> **Administrative Templates** -> **Windows Components** -> **Remote Desktop Services** -> **Remote Desktop Session Host** -> **Azure Virtual Desktop**
6. You should see one or more Azure Virtual Desktop policies, as shown below.



> **TIP**
>
> You can also install administrative templates to the group policy Central Store in your Active Directory domain. For more information about Central Store for Group Policy Administrative Templates, see How to create and manage the Central Store for Group Policy Administrative Templates in Windows.

7. Open the **"Enable screen capture protection"** policy and set it to **"Enabled"**.

Limitations and known issues

- This feature protects the Remote Desktop window from being captured through a specific set of public operating system features and APIs. However, there's no guarantee that this feature will strictly protect content, for example, where someone takes photography of the screen.
- Customers should use the feature together with disabling clipboard, drive, and printer redirection. Disabling redirection will help to prevent the user from copying the captured screen content from the remote session.
- Users can't share the Remote Desktop window using local collaboration software, such as Microsoft Teams, when the feature is enabled. If Microsoft Teams is used, both the local Teams app and Teams running with media optimizations can't share the protected content.

Next steps

- To learn about Azure Virtual Desktop security best practices, see Azure Virtual Desktop security best practices.

Azure Virtual Desktop for Azure Stack HCI (preview)

12/6/2021 • 3 minutes to read • Edit Online

> **IMPORTANT**
>
> Azure Virtual Desktop for Azure Stack HCI is currently in preview. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Azure Virtual Desktop for Azure Stack HCI (preview) lets you deploy Azure Virtual Desktop session hosts to your on-premises Azure Stack HCI infrastructure. You can also use Azure Virtual Desktop for Azure Stack HCI to manage your session hosts from the Azure portal. If you already have an existing on-premises Virtual Desktop Infrastructure (VDI) deployment, Azure Virtual Desktop for Azure Stack HCI can improve your administrator and end-user experience. If you're already using Azure Virtual Desktop in the cloud, you can extend your deployment to your on-premises infrastructure to better meet your performance or data locality needs.

Azure Virtual Desktop for Azure Stack HCI is currently in public preview. Azure Stack HCI doesn't currently support certain important Azure Virtual Desktop features. Because of these limitations, we don't recommend using this feature for production workloads yet.

Key benefits

We've established what Azure Virtual Desktop for Azure Stack HCI is. The question remains: what can it do for you?

With Azure Virtual Desktop for Azure Stack HCI, you can:

- Improve performance for Azure Virtual Desktop users in areas with poor connectivity to the Azure public cloud by giving them session hosts closer to their location.
- Meet data locality requirements by keeping app and user data on-premises. For more information, see Data locations for Azure Virtual Desktop.
- Improve access to legacy on-premises apps and data sources by keeping virtual desktops and apps in the same location.
- Reduce costs and improve user experience with Windows 10 and Windows 11 Enterprise multi-session virtual desktops.
- Simplify your VDI deployment and management compared to traditional on-premises VDI solutions by using the Azure portal.

Pricing

The following things affect how much it costs to run Azure Virtual Desktop for Azure Stack HCI:

- Infrastructure costs. You'll pay monthly service fees for Azure Stack HCI. Learn more at Azure Stack HCI pricing.
- User access rights for Azure Virtual Desktop. The same licenses that grant access to Azure Virtual Desktop in the cloud also apply to Azure Virtual Desktop for Azure Stack HCI. Learn more at Azure Virtual Desktop pricing.

- The Azure Virtual Desktop hybrid service fee. This fee requires you to pay for each active virtual CPU (vCPU) of Azure Virtual Desktop session hosts you're running on Azure Stack HCI. This fee will become active once the preview period ends.

Known issues and limitations

We're aware of the following issues affecting the public preview version of Azure Virtual Desktop for Azure Stack HCI:

- Azure Stack HCI host pools don't currently support the following Azure Virtual Desktop features:
  - Azure Monitor for Azure Virtual Desktop
  - Session host scaling with Azure Automation
  - Autoscale (preview)
  - Start VM on connect
  - Multimedia redirection (preview)
  - Per-user access pricing
- The Azure Virtual Desktop tab in the Azure portal can't create new virtual machines directly on Azure Stack HCI infrastructure. Instead, admins must create on-premises virtual machines separately, then register them with an Azure Virtual Desktop host pool.
- When connecting to a Windows 10 or 11 Enterprise multi-session virtual desktop, users may see activation issues, such as a desktop watermark saying "Activate Windows," even if they have an eligible license.
- Azure Virtual Desktop for Azure Stack HCI doesn't currently support host pools containing both cloud and on-premises session hosts. Each host pool in the deployment must have only one type of host pool.
- Session hosts on Azure Stack HCI don't support certain cloud-only Azure services.
- Because Azure Stack HCI supports so many types of hardware and on-premises networking capabilities that performance and user density may vary widely between session hosts running in the Azure cloud. Azure Virtual Desktop's virtual machine sizing guidelines are broad, so you should only use them for initial performance estimates.

If there are any issues you encounter during the preview that aren't on this list, we encourage you to report them.

Next steps

Now that you're familiar with Azure Virtual Desktop for Azure Stack HCI, learn how to deploy this feature at Set up Azure Virtual Desktop for Azure Stack HCI (preview).

Tag Azure Virtual Desktop resources to manage costs

12/6/2021 • 7 minutes to read • Edit Online

Tagging is a tool available across Azure services that helps you organize resources inside their Azure subscription. Organizing resources makes it easier to track costs across multiple services. Tags also help you understand how much each grouping of Azure resources costs per billing cycle. If you'd like to learn more about tagging in general, see Use tags to organize your Azure resources and management hierarchy. You can also watch a quick video about some other ways to use Azure tags.

How tagging works

You can tag Azure services you manage in the Azure portal or through PowerShell. The tags will appear as key-value pairs of text. As you use tagged Azure resources, the associated tag key-value pair will be attached to the resource usage.

Once your deployment reports tagged usage information to Azure Cost Management, you can use your tagging structure to filter cost data. To learn how to filter by tags in Azure Cost Management, see Quickstart: Explore and analyze costs with cost analysis.

Add, edit, or delete tags

When you apply a new tag to a resource, it won't be visible in Azure Cost Management until its associated Azure resource reports activity. If you apply an existing tag to your resources, this change also won't be visible in Azure Cost Management until the Azure resources report activity.

If you edit a tag name, the associated resources will now associate costs with its new key-value pair. You can still filter data with the old tag, but all new data from after the change will be reported with the new tag.

If you delete a tag, Azure Virtual Desktop will no longer report data associated with the deleted tag to Azure Cost Management. You can still filter with deleted tags for data reported before you deleted the tag.

> **IMPORTANT**
>
> Tagged Azure resources that haven't been active since you applied new or updated tags to them won't report any activity associated with the changed tags to Azure Cost Management. You won't be able to filter for specific tags until their associated activity is reported to Azure Cost Management by the service.

View all existing tags

You can view all existing tags for your Azure services by going to the Azure portal, then opening the **Tags** tab. The Tags tab will show you all tags in objects you have access to. You can also sort tags by their keys or values whenever you need to quickly update a large number of tags at the same time.

What tags can and can't do

Tags only report usage and cost data for Azure resources they're directly assigned to. If you've tagged a resource without tagging the other resources in it, then Azure Virtual Desktop will only report activity related to the top-level tagged resource. You'll also need to tag every resource under that top-level resource if you want your billing data to be accurate.

To learn more about how tags work in Azure Cost Management, see How tags are used in cost and usage data.

For a list of known Azure tag limitations, see Use tags to organize your Azure resources and management hierarchy.

Using tags in Azure Virtual Desktop

Now that you understand the basics of Azure tags, let's go over how you can use them in Azure Virtual Desktop.

You can use Azure tags to organize costs for creating, managing, and deploying virtualized experiences for your customers and users. Tagging can also help you track resources you buy directly through Azure Virtual Desktop and other Azure services connected to Azure Virtual Desktop deployments.

Suggested tags for Azure Virtual Desktop

Because Azure Virtual Desktop can work with other Azure services to support its deployments, there isn't a universal system for tagging deployment resources. What's most important is that you develop a strategy that works for you and your organization. However, we do have some suggestions that might help you, especially if you're new to using Azure.

The following suggestions apply to all Azure Virtual Desktop deployments:

- Become familiar with your purchased Azure services so you understand the extent of what you want to tag. As you learn how to use the Azure portal, keep a list of service groups and objects where you can apply tags. Some resources that you should keep track of include resource groups, virtual machines, disks, and network interface cards (NICs). For a more comprehensive list of cost generating service components you can tag, see Understanding total Azure Virtual Desktop deployment costs.
- Create a cost reporting aggregation to organize your tags. You can either follow a common tagging pattern or create a new pattern that meets your organization's needs.
- Keep your tags consistent wherever you apply them. Even the smallest typo can impact data reporting, so make sure you're adding the exact key-value pair you want to look up later.
- Keep a record of any tags you update or edit. This record will let you combine each tag's historic data as needed. When you edit or update a tag, you should also apply those changes across all resources that include the changed tag.

Suggested tags for Azure Virtual Desktop host pools

Every virtual machine in an Azure Virtual Desktop host pool creates a cost-producing construct. Because host pools are the foundation of an Azure Virtual Desktop deployment, their VMs are typically the main source of costs for Azure Virtual Desktop deployments. If you want to set up a tagging system, we recommend that you start with tagging all the host pools in your deployment to track VM compute costs. Tagging your host pools can help you use filtering in Azure Cost Management to identify these VM costs.

Like with the general suggestions, there's no universal system for tagging host pools. However, we do have a few suggestions to help you organize your host pool tags:

- Tagging host pools while you're creating them is optional, but tagging during the creation process will make it easier for you to view all tagged usage in Azure Cost Management later. Your host pool tags will follow all cost-generating components of the session hosts within your host pool. Learn more about session host-specific costs at Understanding total Azure Virtual Desktop deployment costs.
- If you use the Azure portal to create a new host pool, the creation workflow will give you the chance to add existing tags. These tags will be passed along to all resources you create during the host pool creation process. Tags will also be applied to any session hosts you add to an existing host pool in the Azure portal. However, you'll need to enter the tags manually every time you add a new session host.

- It's unlikely you'll ever get a complete cost report of every supporting Azure service working with your host pools, since configuration options are both limitless and unique to each customer. It's up to you to decide how closely you want to track costs across any Azure services associated with your Azure Virtual Desktop deployment. The more thoroughly you track these costs by tagging, the more accurate your monthly Azure Virtual Desktop cost report will become.
- If you build your tagging system around your host pools, make sure to use key-value pairs that make sense to add to other Azure services later.

Suggested tags for other Azure Virtual Desktop resources

Most Azure Virtual Desktop customers deploy other Azure services to support their deployments. If you want to include the cost of these extra services in your cost report, you should consider the following suggestions:

- If you've already purchased an Azure service or resources that you want to integrate into your Azure Virtual Desktop deployments, you have two options:
  - Separate your purchased Azure services between different Azure subscriptions.
  - Combine all purchased Azure services in the same subscription with your Azure Virtual Desktop tags.

  Separating your services will give you a clearer idea of costs for each service, but may end up being more expensive in the end. You may need to purchase extra storage for these services to make sure your Azure Virtual Desktop has its own designated storage.

  Combining your purchased services is less expensive, but may inflate your cost report because the usage data for shared resources won't be as accurate. To make up for the lack of accuracy, you can add multiple tags to your resources to see shared costs through filters that track different factors.
- If you started building your tagging system with a different Azure service, make sure the key-value pairs you create can be applied to your Azure Virtual Desktop deployment or other services later.

Next steps

If you'd like to learn more about common Azure Virtual Desktop related costs, check out Understanding total Azure Virtual Desktop deployment costs.

If you'd like to learn more about Azure tags, check out the following resources:

- Use tags to organize your Azure resources and management hierarchy
- A video explaining the value of using Azure tags
- How tags are used in cost and usage data
- Develop your naming and tagging strategy for Azure resources
- Define your tagging strategy
- Resource naming and tagging decision guide

If you'd like to learn more about Azure Cost Management, check out the following articles:

- What is Azure Cost Management + Billing?
- Quickstart: Explore and analyze costs with cost analysis

Troubleshooting overview, feedback, and support for Azure Virtual Desktop

12/6/2021 • 4 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

This article provides an overview of the issues you may encounter when setting up an Azure Virtual Desktop environment and provides ways to resolve the issues.

Troubleshoot deployment and connection issues

Azure Monitor for Windows Virtual Desktop is a dashboard built on Azure Monitor workbooks that can quickly troubleshoot and identify issues in your Windows Virtual Desktop environment for you. If you prefer working with Kusto queries, we recommend using the built-in diagnostic feature, Log Analytics, instead.

Report issues

To report issues or suggest features for Azure Virtual Desktop with Azure Resource Manager integration, visit the Azure Virtual Desktop Tech Community. You can use the Tech Community to discuss best practices or suggest and vote for new features.

When you make a post asking for help or propose a new feature, make sure you describe your topic in as much detail as possible. Detailed information can help other users answer your question or understand the feature you're proposing a vote for.

Escalation tracks

Before doing anything else, make sure to check the Azure status page and Azure Service Health to make sure your Azure service is running properly.

Use the following table to identify and resolve issues you may encounter when setting up an environment using Remote Desktop client. Once your environment's set up, you can use our new Diagnostics service to identify issues for common scenarios.

ISSUE	SUGGESTED SOLUTION
Session host pool Azure Virtual Network (VNET) and Express Route settings	Open an Azure support request, then select the appropriate service (under the Networking category).
Session host pool Virtual Machine (VM) creation when Azure Resource Manager templates provided with Azure Virtual Desktop aren't being used	Open an Azure support request, then select **Azure Virtual Desktop** for the service. For issues with the Azure Resource Manager templates that are provided with Azure Virtual Desktop, see Azure Resource Manager template errors section of Host pool creation.

ISSUE	SUGGESTED SOLUTION
Managing Azure Virtual Desktop session host environment from the Azure portal	Open an Azure support request. For management issues when using Remote Desktop Services/Azure Virtual Desktop PowerShell, see Azure Virtual Desktop PowerShell or open an Azure support request, select **Azure Virtual Desktop** for the service, select **Configuration and management** for the problem type, then select **Issues configuring environment using PowerShell** for the problem subtype.
Managing Azure Virtual Desktop configuration tied to host pools and application groups (app groups)	See Azure Virtual Desktop PowerShell, or open an Azure support request, select **Azure Virtual Desktop** for the service, then select the appropriate problem type.
Deploying and manage FSLogix Profile Containers	See Troubleshooting guide for FSLogix products and if that doesn't resolve the issue, Open an Azure support request, select **Azure Virtual Desktop** for the service, select **FSLogix** for the problem type, then select the appropriate problem subtype.
Remote desktop clients malfunction on start	See Troubleshoot the Remote Desktop client and if that doesn't resolve the issue, Open an Azure support request, select **Azure Virtual Desktop** for the service, then select **Remote Desktop clients** for the problem type. If it's a network issue, your users need to contact their network administrator.
Connected but no feed	Troubleshoot using the User connects but nothing is displayed (no feed) section of Azure Virtual Desktop service connections. If your users have been assigned to an app group, open an Azure support request, select **Azure Virtual Desktop** for the service, then select **Remote Desktop Clients** for the problem type.
Feed discovery problems due to the network	Your users need to contact their network administrator.
Connecting clients	See Azure Virtual Desktop service connections and if that doesn't solve your issue, see Session host virtual machine configuration.
Responsiveness of remote applications or desktop	If issues are tied to a specific application or product, contact the team responsible for that product.
Licensing messages or errors	If issues are tied to a specific application or product, contact the team responsible for that product.
Issues with third-party authentication methods or tools	Verify that your third-party provider supports Azure Virtual Desktop scenarios and approach them regarding any known issues.

ISSUE	SUGGESTED SOLUTION
Issues using Log Analytics for Azure Virtual Desktop	For issues with the diagnostics schema, open an Azure support request. For queries, visualization, or other issues in Log Analytics, select the appropriate problem type under Log Analytics.
Issues using Microsoft 365 apps	Contact the Microsoft 365 admin center with one of the Microsoft 365 admin center help options.

Next steps

- To troubleshoot issues while creating a host pool in an Azure Virtual Desktop environment, see host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues with Azure Virtual Desktop client connections, see Azure Virtual Desktop service connections.
- To troubleshoot issues with Remote Desktop clients, see Troubleshoot the Remote Desktop client
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To learn more about the service, see Azure Virtual Desktop environment.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.
- To learn about auditing actions, see Audit operations with Resource Manager.
- To learn about actions to determine errors during deployment, see View deployment operations.

Troubleshoot the Azure Virtual Desktop getting started feature

12/6/2021 • 7 minutes to read • Edit Online

The Azure Virtual Desktop getting started feature uses nested templates to deploy Azure resources for validation and automation in Azure Virtual Desktop. The getting started feature creates either two or three resource groups based on whether the subscription it's running on has existing Active Directory Domain Services (AD DS) or Azure Active Directory Domain Services (Azure AD DS) or not. All resource groups start with the same user-defined prefix.

When you run the nested templates, they create three resource groups and a template that provisions Azure Resource Manager resources. The following lists show each resource group and the templates they run.

The resource group that ends in "-deployment" runs these templates:

- easy-button-roleassignment-job-linked-template
- easy-button-prerequisitecompletion-job-linked-template
- easy-button-prerequisite-job-linked-template
- easy-button-inputvalidation-job-linked-template
- easy-button-deploymentResources-linked-template
- easy-button-prerequisite-user-setup-linked-template

> **NOTE**
> The easy-button-prerequisite-user-setup-linked-template is optional and will only appear if you created a validation user.

The resource group that ends in "-wvd" runs these templates:

- NSG-linkedTemplate
- vmCreation-linkedTemplate
- Workspace-linkedTemplate
- wvd-resources-linked-template
- easy-button-wvdsetup-linked-template

The resource group that ends in "-prerequisite" runs these templates:

- easy-button-prerequisite-resources-linked-template

> **NOTE**
> This resource group is optional, and will only appear if your subscription doesn't have Azure AD DS or AD DS.

No subscriptions

In this issue, you see an error message that says "no subscriptions" when opening the getting started feature. This happens when you try to open the feature without an active Azure subscription.

To fix this issue, check to see if your subscription or the affected user has an active Azure subscription. If they don't, assign the user the Owner Role-based Access Control (RBAC) role on their subscription.

You don’t have permissions

This issue happens when you open the getting started feature and get an error message that says, "You don't have permissions." This message appears when the user running the feature doesn't have Owner permissions on their active Azure subscription.

To fix this issue, sign in with an Azure account that has Owner permissions, then assign the Owner RBAC role to the affected account.

Fields under Virtual Machine tab are grayed out

This issue happens when you open the **Virtual machine** tab and see that the fields under "Do you want users to share this machine?" are grayed out. This issue then prevents you from changing the image type, selecting an image to use, or changing the VM size.

This issue happens when you run the feature with a prefix that was already used to start a deployment. When the feature creates a deployment, it creates an object to represent the deployment in Azure. Certain values in the object, like the image, become attached to that object to prevent multiple objects from using the same images.

To fix this issue, you can either delete all resource groups with the existing prefix or use a new prefix.

Username must not include reserved words

This issue happens when the getting started feature won't accept the new username you enter into the field.

This error message appears because Azure doesn't allow certain words in usernames for public endpoints. For a full list of blocked words, see Resolve reserved resource name errors.

To resolve this issue, either try a new word or add letters to the blocked word to make it unique. For example, if the word "admin" is blocked, try using "AVDadmin" instead.

The value must be between 12 and 72 characters long

This error message appears when entering a password that is either too long or too short to meet the character length requirement. Azure password length and complexity requirements even apply to fields that you later use in Windows, which has less strict requirements.

To resolve this issue, make sure you use an account that follows Microsoft's password guidelines or uses Azure AD Password Protection.

Error messages for easy-button-prerequisite-user-setup-linked-template

If the AD DS VM you're using already has an extension named Microsoft.Powershell.DSC associated with it, you'll see an error message that looks like this:

```
"error": {
        "code": "DeploymentFailed",
        "message": "At least one resource deployment operation failed. Please list deployment operations for
details. Please see https://aka.ms/DeployOperations for usage details.",
        "details": [
            {
                "code": "Conflict",
                "message": "{\r\n  \"status\": \"Failed\",\r\n  \"error\": {\r\n    \"code\":
\"ResourceDeploymentFailure\",\r\n    \"message\": \"The resource operation completed with terminal
provisioning state 'Failed'.\",\r\n    \"details\": [\r\n      {\r\n        \"code\":
\"VMExtensionProvisioningError\",\r\n        \"message\": \"VM has reported a failure when processing
extension 'Microsoft.Powershell.DSC'. Error message: \\\"DSC Configuration 'AddADDSUser' completed with
error(s). Following are the first few: PowerShell DSC resource MSFT_ScriptResource  failed to execute Set-
TargetResource functionality with error message: Some error occurred in DSC CreateUser SetScript:
\\r\\n\\r\\nException             : Microsoft.ActiveDirectory.Management.ADIdentityNotFoundException: Cannot
find an object with \\r\\n                        identity: 'Adam S' under:
'DC=GT090617,DC=onmicrosoft,DC=com'.\\r\\n                           at
Microsoft.ActiveDirectory.Management.Commands.ADFactoryUtil.GetObjectFromIdentitySearcher(\\r\\n
ADObjectSearcher searcher, ADEntity identityObj, String searchRoot, AttributeSetRequest attrs, \\r\\n
CmdletSessionInfo cmdletSessionInfo, String[]& warningMessages)\\r\\n                           at \\r\\n
Microsoft.ActiveDirectory.Management.Commands.ADFactory`1.GetDirectoryObjectFromIdentity(T \\r\\n
identityObj, String searchRoot, Boolean showDeleted)\\r\\n                           at \\r\\n
Microsoft.ActiveDirectory.Management.Commands.SetADGroupMember`1.ValidateMembersParameter()\\r\\nTargetObjec
t          : Adam S\\r\\nCategoryInfo          : ObjectNotFound: (Adam S:ADPrincipal) [Add-ADGroupMember],
ADIdentityNotFoundException\\r\\nFullyQualifiedErrorId :
SetADGroupMember.ValidateMembersParameter,Microsoft.ActiveDirectory.Management.Commands.AddADGro\\r\\n
upMember\\r\\nErrorDetails          : \\r\\nInvocationInfo        :
System.Management.Automation.InvocationInfo\\r\\nScriptStackTrace      : at <ScriptBlock>,
C:\\\\Packages\\\\Plugins\\\\Microsoft.Powershell.DSC\\\\2.83.1.0\\\\DSCWork\\\\DSCADUserCreatio\\r\\n
nScripts_2020-04-28.2\\\\Script-CreateADDSUser.ps1: line 98\\r\\n                        at <ScriptBlock>,
<No file>: line 8\\r\\n                        at ScriptExecutionHelper,
C:\\\\Windows\\\\system32\\\\WindowsPowerShell\\\\v1.0\\\\Modules\\\\PSDesiredStateConfi\\r\\n
guration\\\\DscResources\\\\MSFT_ScriptResource\\\\MSFT_ScriptResource.psm1: line 270\\r\\n
at Set-TargetResource,
C:\\\\Windows\\\\system32\\\\WindowsPowerShell\\\\v1.0\\\\Modules\\\\PSDesiredStateConfigur\\r\\n
ation\\\\DscResources\\\\MSFT_ScriptResource\\\\MSFT_ScriptResource.psm1: line
144\\r\\nPipelineIterationInfo : {}\\r\\nPSMessageDetails      : \\r\\n\\r\\n\\r\\n\\r\\n  The
SendConfigurationApply function did not succeed.\\\"\\r\\n\\r\\nMore information on troubleshooting is
available at https://aka.ms/VMExtensionDSCWindowsTroubleshoot \"\r\n      }\r\n    ]\r\n  }\r\n}"
            }
        ]
    }
```


To resolve this issue, uninstall the Microsoft.Powershell.DSC extension, then run the getting started feature again.

Error messages for easy-button-prerequisite-job-linked-template

If you see an error message like this, that means the resource operation for the easy-button-prerequisite-job-linked-template template didn't complete successfully:

```
{
    "status": "Failed",
    "error": {
        "code": "DeploymentFailed",
        "message": "At least one resource deployment operation failed. Please list deployment operations for
details. Please see https://aka.ms/DeployOperations for usage details.",
        "details": [
            {
                "code": "Conflict",
                "message": "{\r\n  \"status\": \"Failed\",\r\n  \"error\": {\r\n    \"code\":
\"ResourceDeploymentFailure\",\r\n    \"message\": \"The resource operation completed with terminal
provisioning state 'Failed'.\"\r\n  }\r\n}"
            }
        ]
    }
}
```


To make sure this is the issue you're dealing with:

1. Select **easy-button-prerequisite-job-linked-template**, then select **Ok** on the error message window that pops up.
2. Go to **<prefix>-deployment resource group** and select **resourceSetupRunbook**.
3. Select the status, which should say **Failed**.
4. Select the **Exception** tab. You should see an error message that looks like this:

   ```
   The running command stopped because the preference variable "ErrorActionPreference" or common
   parameter is set to Stop: Error while creating and adding validation user <your-username-here> to
   group <your-resource-group-here>
   ```

There currently isn't a way to fix this issue permanently. As a workaround, run The Azure Virtual Desktop getting started feature again, but this time don't create a validation user. After that, create your new users with the manual process only.

Validate that the domain administrator UPN exists for a new profile

To check if the UPN address is causing the issue with the template:

1. Select **easy-button-prerequisite-job-linked-template** and then on the failed step. Confirm the error message.
2. Navigate to the **<prefix>-deployment resource group** and click on the **resourceSetupRunbook**.
3. Select the status, which should say **Failed**.
4. Select the **Output** tab.

If the UPN exists on your new subscription, there are two potential causes for the issue:

- The getting started feature didn't create the domain administrator profile, because the user already exists. To resolve this, run the getting started feature again, but this time enter a username that doesn't already exist in your identity provider.
- The getting started feature didn't create the validation user profile. To resolve this issue, run the getting started feature again, but this time don't create any validation users. After that, create new users with the manual process only.

Error messages for easy-button-inputvalidation-job-linked-template

If there's an issue with the easy-button-inputvalidation-job-linked-template template, you'll see an error message that looks like this:

```
{
    "status": "Failed",
    "error": {
        "code": "ResourceDeploymentFailure",
        "message": "The resource operation completed with terminal provisioning state 'Failed'."
    }
}
```

To make sure this is the issue you've encountered:

1. Open the <prefix>-deployment resource group and look for **inputValidationRunbook**.
2. Under recent jobs there will be a job with failed status. Click on **Failed**.
3. In the **job details** window, select **Exception**.

This error happens when the Azure admin UPN you entered isn't correct. To resolve this issue, make sure you're entering the correct username and password, then try again.

Multiple VMExtensions per handler not supported

When you run the getting started feature on a subscription that has Azure AD DS or AD DS, then the feature will use a Microsoft.Powershell.DSC extension to create validation users and configure FSLogix. However, Windows VMs in Azure can't run more than one of the same type of extension at the same time.

If you try to run multiple versions of Microsoft.Powershell.DSC, you'll get an error message that looks like this:

```
{
    "status": "Failed",
    "error": {
        "code": "BadRequest",
        "message": "Multiple VMExtensions per handler not supported for OS type 'Windows'. VMExtension
'Microsoft.Powershell.DSC' with handler 'Microsoft.Powershell.DSC' already added or specified in input."
    }
}
```

To resolve this issue, before you run the getting started feature, make sure to remove any currently running instance of Microsoft.Powershell.DSC from the domain controller VM.

Failure in easy-button-prerequisitecompletion-job-linked-template

The user group for the validation users is located in the "USERS" container. However, the user group must be synced to Azure AD in order to work properly. If it isn't, you'll get an error message that looks like this:

```
{
    "status": "Failed",
    "error": {
        "code": "ResourceDeploymentFailure",
        "message": "The resource operation completed with terminal provisioning state ‘Failed’."
    }
}
```

To make sure the issue is caused by the validation user group not syncing, open the <prefix>-prerequisites resource group and look for a file named **prerequisiteSetupCompletionRunbook**. Select the runbook, then select **All Logs**.

To resolve this issue:

1. Enable syncing with Azure AD for the "USERS" container.
2. Create the AVDValidationUsers group in an organization unit that's syncing with Azure.

Next steps

Learn more about the getting started feature at Deploy Azure Virtual Desktop with the getting started feature.

Host pool creation

12/6/2021 • 9 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

This article covers issues during the initial setup of the Azure Virtual Desktop tenant and the related session host pool infrastructure.

Provide feedback

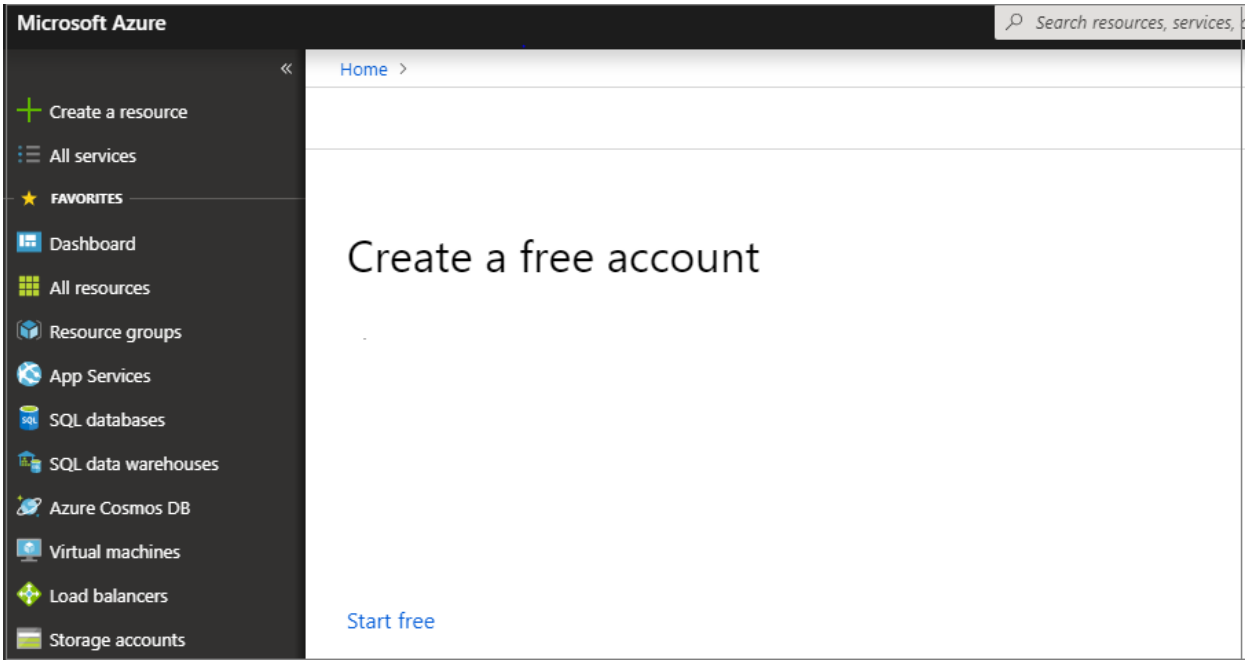
Visit the Azure Virtual Desktop Tech Community to discuss the Azure Virtual Desktop service with the product team and active community members.

Acquiring the Windows 10 Enterprise multi-session image

To use the Windows 10 Enterprise multi-session image, go to the Azure Marketplace, select **Get Started** > **Microsoft Windows 10** > and Windows 10 Enterprise multi-session, Version 1809.

Issues with using the Azure portal to create host pools

Error: "Create a free account" appears when accessing the service



Cause: There aren't active subscriptions in the account you signed in to Azure with, or the account doesn't have permissions to view the subscriptions.

Fix: Sign in to the subscription where you'll deploy the session host virtual machines (VMs) with an account that has at least contributor-level access.

Error: "Exceeding quota limit"

If your operation goes over the quota limit, you can do one of the following things:

- Create a new host pool with the same parameters but fewer VMs and VM cores.
- Open the link you see in the statusMessage field in a browser to submit a request to increase the quota for your Azure subscription for the specified VM SKU.

Error: Can't see user assignments in app groups.

Cause: This error usually happens after you've moved the subscription from 1 Azure Active Directory (AD) tenant to another. If your old assignments are still tied to the old Azure AD tenant, the Azure portal will lose track of them.

Fix: You'll need to reassign users to app groups.

I only see US when setting the location for my service objects

Cause: Azure doesn't currently support that region for the Azure Virtual Desktop service. To learn about which geographies we support, check out Data locations. If Azure Virtual Desktop supports the location but it still doesn't appear when you're trying to select a location, that means your resource provider hasn't updated yet.

Fix: To get the latest list of regions, re-register the resource provider:

1. Go to **Subscriptions** and select the relevant subscription.
2. Select **Resource Provider**.
3. Select **Microsoft.DesktopVirtualization**, then select **Re-register** from the action menu.

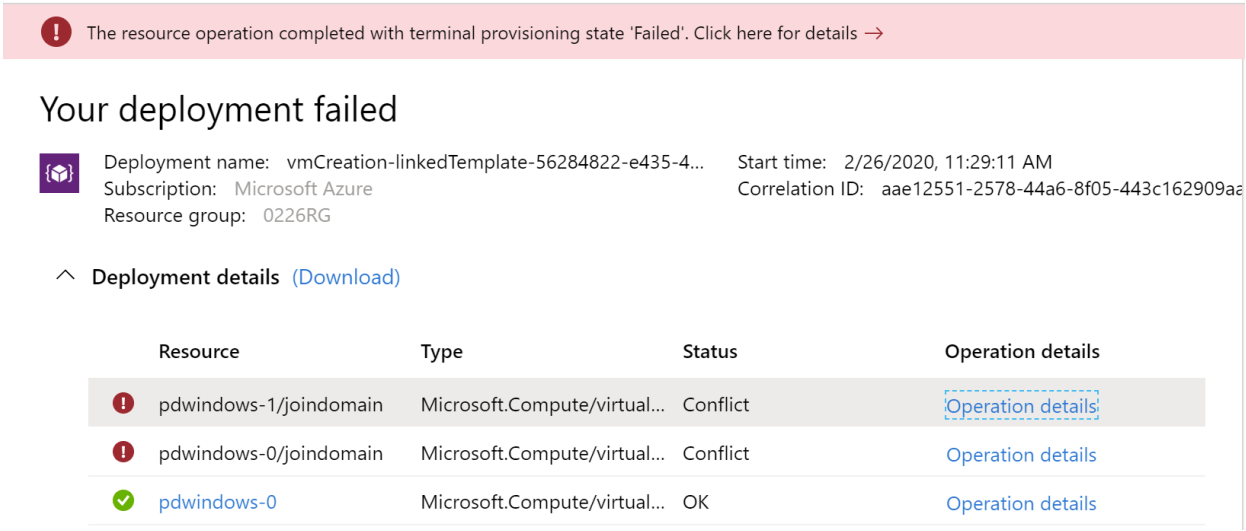
When you re-register the resource provider, you won't see any specific UI feedback or update statuses. The re-registration process also won't interfere with your existing environments.

Azure Resource Manager template errors

Follow these instructions to troubleshoot unsuccessful deployments of Azure Resource Manager templates and PowerShell DSC.

1. Review errors in the deployment using View deployment operations with Azure Resource Manager.
2. If there are no errors in the deployment, review errors in the activity log using View activity logs to audit actions on resources.
3. Once the error is identified, use the error message and the resources in Troubleshoot common Azure deployment errors with Azure Resource Manager to address the issue.
4. Delete any resources created during the previous deployment and retry deploying the template again.

Error: Your deployment failed....<hostname>/joindomain



Example of raw error:

```
 {"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list
deployment operations for details.
 Please see https://aka.ms/arm-debug for usage details.","details":[{"code":"Conflict","message":"{\r\n
\"status\": \"Failed\",\r\n \"error\":
 {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n \"message\": \"The resource operation completed with
terminal provisioning state 'Failed'.
 \",\r\n \"details\": [\r\n {\r\n \"code\": \"VMExtensionProvisioningError\",\r\n \"message\": \"VM has
reported a failure when processing
 extension 'joindomain'. Error message: \\\"Exception(s) occurred while joining Domain
'diamondsg.onmicrosoft.com'\\\".\"\r\n }\r\n ]\r\n }\r\n}"}]}
```


Cause 1: Credentials provided for joining VMs to the domain are incorrect.

Fix 1: See the "Incorrect credentials" error for VMs are not joined to the domain in Session host VM configuration.

Cause 2: Domain name doesn't resolve.

Fix 2: See Error: Domain name doesn't resolve in Session host VM configuration.

Cause 3: Your virtual network (VNET) DNS configuration is set to **Default**.

To fix this, do the following things:

1. Open the Azure portal and go to the **Virtual networks** tab.
2. Find your VNET, then select **DNS servers**.
3. The DNS servers menu should appear on the right side of your screen. On that menu, select **Custom**.
4. Make sure the DNS servers listed under Custom match your domain controller or Active Directory domain. If you don't see your DNS server, you can add it by entering its value into the **Add DNS server** field.

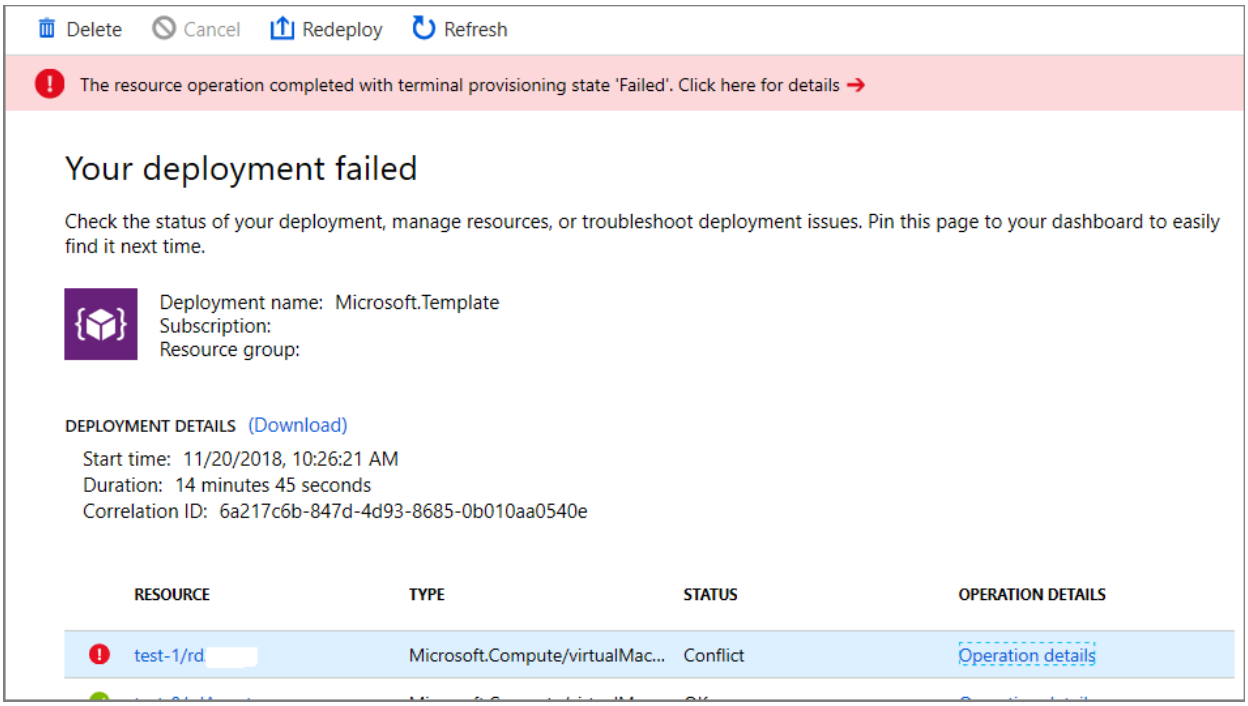
Error: Your deployment failed...\Unauthorized


```
{"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage details.","details":
[{"code":"Unauthorized","message":"{\r\n \"Code\": \"Unauthorized\",\r\n \"Message\": \"The scale operation
is not allowed for this subscription in this region. Try selecting different region or scale option.\",\r\n
\"Target\": null,\r\n \"Details\": [\r\n {\r\n \"Message\": \"The scale operation is not allowed for this
subscription in this region. Try selecting different region or scale option.\"\r\n },\r\n {\r\n \"Code\":
\"Unauthorized\"\r\n },\r\n {\r\n \"ErrorEntity\": {\r\n \"ExtendedCode\": \"52020\",\r\n
\"MessageTemplate\": \"The scale operation is not allowed for this subscription in this region. Try
selecting different region or scale option.\",\r\n \"Parameters\": [\r\n \"default\"\r\n ],\r\n \"Code\":
\"Unauthorized\",\r\n \"Message\": \"The scale operation is not allowed for this subscription in this
region. Try selecting different region or scale option.\"\r\n }\r\n }\r\n ],\r\n \"Innererror\":
null\r\n}"}]}
```


Cause: The subscription you're using is a type that can't access required features in the region where the customer is trying to deploy. For example, MSDN, Free, or Education subscriptions can show this error.

Fix: Change your subscription type or region to one that can access the required features.

Error: VMExtensionProvisioningError

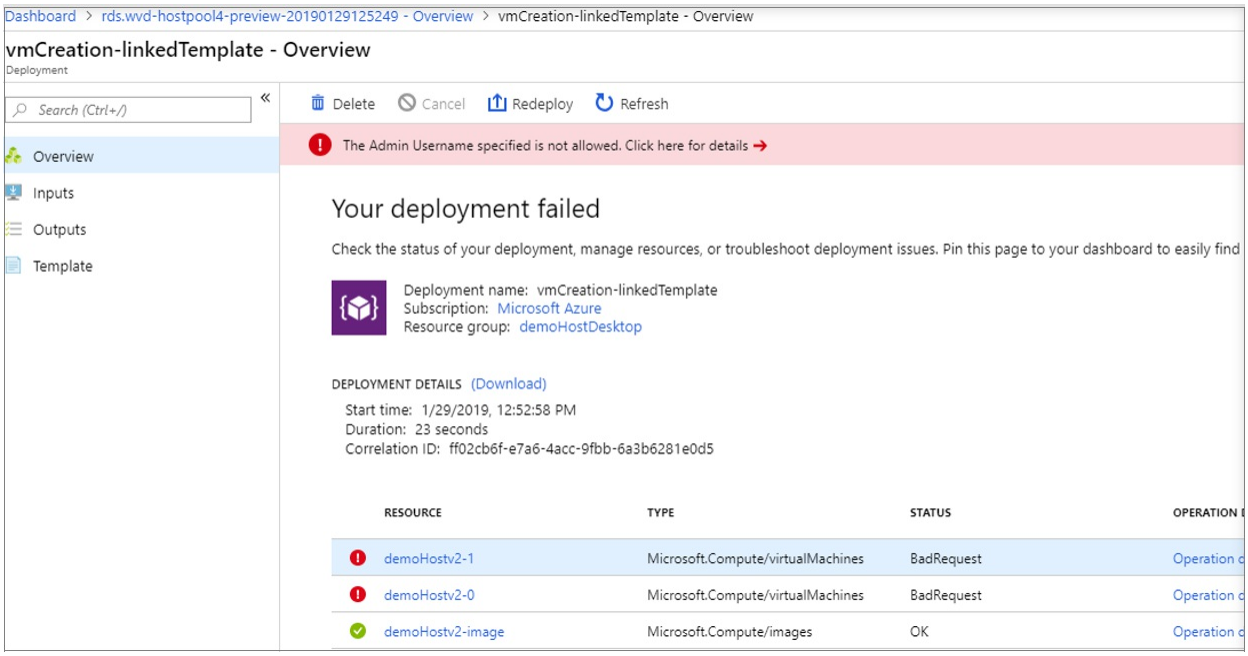


Cause 1: Transient error with the Azure Virtual Desktop environment.

Cause 2: Transient error with connection.

Fix: Confirm Azure Virtual Desktop environment is healthy by signing in using PowerShell. Finish the VM registration manually in Create a host pool with PowerShell.

Error: The Admin Username specified isn't allowed



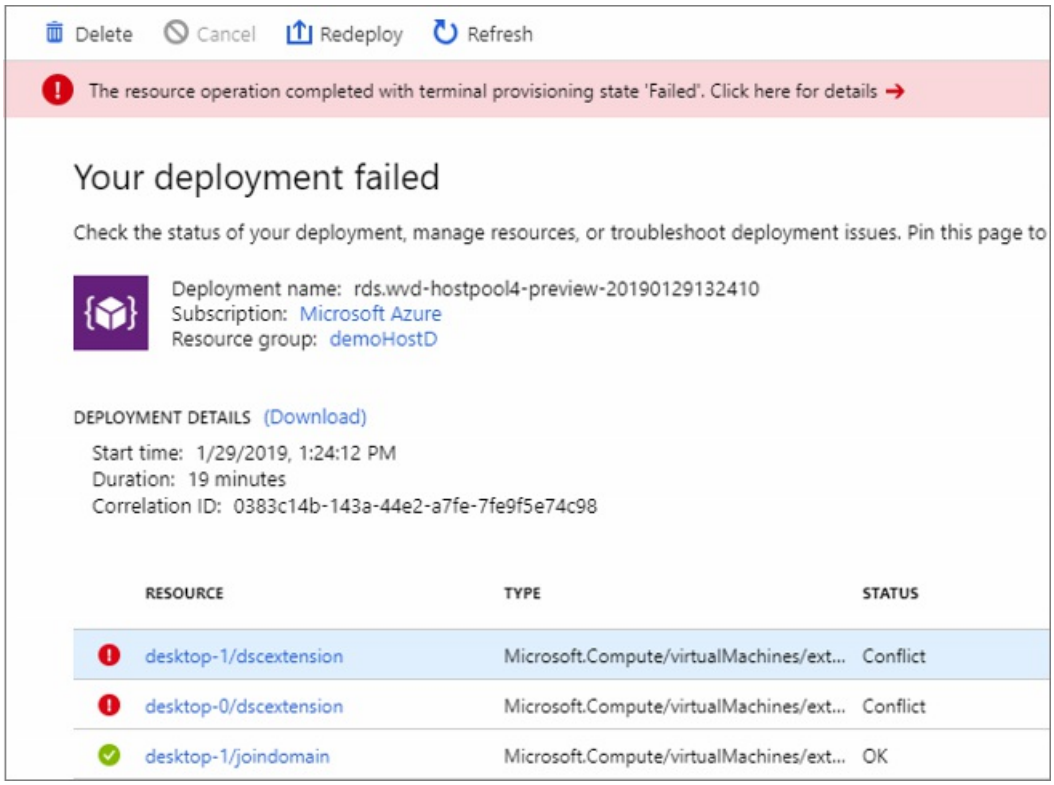
Example of raw error:

```
{ …{ "provisioningOperation":
 "Create", "provisioningState": "Failed", "timestamp": "2019-01-29T20:53:18.904917Z", "duration":
"PT3.0574505S", "trackingId":
 "1f460af8-34dd-4c03-9359-9ab249a1a005", "statusCode": "BadRequest", "statusMessage": { "error": { "code":
"InvalidParameter", "message":
 "The Admin Username specified is not allowed.", "target": "adminUsername" } … }
```

Cause: Password provided contains forbidden substrings (admin, administrator, root).

Fix: Update username or use different users.

Error: VM has reported a failure when processing extension



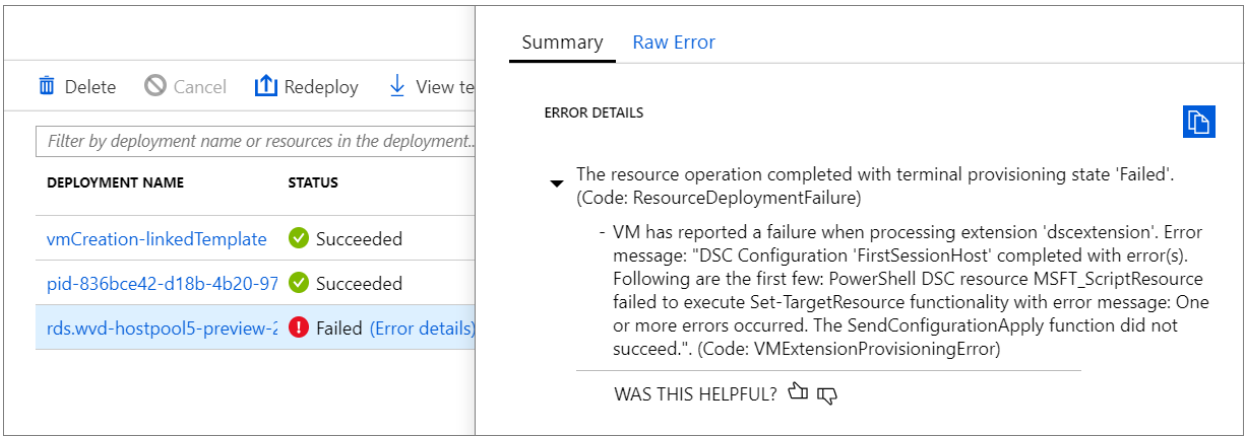
Example of raw error:

```
{ … "code": "ResourceDeploymentFailure", "message":
 "The resource operation completed with terminal provisioning state 'Failed'.", "details": [ { "code":
 "VMExtensionProvisioningError", "message": "VM has reported a failure when processing extension
'dscextension'.
 Error message: \"DSC Configuration 'SessionHost' completed with error(s). Following are the first few:
 PowerShell DSC resource MSFT_ScriptResource failed to execute Set-TargetResource functionality with error
message:
 One or more errors occurred. The SendConfigurationApply function did not succeed.\"." } ] … }
```

Cause: PowerShell DSC extension was not able to get admin access on the VM.

Fix: Confirm username and password have administrative access on the virtual machine and run the Azure Resource Manager template again.

Error: DeploymentFailed – PowerShell DSC Configuration 'FirstSessionHost' completed with Error(s)



Example of raw error:

```
{
    "code": "DeploymentFailed",
   "message": "At least one resource deployment operation failed. Please list
 deployment operations for details. 4 Please see https://aka.ms/arm-debug for usage details.",
 "details": [
         { "code": "Conflict",
         "message": "{\r\n \"status\": \"Failed\",\r\n \"error\": {\r\n \"code\":
         \"ResourceDeploymentFailure\",\r\n \"message\": \"The resource
         operation completed with terminal provisioning state 'Failed'.\",\r\n
         \"details\": [\r\n {\r\n \"code\":
        \"VMExtensionProvisioningError\",\r\n \"message\": \"VM has
              reported a failure when processing extension 'dscextension'.
              Error message: \\\"DSC Configuration 'FirstSessionHost'
              completed with error(s). Following are the first few:
              PowerShell DSC resource MSFT ScriptResource failed to
              execute Set-TargetResource functionality with error message:
              One or more errors occurred. The SendConfigurationApply
              function did not succeed.\\\".\"\r\n }\r\n ]\r\n }\r\n}"  }
```


Cause: PowerShell DSC extension was not able to get admin access on the VM.

Fix: Confirm username and password provided have administrative access on the virtual machine and run the Azure Resource Manager template again.

Error: DeploymentFailed – InvalidResourceReference

Example of raw error:


```
{"code":"DeploymentFailed","message":"At least one resource deployment operation
failed. Please list deployment operations for details. Please see https://aka.ms/arm-
debug for usage details.","details":[{"code":"Conflict","message":"{\r\n \"status\":
\"Failed\",\r\n \"error\": {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n
\"message\": \"The resource operation completed with terminal provisioning state
'Failed'.\",\r\n \"details\": [\r\n {\r\n \"code\": \"DeploymentFailed\",\r\n
\"message\": \"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage
details.\",\r\n \"details\": [\r\n {\r\n \"code\": \"BadRequest\",\r\n \"message\":
\"{\\r\\n \\\"error\\\": {\\r\\n \\\"code\\\": \\\"InvalidResourceReference\\\",\\r\\n
\\\"message\\\": \\\"Resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/virtualNetworks/wvd-vnet/subnets/default
referenced by resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/networkInterfaces/erd. Please make sure that
the referenced resource exists, and that both resources are in the same
region.\\\",\\r\\n\\\"details\\\": []\\r\\n }\\r\\n}\"\r\n }\r\n ]\r\n }\r\n ]\r\n }\r\n}"}]}
```


Cause: Part of the resource group name is used for certain resources being created by the template. Due to the name matching existing resources, the template may select an existing resource from a different group.

Fix: When running the Azure Resource Manager template to deploy session host VMs, make the first two characters unique for your subscription resource group name.

Error: DeploymentFailed – InvalidResourceReference

Example of raw error:

```
{"code":"DeploymentFailed","message":"At least one resource deployment operation
failed. Please list deployment operations for details. Please see https://aka.ms/arm-
debug for usage details.","details":[{"code":"Conflict","message":"{\r\n \"status\":
\"Failed\",\r\n \"error\": {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n
\"message\": \"The resource operation completed with terminal provisioning state
'Failed'.\",\r\n \"details\": [\r\n {\r\n \"code\": \"DeploymentFailed\",\r\n
\"message\": \"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage
details.\",\r\n \"details\": [\r\n {\r\n \"code\": \"BadRequest\",\r\n \"message\":
\"{\\r\\n \\\"error\\\": {\\r\\n \\\"code\\\": \\\"InvalidResourceReference\\\",\\r\\n
\\\"message\\\": \\\"Resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/virtualNetworks/wvd-vnet/subnets/default
referenced by resource
/subscriptions/EXAMPLE/resourceGroups/DEMO/providers/Microsoft.Network/networkInterfaces
/EXAMPLE was not found. Please make sure that the referenced resource exists, and that both
resources are in the same region.\\\",\\r\\n \\\"details\\\": []\\r\\n }\\r\\n}\"\r\n
}\r\n ]\r\n }\r\n ]\r\n }\r\n\
```

Cause: This error is because the NIC created with the Azure Resource Manager template has the same name as another NIC already in the VNET.

Fix: Use a different host prefix.

Error: DeploymentFailed – Error downloading

Example of raw error:

```
\\\"The DSC Extension failed to execute: Error downloading
https://catalogartifact.azureedge.net/publicartifacts/rds.wvd-provision-host-pool-
2dec7a4d-006c-4cc0-965a-02bbe438d6ff-prod
/Artifacts/DSC/Configuration.zip after 29 attempts: The remote name could not be
resolved: 'catalogartifact.azureedge.net'.\\nMore information about the failure can
be found in the logs located under
'C:\\\\WindowsAzure\\\\Logs\\\\Plugins\\\\Microsoft.Powershell.DSC\\\\2.77.0.0' on
the VM.\\\"
```

Cause: This error is due to a static route, firewall rule, or NSG blocking the download of the zip file tied to the Azure Resource Manager template.

Fix: Remove blocking static route, firewall rule, or NSG. Optionally, open the Azure Resource Manager template json file in a text editor, take the link to zip file, and download the resource to an allowed location.

Error: Can't delete a session host from the host pool after deleting the VM

Cause: You need to delete the session host before you delete the VM.

Fix: Put the session host in drain mode, sign out all users from the session host, then delete the host.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues with Azure Virtual Desktop client connections, see Azure Virtual Desktop service connections.
- To troubleshoot issues with Remote Desktop clients, see Troubleshoot the Remote Desktop client
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop

PowerShell.

- To learn more about the service, see Azure Virtual Desktop environment.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.
- To learn about auditing actions, see Audit operations with Resource Manager.
- To learn about actions to determine the errors during deployment, see View deployment operations.

Session host virtual machine configuration

12/6/2021 • 13 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Use this article to troubleshoot issues you're having when configuring the Azure Virtual Desktop session host virtual machines (VMs).

Provide feedback

Visit the Azure Virtual Desktop Tech Community to discuss the Azure Virtual Desktop service with the product team and active community members.

VMs aren't joined to the domain

Follow these instructions if you're having issues joining virtual machines (VMs) to the domain.

- Join the VM manually using the process in Join a Windows Server virtual machine to a managed domain or using the domain join template.
- Try pinging the domain name from a command line on the VM.
- Review the list of domain join error messages in Troubleshooting Domain Join Error Messages.

Error: Incorrect credentials

Cause: There was a typo made when the credentials were entered in the Azure Resource Manager template interface fixes.

Fix: Take one of the following actions to resolve.

- Manually add the VMs to a domain.
- Redeploy the template once credentials have been confirmed. See Create a host pool with PowerShell.
- Join VMs to a domain using a template with Joins an existing Windows VM to AD Domain.

Error: Timeout waiting for user input

Cause: The account used to complete the domain join may have multifactor authentication (MFA).

Fix: Take one of the following actions to resolve.

- Temporarily remove MFA for the account.
- Use a service account.

Error: The account used during provisioning doesn't have permissions to complete the operation

Cause: The account being used doesn't have permissions to join VMs to the domain due to compliance and regulations.

Fix: Take one of the following actions to resolve.

- Use an account that is a member of the Administrator group.
- Grant the necessary permissions to the account being used.

Error: Domain name doesn't resolve

Cause 1: VMs are on a virtual network that's not associated with the virtual network (VNET) where the domain is located.

Fix 1: Create VNET peering between the VNET where VMs were provisioned and the VNET where the domain controller (DC) is running. See Create a virtual network peering - Resource Manager, different subscriptions.

Cause 2: When using Azure Active Directory Domain Services (Azure AD DS), the virtual network doesn't have its DNS server settings updated to point to the managed domain controllers.

Fix 2: To update the DNS settings for the virtual network containing Azure AD DS, see Update DNS settings for the Azure virtual network.

Cause 3: The network interface's DNS server settings don't point to the appropriate DNS server on the virtual network.

Fix 3: Take one of the following actions to resolve, following the steps in [Change DNS servers].

- Change the network interface's DNS server settings to **Custom** with the steps from Change DNS servers and specify the private IP addresses of the DNS servers on the virtual network.
- Change the network interface's DNS server settings to **Inherit from virtual network** with the steps from Change DNS servers, then change the virtual network's DNS server settings with the steps from Change DNS servers.

Azure Virtual Desktop Agent and Azure Virtual Desktop Boot Loader aren't installed

The recommended way to provision VMs is using the Azure portal creation template. The template automatically installs the Azure Virtual Desktop Agent and Azure Virtual Desktop Agent Boot Loader.

Follow these instructions to confirm the components are installed and to check for error messages.

1. Confirm that the two components are installed by checking in **Control Panel** > **Programs** > **Programs and Features**. If **Azure Virtual Desktop Agent** and **Azure Virtual Desktop Agent Boot Loader** aren't visible, they aren't installed on the VM.
2. Open **File Explorer** and navigate to **C:\Windows\Temp\ScriptLog.log**. If the file is missing, it indicates that the PowerShell DSC that installed the two components wasn't able to run in the security context provided.
3. If the file **C:\Windows\Temp\ScriptLog.log** is present, open it and check for error messages.

Error: Azure Virtual Desktop Agent and Azure Virtual Desktop Agent Boot Loader are missing. C:\Windows\Temp\ScriptLog.log is also missing

Cause 1: Credentials provided during input for the Azure Resource Manager template were incorrect or permissions were insufficient.

Fix 1: Manually add the missing components to the VMs using Create a host pool with PowerShell.

Cause 2: PowerShell DSC was able to start and execute but failed to complete as it can't sign in to Azure Virtual Desktop and obtain needed information.

Fix 2: Confirm the items in the following list.

- Make sure the account doesn't have MFA.
- Confirm the host pool's name is accurate and the host pool exists in Azure Virtual Desktop.
- Confirm the account has at least Contributor permissions on the Azure subscription or resource group.

Error: Authentication failed, error in C:\Windows\Temp\ScriptLog.log

Cause: PowerShell DSC was able to execute but couldn't connect to Azure Virtual Desktop.

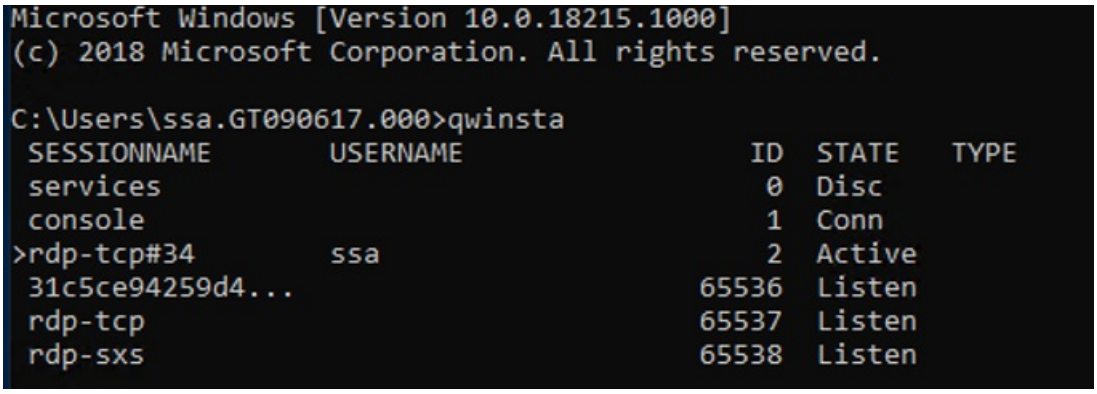
Fix: Confirm the items in the following list.

- Manually register the VMs with the Azure Virtual Desktop service.
- Confirm account used for connecting to Azure Virtual Desktop has permissions on the Azure subscription or resource group to create host pools.
- Confirm account doesn't have MFA.

Azure Virtual Desktop Agent isn't registering with the Azure Virtual Desktop service

When the Azure Virtual Desktop Agent is first installed on session host VMs (either manually or through the Azure Resource Manager template and PowerShell DSC), it provides a registration token. The following section covers troubleshooting issues that apply to the Azure Virtual Desktop Agent and the token.

Error: The status filed in Get-AzWvdSessionHost cmdlet shows status as Unavailable



Cause: The agent isn't able to update itself to a new version.

Fix: Follow these instructions to manually update the agent.

1. Download a new version of the agent on the session host VM.
2. Launch Task Manager and, in the Service Tab, stop the RDAgentBootLoader service.
3. Run the installer for the new version of the Azure Virtual Desktop Agent.
4. When prompted for the registration token, remove the entry INVALID_TOKEN and press next (a new token isn't required).
5. Complete the installation Wizard.
6. Open Task Manager and start the RDAgentBootLoader service.

Error: Azure Virtual Desktop Agent registry entry IsRegistered shows a value of 0

Cause: Registration token has expired.

Fix: Follow these instructions to fix the agent registry error.

1. If there's already a registration token, remove it with Remove-AzWvdRegistrationInfo.
2. Run the **New-AzWvdRegistrationInfo** cmdlet to generate a new token.
3. Confirm that the *-ExpriationTime* parameter is set to three days.

Error: Azure Virtual Desktop agent isn't reporting a heartbeat when running Get-AzWvdSessionHost

Cause 1: RDAgentBootLoader service has been stopped.

Fix 1: Launch Task Manager and, if the Service Tab reports a stopped status for RDAgentBootLoader service,

start the service.

Cause 2: Port 443 may be closed.

Fix 2: Follow these instructions to open port 443.

1. Confirm port 443 is open by downloading the PSPing tool from Sysinternal tools.
2. Install PSPing on the session host VM where the agent is running.
3. Open the command prompt as an administrator and issue the command below:

```
psping rdbroker.wvdselfhost.microsoft.com:443
```

4. Confirm that PSPing received information back from the RDBroker:

```
PsPing v2.10 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
TCP connect to 13.77.160.237:443:
5 iterations (warmup 1) ping test:
Connecting to 13.77.160.237:443 (warmup): from 172.20.17.140:60649: 2.00ms
Connecting to 13.77.160.237:443: from 172.20.17.140:60650: 3.83ms
Connecting to 13.77.160.237:443: from 172.20.17.140:60652: 2.21ms
Connecting to 13.77.160.237:443: from 172.20.17.140:60653: 2.14ms
Connecting to 13.77.160.237:443: from 172.20.17.140:60654: 2.12ms
TCP connect statistics for 13.77.160.237:443:
Sent = 4, Received = 4, Lost = 0 (0% loss),
Minimum = 2.12ms, Maximum = 3.83ms, Average = 2.58ms
```

Troubleshooting issues with the Azure Virtual Desktop side-by-side stack

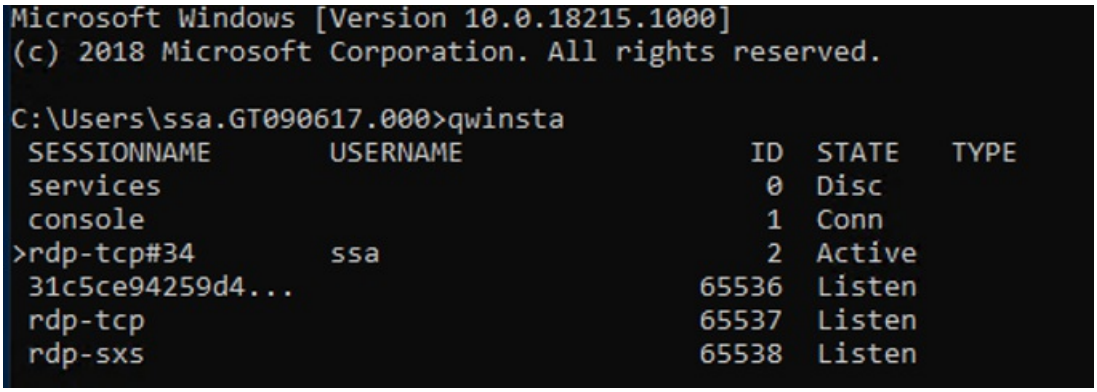
The Azure Virtual Desktop side-by-side stack is automatically installed with Windows Server 2019 and newer. Use Microsoft Installer (MSI) to install the side-by-side stack on Microsoft Windows Server 2016 or Windows Server 2012 R2. For Microsoft Windows 10, the Azure Virtual Desktop side-by-side stack is enabled with **enablesxstackrs.ps1**.

There are three main ways the side-by-side stack gets installed or enabled on session host pool VMs:

- With the Azure portal creation template
- By being included and enabled on the master image
- Installed or enabled manually on each VM (or with extensions/PowerShell)

If you're having issues with the Azure Virtual Desktop side-by-side stack, type the **qwinsta** command from the command prompt to confirm that the side-by-side stack is installed or enabled.

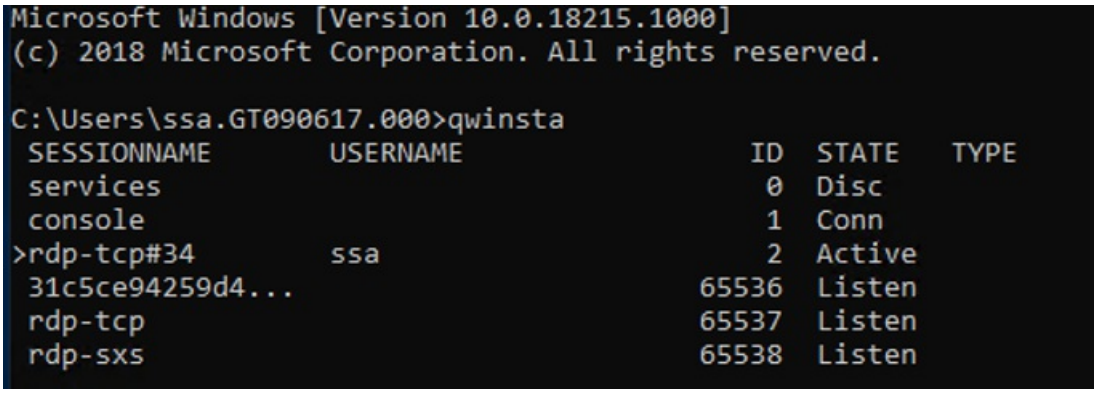
The output of **qwinsta** will list **rdp-sxs** in the output if the side-by-side stack is installed and enabled.



Examine the registry entries listed below and confirm that their values match. If registry keys are missing or values are mismatched, make sure you're running a supported operating system. If you are, follow the instructions in Create a host pool with PowerShell on how to reinstall the side-by-side stack.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\rds-sxs\"fEnableWinstation":DWORD=1

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\ClusterSettings\"SessionDirectoryListener":rdp-sxs
```

Error: O_REVERSE_CONNECT_STACK_FAILURE



Cause: The side-by-side stack isn't installed on the session host VM.

Fix: Follow these instructions to install the side-by-side stack on the session host VM.

1. Use Remote Desktop Protocol (RDP) to get directly into the session host VM as local administrator.
2. Install the side-by-side stack using Create a host pool with PowerShell.

How to fix an Azure Virtual Desktop side-by-side stack that malfunctions

There are known circumstances that can cause the side-by-side stack to malfunction:

- Not following the correct order of the steps to enable the side-by-side stack
- Auto update to Windows 10 Enhanced Versatile Disc (EVD)
- Missing the Remote Desktop Session Host (RDSH) role
- Running enablesxsstackrc.ps1 multiple times
- Running enablesxsstackrc.ps1 in an account that doesn't have local admin privileges

The instructions in this section can help you uninstall the Azure Virtual Desktop side-by-side stack. Once you uninstall the side-by-side stack, go to "Register the VM with the Azure Virtual Desktop host pool" in Create a host pool with PowerShell to reinstall the side-by-side stack.

The VM used to run remediation must be on the same subnet and domain as the VM with the malfunctioning side-by-side stack.

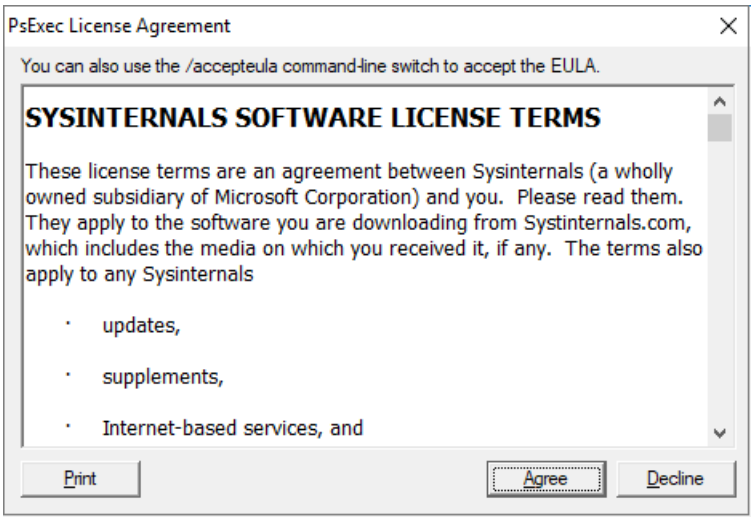
Follow these instructions to run remediation from the same subnet and domain:

1. Connect with standard Remote Desktop Protocol (RDP) to the VM from where fix will be applied.
2. Download PsExec from https://docs.microsoft.com/sysinternals/downloads/psexec.
3. Unzip the downloaded file.
4. Start command prompt as local administrator.
5. Navigate to folder where PsExec was unzipped.
6. From command prompt, use the following command:

```
psexec.exe \\<VMname> cmd
```

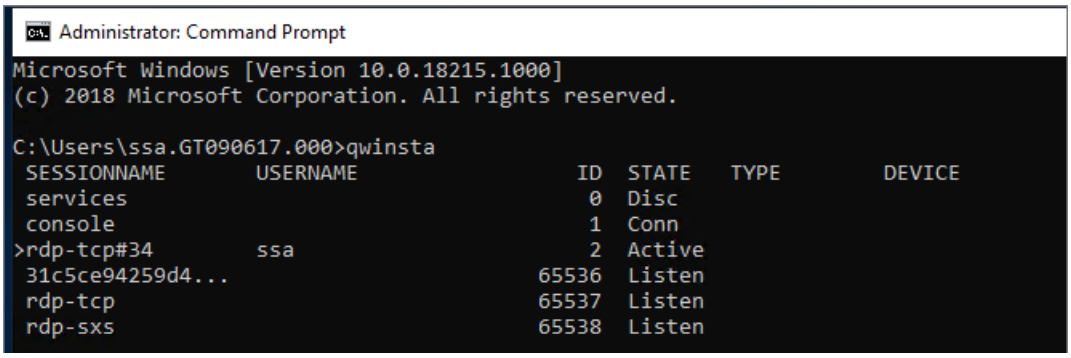
> **NOTE**
> VMname is the machine name of the VM with the malfunctioning side-by-side stack.

7. Accept the PsExec License Agreement by clicking Agree.



> **NOTE**
> This dialog will show up only the first time PsExec is run.

8. After the command prompt session opens on the VM with the malfunctioning side-by-side stack, run qwinsta and confirm that an entry named rdp-sxs is available. If not, a side-by-side stack isn't present on the VM so the issue isn't tied to the side-by-side stack.



9. Run the following command, which will list Microsoft components installed on the VM with the malfunctioning side-by-side stack.

```
wmic product get name
```

10. Run the command below with product names from step above.

```
wmic product where name="<Remote Desktop Services Infrastructure Agent>" call uninstall
```

11. Uninstall all products that start with "Remote Desktop."
12. After all Azure Virtual Desktop components have been uninstalled, follow the instructions for your operating system:
13. If your operating system is Windows Server, restart the VM that had the malfunctioning side-by-side stack (either with Azure portal or from the PsExec tool).

If your operating system is Microsoft Windows 10, continue with the instructions below:

14. From the VM running PsExec, open File Explorer and copy disablesxsstackrc.ps1 to the system drive of the VM with the malfunctioned side-by-side stack.

```
\\<VMname>\c$\
```

> **NOTE**
> VMname is the machine name of the VM with the malfunctioning side-by-side stack.

15. The recommended process: from the PsExec tool, start PowerShell and navigate to the folder from the previous step and run disablesxsstackrc.ps1. Alternatively, you can run the following cmdlets:

```
Remove-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\ClusterSettings" -
Name "SessionDirectoryListener" -Force
Remove-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\rdp-sxs" -
Recurse -Force
Remove-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations" -Name
"ReverseConnectionListener" -Force
```

16. When the cmdlets are done running, restart the VM with the malfunctioning side-by-side stack.

Remote Desktop licensing mode isn't configured

If you sign in to Windows 10 Enterprise multi-session using an administrative account, you might receive a

notification that says, "Remote Desktop licensing mode isn't configured, Remote Desktop Services will stop working in X days. On the Connection Broker server, use Server Manager to specify the Remote Desktop licensing mode."

If the time limit expires, an error message will appear that says, "The remote session was disconnected because there are no Remote Desktop client access licenses available for this computer."

If you see either of these messages, it means the image doesn't have the latest Windows updates installed or you're setting the Remote Desktop licensing mode through group policy. Follow the steps in the next sections to check the group policy setting, identify the version of Windows 10 Enterprise multi-session, and install the corresponding update.

> **NOTE**
> Azure Virtual Desktop only requires an RDS client access license (CAL) when your host pool contains Windows Server session hosts. To learn how to configure an RDS CAL, see License your RDS deployment with client access licenses.

Disable the Remote Desktop licensing mode group policy setting

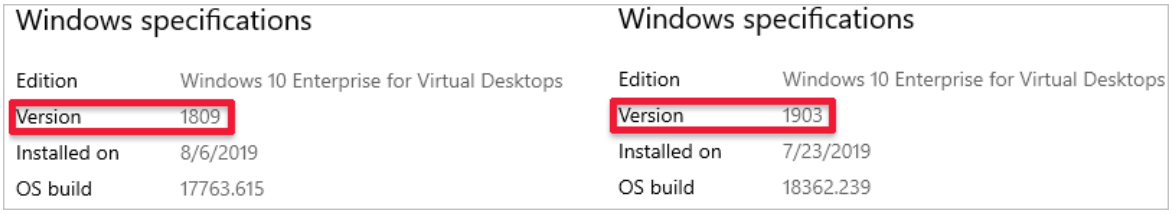
Check the group policy setting by opening the Group Policy Editor in the VM and navigating to **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Licensing** > **Set the Remote Desktop licensing mode**. If the group policy setting is **Enabled**, change it to **Disabled**. If it's already disabled, then leave it as-is.

> **NOTE**
> If you set group policy through your domain, disable this setting on policies that target these Windows 10 Enterprise multi-session VMs.

Identify which version of Windows 10 Enterprise multi-session you're using

To check which version of Windows 10 Enterprise multi-session you have:

1. Sign in with your admin account.
2. Enter "About" into the search bar next to the Start menu.
3. Select **About your PC**.
4. Check the number next to "Version." The number should be either "1809" or "1903," as shown in the following image.



Now that you know your version number, skip ahead to the relevant section.

Version 1809

If your version number says "1809," install the KB4516077 update.

Version 1903

Redeploy the host operating system with the latest version of the Windows 10, version 1903 image from the Azure Gallery.

We couldn't connect to the remote PC because of a security error

If your users see an error that says, "We couldn't connect to the remote PC because of a security error. If this keeps happening, ask your admin or tech support for help," validate any existing policies that change default RDP permissions. One policy that might cause this error to appear is "Allow log on through Remote Desktop Services security policy."

To learn more about this policy, see Allow log on through Remote Desktop Services.

I can't deploy the golden image

Golden images must not include the Azure Virtual Desktop agent. You can install the agent only after you deploy the golden image.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while creating a host pool in an Azure Virtual Desktop environment, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues with Azure Virtual Desktop client connections, see Azure Virtual Desktop service connections.
- To troubleshoot issues with Remote Desktop clients, see Troubleshoot the Remote Desktop client
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To learn more about the service, see Azure Virtual Desktop environment.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.
- To learn about auditing actions, see Audit operations with Resource Manager.
- To learn about actions to determine the errors during deployment, see View deployment operations.

Management issues

12/6/2021 • 2 minutes to read • Edit Online

This article describes common management errors and gives suggestions for how to solve them.

Common management errors

The following table lists error messages that appear due to management-related issues and suggestions for how to solve them.

ERROR MESSAGE	SUGGESTED SOLUTION
Failed to create registration key	Registration token couldn't be created. Try creating it again with a shorter expiry time (between 1 hour and 1 month).
Failed to delete registration key	Registration token couldn't be deleted. Try deleting it again. If it still doesn't work, use PowerShell to check if the token is still there. If it's there, delete it with PowerShell.
Failed to change session host drain mode	Couldn't change drain mode on the VM. Check the VM status. If the VM isn't available, you can't change drain mode.
Failed to disconnect user sessions	Couldn't disconnect the user from the VM. Check the VM status. If the VM isn't available, you can't disconnect the user session. If the VM is available, check the user session status to see if it's disconnected.
Failed to log off all user(s) within the session host	Could not sign users out of the VM. Check the VM status. If unavailable, users can't be signed out. Check user session status to see if they're already signed out. You can force sign out with PowerShell.
Failed to unassign user from application group	Could not unpublish an app group for a user. Check to see if user is available on Azure AD. Check to see if the user is part of a user group that the app group is published to.
There was an error retrieving the available locations	Check location of VM used in the create host pool wizard. If image is not available in that location, add image in that location or choose a different VM location.

Error: Can't add user assignments to an app group

After assigning a user to an app group, the Azure portal displays a warning that says "Session Ending" or "Experiencing Authentication Issues - Extension Microsoft_Azure_WVD." The assignment page then doesn't load, and after that, pages stop loading throughout the Azure portal (for example, Azure Monitor, Log Analytics, Service Health, and so on).

This issue usually appears because there's a problem with the conditional access policy. The Azure portal is trying to obtain a token for Microsoft Graph, which is dependent on SharePoint Online. The customer has a conditional access policy called "Microsoft Office 365 Data Storage Terms of Use" that requires users to accept the terms of use to access data storage. However, they haven't signed in yet, so the Azure portal can't get the

token.

To solve this issue, before signing in to the Azure portal, the admin first needs to sign in to SharePoint and accept the Terms of Use. After that, they should be able to sign in to the Azure portal like normal.

Next steps

To review common error scenarios that the diagnostics feature can identify for you, see Identify and diagnose issues.

Azure Virtual Desktop PowerShell

12/6/2021 • 3 minutes to read • Edit Online

> **IMPORTANT**
> This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Use this article to resolve errors and issues when using PowerShell with Azure Virtual Desktop. For more information on Remote Desktop Services PowerShell, see Azure Virtual Desktop PowerShell.

Provide feedback

Visit the Azure Virtual Desktop Tech Community to discuss the Azure Virtual Desktop service with the product team and active community members.

PowerShell commands used during Azure Virtual Desktop setup

This section lists PowerShell commands that are typically used while setting up Azure Virtual Desktop and provides ways to resolve issues that may occur while using them.

Error: New-AzRoleAssignment: The provided information does not map to an AD object ID

```
New-AzRoleAssignment -SignInName "admins@contoso.com" -RoleDefinitionName "Desktop Virtualization User" -
ResourceName "0301HP-DAG" -ResourceGroupName 0301RG -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Cause: The user specified by the *-SignInName* parameter can't be found in the Azure Active Directory tied to the Azure Virtual Desktop environment.

Fix: Make sure of the following things.

- The user should be synced to Azure Active Directory.
- The user shouldn't be tied to business-to-consumer (B2C) or business-to-business (B2B) commerce.
- The Azure Virtual Desktop environment should be tied to correct Azure Active Directory.

Error: New-AzRoleAssignment: "The client with object id does not have authorization to perform action over scope (code: AuthorizationFailed)"

Cause 1: The account being used doesn't have Owner permissions on the subscription.

Fix 1: A user with Owner permissions needs to execute the role assignment. Alternatively, the user needs to be assigned to the User Access Administrator role to assign a user to an application group.

Cause 2: The account being used has Owner permissions but isn't part of the environment's Azure Active Directory or doesn't have permissions to query the Azure Active Directory where the user is located.

Fix 2: A user with Active Directory permissions needs to execute the role assignment.

Error: New-AzWvdHostPool -- the location is not available for resource type

```
New-AzWvdHostPool_CreateExpanded: The provided location 'southeastasia' is not available for resource type
'Microsoft.DesktopVirtualization/hostpools'. List of available regions for the resource type is
'eastus,eastus2,westus,westus2,northcentralus,southcentralus,westcentralus,centralus'.
```

Cause: Azure Virtual Desktop supports selecting the location of host pools, application groups, and workspaces to store service metadata in certain locations. Your options are restricted to where this feature is available. This error means that the feature isn't available in the location you chose.

Fix: In the error message, a list of supported regions will be published. Use one of the supported regions instead.

Error: New-AzWvdApplicationGroup must be in same location as host pool

```
New-AzWvdApplicationGroup_CreateExpanded: ActivityId: e5fe6c1d-5f2c-4db9-817d-e423b8b7d168 Error:
ApplicationGroup must be in same location as associated HostPool
```

Cause: There's a location mismatch. All host pools, application groups, and workspaces have a location to store service metadata. Any objects you create that are associated with each other must be in the same location. For example, if a host pool is in `eastus`, then you also need to create the application groups in `eastus`. If you create a workspace to register these application groups to, that workspace needs to be in `eastus` as well.

Fix: Retrieve the location the host pool was created in, then assign the application group you're creating to that same location.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while setting up your Azure Virtual Desktop environment and host pools, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues with Azure Virtual Desktop client connections, see Azure Virtual Desktop service connections.
- To troubleshoot issues with Remote Desktop clients, see Troubleshoot the Remote Desktop client
- To learn more about the service, see Azure Virtual Desktop environment.
- To learn about auditing actions, see Audit operations with Resource Manager.
- To learn about actions to determine the errors during deployment, see View deployment operations.

Troubleshoot common Azure Virtual Desktop Agent issues

12/6/2021 • 16 minutes to read • Edit Online

The Azure Virtual Desktop Agent can cause connection issues because of multiple factors:

- An error on the broker that makes the agent stop the service.
- Problems with updates.
- Issues with installing during the agent installation, which disrupts connection to the session host.

This article will guide you through solutions to these common scenarios and how to address connection issues.

> **NOTE**
> For troubleshooting issues related to session connectivity and the Azure Virtual Desktop agent, we recommend you review the event logs in **Event Viewer** > **Windows Logs** > **Application**. Look for events that have one of the following sources to identify your issue:
>
> - WVD-Agent
> - WVD-Agent-Updater
> - RDAgentBootLoader
> - MsiInstaller

Error: The RDAgentBootLoader and/or Remote Desktop Agent Loader has stopped running

If you're seeing any of the following issues, this means that the boot loader, which loads the agent, was unable to install the agent properly and the agent service isn't running:

- **RDAgentBootLoader** is either stopped or not running.
- There's no status for **Remote Desktop Agent Loader**.

To resolve this issue, start the RDAgent boot loader:

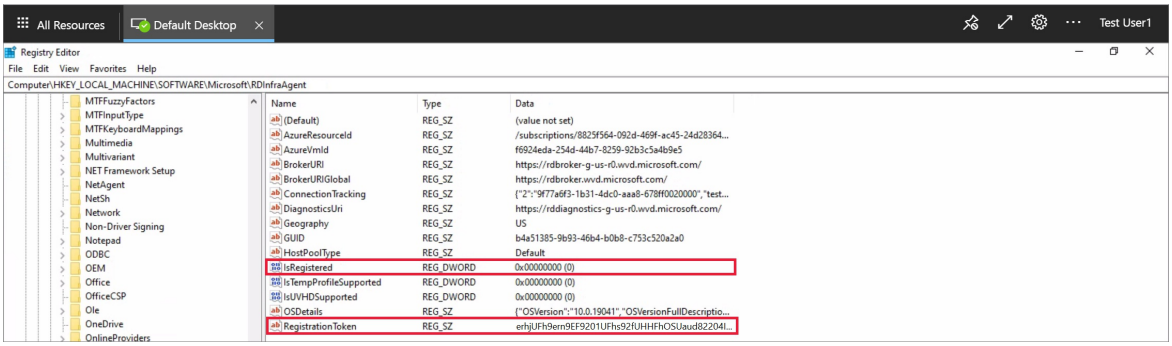
1. In the Services window, right-click **Remote Desktop Agent Loader**.
2. Select **Start**. If this option is greyed out for you, you don't have administrator permissions and will need to get them to start the service.
3. Wait 10 seconds, then right-click **Remote Desktop Agent Loader**.
4. Select **Refresh**.
5. If the service stops after you started and refreshed it, you may have a registration failure. For more information, see INVALID_REGISTRATION_TOKEN.

Error: INVALID_REGISTRATION_TOKEN

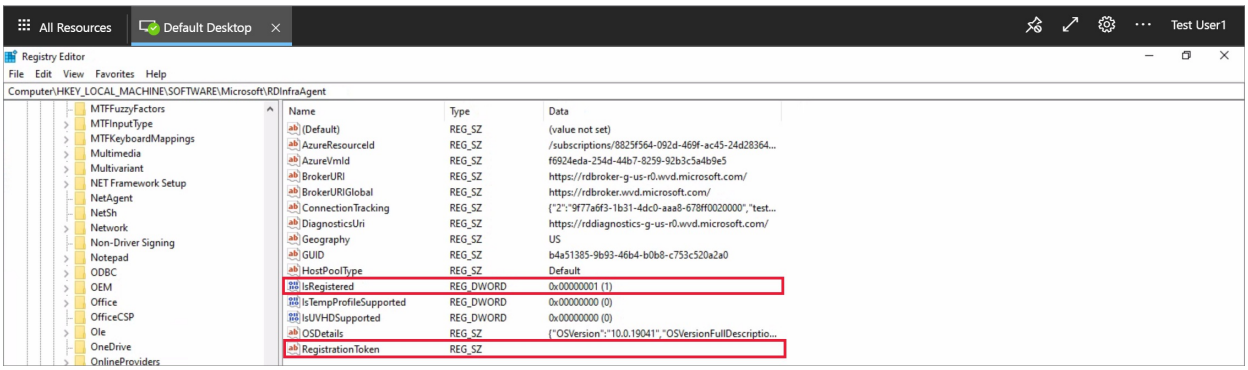
Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277, that says **INVALID_REGISTRATION_TOKEN** in the description, the registration token that you have isn't recognized as valid.

To resolve this issue, create a valid registration token:

1. To create a new registration token, follow the steps in the Generate a new registration key for the VM section.
2. Open the Registry Editor.
3. Go to **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **Microsoft** > **RDInfraAgent**.
4. Select **IsRegistered**.
5. In the **Value data:** entry box, type **0** and select **Ok**.
6. Select **RegistrationToken**.
7. In the **Value data:** entry box, paste the registration token from step 1.



8. Open a command prompt as an administrator.
9. Enter **net stop RDAgentBootLoader**.
10. Enter **net start RDAgentBootLoader**.
11. Open the Registry Editor.
12. Go to **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **Microsoft** > **RDInfraAgent**.
13. Verify that **IsRegistered** is set to 1 and there is nothing in the data column for **RegistrationToken**.



Error: Agent cannot connect to broker with INVALID_FORM

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277 that says "INVALID_FORM" in the description, something went wrong with the communication between the agent and the broker. The agent can't connect to the broker or reach a particular URL because of certain firewall or DNS settings.

To resolve this issue, check that you can reach BrokerURI and BrokerURIGlobal:

1. Open theRegistryEditor.
2. Go to **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **Microsoft** > **RDInfraAgent**.

3. Make note of the values for **BrokerURI** and **BrokerURIGlobal**.

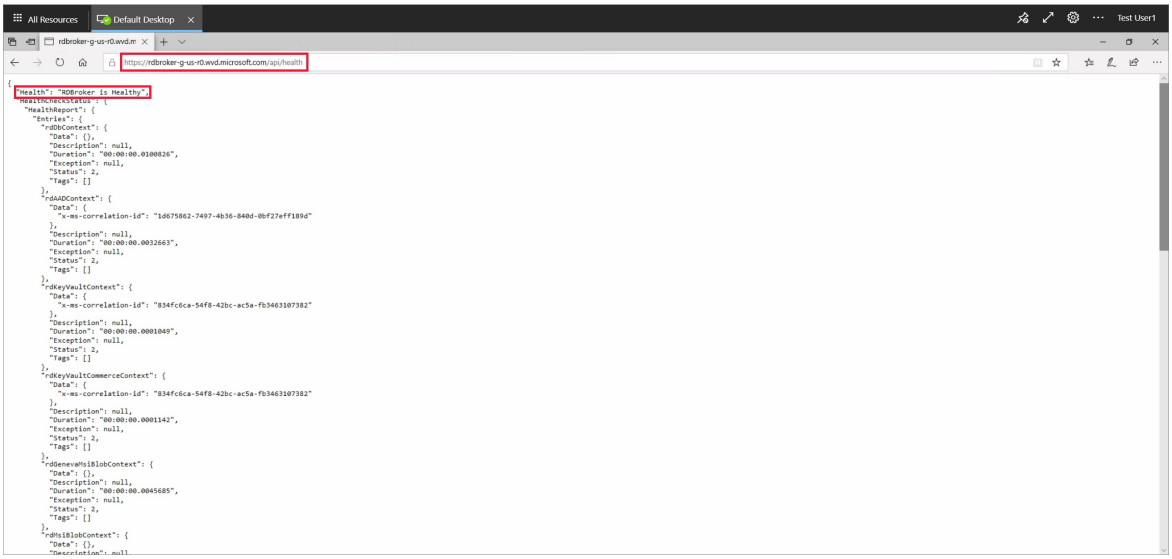
4. Open a browser and go to *<BrokerURI>api/health*.

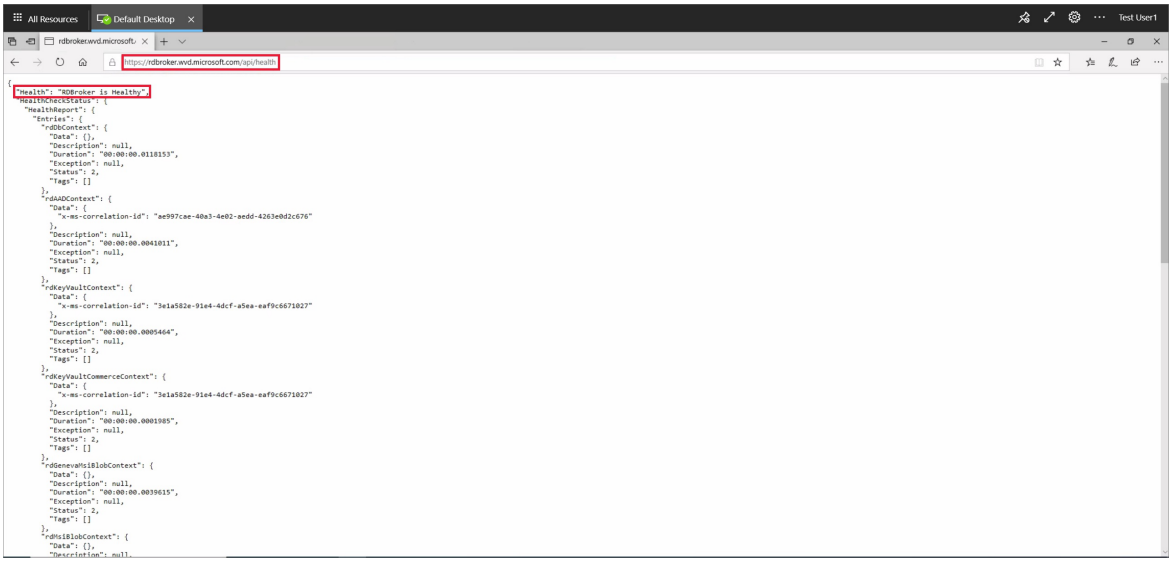
   - Make sure you use the value from step 3 in the **BrokerURI**. In this section's example, it would be https://rdbroker-g-us-r0.wvd.microsoft.com/api/health.

5. Open another tab in the browser and go to *<BrokerURIGlobal>api/health*.

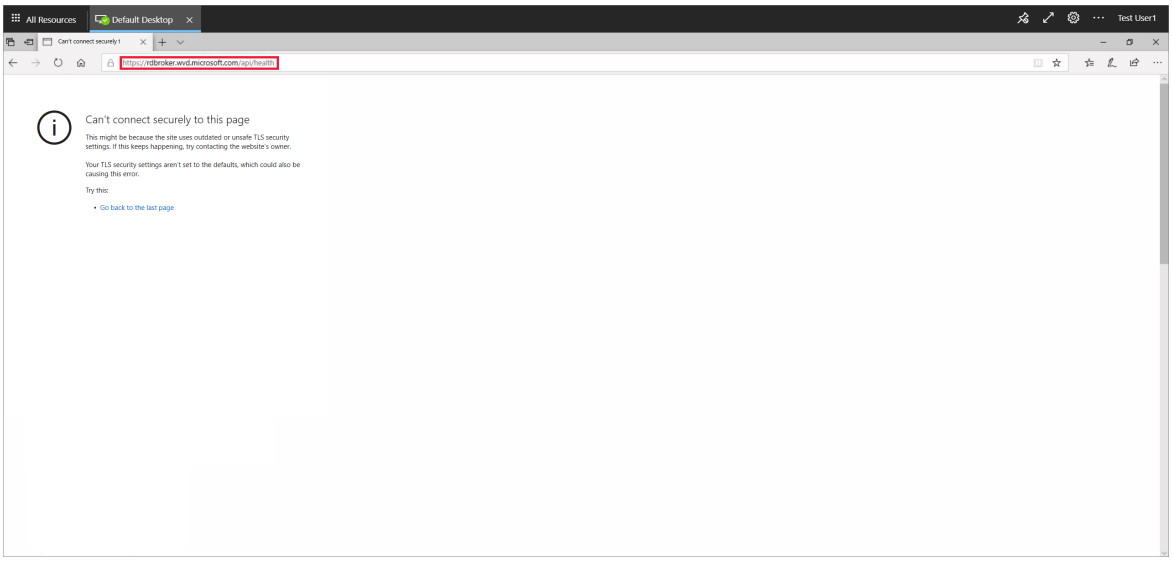
   - Make sure you use the value from step 3 in the **BrokerURIGlobal** link. In this section's example, it would be https://rdbroker.wvd.microsoft.com/api/health.

6. If the network isn't blocking broker connection, both pages will load successfully and will show a message that says **"RD Broker is Healthy"** as shown in the following screenshots.





7. If the network is blocking broker connection, the pages will not load, as shown in the following screenshot.



8. If the network is blocking these URLs, you will need to unblock the required URLs. For more information, see Required URL List.

9. If this does not resolve your issue, make sure that you do not have any group policies with ciphers that block the agent to broker connection. Azure Virtual Desktop uses the same TLS 1.2 ciphers as Azure Front Door. For more information, see Connection Security.

Error: 3703

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3703 that says "RD Gateway Url: is not accessible" in the description, the agent is unable to reach the gateway URLs. To successfully connect to your session host and allow network traffic to these endpoints to bypass restrictions, you must unblock the URLs from the Required URL List. Also, make sure your firewall or proxy settings don't block these URLs. Unblocking these URLs is required to use Azure Virtual Desktop.

To resolve this issue, verify that your firewall and/or DNS settings are not blocking these URLs:

1. Use Azure Firewall to protect Azure Virtual Desktop deployments..
2. Configure your Azure Firewall DNS settings.

Error: 3019

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3019, this means the agent can't reach the web socket transport URLs. To successfully connect to your session host and allow network traffic

to bypass these restrictions, you must unblock the URLs listed in the the Required URL list. Work with the Azure Networking team to make sure your firewall, proxy, and DNS settings aren't blocking these URLs. You can also check your network trace logs to identify where the Azure Virtual Desktop service is being blocked. If you open a support request for this particular issue, make sure to attach your network trace logs to the request.

Error: InstallationHealthCheckFailedException

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277 that says "InstallationHealthCheckFailedException" in the description, that means the stack listener isn't working because the terminal server has toggled the registry key for the stack listener.

To resolve this issue:

1. Check to see if the stack listener is working.
2. If the stack listener isn't working, manually uninstall and reinstall the stack component.

Error: ENDPOINT_NOT_FOUND

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277 that says "ENDPOINT_NOT_FOUND" in the description that means the broker couldn't find an endpoint to establish a connection with. This connection issue can happen for one of the following reasons:

- There aren't VMs in your host pool
- The VMs in your host pool aren't active
- All VMs in your host pool have exceeded the max session limit
- None of the VMs in your host pool have the agent service running on them

To resolve this issue:

1. Make sure the VM is powered on and hasn't been removed from the host pool.
2. Make sure that the VM hasn't exceeded the max session limit.
3. Make sure the agent service is running and the stack listener is working.
4. Make sure the agent can connect to the broker.
5. Make sure your VM has a valid registration token.
6. Make sure the VM registration token hasn't expired.

Error: InstallMsiException

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277, that says **InstallMsiException** in the description, the installer is already running for another application while you're trying to install the agent, or a policy is blocking the msiexec.exe program from running.

To resolve this issue, disable the following policy:

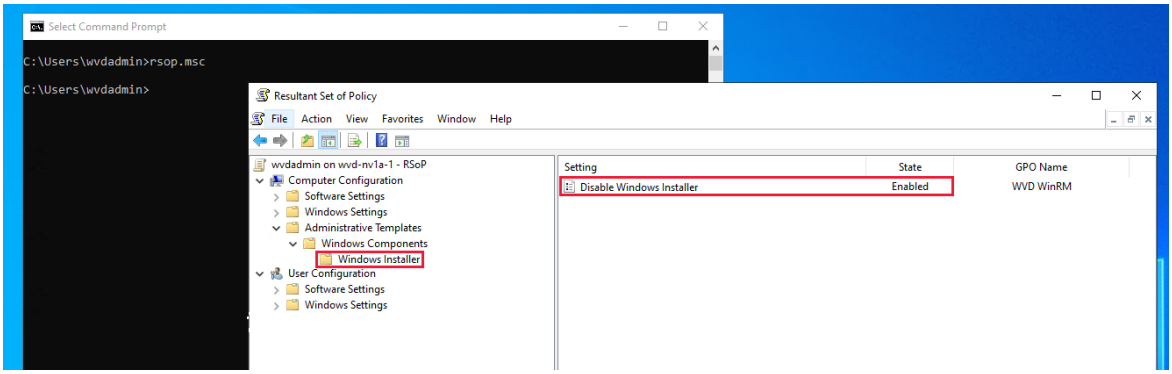
- Turn off Windows Installer
  - Category Path: Computer Configuration\Administrative Templates\Windows Components\Windows Installer

> **NOTE**
> This isn't a comprehensive list of policies, just the ones we're currently aware of.

To disable a policy:

1. Open a command prompt as an administrator.

2. Enter and run **rsop.msc**.
3. In the **Resultant Set of Policy** window that pops up, go to the category path.
4. Select the policy.
5. Select **Disabled**.
6. Select **Apply**.



Error: Win32Exception

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277, that says **InstallMsiException** in the description, a policy is blocking cmd.exe from launching. Blocking this program prevents you from running the console window, which is what you need to use to restart the service whenever the agent updates.

To resolve this issue, disable the following policy:

- Prevent access to the command prompt
  - Category Path: User Configuration\Administrative Templates\System

> **NOTE**
> This isn't a comprehensive list of policies, just the ones we're currently aware of.

To disable a policy:

1. Open a command prompt as an administrator.
2. Enter and run **rsop.msc**.
3. In the **Resultant Set of Policy** window that pops up, go to the category path.
4. Select the policy.
5. Select **Disabled**.
6. Select **Apply**.

Error: Stack listener isn't working on Windows 10 2004 VM

Run **qwinsta** in your command prompt and make note of the version number that appears next to **rdp-sxs**. If you're not seeing the **rdp-tcp** and **rdp-sxs** components say **Listen** next to them or they aren't showing up at all after running **qwinsta**, it means that there's a stack issue. Stack updates get installed along with agent updates, and when this installation goes awry, the Azure Virtual Desktop Listener won't work.

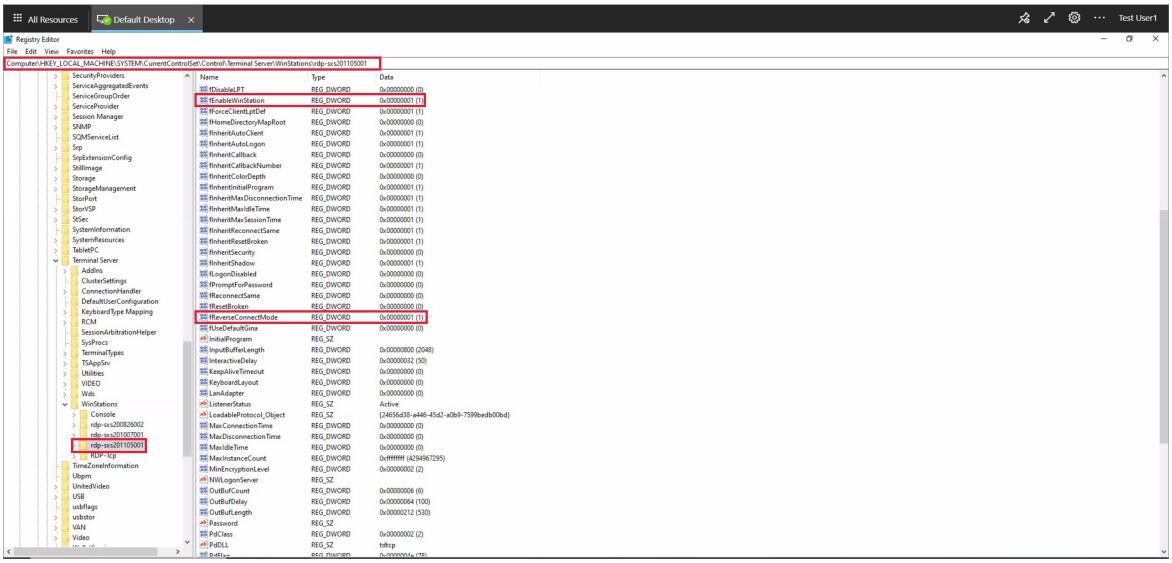
To resolve this issue:

1. Open the Registry Editor.

2. Go to **HKEY_LOCAL_MACHINE** > **SYSTEM** > **CurrentControlSet** > **Control** > **Terminal Server** > **WinStations**.

3. Under **WinStations** you may see several folders for different stack versions, select the folder that matches the version information you saw when running **qwinsta** in your Command Prompt.

4. Find **fReverseConnectMode** and make sure its data value is **1**. Also make sure that **fEnableWinStation** is set to **1**.



5. If **fReverseConnectMode** isn't set to **1**, select **fReverseConnectMode** and enter **1** in its value field.

6. If **fEnableWinStation** isn't set to **1**, select **fEnableWinStation** and enter **1** into its value field.

7. Restart your VM.

> **NOTE**
>
> To change the **fReverseConnectMode** or **fEnableWinStation** mode for multiple VMs at a time, you can do one of the following two things:
>
> - Export the registry key from the machine that you already have working and import it into all other machines that need this change.
> - Create a group policy object (GPO) that sets the registry key value for the machines that need the change.

7. Go to **HKEY_LOCAL_MACHINE** > **SYSTEM** > **CurrentControlSet** > **Control** > **Terminal Server** > **ClusterSettings**.
8. Under **ClusterSettings**, find **SessionDirectoryListener** and make sure its data value is **rdp-sxs...**.
9. If **SessionDirectoryListener** isn't set to **rdp-sxs...**, you'll need to follow the steps in the Uninstall the agent and boot loader section to first uninstall the agent, boot loader, and stack components, and then Reinstall the agent and boot loader. This will reinstall the side-by-side stack.

Error: DownloadMsiException

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3277, that says **DownloadMsiException** in the description, there isn't enough space on the disk for the RDAgent.

To resolve this issue, make space on your disk by:

- Deleting files that are no longer in user
- Increasing the storage capacity of your VM

Error: Agent fails to update with MissingMethodException

Go to **Event Viewer** > **Windows Logs** > **Application**. If you see an event with ID 3389 that says "MissingMethodException: Method not found" in the description, that means the Azure Virtual Desktop agent didn't update successfully and reverted to an earlier version. This may be because the version number of the .NET framework currently installed on your VMs is lower than 4.7.2. To resolve this issue, you need to upgrade the .NET to version 4.7.2 or later by following the installation instructions in the .NET Framework documentation.

Error: VMs are stuck in Unavailable or Upgrading state

Open a PowerShell window as an administrator and run the following cmdlet:

```
Get-AzWvdSessionHost -ResourceGroupName <resourcegroupname> -HostPoolName <hostpoolname> | Select-Object *
```

If the status listed for the session host or hosts in your host pool always says "Unavailable" or "Upgrading," the agent or stack didn't install successfully.

To resolve this issue, reinstall the side-by-side stack:

1. Open a command prompt as an administrator.
2. Enter **net stop RDAgentBootLoader**.
3. Go to **Control Panel** > **Programs** > **Programs and Features**.
4. Uninstall the latest version of the **Remote Desktop Services SxS Network Stack** or the version listed in **HKEY_LOCAL_MACHINE** > **SYSTEM** > **CurrentControlSet** > **Control** > **Terminal Server** > **WinStations** under **ReverseConnectListener**.
5. Open a console window as an administrator and go to **Program Files** > **Microsoft RDInfra**.
6. Select the **SxSStack** component or run the `msiexec /i SxsStack-<version>.msi` command to install the MSI.
7. Restart your VM.
8. Go back to the command prompt and run the **qwinsta** command.
9. Verify that the stack component installed in step 6 says **Listen** next to it.

- If so, enter **net start RDAgentBootLoader** in the command prompt and restart your VM.
- If not, you will need to re-register your VM and reinstall the agent component.

Error: Connection not found: RDAgent does not have an active connection to the broker

Your VMs may be at their connection limit, so the VM can't accept new connections.

To resolve this issue:

- Decrease the max session limit. This ensures that resources are more evenly distributed across session hosts and will prevent resource depletion.
- Increase the resource capacity of the VMs.

Error: Operating a Pro VM or other unsupported OS

The side-by-side stack is only supported by Windows Enterprise or Windows Server SKUs, which means that operating systems like Pro VM aren't. If you don't have an Enterprise or Server SKU, the stack will be installed on your VM but won't be activated, so you won't see it show up when you run **qwinsta** in your command line.

To resolve this issue, create a VM that is Windows Enterprise or Windows Server.

1. Go to Virtual machine details and follow steps 1-12 to set up one of the following recommended images:

- Windows 10 Enterprise multi-session, version 1909
- Windows 10 Enterprise multi-session, version 1909 + Microsoft 365 Apps
- Windows Server 2019 Datacenter
- Windows 10 Enterprise multi-session, version 2004
- Windows 10 Enterprise multi-session, version 2004 + Microsoft 365 Apps

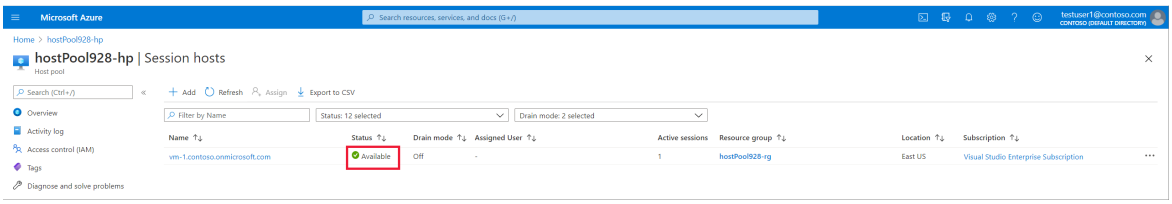
2. Select **Review and Create**.

Error: NAME_ALREADY_REGISTERED

The name of your VM has already been registered and is probably a duplicate.

To resolve this issue:

1. Follow the steps in the Remove the session host from the host pool section.
2. Create another VM. Make sure to choose a unique name for this VM.
3. Go to the Azure portal and open the **Overview** page for the host pool your VM was in.
4. Open the **Session Hosts** tab and check to make sure all session hosts are in that host pool.
5. Wait for 5-10 minutes for the session host status to say **Available**.



Your issue isn't listed here or wasn't resolved

If you can't find your issue in this article or the instructions didn't help you, we recommend you uninstall, reinstall, and re-register Azure Virtual Desktop Agent. The instructions in this section will show you how to reregister your VM to the Azure Virtual Desktop service by uninstalling all agent, boot loader, and stack components, removing the session host from the host pool, generating a new registration key for the VM, and reinstalling the agent and boot loader. If one or more of the following scenarios apply to you, follow these instructions:

- Your VM is stuck in **Upgrading** or **Unavailable**
- Your stack listener isn't working and you're running on Windows 10 1809, 1903, or 1909
- You're receiving an **EXPIRED_REGISTRATION_TOKEN** error
- You're not seeing your VMs show up in the session hosts list
- You don't see the **Remote Desktop Agent Loader** in the Services window
- You don't see the **RdAgentBootLoader** component in the Task Manager
- You're receiving a **Connection Broker couldn't validate the settings** error on custom image VMs
- The instructions in this article didn't resolve your issue

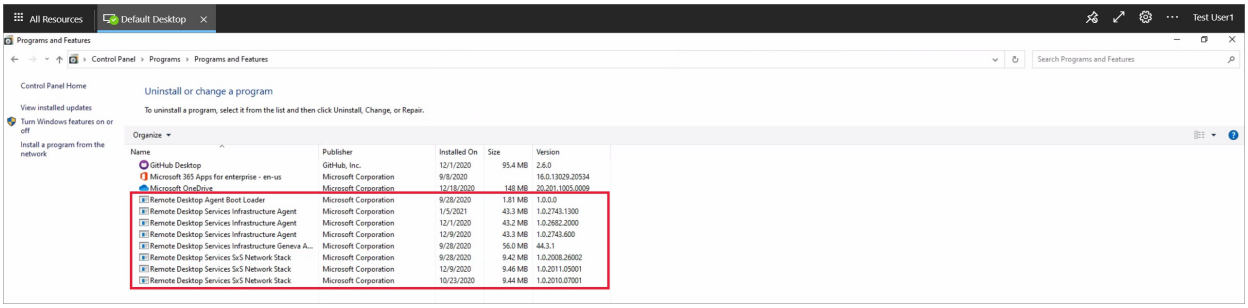
Step 1: Uninstall all agent, boot loader, and stack component programs

Before reinstalling the agent, boot loader, and stack, you must uninstall any existing component programs from your VM. To uninstall all agent, boot loader, and stack component programs:

1. Sign in to your VM as an administrator.
2. Go to **Control Panel** > **Programs** > **Programs and Features**.
3. Remove the following programs:
   - Remote Desktop Agent Boot Loader

- Remote Desktop Services Infrastructure Agent
- Remote Desktop Services Infrastructure Geneva Agent
- Remote Desktop Services SxS Network Stack

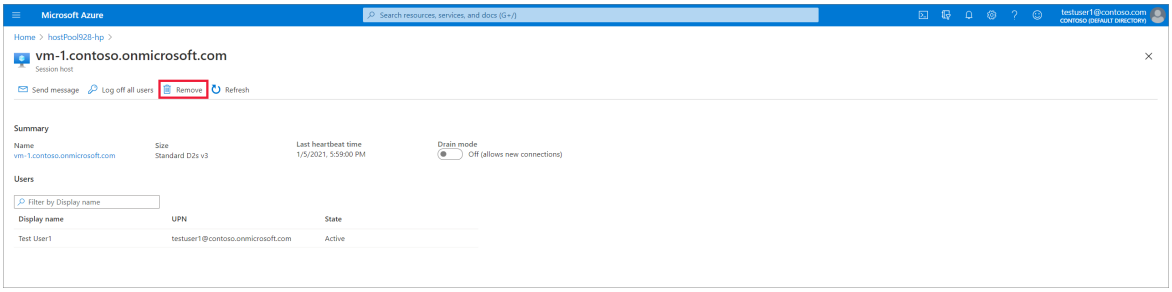
> **NOTE**
> You may see multiple instances of these programs. Make sure to remove all of them.



Step 2: Remove the session host from the host pool

When you remove the session host from the host pool, the session host is no longer registered to that host pool. This acts as a reset for the session host registration. To remove the session host from the host pool:

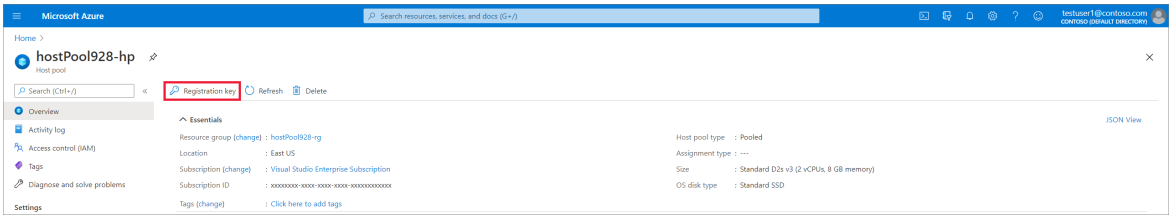
1. Go to the **Overview** page for the host pool that your VM is in, in the Azure portal.
2. Go to the **Session Hosts** tab to see the list of all session hosts in that host pool.
3. Look at the list of session hosts and select the VM that you want to remove.
4. Select **Remove**.



Step 3: Generate a new registration key for the VM

You must generate a new registration key that is used to re-register your VM to the host pool and to the service. To generate a new registration key for the VM:

1. Open the Azure portal and go to the **Overview** page for the host pool of the VM you want to edit.
2. Select **Registration key**.



3. Open the **Registration key** tab and select **Generate new key**.
4. Enter the expiration date and then select **Ok**.

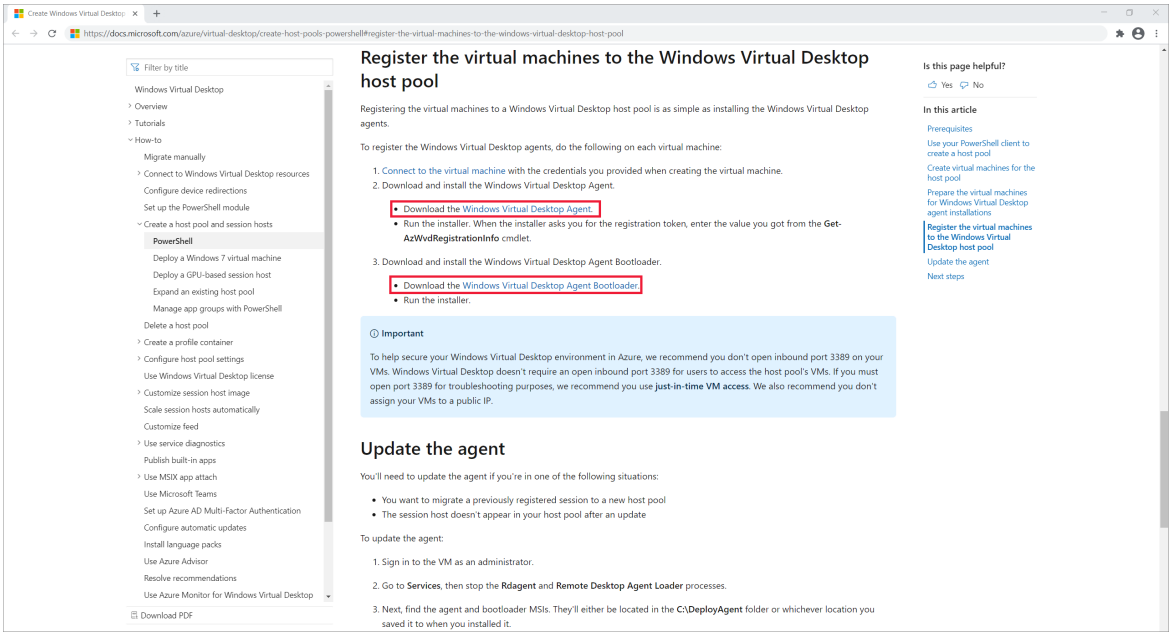
> **NOTE**
>
> The expiration date can be no less than an hour and no longer than 27 days from its generation time and date. We highly recommend you set the expiration date to the 27 day maximum.

5. Copy the newly generated key to your clipboard. You'll need this key later.

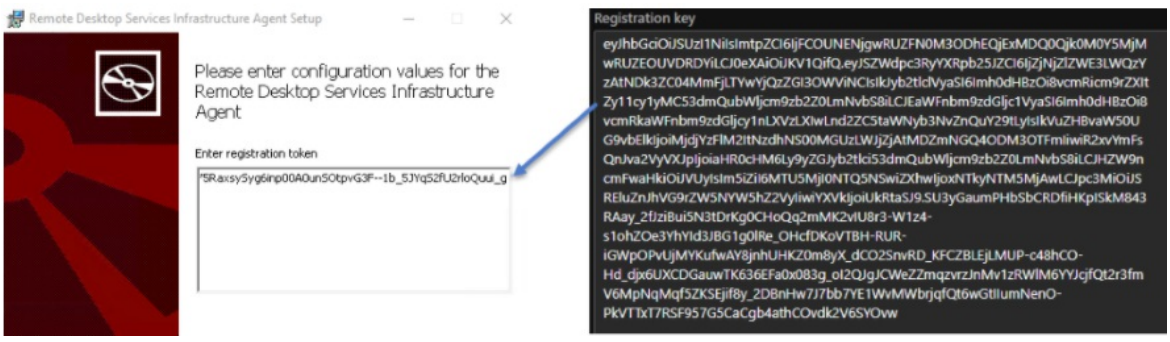
Step 4: Reinstall the agent and boot loader

By reinstalling the most updated version of the agent and boot loader, the side-by-side stack and Geneva monitoring agent automatically get installed as well. To reinstall the agent and boot loader:

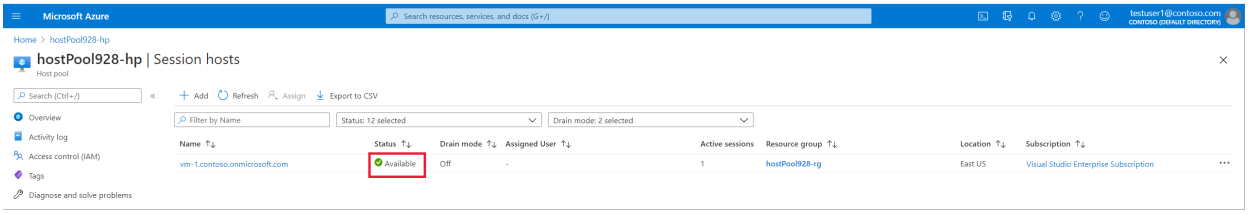
1. Sign in to your VM as an administrator and use the correct version of the agent installer for your deployment depending on which version of Windows your VM is running. If you have a Windows 10 VM, follow the instructions in Register virtual machines to download the **Azure Virtual Desktop Agent** and the **Azure Virtual Desktop Agent Bootloader**. If you have a Windows 7 VM, follow steps 13-14 in Register virtual machines to download the **Azure Virtual Desktop Agent** and the **Azure Virtual Desktop Agent Manager**.



2. Right-click the agent and boot loader installers you downloaded.
3. Select **Properties**.
4. Select **Unblock**.
5. Select **Ok**.
6. Run the agent installer.
7. When the installer asks you for the registration token, paste the registration key from your clipboard.



8. Run the boot loader installer.
9. Restart your VM.
10. Go to the Azure portal and open the **Overview** page for the host pool your VM belongs to.
11. Go to the **Session Hosts** tab to see the list of all session hosts in that host pool.
12. You should now see the session host registered in the host pool with the status **Available**.



Next steps

If the issue continues, create a support case and include detailed information about the problem you're having and any actions you've taken to try to resolve it. The following list includes other resources you can use to troubleshoot issues in your Azure Virtual Desktop deployment.

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while creating a host pool in a Azure Virtual Desktop environment, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues with Azure Virtual Desktop client connections, see Azure Virtual Desktop service connections.
- To troubleshoot issues with Remote Desktop clients, see Troubleshoot the Remote Desktop client.
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To learn more about the service, see Azure Virtual Desktop environment.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.
- To learn about auditing actions, see Audit operations with Resource Manager.
- To learn about actions to determine the errors during deployment, see View deployment operations.

Azure Virtual Desktop service connections

12/6/2021 • 2 minutes to read • Edit Online

IMPORTANT

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see this article.

Use this article to resolve issues with Azure Virtual Desktop client connections.

Provide feedback

You can give us feedback and discuss the Azure Virtual Desktop Service with the product team and other active community members at the Azure Virtual Desktop Tech Community.

User connects but nothing is displayed (no feed)

A user can start Remote Desktop clients and is able to authenticate, however the user doesn't see any icons in the web discovery feed.

1. Confirm that the user reporting the issues has been assigned to application groups by using this command line:

   ```
   Get-AzRoleAssignment -SignInName <userupn>
   ```

2. Confirm that the user is signing in with the correct credentials.
3. If the web client is being used, confirm that there are no cached credentials issues.
4. If the user is part of an Azure Active Directory (AD) user group, make sure the user group is a security group instead of a distribution group. Azure Virtual Desktop doesn't support Azure AD distribution groups.

User loses existing feed and no remote resource is displayed (no feed)

This error usually appears after a user moved their subscription from one Azure AD tenant to another. As a result, the service loses track of their user assignments, since those are still tied to the old Azure AD tenant.

To resolve this, all you need to do is reassign the users to their app groups.

This could also happen if a CSP Provider created the subscription and then transferred to the customer. To resolve this re-register the Resource Provider.

1. Sign in to the Azure portal.
2. Go to **Subscription**, then select your subscription.
3. In the menu on the left side of the page, select **Resource provider**.
4. Find and select **Microsoft.DesktopVirtualization**, then select **Re-register**.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while creating a Azure Virtual Desktop environment and host pool in a Azure Virtual Desktop environment, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.

Troubleshoot the Remote Desktop client

12/6/2021 • 3 minutes to read • Edit Online

This article describes common issues with the Remote Desktop client and how to fix them.

Remote Desktop client for Windows 7 or Windows 10 stops responding or cannot be opened

Starting with version 1.2.790, you can reset the user data from the About page or using a command.

Use the following command to remove your user data, restore default settings and unsubscribe from all Workspaces.

```
msrdcw.exe /reset [/f]
```

If you're using an earlier version of the Remote Desktop client, we recommend you uninstall and reinstall the client.

Web client won't open

First, test your internet connection by opening another website in your browser; for example, www.bing.com.

Use **nslookup** to confirm DNS can resolve the FQDN:

```
nslookup rdweb.wvd.microsoft.com
```

Try connecting with another client, like Remote Desktop client for Windows 7 or Windows 10, and check to see if you can open the web client.

Can't open other websites while connected to the web client

If you can't open other websites while you're connected to the web client, there might be network connection problems or a network outage. We recommend you contact network support.

Nslookup can't resolve the name

If nslookup can't resolve the name, then there might be network connection problems or a network outage. We recommend you contact network support.

Your client can't connect but other clients on your network can connect

If your browser starts acting up or stops working while you're using the web client, follow these instructions to troubleshoot it:

1. Restart the browser.
2. Clear browser cookies. See How to delete cookie files in Internet Explorer.
3. Clear browser cache. See clear browser cache for your browser.
4. Open browser in Private mode.

Client doesn't show my resources

First, check the Azure Active Directory account you're using. If you've already signed in with a different Azure Active Directory account than the one you want to use for Azure Virtual Desktop, you should either sign out or

use a private browser window.

If you're using Azure Virtual Desktop (classic), use the web client link in this article to connect to your resources.

If that doesn't work, make sure your app group is associated with a workspace.

Web client stops responding or disconnects

Try connecting using another browser or client.

Other browsers and clients also malfunction or fail to open

If issues continue even after you've switched browsers, the problem may not be with your browser, but with your network. We recommend you contact network support.

Web client keeps prompting for credentials

If the Web client keeps prompting for credentials, follow these instructions:

1. Confirm the web client URL is correct.
2. Confirm that the credentials you're using are for the Azure Virtual Desktop environment tied to the URL.
3. Clear browser cookies. For more information, see How to delete cookie files in Internet Explorer.
4. Clear browser cache. For more information, see Clear browser cache for your browser.
5. Open your browser in Private mode.

Windows client blocks Azure Virtual Desktop (classic) feed

If the Windows client feed won't show Azure Virtual Desktop (classic) apps, follow these instructions:

1. Check if the Conditional Access policy includes the app IDs associated with Azure Virtual Desktop (classic).
2. Check if the Conditional Access policy blocks all access except Azure Virtual Desktop (classic) app IDs. If so, you'll need to add the app ID **9cdead84-a844-4324-93f2-b2e6bb768d07** to the policy to allow the client to discover the feeds.

If you can't find the app ID 9cdead84-a844-4324-93f2-b2e6bb768d07 in the list, you'll need to register the Azure Virtual Desktop resource provider. To register the resource provider:

1. Sign in to the Azure portal.
2. Go to **Subscription**, then select your subscription.
3. In the menu on the left side of the page, select **Resource provider**.
4. Find and select **Microsoft.DesktopVirtualization**, then select **Re-register**.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To troubleshoot issues while creating a Azure Virtual Desktop environment and host pool in a Azure Virtual Desktop environment, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.

Diagnose graphics performance issues in Remote Desktop

12/6/2021 • 4 minutes to read • Edit Online

To diagnose experience quality issues with your remote sessions, counters have been provided under the RemoteFX Graphics section of Performance Monitor. This article helps you pinpoint and fix graphics-related performance bottlenecks during Remote Desktop Protocol (RDP) sessions using these counters.

Find your remote session name

You'll need your remote session name to identify the graphics performance counters. Follow the instructions in this section to identify your instance of each counter.

1. Open the Windows command prompt from your remote session.
2. Run the **qwinsta** command and find your session name.
   - If your session is hosted in a multi-session virtual machine (VM): Your instance of each counter is suffixed by the same number that suffixes your session name, such as "rdp-tcp 37."
   - If your session is hosted in a VM that supports virtual Graphics Processing Units (vGPU): Your instance of each counter is stored on the server instead of in your VM. Your counter instances include the VM name instead of the number in the session name, such as "Win8 Enterprise VM."

> **NOTE**
> While counters have RemoteFX in their names, they include remote desktop graphics in vGPU scenarios as well.

Access performance counters

After you've determined your remote session name, follow these instructions to collect the RemoteFX Graphics performance counters for your remote session.

1. Select **Start** > **Administrative Tools** > **Performance Monitor**.
2. In the **Performance Monitor** dialog box, expand **Monitoring Tools**, select **Performance Monitor**, and then select **Add**.
3. In the **Add Counters** dialog box, from the **Available Counters** list, expand the section for RemoteFX Graphics.
4. Select the counters to be monitored.
5. In the **Instances of selected object** list, select the specific instances to be monitored for the selected counters and then select **Add**. To select all available counter instances, select **All instances**.
6. After adding the counters, select **OK**.

The selected performance counters will appear on the Performance Monitor screen.

> **NOTE**
> Each active session on a host has its own instance of each performance counter.

Diagnose issues

Graphics-related performance issues generally fall into four categories:

- Low frame rate
- Random stalls
- High input latency
- Poor frame quality

Addressing low frame rate, random stalls, and high input latency

First check the Output Frames/Second counter. It measures the number of frames made available to the client. If this value is less than the Input Frames/Second counter, frames are being skipped. To identify the bottleneck, use the Frames Skipped/Second counters.

There are three types of Frames Skipped/Second counters:

- Frames Skipped/Second (Insufficient Server Resources)
- Frames Skipped/Second (Insufficient Network Resources)
- Frames Skipped/Second (Insufficient Client Resources)

A high value for any of the Frames Skipped/Second counters implies that the problem is related to the resource the counter tracks. For example, if the client doesn't decode and present frames at the same rate the server provides the frames, the Frames Skipped/Second (Insufficient Client Resources) counter will be high.

If the Output Frames/Second counter matches the Input Frames/Second counter, yet you still notice unusual lag or stalling, Average Encoding Time may be the culprit. Encoding is a synchronous process that occurs on the server in the single-session (vGPU) scenario and on the VM in the multi-session scenario. Average Encoding Time should be under 33 ms. If Average Encoding Time is under 33 ms but you still have performance issues, there may be an issue with the app or operating system you are using.

For more information about diagnosing app-related issues, see User Input Delay performance counters.

Because RDP supports an Average Encoding Time of 33 ms, it supports an input frame rate up to 30 frames/second. Note that 33 ms is the maximum supported frame rate. In many cases, the frame rate experienced by the user will be lower, depending on how often a frame is provided to RDP by the source. For example, tasks like watching a video require a full input frame rate of 30 frames/second, but less computationally intensive tasks like infrequently editing a document result in a much lower value for Input Frames/Second with no degradation in the user's experience quality.

Addressing poor frame quality

Use the Frame Quality counter to diagnose frame quality issues. This counter expresses the quality of the output frame as a percentage of the quality of the source frame. The quality loss may be due to RemoteFX, or it may be inherent to the graphics source. If RemoteFX caused the quality loss, the issue may be a lack of network or server resources to send higher-fidelity content.

Mitigation

If server resources are causing the bottleneck, try one of the following approaches to improve performance:

- Reduce the number of sessions per host.
- Increase the memory and compute resources on the server.
- Drop the resolution of the connection.

If network resources are causing the bottleneck, try one of the following approaches to improve network availability per session:

- Reduce the number of sessions per host.
- Use a higher-bandwidth network.

- Drop the resolution of the connection.

If client resources are causing the bottleneck, try one of the following approaches to improve performance:

- Install the most recent Remote Desktop client.
- Increase memory and compute resources on the client machine.

> **NOTE**
> We currently don't support the Source Frames/Second counter. For now, the Source Frames/Second counter will always display 0.

Next steps

- To create a GPU optimized Azure virtual machine, see Configure graphics processing unit (GPU) acceleration for Azure Virtual Desktop environment.
- For an overview of troubleshooting and escalation tracks, see Troubleshooting overview, feedback, and support.
- To learn more about the service, see Windows Desktop environment.

Connections to Azure AD-joined VMs

12/6/2021 • 4 minutes to read • Edit Online

> **IMPORTANT**
> This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects.

Use this article to resolve issues with connections to Azure Active Directory (Azure AD)-joined VMs in Azure Virtual Desktop.

Provide feedback

Visit the Azure Virtual Desktop Tech Community to discuss the Azure Virtual Desktop service with the product team and active community members.

All clients

Your account is configured to prevent you from using this device

If you come across an error saying **Your account is configured to prevent you from using this device. For more information, contact your system administrator**, ensure the user account was given the Virtual Machine User Login role on the VMs.

I can't sign in, even though I'm using the right credentials

If you can't sign in and keep receiving an error message that says your credentials are incorrect, first make sure you're using the right credentials. If you keep seeing error messages, ask yourself the following questions:

- Does your Conditional Access policy exclude multifactor authentication requirements for the Azure Windows VM sign-in cloud application?
- Have you assigned the **Virtual Machine User Login** role-based access control (RBAC) permission to the VM or resource group for each user?

If you answered "no" to either of these questions, follow the instructions in Enable multifactor authentication to reconfigure your multifactor authentication.

> **WARNING**
> VM sign-ins don't support per-user enabled or enforced Azure AD multifactor authentication. If you try to sign in with multifactor authentication on a VM, you won't be able to sign in and will receive an error message.

If you can access your Azure AD sign-in logs through Log Analytics, you can see if you've enabled multifactor authentication and which Conditional Access policy is triggering the event. The events shown are non-interactive user login events for the VM, which means the IP address will appear to come from the external IP address that your VM accesses Azure AD from.

You can access your sign-in logs by running the following Kusto query:

```
let UPN = "userupn";
AADNonInteractiveUserSignInLogs
| where UserPrincipalName == UPN
| where AppId == "38aa3b87-a06d-4817-b275-7a316988d93b"
| project ['Time']=(TimeGenerated), UserPrincipalName, AuthenticationRequirement, ['MFA
Result']=ResultDescription, Status, ConditionalAccessPolicies, DeviceDetail, ['Virtual Machine
IP']=IPAddress, ['Cloud App']=ResourceDisplayName
| order by ['Time'] desc
```

Windows Desktop client

The logon attempt failed

If you come across an error saying **The logon attempt failed** on the Windows Security credential prompt, verify the following:

- You are on a device that is Azure AD-joined or hybrid Azure AD-joined to the same Azure AD tenant as the session host OR
- You are on a device running Windows 10 2004 or later that is Azure AD registered to the same Azure AD tenant as the session host
- The PKU2U protocol is enabled on both the local PC and the session host
- Per-user multifactor authentication is disabled for the user account as it's not supported for Azure AD-joined VMs.

The sign-in method you're trying to use isn't allowed

If you come across an error saying **The sign-in method you're trying to use isn't allowed. Try a different sign-in method or contact your system administrator**, you have Conditional Access policies restricting access. Follow the instructions in Enable multifactor authentication to enable multifactor authentication for your Azure AD-joined VMs.

Web client

Sign in failed. Please check your username and password and try again

If you come across an error saying **Oops, we couldn't connect to NAME. Sign in failed. Please check your username and password and try again.** when using the web client, ensure that you enabled connections from other clients.

We couldn't connect to the remote PC because of a security error

If you come across an error saying **Oops, we couldn't connect to NAME. We couldn't connect to the remote PC because of a security error. If this keeps happening, ask your admin or tech support for help.**, you have Conditional Access policies restricting access. Follow the instructions in Enable multifactor authentication to enable multifactor authentication for your Azure AD-joined VMs.

Android client

Error code 2607 - We couldn't connect to the remote PC because your credentials did not work

If you come across an error saying **We couldn't connect to the remote PC because your credentials did not work. The remote machine is AADJ joined.** with error code 2607 when using the Android client, ensure that you enabled connections from other clients.

Next steps

- For an overview on troubleshooting Azure Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.

- To troubleshoot issues while creating an Azure Virtual Desktop environment and host pool in an Azure Virtual Desktop environment, see Environment and host pool creation.
- To troubleshoot issues while configuring a virtual machine (VM) in Azure Virtual Desktop, see Session host virtual machine configuration.
- To troubleshoot issues related to the Azure Virtual Desktop agent or session connectivity, see Troubleshoot common Azure Virtual Desktop Agent issues.
- To troubleshoot issues when using PowerShell with Azure Virtual Desktop, see Azure Virtual Desktop PowerShell.
- To go through a troubleshoot tutorial, see Tutorial: Troubleshoot Resource Manager template deployments.

Troubleshoot Azure Monitor for Azure Virtual Desktop

12/6/2021 • 3 minutes to read • Edit Online

This article presents known issues and solutions for common problems in Azure Monitor for Azure Virtual Desktop.

Issues with configuration and setup

If the configuration workbook isn't working properly to automate setup, you can use these resources to set up your environment manually:

- To manually enable diagnostics or access the Log Analytics workspace, see Send Azure Virtual Desktop diagnostics to Log Analytics.
- To install the Log Analytics extension on a session host manually, see Log Analytics virtual machine extension for Windows.
- To set up a new Log Analytics workspace, see Create a Log Analytics workspace in the Azure portal.
- To add, remove, or edit performance counters, see Configuring performance counters.
- To configure Windows Event Logs for a Log Analytics workspace, see Collect Windows event log data sources with Log Analytics agent.

My data isn't displaying properly

If your data isn't displaying properly, check the following common solutions:

- First, make sure you've set up correctly with the configuration workbook as described in Use Azure Monitor for Azure Virtual Desktop to monitor your deployment. If you're missing any counters or events, the data associated with them won't appear in the Azure portal.
- Check your access permissions & contact the resource owners to request missing permissions; anyone monitoring Azure Virtual Desktop requires the following permissions:
  - Read-access to the Azure subscriptions that hold your Azure Virtual Desktop resources
  - Read-access to the subscription's resource groups that hold your Azure Virtual Desktop session hosts
  - Read-access to whichever Log Analytics workspaces you're using
- You may need to open outgoing ports in your server's firewall to allow Azure Monitor and Log Analytics to send data to the portal. To learn how to do this, see the following articles: - Azure Monitor Outgoing ports - Log Analytics Firewall Requirements.
- Not seeing data from recent activity? You may want to wait for 15 minutes and refresh the feed. Azure Monitor has a 15-minute latency period for populating log data. To learn more, see Log data ingestion time in Azure Monitor.

If you're not missing any information but your data still isn't displaying properly, there may be an issue in the query or the data sources. Review known issues and limitations.

I want to customize Azure Monitor for Azure Virtual Desktop

Azure Monitor for Azure Virtual Desktop uses Azure Monitor Workbooks. Workbooks lets you save a copy of the Azure Virtual Desktop workbook template and make your own customizations.

By design, custom Workbook templates will not automatically adopt updates from the products group. For more

information, see Troubleshooting workbook-based insights and the Workbooks overview.

I can't interpret the data

Learn more about data terms at the Azure Monitor for Window Virtual Desktop glossary.

The data I need isn't available

If you want to monitor more Performance counters or Windows Event Logs, you can enable them to send diagnostics info to your Log Analytics workspace and monitor them in **Host Diagnostics: Host browser**.

- To add performance counters, see Configuring performance counters
- To add Windows Events, see Configuring Windows Event Logs

Can't find a data point to help diagnose an issue? Send us feedback!

- To learn how to leave feedback, see Troubleshooting overview, feedback, and support for Azure Virtual Desktop.
- You can also leave feedback for Azure Virtual Desktop at the Azure Virtual Desktop feedback hub.

Known issues and limitations

The following are issues and limitations we're aware of and working to fix:

- You can only monitor one host pool at a time.
- To save favorite settings, you have to save a custom template of the workbook. Custom templates won't automatically adopt updates from the product group.
- The configuration workbook will sometimes show "query failed" errors when loading your selections. Refresh the query, reenter your selection if needed, and the error should resolve itself.
- Some error messages aren't phrased in a user-friendly way, and not all error messages are described in documentation.
- The total sessions performance counter can over-count sessions by a small number and your total sessions may appear to go above your Max Sessions limit.
- Available sessions count doesn't reflect scaling policies on the host pool.
- Do you see contradicting or unexpected connection times? While rare, a connection's completion event can go missing and can impact some visuals and metrics.
- Time to connect includes the time it takes users to enter their credentials; this correlates to the experience but in some cases can show false peaks.

Next steps

- To get started, see Use Azure Monitor for Azure Virtual Desktop to monitor your deployment.
- To estimate, measure, and manage your data storage costs, see Estimate Azure Monitor costs.
- Check out our glossary to learn more about terms and concepts related to Azure Monitor for Azure Virtual Desktop.

Troubleshoot Azure Files authorization

12/6/2021 • 2 minutes to read • Edit Online

This article describes common issues related to Azure Files authentication with Azure Active Directory (AD), and suggestions for how to fix them.

My group membership isn't working

When you add a virtual machine (VM) to an Active Directory Domain Services (AD DS) group, you must restart that VM to activate its membership within the service.

I can't add my storage account to my AD DS

First, check Unable to mount Azure Files with AD credentials to see if your problem is listed there.

Here are the most common reasons users may come across issues:

- Ignoring any warning messages that appear when creating the account in PowerShell. Ignoring warnings may cause the new account to have incorrectly configured settings. To fix this issue, you should delete the domain account that represents the storage account and try again.
- The account is using an incorrect organizational unit (OU). To fix this issue, reenter the OU information with the following syntax:

  ```
  DC=ouname,DC=domainprefix,DC=topleveldomain
  ```

  For example:

  ```
  DC=storageAccounts,DC=wvdcontoso,DC=com
  ```

- If the storage account doesn't instantly appear in your Azure AD, don't worry. It usually takes 30 minutes for a new storage account to sync with Azure AD, so be patient. If the sync doesn't happen after 30 minutes, see the next section.

My AD DS group won't sync to Azure AD

If your storage account doesn't automatically sync with Azure AD after 30 minutes, you'll need to force the sync by using this script.

My storage account says it needs additional permissions

If your storage account needs additional permissions, you may not have permission to access MSIX app attach and FSLogix. To fix this issue, make sure you've assigned one of these permissions to your account:

- The **Storage File Data SMB Share Contributor** RBAC permission.
- The **Read & Execute** and **List folder content** NTFS permissions.

Next steps

If you need to refresh your memory about the Azure Files setup process, see Authorize an account for Azure

Files.

Security best practices

12/6/2021 • 10 minutes to read • Edit Online

Azure Virtual Desktop is a managed virtual desktop service that includes many security capabilities for keeping your organization safe. In a Azure Virtual Desktop deployment, Microsoft manages portions of the services on the customer's behalf. The service has many built-in advanced security features, such as Reverse Connect, which reduce the risk involved with having remote desktops accessible from anywhere.

This article describes additional steps you can take as an admin to keep your customers' Azure Virtual Desktop deployments secure.

Security responsibilities

What makes cloud services different from traditional on-premises virtual desktop infrastructures (VDIs) is how they handle security responsibilities. For example, in a traditional on-premises VDI, the customer would be responsible for all aspects of security. However, in most cloud services, these responsibilities are shared between the customer and the company.

When you use Azure Virtual Desktop, it's important to understand that while some components come already secured for your environment, you'll need to configure other areas yourself to fit your organization's security needs.

Here are the security needs you're responsible for in your Azure Virtual Desktop deployment:

SECURITY NEED	IS THE CUSTOMER RESPONSIBLE FOR THIS?
Identity	Yes
User devices (mobile and PC)	Yes
App security	Yes
Session host OS	Yes
Deployment configuration	Yes
Network controls	Yes
Virtualization control plane	No
Physical hosts	No
Physical network	No
Physical datacenter	No

The security needs the customer isn't responsible for are handled by Microsoft.

Azure security best practices

Azure Virtual Desktop is a service under Azure. To maximize the safety of your Azure Virtual Desktop

deployment, you should make sure to secure the surrounding Azure infrastructure and management plane as well. To secure your infrastructure, consider how Azure Virtual Desktop fits into your larger Azure ecosystem. To learn more about the Azure ecosystem, see Azure security best practices and patterns.

This section describes best practices for securing your Azure ecosystem.

Enable Microsoft Defender for Cloud

We recommend enabling Microsoft Defender for Cloud's enhanced security features to:

- Manage vulnerabilities.
- Assess compliance with common frameworks like PCI.
- Strengthen the overall security of your environment.

To learn more, see Enable enhanced security features.

Improve your Secure Score

Secure Score provides recommendations and best practice advice for improving your overall security. These recommendations are prioritized to help you pick which ones are most important, and the Quick Fix options help you address potential vulnerabilities quickly. These recommendations also update over time, keeping you up to date on the best ways to maintain your environment's security. To learn more, see Improve your Secure Score in Microsoft Defender for Cloud.

Azure Virtual Desktop security best practices

Azure Virtual Desktop has many built-in security controls. In this section, you'll learn about security controls you can use to keep your users and data safe.

Require multi-factor authentication

Requiring multi-factor authentication for all users and admins in Azure Virtual Desktop improves the security of your entire deployment. To learn more, see Enable Azure AD Multi-Factor Authentication for Azure Virtual Desktop.

Enable Conditional Access

Enabling Conditional Access lets you manage risks before you grant users access to your Azure Virtual Desktop environment. When deciding which users to grant access to, we recommend you also consider who the user is, how they sign in, and which device they're using.

Collect audit logs

Enabling audit log collection lets you view user and admin activity related to Azure Virtual Desktop. Some examples of key audit logs are:

- Azure Activity Log
- Azure Active Directory Activity Log
- Azure Active Directory
- Session hosts
- Azure Virtual Desktop Diagnostic Log
- Key Vault logs

Use RemoteApps

When choosing a deployment model, you can either provide remote users access to entire virtual desktops or only select applications. Remote applications, or RemoteApps, provide a seamless experience as the user works with apps on their virtual desktop. RemoteApps reduce risk by only letting the user work with with a subset of the remote machine exposed by the application.

Monitor usage with Azure Monitor

Monitor your Azure Virtual Desktop service's usage and availability with Azure Monitor. Consider creating service health alerts for the Azure Virtual Desktop service to receive notifications whenever there's a service impacting event.

Session host security best practices

Session hosts are virtual machines that run inside an Azure subscription and virtual network. Your Azure Virtual Desktop deployment's overall security depends on the security controls you put on your session hosts. This section describes best practices for keeping your session hosts secure.

Enable endpoint protection

To protect your deployment from known malicious software, we recommend enabling endpoint protection on all session hosts. You can use either Windows Defender Antivirus or a third-party program. To learn more, see Deployment guide for Windows Defender Antivirus in a VDI environment.

For profile solutions like FSLogix or other solutions that mount VHD files, we recommend excluding VHD file extensions.

Install an endpoint detection and response product

We recommend you install an endpoint detection and response (EDR) product to provide advanced detection and response capabilities. For server operating systems with Microsoft Defender for Cloud enabled, installing an EDR product will deploy Defender ATP. For client operating systems, you can deploy Defender ATP or a third-party product to those endpoints.

Enable threat and vulnerability management assessments

Identifying software vulnerabilities that exist in operating systems and applications is critical to keeping your environment secure. Microsoft Defender for Cloud can help you identify problem spots through vulnerability assessments for server operating systems. You can also use Defender ATP, which provides threat and vulnerability management for desktop operating systems. You can also use third-party products if you're so inclined, although we recommend using Microsoft Defender for Cloud and Defender ATP.

Patch software vulnerabilities in your environment

Once you identify a vulnerability, you must patch it. This applies to virtual environments as well, which includes the running operating systems, the applications that are deployed inside of them, and the images you create new machines from. Follow your vendor patch notification communications and apply patches in a timely manner. We recommend patching your base images monthly to ensure that newly deployed machines are as secure as possible.

Establish maximum inactive time and disconnection policies

Signing users out when they're inactive preserves resources and prevents access by unauthorized users. We recommend that timeouts balance user productivity as well as resource usage. For users that interact with stateless applications, consider more aggressive policies that turn off machines and preserve resources. Disconnecting long running applications that continue to run if a user is idle, such as a simulation or CAD rendering, can interrupt the user's work and may even require restarting the computer.

Set up screen locks for idle sessions

You can prevent unwanted system access by configuring Azure Virtual Desktop to lock a machine's screen during idle time and requiring authentication to unlock it.

Establish tiered admin access

We recommend you don't grant your users admin access to virtual desktops. If you need software packages, we recommend you make them available through configuration management utilities like Microsoft Endpoint Manager. In a multi-session environment, we recommend you don't let users install software directly.

Consider which users should access which resources

Consider session hosts as an extension of your existing desktop deployment. We recommend you control access to network resources the same way you would for other desktops in your environment, such as using network segmentation and filtering. By default, session hosts can connect to any resource on the internet. There are several ways you can limit traffic, including using Azure Firewall, Network Virtual Appliances, or proxies. If you need to limit traffic, make sure you add the proper rules so that Azure Virtual Desktop can work properly.

Manage Office Pro Plus security

In addition to securing your session hosts, it's important to also secure the applications running inside of them. Office Pro Plus is one of the most common applications deployed in session hosts. To improve the Office deployment security, we recommend you use the Security Policy Advisor for Microsoft 365 Apps for enterprise. This tool identifies policies that can you can apply to your deployment for more security. Security Policy Advisor also recommends policies based on their impact to your security and productivity.

Other security tips for session hosts

By restricting operating system capabilities, you can strengthen the security of your session hosts. Here are a few things you can do:

- Control device redirection by redirecting drives, printers, and USB devices to a user's local device in a remote desktop session. We recommend that you evaluate your security requirements and check if these features ought to be disabled or not.
- Restrict Windows Explorer access by hiding local and remote drive mappings. This prevents users from discovering unwanted information about system configuration and users.
- Avoid direct RDP access to session hosts in your environment. If you need direct RDP access for administration or troubleshooting, enable just-in-time access to limit the potential attack surface on a session host.
- Grant users limited permissions when they access local and remote file systems. You can restrict permissions by making sure your local and remote file systems use access control lists with least privilege. This way, users can only access what they need and can't change or delete critical resources.
- Prevent unwanted software from running on session hosts. You can enable App Locker for additional security on session hosts, ensuring that only the apps you allow can run on the host.

Azure Virtual Desktop support for Trusted Launch

Trusted launch are Gen2 Azure VMs with enhanced security features aimed to protect against “bottom of the stack” threats through attack vectors such as rootkits, boot kits, and kernel-level malware. The following are the enhanced security features of trusted launch, all of which are supported in Azure Virtual Desktop. To learn more about trusted launch, visit Trusted launch for Azure virtual machines (preview).

Secure Boot

Secure Boot is a mode that platform firmware supports that protects your firmware from malware-based rootkits and boot kits. This mode only allows signed OSes and drivers to start up the machine.

Monitor boot integrity using Remote Attestation

Remote attestation is a great way to check the health of your VMs. Remote attestation verifies that Measured Boot records are present, genuine, and originate from the Virtual Trusted Platform Module (vTPM). As a health check, it provides cryptographic certainty that a platform started up correctly.

vTPM

A vTPM is a virtualized version of a hardware Trusted Platform Module (TPM), with a virtual instance of a TPM per VM. vTPM enables remote attestation by performing integrity measurement of the entire boot chain of the VM (UEFI, OS, system, and drivers).

We recommend enabling vTPM to use remote attestation on your VMs. With vTPM enabled, you can also enable BitLocker functionality, which provides full-volume encryption to protect data at rest. Any features using vTPM will result in secrets bound to the specific VM. When users connect to the Azure Virtual Desktop service in a pooled scenario, users can be redirected to any VM in the host pool. Depending on how the feature is designed this may have an impact.

> **NOTE**
> BitLocker should not be used to encrypt the specific disk where you're storing your FSLogix profile data.

Virtualization-based Security

Virtualization-based Security (VBS) uses the hypervisor to create and isolate a secure region of memory that's inaccessible to the OS. Hypervisor-Protected Code Integrity (HVCI) and Windows Defender Credential Guard both use VBS to provide increased protection from vulnerabilities.

Hypervisor-Protected Code Integrity

HVCI is a powerful system mitigation that uses VBS to protect Windows kernel-mode processes against injection and execution of malicious or unverified code.

Windows Defender Credential Guard

Windows Defender Credential Guard uses VBS to isolate and protect secrets so that only privileged system software can access them. This prevents unauthorized access to these secrets and credential theft attacks, such as Pass-the-Hash attacks.

Deploy Trusted Launch in your Azure Virtual Desktop environment

Azure Virtual Desktop doesn't currently support automatically configuring Trusted Launch during the host pool setup process. To use trusted launch in your Azure Virtual Desktop environment, you'll need to deploy Trusted Launch normally and then manually add the virtual machine to your desired host pool.

Nested virtualization

The following operating systems support running nested virtualization on Azure Virtual Desktop:

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows 10 Enterprise
- Windows 10 Enterprise multi-session

Windows Defender Application Control

The following operating systems support using Windows Defender Application Control with Azure Virtual Desktop:

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows 10 Enterprise
- Windows 10 Enterprise multi-session

NOTE

When using Windows Defender Access Control, we recommend only targeting policies at the device level. Although it's possible to target policies to individual users, once the policy is applied, it affects all users on the device equally.

Next steps

To learn how to enable multi-factor authentication, see Set up multi-factor authentication.

Estimate Azure Monitor costs

12/6/2021 • 11 minutes to read • Edit Online

Azure Monitor Logs is a service that collects, indexes, and stores data generated by your environment. Because of this, the Azure Monitor pricing model is based on the amount of data that's brought into and processed (or "ingested") by your Log Analytics workspace in gigabytes per day. The cost of a Log Analytics workspace isn't only based on the volume of data collected, but also which Azure payment plan you've selected and how long you choose to store the data your environment generates.

This article will explain the following things to help you understand how pricing in Azure Monitor works:

- How to estimate data ingestion and storage costs upfront before you enable this feature
- How to measure and control your ingestion and storage to reduce costs when using this feature

> **NOTE**
> All sizes and pricing listed in this article are just examples to demonstrate how estimation works. For a more accurate assessment based on your Azure Monitor Log Analytics pricing model and Azure region, see Azure Monitor pricing.

Estimate data ingestion and storage costs

We recommend you use a predefined set of data written as logs in your Log Analytics workspace. In the following example estimates, we'll look at billable data in the default configuration

The predefined datasets for Azure Monitor for Azure Virtual Desktop include:

- Performance counters from the session hosts
- Windows Event Logs from the session hosts
- Azure Virtual Desktop diagnostics from the service infrastructure

Your data ingestion and storage costs depend on your environment size, health, and usage. The example estimates we'll use in this article to calculate the cost ranges you can expect are based on healthy virtual machines running light to power usage, based on our virtual machine sizing guidelines, to calculate a range of data ingestion and storage costs you could expect.

The light usage VM we'll be using in our example includes the following components:

- 4 vCPUs, 1 disk
- 16 sessions per day
- An average session duration of 2 hours (120 minutes)
- 100 processes per session

The power usage VM we'll be using in our example includes the following components:

- 6 vCPUs, 1 disk
- 6 sessions per day
- Average session duration of 4 hours (240 minutes)
- 200 processes per session

Estimating performance counter ingestion

Performance counters show how the system resources are performing. Performance counter data ingestion depends on your environment size and usage. In most cases, performance counters should make up 80 to 99% of your data ingestion for Azure Monitor for Azure Virtual Desktop.

Before you start estimating, it's important that you understand that each performance counter sends data at a specific frequency. We set a default sample rate-per-minute (you can also edit this rate in your settings), but that rate will be applied at different multiplying factors depending on the counter. The following factors affect the rate:

- For the per virtual machine (VM) factor, each counter sends data per VM in your environment at the default sample rate per minute while the VM is running. You can estimate the number of records these counters send per day by multiplying the default sample rate per minute by the number of VMs in your environment, then multiplying that number by the average VM running time per day.

  To summarize:

  Default sample rate per minute × number of CPU cores in the VM SKU × number of VMs × average VM running time per day = number of records sent per day

- For the per CPU factor, each counter sends at the default sample rate per minute per vCPU in each VM in your environment while the VM is running. You can estimate the number of records the counters will send per day by multiplying the default sample rate per minute by the number of CPU cores in the VM SKU, then multiplying that number by the number of minutes the VM runs and the number of VMs in your environment.

  To summarize:

  Default sample rate per minute × number of CPU cores in the VM SKU × number of minutes the VM runs × number of VMs = number of records sent per day

- For the per disk factor, each counter sends data at the default sample rate for each disk in each VM in your environment. The number of records these counters will send per day equals the default sample rate per minute multiplied by number of disks in the VM SKU, multiplied by 60 minutes per hour, and finally multiplied by the average active hours for a VM.

  To summarize:

  Default sample rate per minute × number of disks in VM SKU × 60 minutes per hour × number of VMs × average VM running time per day = number of records sent per day

- For the per session factor, each counter sends data at the default sample rate for each session in your environment while the session is connected. You can estimate the number of records these counters will send per day can by multiplying the default sample rate per minute by the average number of sessions per day and the average session duration.

  To summarize:

  Default sample rate per minute × sessions per day × average session duration = number of records sent per day

- For the per-process factor, each counter sends data at the default rate for each process in each session in your environment. You can estimate the number of records these counters will send per day by multiplying the default sample rate per minute by the average number of sessions per day, then multiplying that by the average session duration and the average number of processes per session.

  To summarize:

  Default sample rate per minute × sessions per day × average session duration × average number of processes per session = number of records sent per day

The following table lists the 20 performance counters Azure Monitor for Azure Virtual Desktop collects and their default rates:

COUNTER NAME	DEFAULT SAMPLE RATE	FREQUENCY FACTOR
Logical Disk(C:)\% free space	60 seconds	Per disk
Logical Disk(C:)\Avg. Disk Queue Length	30 seconds	Per disk
Logical Disk(C:)\Avg. Disk sec/Transfer	60 seconds	Per disk
Logical Disk(C:)\Current Disk Queue Length	30 seconds	Per disk
Memory(*)\Available Mbytes	30 seconds	Per VM
Memory(*)\Page Faults/sec	30 seconds	Per VM
Memory(*)\Pages/sec	30 seconds	Per VM
Memory(*)\% Committed Bytes in Use	30 seconds	Per VM
PhysicalDisk(*)\Avg. Disk Queue Length	30 seconds	Per disk
PhysicalDisk(*)\Avg. Disk sec/Read	30 seconds	Per disk
PhysicalDisk(*)\Avg. Disk sec/Transfer	30 seconds	Per disk
PhysicalDisk(*)\Avg. Disk sec/Write	30 seconds	Per disk
Processor Information(_Total)\% Processor Time	30 seconds	Per core/CPU
Terminal Services(*)\Active Sessions	60 seconds	Per VM
Terminal Services(*)\Inactive Sessions	60 seconds	Per VM
Terminal Services(*)\Total Sessions	60 seconds	Per VM
User Input Delay per Process(*)\Max Input Delay	30 seconds	Per process
User Input Delay per Session(*)\Max Input Delay	30 seconds	Per session
RemoteFX Network(*)\Current TCP RTT	30 seconds	Per VM
RemoteFX Network(*)\Current UDP Bandwidth	30 seconds	Per VM

If we estimate each record size to be 200 bytes, an example VM running a light workload on the default sample rate would send roughly 90 megabytes of performance counter data per day per VM. Meanwhile, an example

VM running a power workload would send roughly 130 megabytes of performance counter data per day per VM. However, record size and environment usage can vary, so the megabytes per day your deployment uses may be different.

To learn more about input delay performance counters, see User Input Delay performance counters.

Estimating Windows Event Log ingestion

Windows Event Logs are data sources collected by Log Analytics agents on Windows virtual machines. You can collect events from standard logs like System and Application as well as custom logs created by applications you need to monitor.

These are the default Windows Events for Azure Monitor for Azure Virtual Desktop:

- Application
- Microsoft-Windows-TerminalServices-RemoteConnectionManager/Admin
- Microsoft-Windows-TerminalServices-LocalSessionManager/Operational
- System
- Microsoft-FSLogix-Apps/Operational
- Microsoft-FSLogix-Apps/Admin

Windows Events send whenever the terms of the event are met in the environment. Machines in healthy states will send fewer events than machines in unhealthy states. Since event count is unpredictable, we use a range of 1,000 to 10,000 events per VM per day based on examples from healthy environments for this estimate. For example, if we estimate each event record size in this example to be 1,500 bytes, this comes out to roughly 2 to 15 megabytes of event data per day for the specified environment.

To learn more about Windows events, see Windows event records properties.

Estimating diagnostics ingestion

The diagnostics service creates activity logs for both user and administrative actions.

These are the names of the activity logs the diagnostic counter tracks:

- WVDCheckpoints
- WVDConnections
- WVDErrors
- WVDFeeds
- WVDManagement
- WVDAgentHealthStatus

The service sends diagnostic information whenever the environment meets the terms required to make a record. Since diagnostic record count is unpredictable, we use a range of 500 to 1000 events per VM per day based on examples from healthy environments for this estimate.

For example, if we estimate each diagnostic record size in this example to be 200 bytes, then the total ingested data would be less than 1 MB per VM per day.

To learn more about the activity log categories, see Azure Virtual Desktop diagnostics.

Estimating total costs

Finally, let's estimate the total cost. In this example, let's say we come up with the following results based on the example values in the previous sections:

DATA SOURCE	SIZE ESTIMATE PER DAY (IN MEGABYTES)
Performance counters	90-130
Events	2-15
Azure Virtual Desktop diagnostics	< 1

In this example, the total ingested data for Azure Monitor for Azure Virtual Desktop is between 92 to 145 megabytes per VM per day. In other words, every 31 days, each VM ingests roughly 3 to 5 gigabytes of data.

Using the default Pay-as-you-go model for Log Analytics pricing, you can estimate the Azure Monitor data collection and storage cost per month. Depending on your data ingestion, you may also consider the Capacity Reservation model for Log Analytics pricing.

Manage your data ingestion to reduce costs

This section will explain how to measure and manage data ingestion to reduce costs.

To learn about managing rights and permissions to the workbook, see Access control.

> **NOTE**
> Removing data points will impact their corresponding visuals in Azure Monitor for Azure Virtual Desktop.

Log Analytics settings

Here are some suggestions to optimize your Log Analytics settings to manage data ingestion:

- Use a designated Log Analytics workspace for your Azure Virtual Desktop resources to ensure that Log Analytics only collects performance counters and events for the virtual machines in your Azure Virtual Desktop deployment.
- Adjust your Log Analytics storage settings to manage costs. You can reduce the retention period, evaluate whether a fixed storage pricing tier would be more cost-effective, or set boundaries on how much data you can ingest to limit impact of an unhealthy deployment. To learn more, see Manage usage and costs for Azure Monitor Logs.

Remove excess data

Our default configuration is the only set of data we recommend for Azure Monitor for Azure Virtual Desktop. You always have the option to add additional data points and view them in the Host Diagnostics: Host browser or build custom charts for them, however added data will increase your Log Analytics cost. These can be removed for cost savings.

Measure and manage your performance counter data

Your true monitoring costs will depend on your environment size, usage, and health. To understand how to measure data ingestion in your Log Analytics workspace, see Understanding ingested log data volume.

The performance counters the session hosts use will probably be your largest source of ingested data for Azure Monitor for Azure Virtual Desktop. The following custom query template for a Log Analytics workspace can track frequency and megabytes ingested per performance counter over the last day:

```
let WVDHosts = dynamic(['Host1.MyCompany.com', 'Host2.MyCompany.com']);
Perf
| where TimeGenerated > ago(1d)
| where Computer in (WVDHosts)
| extend PerfCounter = strcat(ObjectName, ":", CounterName)
| summarize Records = count(TimeGenerated), InstanceNames = dcount(InstanceName), Bytes=sum(_BilledSize) by
PerfCounter
| extend Billed_MBytes = Bytes / (1024 * 1024), BytesPerRecord = Bytes / Records
| sort by Records desc
```

> **NOTE**
> Make sure to replace the template's placeholder values with the values your environment uses, otherwise the query won't work.

This query will show all performance counters you have enabled on the environment, not just the default ones for Azure Monitor for Azure Virtual Desktop. This information can help you understand which areas to target to reduce costs, like reducing a counter's frequency or removing it altogether.

You can also reduce costs by removing performance counters. To learn how to remove performance counters or edit existing counters to reduce their frequency, see Configuring performance counters.

Manage Windows Event Logs

Windows Events are unlikely to cause a spike in data ingestion when all hosts are healthy. An unhealthy host can increase the number of events sent to the log, but the information can be critical to fixing the host's issues. We recommend keeping them. To learn more about how to manage Windows Event Logs, see Configuring Windows Event logs.

Manage diagnostics

Azure Virtual Desktop diagnostics should make up less than 1% of your data storage costs, so we don't recommend removing them. To manage Azure Virtual Desktop diagnostics, Use Log Analytics for the diagnostics feature.

Next steps

Learn more about Azure Monitor for Azure Virtual Desktop at these articles:

- Use Azure Monitor for Azure Virtual Desktop to monitor your deployment.
- Use the glossary to learn more about terms and concepts.
- If you encounter a problem, check out our troubleshooting guide for help.
- Check out Monitoring usage and estimated costs in Azure Monitor to learn more about managing your monitoring costs.

Introduction to Azure Advisor

12/6/2021 • 3 minutes to read • Edit Online

Learn about the key capabilities of Azure Advisor and get answers to frequently asked questions.

What is Advisor?

Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, Reliability (formerly called High availability), and security of your Azure resources.

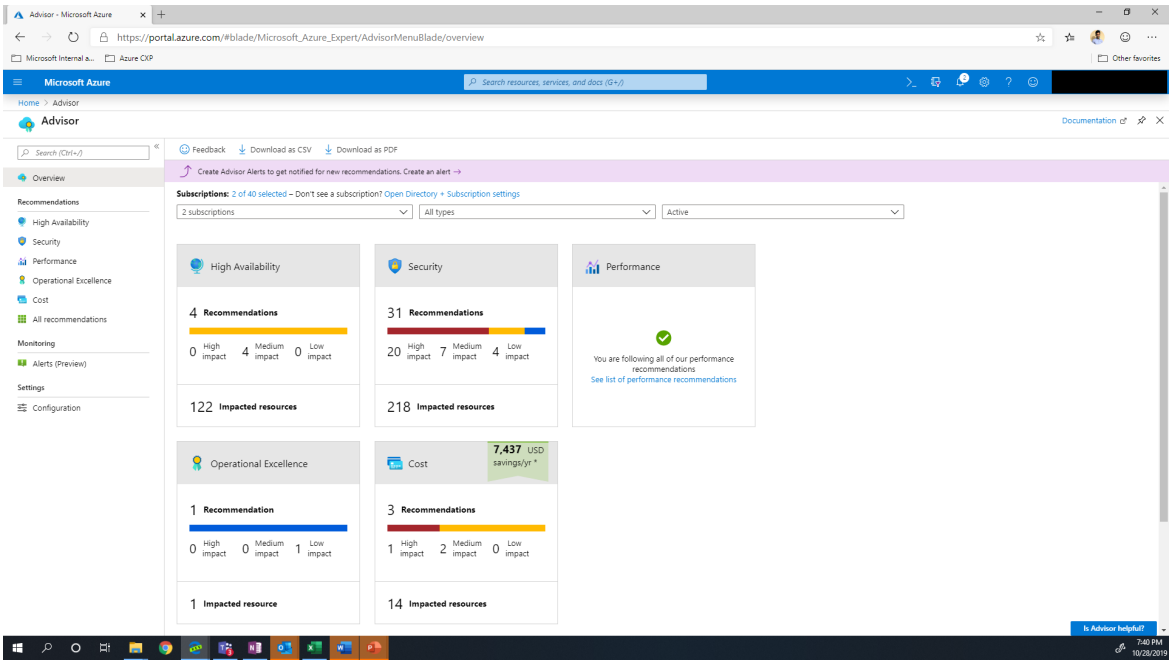
With Advisor, you can:

- Get proactive, actionable, and personalized best practices recommendations.
- Improve the performance, security, and reliability of your resources, as you identify opportunities to reduce your overall Azure spend.
- Get recommendations with proposed actions inline.

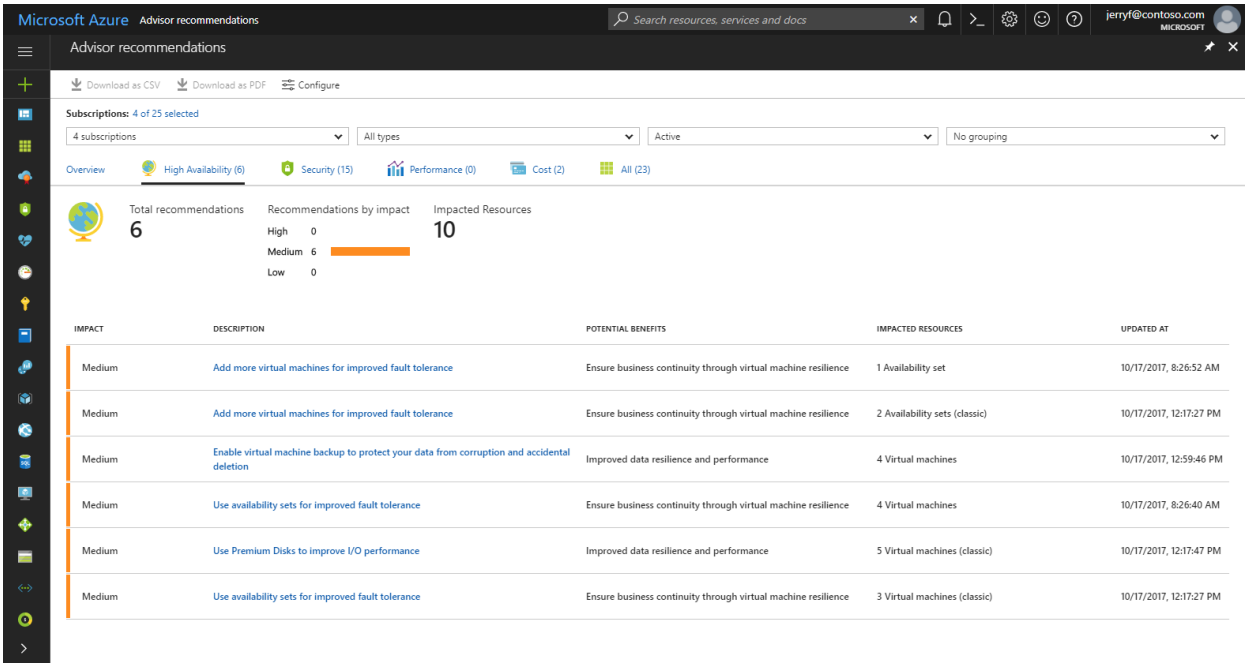
You can access Advisor through the Azure portal. Sign in to the portal, locate **Advisor** in the navigation menu, or search for it in the **All services** menu.

The Advisor dashboard displays personalized recommendations for all your subscriptions. You can apply filters to display recommendations for specific subscriptions and resource types. The recommendations are divided into five categories:

- **Reliability (formerly called High Availability)**: To ensure and improve the continuity of your business-critical applications. For more information, see Advisor Reliability recommendations.
- **Security**: To detect threats and vulnerabilities that might lead to security breaches. For more information, see Advisor Security recommendations.
- **Performance**: To improve the speed of your applications. For more information, see Advisor Performance recommendations.
- **Cost**: To optimize and reduce your overall Azure spending. For more information, see Advisor Cost recommendations.
- **Operational Excellence**: To help you achieve process and workflow efficiency, resource manageability and deployment best practices. . For more information, see Advisor Operational Excellence recommendations.



You can click a category to display the list of recommendations within that category, and select a recommendation to learn more about it. You can also learn about actions that you can perform to take advantage of an opportunity or resolve an issue.



Select the recommended action for a recommendation to implement the recommendation. A simple interface will open that enables you to implement the recommendation or refer you to documentation that assists you with implementation. Once you implement a recommendation, it can take up to a day for Advisor to recognize that.

If you do not intend to take immediate action on a recommendation, you can postpone it for a specified time period or dismiss it. If you do not want to receive recommendations for a specific subscription or resource group, you can configure Advisor to only generate recommendations for specified subscriptions and resource groups.

Frequently asked questions

How do I access Advisor?

You can access Advisor through the Azure portal. Sign in to the portal, locate **Advisor** in the navigation menu,

or search for it in the **All services** menu.

You can also view Advisor recommendations through the virtual machine resource interface. Choose a virtual machine, and then scroll to Advisor recommendations in the menu.

What permissions do I need to access Advisor?

You can access Advisor recommendations as *Owner*, *Contributor*, or *Reader* of a subscription, Resource Group or Resource.

What resources does Advisor provide recommendations for?

Advisor provides recommendations for Application Gateway, App Services, availability sets, Azure Cache, Azure Data Factory, Azure Database for MySQL, Azure Database for PostgreSQL, Azure Database for MariaDB, Azure ExpressRoute, Azure Cosmos DB, Azure public IP addresses, Azure Synapse Analytics, SQL servers, storage accounts, Traffic Manager profiles, and virtual machines.

Azure Advisor also includes your recommendations from Microsoft Defender for Cloud which may include recommendations for additional resource types.

Can I postpone or dismiss a recommendation?

To postpone or dismiss a recommendation, click the **Postpone** link. You can specify a postpone period or select **Never** to dismiss the recommendation.

Next steps

To learn more about Advisor recommendations, see:

- Get started with Advisor
- Advisor score
- Advisor Reliability recommendations
- Advisor Security recommendations
- Advisor Performance recommendations
- Advisor Cost recommendations
- Advisor operational excellence recommendations