

Contents

Windows Virtual Desktop

Overview

[What is Windows Virtual Desktop?](#)

[What's new?](#)

[Windows Virtual Desktop \(classic\) documentation](#)

Tutorials

- [1. Create a host pool with the Azure portal](#)
- [2. Manage app groups](#)
- [3. Create a host pool to validate service updates](#)
- [4. Set up service alerts](#)

How-to

[Connect to Windows Virtual Desktop resources](#)

[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](#)

[Set up the PowerShell module](#)

[Create a host pool and session hosts](#)

[PowerShell](#)

[Deploy a Windows 7 virtual machine](#)

[Deploy a GPU-based session host](#)

[Expand an existing host pool](#)

[Manage app groups with PowerShell](#)

[Delete a host pool](#)

[Create a profile container](#)

[Use a VM-based file share](#)

[Use Azure NetApp Files](#)

[Use Azure Files and Azure AD DS](#)

Use Azure Files and AD DS

Configure host pool settings

RDP properties

Load-balancing for pooled host pools

Personal desktop assignment type

Use Windows Virtual Desktop license

Customize session host image

Set up a master VHD image

Install Office on a master VHD image

Scale session hosts automatically

Customize feed

Use service diagnostics

Use diagnostics with Log Analytics

Publish built-in apps

Set up MSIX app attach

Use Microsoft Teams

Set up Azure Multi-Factor Authentication

Configure automatic updates

Install language packs

Concepts

Safe URL list

Windows Virtual Desktop environment

Determine user connection latency

Delegated access in Windows Virtual Desktop

Host pool load-balancing methods

FSLogix profile containers and Azure files

Storage options for FSLogix profile containers

Partner integrations

Windows Virtual Desktop FAQ

Windows 10 Enterprise multi-session FAQ

MSIX app attach

What is MSIX app attach?

[Glossary](#)

[FAQ](#)

[Data locations](#)

[Troubleshoot](#)

[Troubleshooting overview, feedback, and support](#)

[Identify and diagnose issues](#)

[Host pool creation](#)

[Session host virtual machine configuration](#)

[Remote Desktop service connections](#)

[Remote Desktop client issues](#)

[Windows Virtual Desktop PowerShell](#)

[Diagnosing graphics performance issues](#)

[Reference](#)

[Security best practices](#)

[Linux support](#)

[Supported RDP file settings](#)

[Network guidance](#)

[Virtual machine sizing guidance](#)

[Azure command-line interface \(CLI\)](#)

[Azure example scenarios](#)

[Resources](#)

[PowerShell](#)

[REST API reference](#)

[Experience estimator](#)

[Pricing calculator](#)

[Learning path](#)

[How-to videos](#)

[Tech Community support group](#)

[UserVoice forum](#)

[Microsoft 365 roadmap](#)

[Azure Resource Manager templates](#)

What is Windows Virtual Desktop?

8/25/2020 • 6 minutes to read • [Edit Online](#)

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

Here's what you can do when you run Windows 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 Windows Virtual Desktop, why it's unique, and what's new in this video:

<https://www.youtube.com/embed/NQFtl3JLtaU>

For more videos about Windows Virtual Desktop, see [our playlist](#).

Key capabilities

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

- Create a full desktop virtualization environment in your Azure subscription without having to run any additional 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 Windows 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, Windows 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 Windows Virtual Desktop client to connect users to their published Windows desktops and applications. Connect from any device through either a native application on your device or the Windows 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 Windows 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	RDS Client Access License (CAL) with Software Assurance

Your infrastructure needs the following things to support Windows 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 Windows 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 Windows Virtual Desktop VM is joined to Azure AD Domain Services domain.
 - A Azure AD Domain Services domain. User is sourced from Azure Active Directory, and the Windows 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 Windows Virtual Desktop:

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

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

- [Standard domain-joined](#) or [Hybrid AD-joined](#). Virtual machines can't be Azure AD-joined.
- 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 Windows Virtual Desktop deployment to work as intended, see our [Safe URL list](#).

Windows Virtual Desktop comprises 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 collocated in the same Azure region as the management service.

Supported Remote Desktop clients

The following Remote Desktop clients support Windows Virtual Desktop:

- [Windows Desktop](#)
- [Web](#)
- [macOS](#)
- [iOS](#)
- [Android](#)

IMPORTANT

Windows Virtual Desktop doesn't support the RemoteApp and Desktop Connections (RADC) client or the Remote Desktop Connection (MSTSC) client.

IMPORTANT

Windows Virtual Desktop doesn't currently support the Remote Desktop client from the Windows Store. Support for this client will be added in a future release.

To learn more about URLs you must unblock to use the clients, see the [Safe URL list](#).

Supported virtual machine OS images

Windows Virtual Desktop supports the following x64 operating system images:

- Windows 10 Enterprise multi-session, version 1809 or later
- Windows 10 Enterprise, version 1809 or later

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

Windows Virtual Desktop does not support x86 (32-bit), Windows 10 Enterprise N, 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 10 Enterprise (multi-session), version 2004	Yes	Yes	Yes	Yes
Windows 10 Enterprise (multi-session), version 1909	Yes	Yes	Yes	Yes
Windows 10 Enterprise (multi-session), version 1903	Yes	Yes	No	No
Windows 10 Enterprise (multi-session), version 1809	Yes	Yes	No	No
Windows 7 Enterprise	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 Windows Virtual Desktop (classic), you can get started with our tutorial at [Create a tenant in Windows Virtual Desktop](#).

If you're using the Windows 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 Windows Virtual Desktop?

8/25/2020 • 5 minutes to read • [Edit Online](#)

Windows Virtual Desktop updates on a regular basis. 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.

July 2020

July was when Windows 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 "Windows Virtual Desktop (Classic)," while the "Spring 2020 release" is now just "Windows 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 virtual machines (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 Windows 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 **Windows 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 Windows 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 Windows 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 Windows Virtual Desktop. Most importantly, Windows 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 Windows Virtual Desktop](#).

June 2020

Last month, we introduced Windows 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 Windows Virtual Desktop.

Windows Virtual Desktop is now integrated with Azure Resource Manager

Windows Virtual Desktop is now integrated into Azure Resource Manager. In the latest update, all Windows 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:

- Windows 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 Windows 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 Windows 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 virtual machine (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 Windows 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 Windows 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 Windows Virtual Desktop (Preview)

We've made some improvements to Microsoft Teams for Windows Virtual Desktop. Most importantly, Windows 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 Windows Virtual Desktop roadmap](#).

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

- [Windows](#)
- [macOS](#)
- [iOS](#)
- [Android](#)
- [Web](#)

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

8/25/2020 • 6 minutes to read • [Edit Online](#)

IMPORTANT

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

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

What you need to set up a tenant

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

- The [Azure Active Directory](#) tenant ID for Windows 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 Windows 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 Windows 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 Windows Virtual Desktop

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

Granting permissions to the Windows 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 [Windows 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 Windows 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 [Windows 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
```

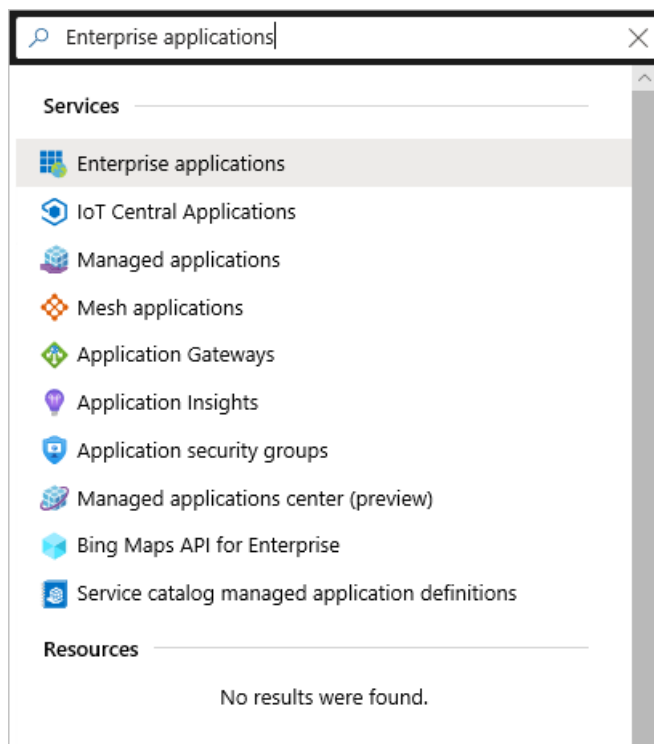
6. Sign in to the Windows Virtual Desktop consent page as global administrator, as you did in step 2.
7. Select **Accept**.

Assign the TenantCreator application role

Assigning an Azure Active Directory user the TenantCreator application role allows that user to create a Windows 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 **Windows Virtual Desktop**. You'll see the two applications that you provided consent for in the previous section. Of these two apps, select **Windows Virtual Desktop**.

NAME	HOMEPAGE URL	OBJECT ID	APPLICATION ID
Windows Virtual Desktop	https://mrs-Pro...	d4fbe527-f98f-4...	5a0aa725-4958-4...
Windows Virtual Desktop Client		62162f31-55cf-4...	fa4345a4-a730-4...

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 Windows Virtual Desktop tenant. Continue following these instructions to add the **TenantCreator** role to a user.

Windows Virtual Desktop - Users and groups

Enterprise Application

+

Add user

Edit

Remove

Update Credentials

Columns

The application will appear on the access panel for assigned users. Set 'visible to users?' to no in properties to prevent this.

First 100 shown, to search all users & groups, enter a display name.

DISPLAY NAME	OBJECT TYPE	ROLE ASSIGNED
AD Admin	User	Default Access

Overview

Getting started

Manage

Properties

Owners

Users and groups

Provisioning

Self-service

Security

Conditional Access

Permissions

Token encryption (Preview)

Activity

Sign-ins

Audit logs

Access reviews

- Select **Add user**, and then select **Users and groups** in the **Add Assignment** tab.
- Search for a user account that will create your Windows 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 Windows Virtual Desktop. We recommend using a domain-specific account like `admin@contoso.com` or `admin@contoso.onmicrosoft.com` instead.

Add Assignment

Contoso

Users and groups

None Selected

Select Role

TenantCreator

Assign

Select member or invite an external user

Search by name or email address

AD AAD DC Administrators

AD Admin admin@contoso.com

Adatum - User1 user1@adatum.com

Selected members:

AD Admin admin@contoso.com

Remove

Select

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 **Windows 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 Windows Virtual Desktop tenant.

Windows Virtual Desktop - Users and groups

Enterprise Application

« + Add user Edit Remove Update Credentials Columns

i The application will appear on the access panel for assigned users. Set 'visible to users?' to no in properties to prevent this. →

First 100 shown, to search all users & groups, enter a display name.

DISPLAY NAME	OBJECT TYPE	ROLE ASSIGNED
AD Admin	User	Default Access
AD Admin	User	TenantCreator

Manage

- Overview
- Getting started
- Properties
- Owners
- Users and groups**
- Provisioning
- Self-service

Security

- Conditional Access
- Permissions
- Token encryption (Preview)

Activity

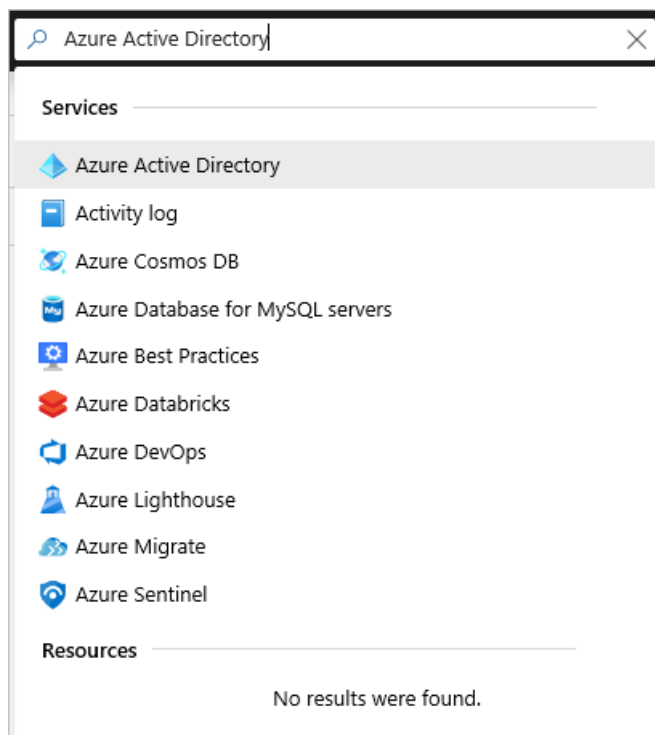
- Sign-ins
- Audit logs
- Access reviews

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

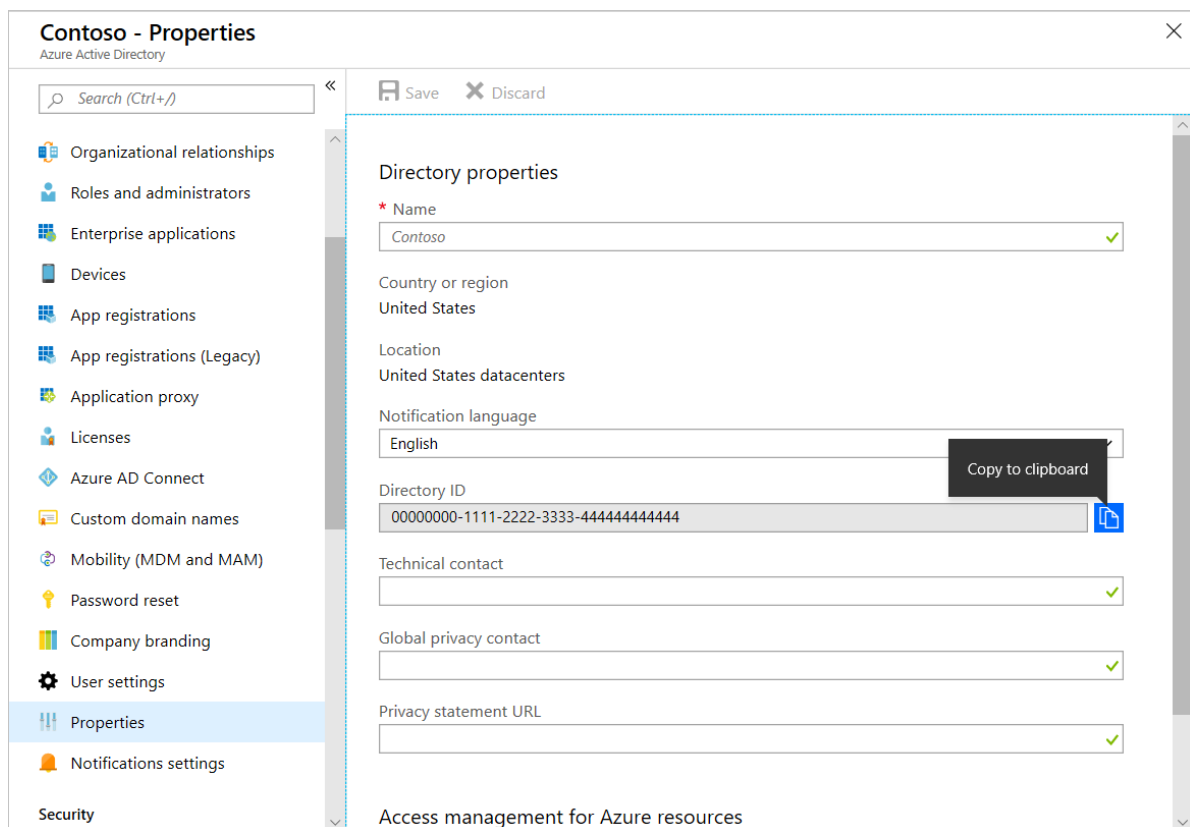
- Your Azure Active Directory tenant ID (or **Directory ID**)
- Your Azure subscription ID

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**.

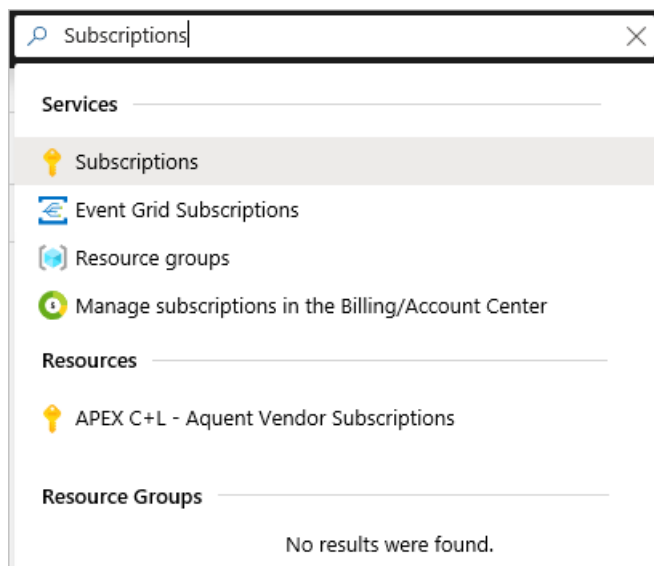


2. Scroll down until you find **Properties**, and then select it.
3. Look for **Directory ID**, and then select the clipboard icon. Paste it in a handy location so you can use it later as the **AadTenantId** value.

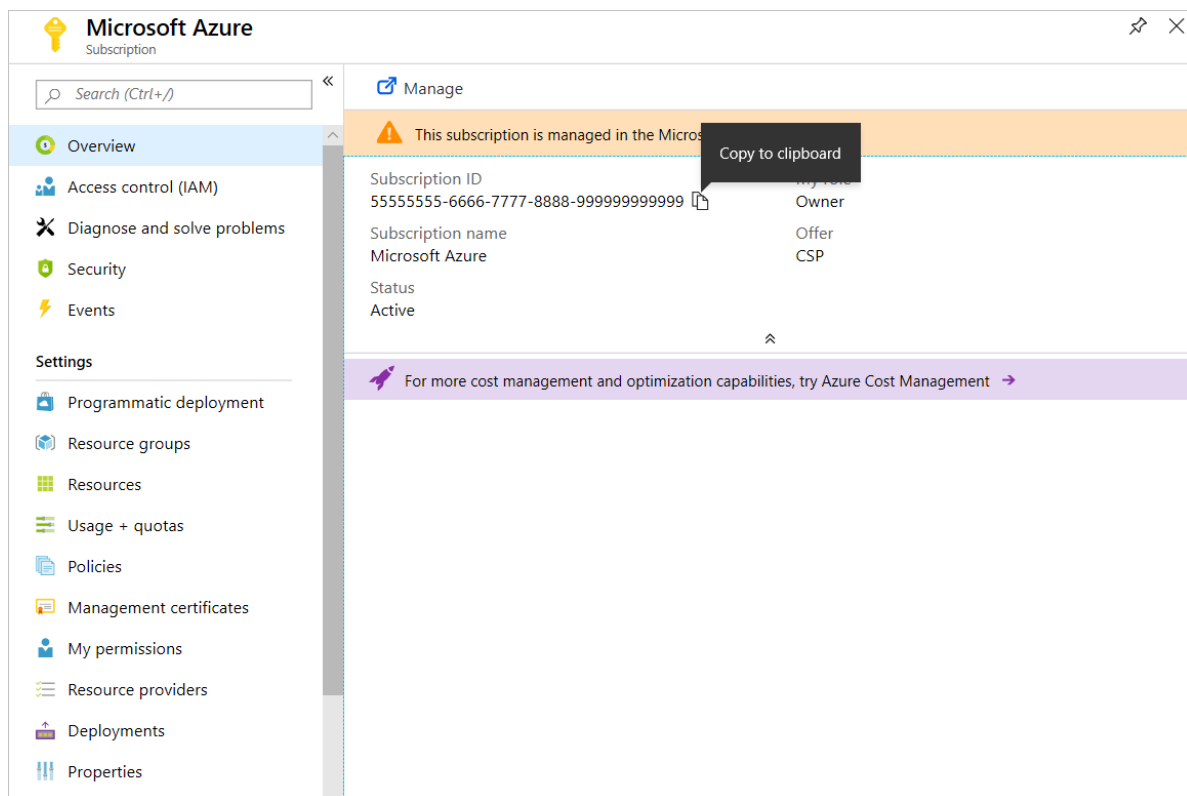


To find your Azure subscription ID:

1. In the same [Azure portal](#) session, search for and select **Subscriptions**.



2. Select the Azure subscription you want to use to receive Windows 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 Windows Virtual Desktop tenant

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

First, [download and import the Windows Virtual Desktop module](#) to use in your PowerShell session if you haven't already.

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

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

After that, create a new Windows 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 Windows Virtual Desktop tenant should be globally unique. For example, let's say you're the Windows 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 Windows Virtual Desktop. The service principal will allow you to successfully deploy the Windows 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 Windows Virtual Desktop.

[Create service principals and role assignments with PowerShell](#)

Tutorial: Create a host pool with the Azure portal

8/25/2020 • 8 minutes to read • [Edit Online](#)

IMPORTANT

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

Host pools are a collection of one or more identical virtual machines (VMs) within Windows Virtual Desktop environments. Each host pool can contain an app group that users can interact with as they would on a physical desktop.

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

Prerequisites

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

- The VM image name
- VM configuration
- Domain and network properties
- Windows 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.

Also, 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.

When you create a Windows 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 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

To start creating your new host pool:

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. Enter **Windows Virtual Desktop** into the search bar, then find and select **Windows Virtual Desktop** under Services.

3. In the **Windows 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.

Project details

Subscription *	Microsoft Azure
Resource group *	Select a resource group
	Create new
Host pool name *	
Location *	(US) East US

Metadata will be stored in East US

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.

Host pool type *	Personal
Assignment type	Automatic
	Automatic
	Direct

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.

Host pool type *	Pooled
Max session limit	Max # of users
Load balancing algorithm	Breadth-first
	Breadth-first
	Depth-first

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 completed the first part, 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.

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

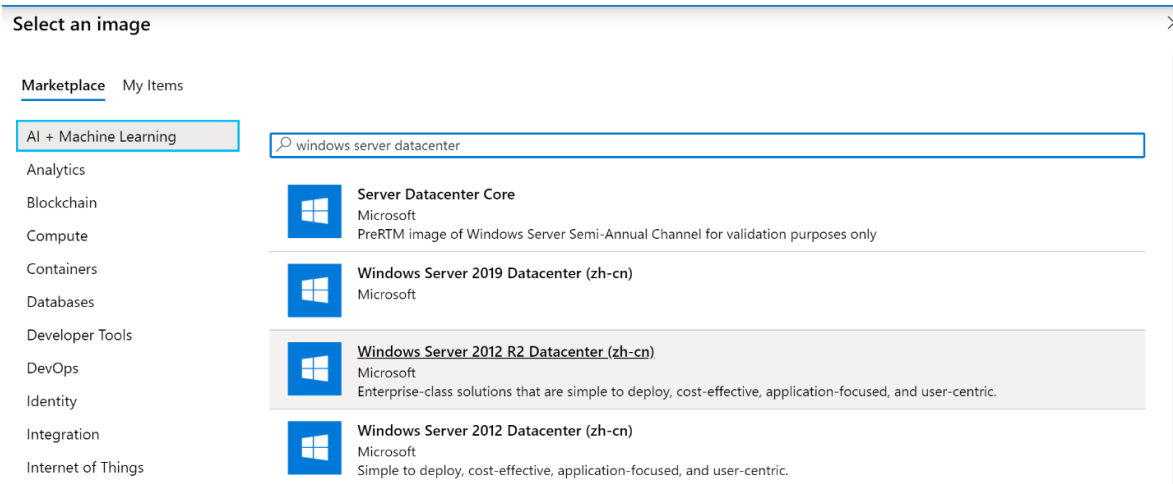
1. Under **Resource group**, choose the resource group where you want to create the virtual machines. This can be a different resource group than the one you used for the host pool.
2. Choose the **Virtual machine location** where you want to create the virtual machines. They can be the same or different from the region you selected for the host pool.
3. Next, 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.
4. Under **Number of VMs**, provide the number of VMs you want to create for your host pool.

NOTE

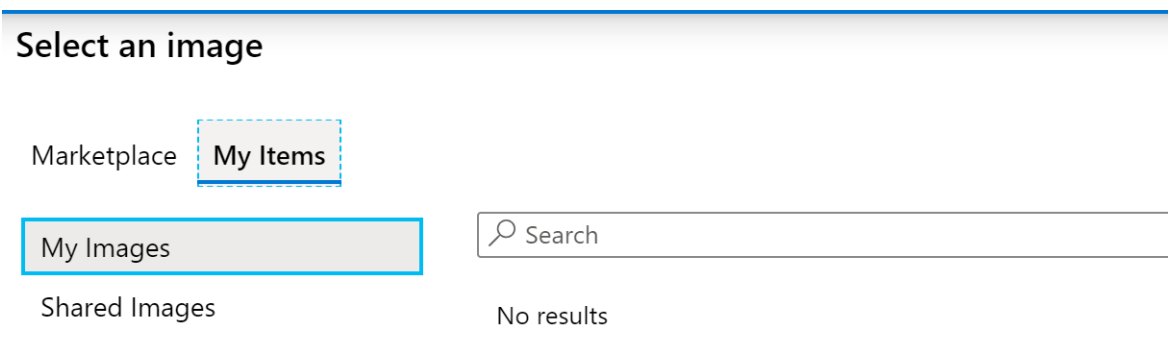
The setup process can create up to 400 VMs while setting up your host pool, and each VM setup process creates four objects in your resource group. Since the creation process doesn't check your subscription quota, make sure the number of VMs you enter is within the Azure VM and API limits for your resource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

5. After that, provide a **Name prefix** to name the virtual machines the setup process creates. The suffix will be with numbers starting from 0.
6. 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 **Browse all images and disks**, which lets you select either another image in your gallery or an image provided by Microsoft and other publishers.



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



- If you choose **Storage Blob**, you can leverage 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.
7. Choose what kind of OS disks you want your VMs to use: Standard SSD, Premium SSD, or Standard HDD.
 8. 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. Next, select whether or not you want a public IP for the virtual machines. We recommend you select **No**, because a private IP is more secure.
 9. 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.

Network security group ⓘ Basic

Public inbound ports ⓘ ☒ Yes ☐ No

Inbound ports to allow

Specify domain or unit ⓘ

Domain to join * ⓘ

Organizational Unit path ⓘ

Select one or more ports

- ☐ HTTP (80)
- ☐ HTTPS (443)
- ☐ SSH (22)
- ☐ RDP (3389)

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

- After that, select whether you want the virtual machines to be joined to a specific domain and organizational unit. If you choose **Yes**, specify the domain to join. You can optionally add a specific organizational unit you want the virtual machines to be in. If you choose **No**, the VMs will be joined to the domain matching the suffix of the **AD domain join UPN**.
- Under Administrator account, enter the credentials for the Active Directory Domain admin of the virtual network you selected.
- 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.

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!

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 Windows Virtual Desktop, head to our next tutorial:

[Manage app groups tutorial](#)

Tutorial: Manage app groups with the Azure portal

8/25/2020 • 4 minutes to read • [Edit Online](#)

IMPORTANT

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

The default app group created for a new Windows 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.

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](#).
2. Search for and select **Windows 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.

Create an application group

Basics Assignments Applications Workspace Tags Review + create

Subscription * ⓘ Microsoft Azure ▼

Resource group * ⓘ Select a resource group ▼
[Create new](#)

Host pool * ⓘ 0224HP ▼

Location ⓘ West US ▼
Metadata stored in same location as host pool

Application group type

RemoteApp application groups are where you can add applications. A Desktop application group will grant full desktop access.

Application group type * ⓘ ☒ RemoteApp ☐ Desktop

Application group name *

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

Application group type

RemoteApp application groups are where you can add applications. A Desktop application group will grant full desktop access.

Application group type * ⓘ ☒ RemoteApp ☐ Desktop

Application group name *

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.

Select Azure AD users or user groups ✕

Select member or invite an external user ⓘ

Search by name or email address ✓

AC	Accountants
AD	ADSyncAdmins
AD	ADSyncAdmins
AD	ADSyncBrowse

Selected members:

RD	RdsDemoUser1 RdsDemoUser1 @RdsPTTen...	Remove
AC	Accountants	Remove

Select

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.

Add application



Select an application from your start menu or add from a file path.

Application source *

Start menu



Application *

Character Map



Display name

Character Map

Description

Application path

C:\windows\system32\charmap.exe

Icon path

C:\windows\system32\charmap.exe

Icon index

0

Show in web feed

☐ No ☒ Yes

Require command line

☒ No ☐ Yes

Save

Cancel

- 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**.

Add application



Select an application from your start menu or add from a file path.

Application source *	<input type="text" value="File path"/>
Application path *	<input type="text" value="C:\windows\system32\charmap.exe"/>
Application name *	<input type="text" value="Character Map"/> ✓
Display name	<input type="text" value="false"/> ✓
Icon path *	<input type="text" value="C:\windows\system32\charmap.exe"/>
Icon index *	<input type="text" value="0"/>
Description	<div></div>
Show in web feed	<input type="radio"/> No <input checked="" type="radio"/> Yes
Require command line	<input checked="" type="radio"/> No <input type="radio"/> Yes

Save

Cancel

- Repeat this process for every application you want to add to the application group.
- Next, select **Next: Workspace** > .
- 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**.
- 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.

[Basics](#) [Assignments](#) [Applications](#) [Workspace](#) [Tags](#) [Review + create](#)

To save some time, you can register the default desktop application group from this host pool, with a new or pre-existing workspace.

Register application group

☐ No ☒ Yes

Register application group ⓘ

0224WS

i Another application group in 0224HP has already been registered, so this app group will also be registered to that 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.

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

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Host pools are a collection of one or more identical virtual machines within Windows Virtual Desktop tenant environments. Before deploying host pools to your production environment, we highly recommend you create a validation host pool. Updates are applied first to validation host pools, letting you monitor service updates before rolling them out to your production environment. Without a validation host pool, you may not discover changes that introduce errors, which could result in downtime for users in your production environment.

To ensure your apps work with the latest updates, the validation host pool should be as similar to host pools in your production environment as possible. Users should connect as frequently to the validation host pool as they do to the production 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 [Windows Virtual Desktop troubleshooting articles](#).

NOTE

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

IMPORTANT

Windows Virtual Desktop with Azure Resource Management integration currently has trouble enabling and disabling validation environments. We'll update this article when we've resolved the issue.

Prerequisites

Before you begin, follow the instructions in [Set up the Windows Virtual Desktop PowerShell module](#) to set up your PowerShell module and sign in to Azure.

Create your host pool

You can create a host pool by following the instructions in any of these articles:

- [Tutorial: Create a host pool with Azure Marketplace](#)
- [Create a host pool with PowerShell](#)

Define your host pool as a validation host pool

Run the following PowerShell cmdlets to define the new host pool as a validation host pool. Replace the values in brackets with the values relevant to your session:

```
Update-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> -ValidationEnvironment:$true
```

Run the following PowerShell cmdlet to confirm that the validation property has been set. Replace the values in brackets with the values relevant to your session.

```
Get-AzWvdHostPool -ResourceGroupName <resourcegroupname> -Name <hostpoolname> | Format-List
```

The results from the cmdlet should look similar to this output:

```
HostPoolName      : hostpoolname
FriendlyName      :
Description       :
Persistent        : False
CustomRdpProperty : use multimon:i:0;
MaxSessionLimit   : 10
LoadBalancerType  : BreadthFirst
ValidationEnvironment : True
```

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 small group of users signing 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 Windows Virtual Desktop deployment.

[Set up service alerts](#)

Tutorial: Set up service alerts

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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 Windows 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 Windows 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 Windows Virtual Desktop. To learn about how to sign in to Windows Virtual Desktop, continue to the [Connect to Windows Virtual Desktop How-tos](#).

[Connect to the Remote Desktop client on Windows 7 and Windows 10](#)

Connect with the Windows Desktop client

8/25/2020 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 7, Windows 10, and Windows 10 IoT Enterprise

IMPORTANT

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

You can access Windows Virtual Desktop resources on devices with Windows 7, Windows 10, and Windows 10 IoT Enterprise using the Windows Desktop client. The client doesn't support Windows 8 or Windows 8.1.

NOTE

The Windows client automatically defaults to Windows Virtual Desktop (classic). However, if the client detects that the user also has Azure Resource Manager resources, it automatically adds the resources or notifies the user that they are available.

IMPORTANT

Windows Virtual Desktop doesn't support the RemoteApp and Desktop Connections (RADC) client or the Remote Desktop Connection (MSTSC) client.

IMPORTANT

Windows Virtual Desktop doesn't currently support the Remote Desktop client from the Windows Store.

Install the Windows Desktop client

Choose the client that matches your version of Windows:

- [Windows 64-bit](#)
- [Windows 32-bit](#)
- [Windows ARM64](#)

You can install the client for the current user, which doesn't require admin rights, or 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**.

Subscribe to a Workspace

There are two ways you can subscribe to a Workspace. The client can try to discover the resources available to you from your work or school account or you can directly specify the URL where your resources are for cases where the client is unable to find them. Once you've subscribed to a Workspace, you can launch resources with one of the following methods:

- Go to the Connection Center and double-click a resource to launch it.

- You can also go to the Start menu and look for a folder with the Workspace name or enter the resource name in the search bar.

Subscribe with a user account

1. From the main page of the client, select **Subscribe**.
2. Sign in with your user account when prompted.
3. The resources will appear in the Connection Center, and are grouped by workspace.

Subscribe with a URL

1. From the main page of the client, select **Subscribe with URL**.
2. Enter the Workspace URL or your email address:
 - If you use the **Workspace URL**, use the one your admin gave you. If accessing resources from Windows Virtual Desktop, you can use one of the following URLs:
 - Windows Virtual Desktop (classic):
`https://rdweb.wvd.microsoft.com/api/feeddiscovery/webfeeddiscovery.aspx`
 - Windows Virtual Desktop: `https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery`
 - If you're using the **Email** field instead, enter your email address. This tells the client to search for a URL associated with your email address if your admin has set up [email discovery](#).
3. Select **Next**.
4. Sign in with your user account when prompted.
5. The resources should appear in the Connection Center, grouped by workspace.

Next steps

To learn more about how to use the Windows Desktop client, check out [Get started with the Windows Desktop client](#).

Connect to Windows Virtual Desktop with the web client

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

The web client lets you access your Windows 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

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

BROWSER	SUPPORTED OS	NOTES
Microsoft Edge	Windows	
Internet Explorer	Windows	Version 11 or later
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 Windows Virtual Desktop web client at <https://rdweb.wvd.microsoft.com/arm/webclient> and sign in with your user account.

NOTE

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

NOTE

If you've already signed in with a different Azure Active Directory account than the one you want to use for Windows 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 Windows Virtual Desktop with the Android client

8/25/2020 • 2 minutes to read • [Edit Online](#)

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

IMPORTANT

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

You can access Windows 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 **Remote Resource Feed**.
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.
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 Windows Virtual Desktop with the macOS client

8/25/2020 • 2 minutes to read • [Edit Online](#)

Applies to: macOS 10.12 or later

IMPORTANT

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

You can access Windows 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.vwd.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.
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 Windows Virtual Desktop with the iOS client

8/25/2020 • 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 Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows Virtual Desktop (classic) without Azure Resource Manager objects, see [this article](#).

You can access Windows 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.
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.

Set up the PowerShell module for Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

The Windows 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 Windows Virtual Desktop.

Set up your PowerShell environment

To get started with using the module, first install the [latest version of PowerShell Core](#). Windows 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
```

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.insightsIÇ³}

Location : southeastasia

DisplayName : Southeast Asia

Providers : {Microsoft.RecoveryServices, Microsoft.ManagedIdentity,  
Microsoft.SqlVirtualMachine, microsoft.insightsIÇ³}

Location : centralus

DisplayName : Central US

Providers : {Microsoft.RecoveryServices, Microsoft.DesktopVirtualization,  
Microsoft.ManagedIdentity, Microsoft.SqlVirtualMachineIÇ³}

Location : eastus

DisplayName : East US

Providers : {Microsoft.RecoveryServices, Microsoft.DesktopVirtualization,  
Microsoft.ManagedIdentity, Microsoft.SqlVirtualMachineIÇ³}
```

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 Windows Virtual Desktop. Here are some of the places you can use your module:

- Run through our to set up your very own Windows Virtual Desktop environment.
- [Create a host pool with PowerShell](#)
- [Configure the Windows 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 a Windows Virtual Desktop host pool with PowerShell

8/25/2020 • 5 minutes to read • [Edit Online](#)

IMPORTANT

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

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

Prerequisites

This article assumes you've already followed the instructions in [Set up the PowerShell module](#).

Use your PowerShell client to create a host pool

Run the following cmdlet to sign in to the Windows 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 -ExpirationHours 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 Windows 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 Windows 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 Windows 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 Windows Virtual Desktop agent installations

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

- You must domain-join the machine. This allows incoming Windows 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 Windows 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](#).

Register the virtual machines to the Windows Virtual Desktop host pool

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

To register the Windows 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 Windows Virtual Desktop Agent.
 - Download the [Windows 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 Windows Virtual Desktop Agent Bootloader.
 - Download the [Windows Virtual Desktop Agent Bootloader](#).
 - Run the installer.

IMPORTANT

To help secure your Windows Virtual Desktop environment in Azure, we recommend you don't open inbound port 3389 on your VMs. Windows 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 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-verx.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 Windows 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 Windows Virtual Desktop, see the Manage app groups tutorial.

[Manage app groups tutorial](#)

Deploy a Windows 7 virtual machine on Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

The process to deploy a Windows 7 virtual machine (VM) on Windows 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 Windows Virtual Desktop.

To set up a Windows 7 VM on Windows 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 Windows Virtual Desktop Agent for Windows 7.](#)

14. [Download the Windows Virtual Desktop Agent Manager for Windows 7.](#)

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

16. Open the Windows 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 Windows Virtual Desktop deployment is now ready to use. [Download the latest version of the Windows Virtual Desktop client](#) to get started.

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

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

8/25/2020 • 5 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows 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 Windows Virtual Desktop tenant configured.

Select a GPU optimized Azure virtual machine size

Azure offers a number of [GPU optimized virtual machine sizes](#). 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.

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](#).

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

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

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 Windows Virtual Desktop](#).

Install supported graphics drivers in your virtual machine

To take advantage of the GPU capabilities of Azure N-series VMs in Windows Virtual Desktop, you must install the appropriate graphics drivers. Follow the instructions at [Supported operating systems and drivers](#) to install drivers from the appropriate graphics vendor, either manually or using an Azure VM extension.

Only drivers distributed by Azure are supported for Windows Virtual Desktop. Additionally, for Azure VMs with NVIDIA GPUs, only [NVIDIA GRID drivers](#) are supported for Windows Virtual Desktop.

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

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 Windows 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 Windows 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

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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 **Windows 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 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

You can't edit the size or image of the VMs because it's important to ensure that all VMs in the host pool are the same size.

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 **Administrator account**, enter the Active Directory domain username and password associated with the virtual network you selected. These credentials will be used to join the virtual machines to the virtual network.

NOTE

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

11. Select the **Tag** tab if you have any tags that you want to group the virtual machines with. Otherwise, skip this tab.
12. 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 Windows 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

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

The default app group created for a new Windows 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

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

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.DesktopVirtualizatio
n/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/startmenuitems
/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/startmenuitems
/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

8/25/2020 • 2 minutes to read • [Edit Online](#)

All host pools created in Windows 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.

Delete a host pool with PowerShell

To delete a host pool using PowerShell, you first need to delete all app groups in the host pool. To delete all app groups, run the following PowerShell cmdlet:

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

Next, run this cmdlet to delete the host pool:

```
Remove-AzWvdHostPool -Name <hostpoolname> -ResourceGroupName <resourcegroupname> -Force:$true
```

This cmdlet removes all existing user sessions on the host pool's session host. It also unregisters the session host from the host pool. Any related virtual machines (VMs) will still exist within your subscription.

Delete a host pool with the Azure portal

To delete a host pool in the Azure portal:

1. Sign in to the [Azure portal](#).
2. Search for and select **Windows 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 Windows 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

8/25/2020 • 3 minutes to read • [Edit Online](#)

The Windows 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 Windows 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 Windows Virtual Desktop Active Directory users to an [Active Directory security group](#). This security group will be used to authenticate the Windows 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 Windows 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 Windows Virtual Desktop environment in Azure, we recommend you don't open inbound port 3389 on your VMs. Windows 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

8/25/2020 • 8 minutes to read • [Edit Online](#)

We recommend using FSLogix profile containers as a user profile solution for the [Windows 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 Windows 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 Windows 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 Windows Virtual Desktop.

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

The instructions in this guide are specifically for Windows 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 Windows 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:

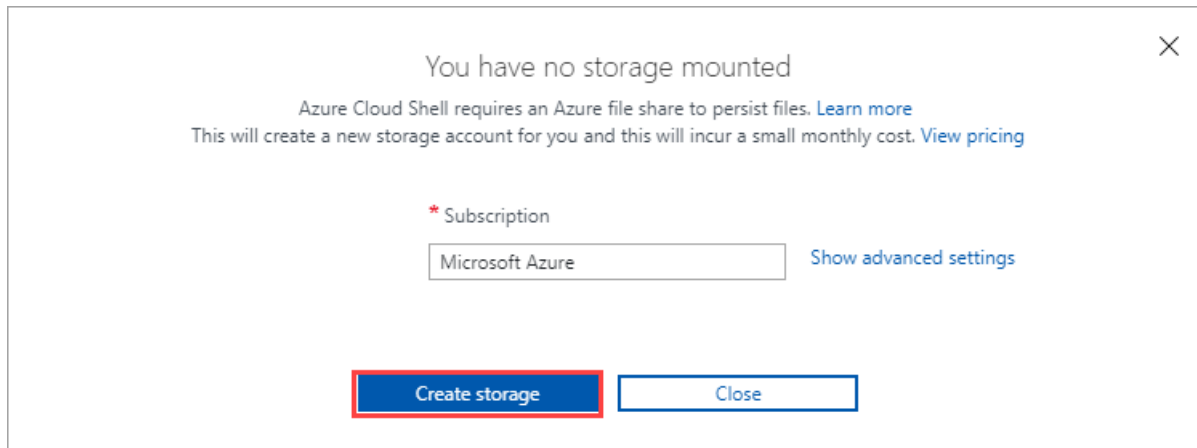
- Set up and configure Windows Virtual Desktop
- Provision a Windows Virtual Desktop host pool
- [Enable your Azure NetApp Files subscription](#)

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 Windows Virtual Desktop.

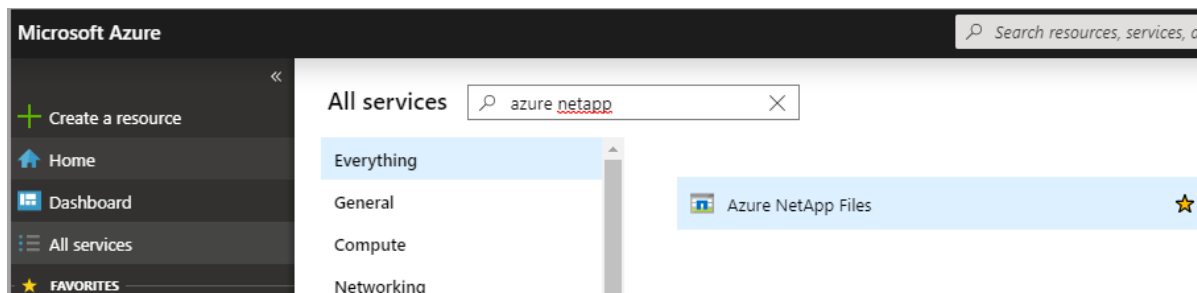


5. Once Azure Cloud Shell loads, run the following two cmdlets.

```
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

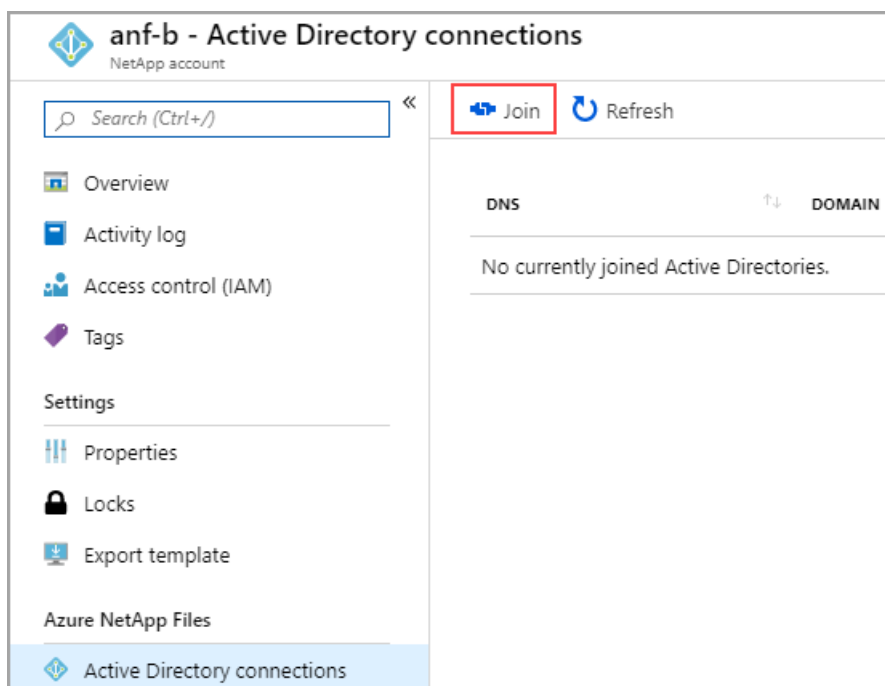
The Premium setting provides the minimum throughput available for a Premium Service level, which is 256 MBps. You may need to adjust this throughput for a production environment. Final throughput is based on the relationship described in [Throughput limits](#).

- 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.

anf-Vol (anf-b/capPool/anf-Vol)
Volume

Search (Ctrl+/) «

Overview

Activity log

Access control (IAM)

Tags

Settings

Properties

Usage

Resize Delete

Resource group : anf-rg

Mount path : \\anf-SMB-3863.gt1107.onmicrosoft.com\\anf-Voll

Subscription : Microsoft Azure

Subscription ID : 00000000-0000-0000-0000-000000000000

Virtual network/subnet : adVNET/anf-SUBNET

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 **DeleteLocalProfileWhenVHDSshouldApply** with a **DWORD** value of 1 to avoid problems with existing local profiles before you sign in.

WARNING

Be careful when creating the **DeleteLocalProfileWhenVHDSshouldApply** 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 Windows Virtual Desktop.
2. Run the following cmdlets:

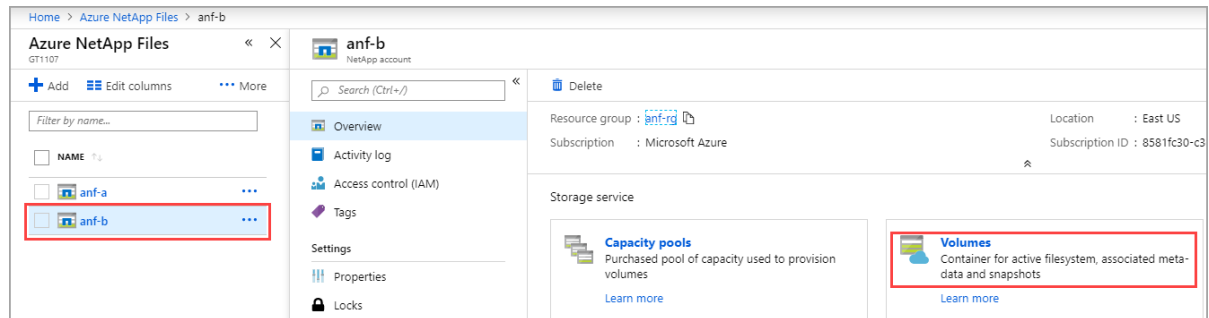
```
Import-Module Microsoft.RdInfra.RdPowershell
# (Optional) Install-Module Microsoft.RdInfra.RdPowershell
$brokerurl = "https://rdbroker.wvd.microsoft.com"
Add-RdsAccount -DeploymentUrl $brokerurl
```

3. When prompted for credentials, enter the credentials for the user with the Tenant Creator or RDS Owner/RDS Contributor roles on the Windows 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
```

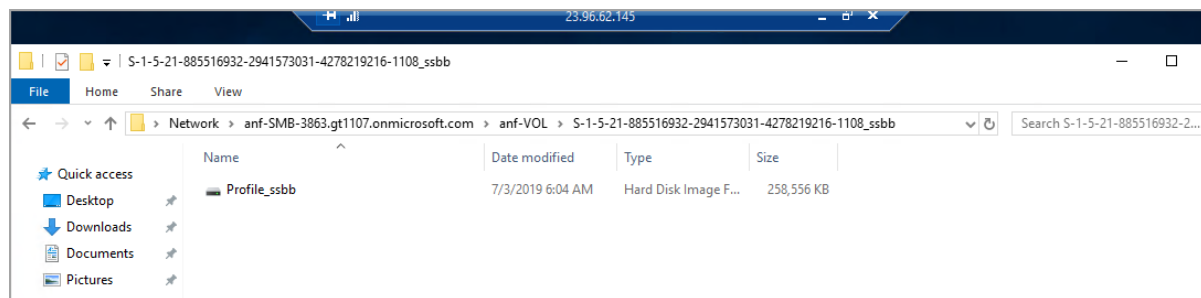
Make sure users can access the Azure NetApp File share

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](#).

Create a profile container with Azure Files and Azure AD DS

8/25/2020 • 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.

Connect to virtual machine

sh

 To improve security, enable just-in-time access on this VM. →

RDP

SSH

To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

◦

* IP address

* Port number

3389

Download RDP File

Having trouble connecting to this VM?

- [Diagnose and solve problems](#)
- [Troubleshoot connection](#)
- [Serial console](#)

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> <storage-account-key> /user:Azure\<storage-account-name>
```

- 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 Windows 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

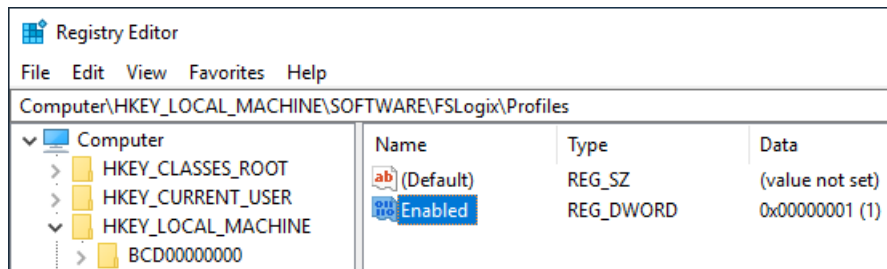
If there are multiple VMs in the host pool, you'll need to repeat steps 1 through 5 for each VM.

6. Run **Registry Editor** (RegEdit) as an administrator.
7. Navigate to **Computer > HKEY_LOCAL_MACHINE > software > FSLogix**, right-click on **FSLogix**, select

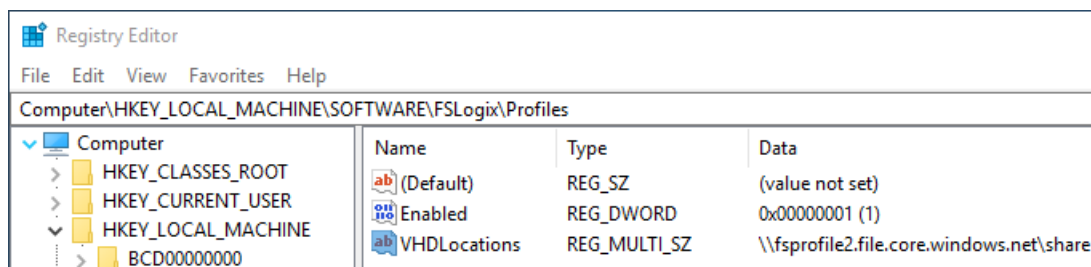
New, and then select **Key**.

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 Windows 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 Windows 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 Windows 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

8/25/2020 • 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 Windows 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 Windows Virtual Desktop host pool.
 - For **Performance**, select **Standard**. (Depending on your IOPS requirements. For more information, see [Storage options for FSLogix profile containers in Windows 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.

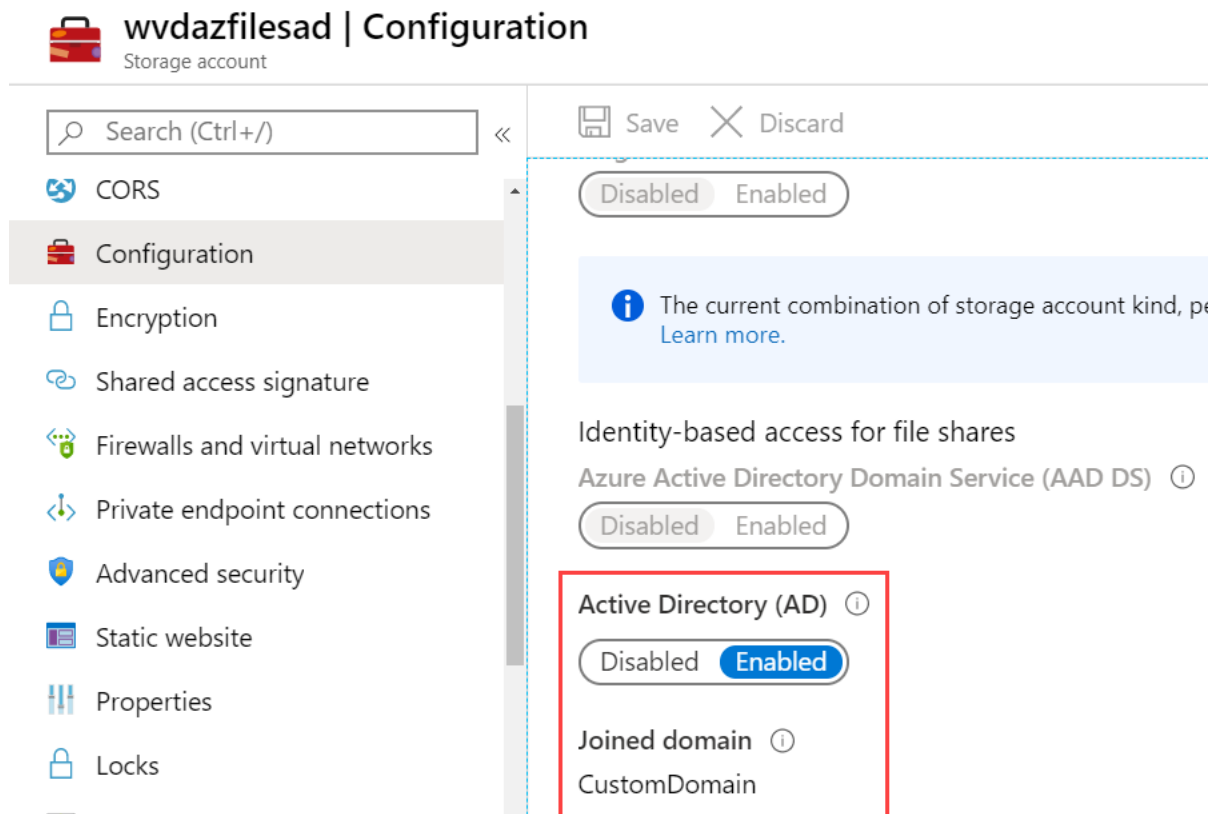
To create a file share:

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 Azure 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 Windows 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 Windows 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 role-based access control (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 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-pat> <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 Windows 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 with the letter of the drive you used to map the drive.
- Replace 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 Windows 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 Windows 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](#).

Customize Remote Desktop Protocol (RDP) properties for a host pool

8/25/2020 • 3 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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. You can customize RDP properties in Windows 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.

Prerequisites

Before you begin, follow the instructions in [Set up the Windows 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 **windows virtual desktop** into the search bar.
3. Under Services, select **Windows Virtual Desktop**.
4. At the Windows 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 **Properties** in the menu on the left side of the screen.
7. On the **Properties** tab, go to **RDP settings** to start editing the RDP properties. Properties should be in a semicolon-separated format like the PowerShell examples.
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 Windows 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 Windows Virtual Desktop load-balancing method

8/25/2020 • 2 minutes to read • [Edit Online](#)

Configuring the load-balancing method for a host pool allows you to adjust the Windows 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 Windows 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. 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 ###
```

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 **Windows Virtual Desktop** under Services.
3. In the Windows 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

8/25/2020 • 3 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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 Windows 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 Windows 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 Windows 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 **Windows Virtual Desktop** into the search bar.
3. Under **Services**, select **Windows Virtual Desktop**.
4. At the Windows 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 Windows 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

8/25/2020 • 2 minutes to read • [Edit Online](#)

Customers who are properly licensed to run Windows 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 [Windows Virtual Desktop pricing](#).

Ways to use your Windows Virtual Desktop license

Windows 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 Windows 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
```

Prepare and customize a master VHD image

8/25/2020 • 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 Windows Virtual Desktop-specific configuration that can be used with your organization's existing processes.

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](#).

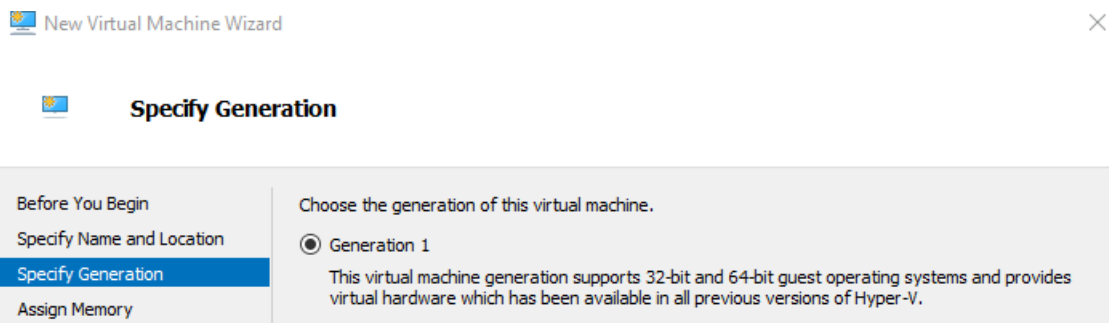
The second option is to create the image locally by downloading the image, provisioning a Hyper-V VM, and customizing it to suit your needs, which we cover in the following section.

Local image creation

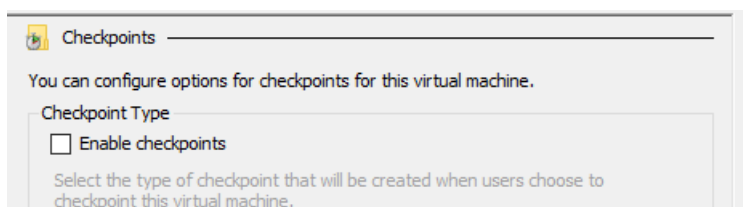
Once you've downloaded the image to a local location, open **Hyper-V Manager** to create a VM with the VHD you copied. The following instructions are a simple version, but you can find more detailed instructions in [Create a virtual machine in Hyper-V](#).

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.



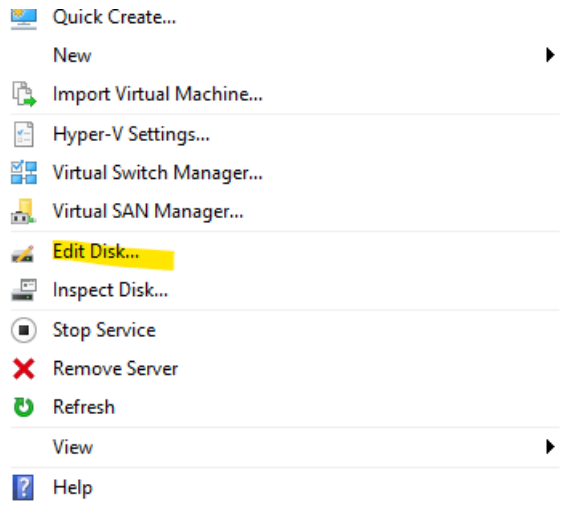
You can also run the following cmdlet in PowerShell to disable checkpoints.

```
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:

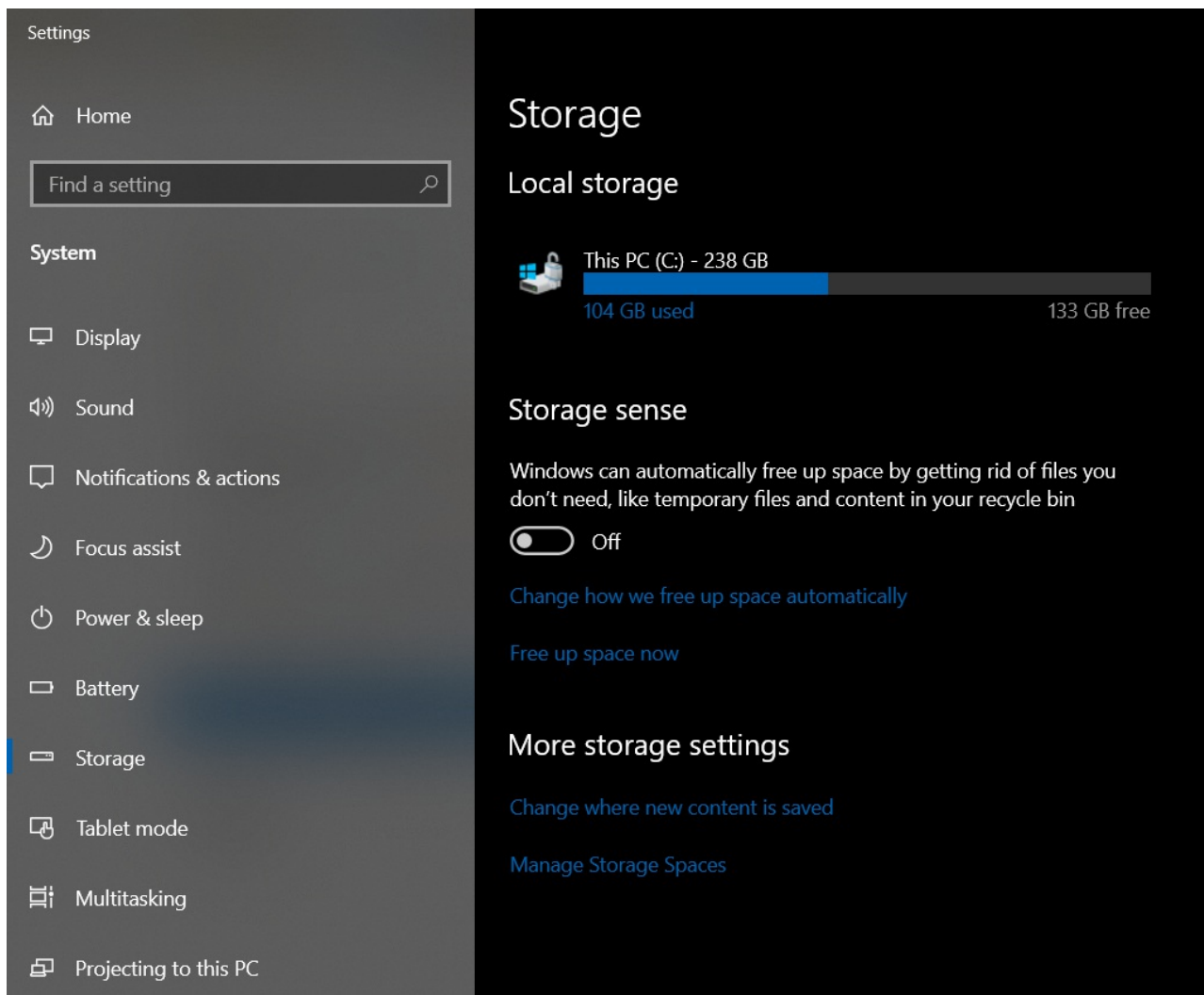
1. On the Active Directory server, open the **Group Policy Management Console**.
2. Expand your domain and Group Policy Objects.
3. Right-click the **Group Policy Object** that you created for the group policy settings and select **Edit**.
4. In the **Group Policy Management Editor**, navigate to **Computer 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 Windows 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.

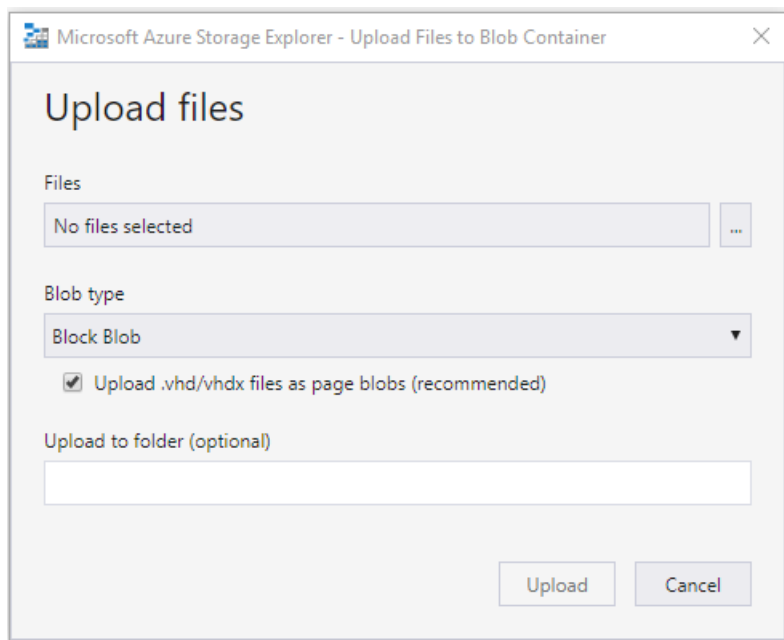
After preparing the image for upload, make sure the VM remains in the off or deallocated state.

Upload master image to a storage account in Azure

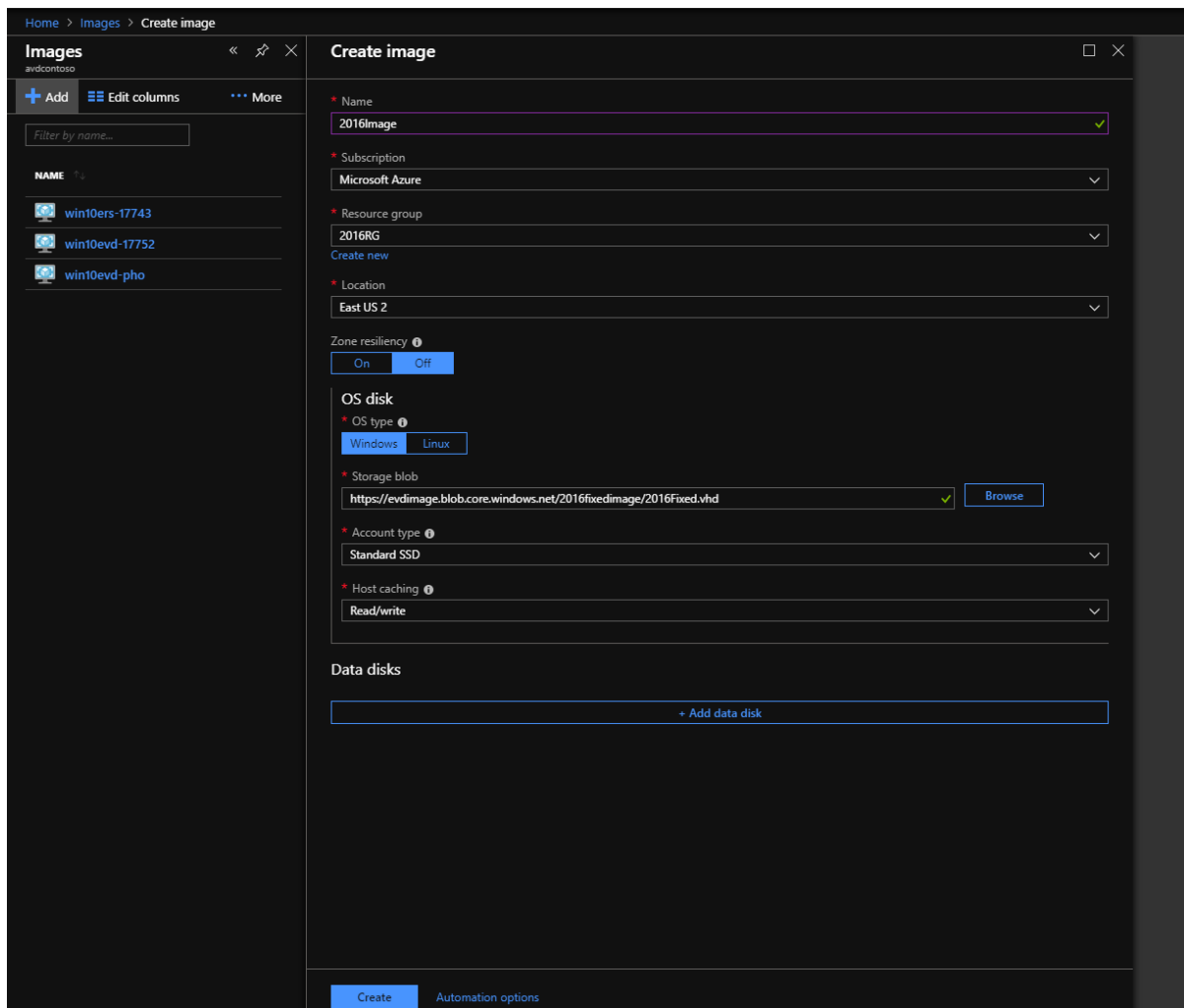
This section only applies when the master image was created locally.

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.

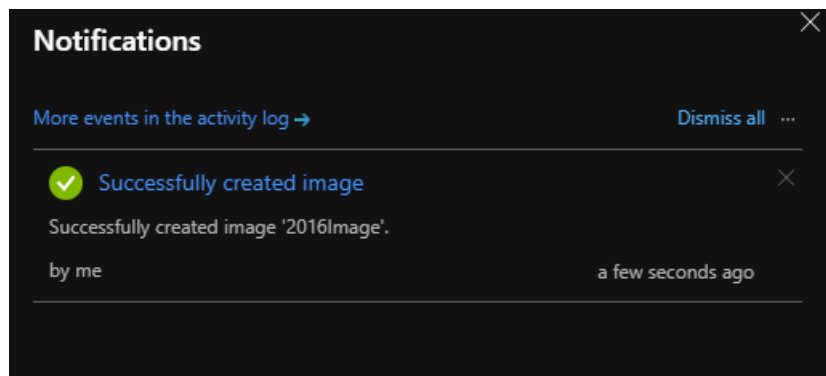
1. Convert the VM image (VHD) to Fixed if you haven't already. If you don't convert the image to Fixed, you can't successfully create the image.
2. Upload the VHD to a blob container in your storage account. You can upload quickly with the [Storage Explorer tool](#). To learn more about the Storage Explorer 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 Windows Virtual Desktop load-balancing method](#)

Install Office on a master VHD image

8/25/2020 • 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 Windows 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 Windows 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="0365ProPlusRetail">
      <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

Windows Virtual Desktop doesn't support Skype for Business.

For help with installing Microsoft Teams, see [Use Microsoft Teams on Windows Virtual desktop](#). Media optimization for Microsoft Teams on Windows Virtual Desktop is available in preview.

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

8/25/2020 • 15 minutes to read • [Edit Online](#)

You can reduce your total Windows 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 Windows Virtual Desktop environment. To learn how to use the scaling tool, skip ahead to [Prerequisites](#).

Report issues

Issue reports for the scaling tool are currently being handled on GitHub instead of Microsoft Support. If you encounter any issue with the scaling tool, get the necessary information as described in the [Reporting issues](#) section and open a GitHub issue labeled with "4a-WVD-scaling-logicapps" on the [RDS GitHub page](#).

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 Windows 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](#).

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:

- A [Windows Virtual Desktop host pool](#)
- Session host pool VMs configured and registered with the Windows 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"
```

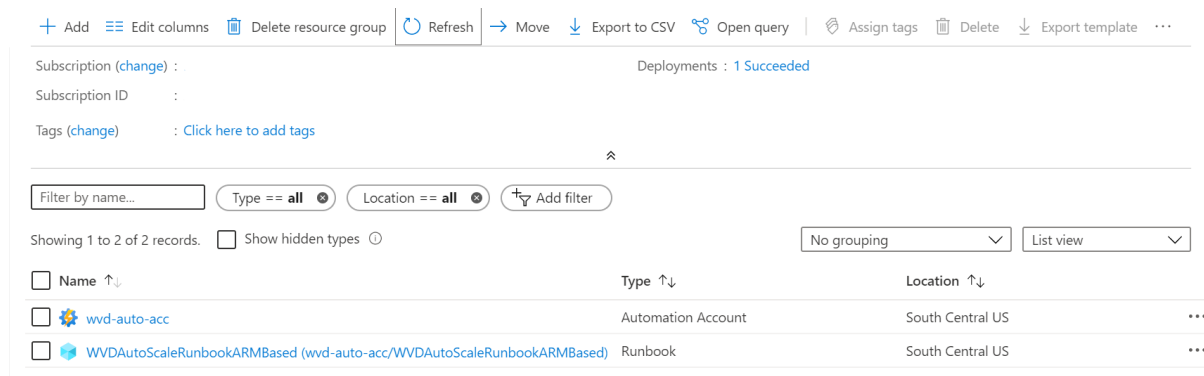
4. Run the following cmdlet to execute the script and create the Azure Automation account. You can either fill in values for the parameters or comment them to use their defaults.

```
$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:



+ Add Edit columns Delete resource group Refresh Move Export to CSV Open query Assign tags Delete Export template ...		
Subscription (change) : Deployments : 1 Succeeded		
Subscription ID :		
Tags (change) : Click here to add tags		
⌵		
Filter by name...	Type == all	Location == all Add filter
Showing 1 to 2 of 2 records. <input type="checkbox"/> Show hidden types No grouping List view		
<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/> wvd-auto-acc	Automation Account	South Central US ...
<input type="checkbox"/> WVDAutoScaleRunbookARMBased (wvd-auto-acc/WVDAutoScaleRunbookARMBased)	Runbook	South Central US ...

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
```

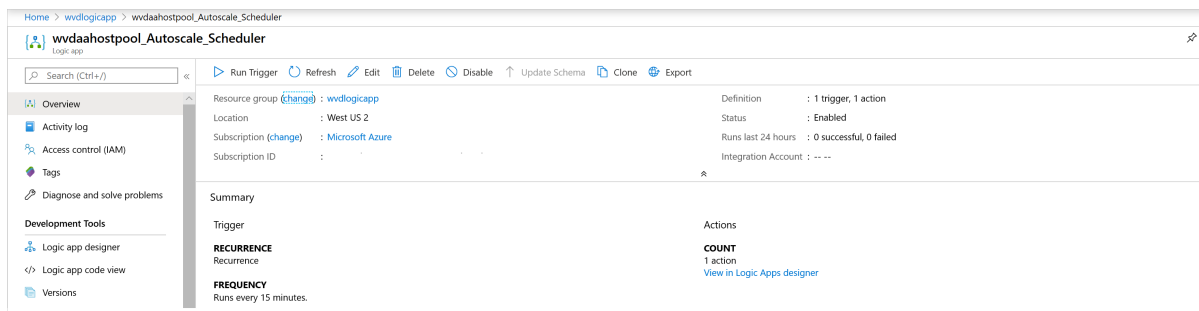
"Select the Azure RunAs connection asset"

```
$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.

Logic Apps Designer

Save Discard Run Designer Code view Parameters Templates Connectors Help

Recurrence

Interval

15

Frequency

Minute

Add new parameter

HTTP

Method

POST

URI

https://8a97ecd2-fe92-4222-ae5e-540e311bcd99.webhook.scus.azure-automation.net/webhooks?token=

Headers

Enter key

Enter value

Queries

Enter key

Enter value

Body

{
 "BeginPeakTime": "09:00",
 "EndPeakTime": "17:00",
 "TimeDifference": "-7:00",
 "SessionThresholdPerCPU": 1,
 "MinimumNumberOfRDSH": 1,
 "LimitSecondsToForceLogOffUser": 900,
 "LogOffMessageTitle": "Machine is about to shutdown.",
 "LogOffMessageBody": "Your session will be logged off. Please save and close everything.",
 "MaintenanceTagName": ""
}

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.

Search runbooks...

Status : All

Time span : All

Runbook	Job created	Status	Ran on
WVDAutoScaleRunbookARMBased	5/29/2020, 4:54:50 PM	✓ Completed	Azure
WVDAutoScaleRunbookARMBased	5/29/2020, 4:49:19 PM	✓ Completed	Azure
WVDAutoScaleRunbookARMBased	5/29/2020, 4:48:08 PM	✓ Completed	Azure

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.



WVDAutoScaleRunbookARMBased 7/9/2020, 3:49 PM

Job

▶ Resume ☐ Stop || Suspend Refresh

Id : 4bb31f10-5f4a-4951-99c2-94321ee2b0ba

Created : 7/9/2020, 3:49:45 PM

Status : Completed

Last Update : 7/9/2020, 3:52:25 PM

Ran ... : Azure

Runbook : [WVDAutoScaleRunbookARMBased](#)

Ran ... : User

Source snaps... : [View source snapshot](#)

Input Output Errors Warnings All Logs Exception

Errors Warnings

0 3

Type : **Any**

Search logs...

Time	Type	Details
7/9/2020, 3:49:57 PM	Output	2020-07-09 15:49:57 [340] Request params: AADTenantId : BeginPeakTime :
7/9/2020, 3:49:58 PM	Output	2020-07-09 15:49:58 [343] Log analytics is enabled
7/9/2020, 3:49:58 PM	Output	2020-07-09 15:49:58 [355] Get auto connection from asset: 'AzureRunAsConnection'
7/9/2020, 3:50:05 PM	Output	2020-07-09 15:50:05 [368] Successfully authenticated with Azure using service principal: Name : Account : 16dbd281-
7/9/2020, 3:50:05 PM	Output	2020-07-09 15:50:05 [398] Get Hostpool info of ' -hp-0' in resource group ' -wvd-rg-0'
7/9/2020, 3:50:08 PM	Output	2020-07-09 15:50:08 [413] Get all session hosts
7/9/2020, 3:50:09 PM	Output	2020-07-09 15:50:09 [421] Get number of user sessions in Hostpool
7/9/2020, 3:50:10 PM	Output	2020-07-09 15:50:10 [433] HostPool info: ApplicationGroupReference : /subscriptions/
7/9/2020, 3:50:10 PM	Output	2020-07-09 15:50:10 [434] Number of session hosts in the HostPool: 5
7/9/2020, 3:50:10 PM	Output	2020-07-09 15:50:10 [456] Using current time: 2020-07-09 15:50:10, begin peak time: 2020-07-09 03:00:00, end peak
7/9/2020, 3:50:10 PM	Output	2020-07-09 15:50:10 [463] Off peak hours

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
```

Customize the feed for Windows Virtual Desktop users

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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 Windows 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-EMM1HWVW3F7LLsNt63Zz2K10RE0_64
IconIndex           : 0
IconPath             : C:\Program Files\Windows NT\Accessories\wordpad.exe
Id                  :
/subscriptions/<subid>/resourcegroups/0301RG/providers/Microsoft.DesktopVirtualization/applicationgroups/0301RAG/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 **Windows Virtual Desktop**.

3. Under Services, select **Windows Virtual Desktop**.
4. On the Windows 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.
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 Windows Virtual Desktop client to test it out. To do so, continue to the [Connect to Windows 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

8/25/2020 • 6 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows 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 Windows 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 Windows Virtual Desktop won't show up in diagnostics results because the diagnostics role service itself is part of Windows Virtual Desktop. Windows Virtual Desktop connection issues can happen when the user is experiencing network connectivity issues.

Azure Monitor lets you analyze Windows 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 Windows 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 Windows 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 Windows 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 **Windows 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 Windows 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 reference](#). All tables related to Windows 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 **Windows 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 ResourceAlias 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](#)

Publish built-in apps in Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

This article will tell you how to publish apps in your Windows 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

Windows 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 Windows Virtual Desktop users](#).
- Learn about the MSIX app attach feature at [Set up MSIX app attach](#).

Set up MSIX app attach

8/25/2020 • 11 minutes to read • [Edit Online](#)

IMPORTANT

MSIX app attach is currently in public preview. This preview version is provided without a service level agreement, and we don't recommend using it 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](#).

This topic will walk you through how to set up MSIX app attach in a Windows Virtual Desktop environment.

Requirements

Before you get started, here's what you need to configure MSIX app attach:

- Access to the Windows Insider portal to obtain the version of Windows 10 with support for the MSIX app attach APIs.
- A functioning Windows Virtual Desktop deployment. To learn how to deploy Windows Virtual Desktop (classic), see [Create a tenant in Windows Virtual Desktop](#). To learn how to deploy Windows Virtual Desktop with Azure Resource Manager integration, see [Create a host pool with the Azure portal](#).
- The MSIX packaging tool.
- A network share in your Windows Virtual Desktop deployment where the MSIX package will be stored.

Get the OS image

First, you need to get the OS image. You can get the OS image through the Azure portal. However, if you're a member of the Windows Insider program, you have the option to use the Windows Insider portal instead.

Get the OS image from the Azure portal

To get the OS image from the Azure portal:

1. Open the [Azure portal](#) and sign in.
2. Go to **Create a virtual machine**.
3. In the **Basic** tab, select **Windows 10 enterprise multi-session, version 2004**.
4. Follow the rest of the instructions to finish creating the virtual machine.

NOTE

You can use this VM to directly test MSIX app attach. To learn more, skip ahead to [Generate a VHD or VHDX package for MSIX](#). Otherwise, keep reading this section.

Get the OS image from the Windows Insider portal

To get the OS image from the Windows Insider Portal:

1. Open the [Windows Insider portal](#) and sign in.

NOTE

You must be member of the Windows Insider program to access the Windows Insider portal. To learn more about the Windows Insider program, check out our [Windows Insider documentation](#).

2. Scroll down to the **Select edition** section and select **Windows 10 Insider Preview Enterprise (FAST) – Build 19041** or later.
3. Select **Confirm**, then select the language you wish to use, and then select **Confirm** again.

NOTE

At the moment, English is the only language that has been tested with the feature. You can select other languages, but they may not display as intended.

4. When the download link is generated, select the **64-bit Download** and save it to your local hard disk.

Prepare the VHD image for Azure

Next, you'll need to create a master VHD image. If you haven't created your master VHD image yet, go to [Prepare and customize a master VHD image](#) and follow the instructions there.

After you've created your master VHD image, 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

rem Disable Windows Update:

sc config wuauserv start=disabled
```

After you've disabled automatic updates, you must enable Hyper-V because you'll be using the Mount-VHD command to stage and and Dismount-VHD to destage.

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

NOTE

This change will require that you restart the virtual machine.

Next, prepare the VM VHD for Azure and upload the resulting VHD disk to Azure. To learn more, see [Prepare and customize a master VHD image](#).

Once you've uploaded the VHD to Azure, create a host pool that's based on this new image by following the instructions in the [Create a host pool by using the Azure Marketplace](#) tutorial.

Prepare the application for MSIX app attach

If you already have an MSIX package, skip ahead to [Configure Windows Virtual Desktop infrastructure](#). 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.

Generate a VHD or VHDX package for MSIX

Packages are in VHD or VHDX format to optimize performance. MSIX requires VHD or VHDX packages to work properly.

To generate a VHD or VHDX package for MSIX:

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. Run the following cmdlet in PowerShell to create a VHD:

```
New-VHD -SizeBytes <size>MB -Path c:\temp\<name>.vhd -Dynamic -Confirm:$false
```

NOTE

Make sure the size of VHD is large enough to hold the expanded MSIX.*

5. Run the following cmdlet to mount the newly created VHD:

```
$vhdObject = Mount-VHD c:\temp\<name>.vhd -Passthru
```

6. Run this cmdlet to initialize the VHD:

```
$disk = Initialize-Disk -Passthru -Number $vhdObject.Number
```

7. Run this cmdlet to create a new partition:

```
$partition = New-Partition -AssignDriveLetter -UseMaximumSize -DiskNumber $disk.Number
```

8. Run this cmdlet to format the partition:

```
Format-Volume -FileSystem NTFS -Confirm:$false -DriveLetter $partition.DriveLetter -Force
```

9. Create a parent folder on the mounted VHD. This step is mandatory as the MSIX app attach requires a parent folder. You can name the parent folder whatever you like.

Expand MSIX

After that, you'll need to "expand" the MSIX image by unpacking it. To unpack the MSIX image:

1. Open a command prompt as Administrator and navigate to the folder where you downloaded and

unzipped the msixmgr tool.

2. Run the following cmdlet to unpack the MSIX into the VHD you created and mounted in the previous section.

```
msixmgr.exe -Unpack -packagePath <package>.msix -destination "f:\<name of folder you created earlier>"  
-applyacls
```

The following message should appear once unpacking is done:

```
Successfully unpacked and applied ACLs for package: <package name>.msix
```

NOTE

If using packages from the Microsoft Store for Business (or Education) within your network, or on devices that are not connected to the internet, you will need to obtain the package licenses from the Store and install them to run the app successfully. See [Use packages offline](#).

3. Navigate to the mounted VHD and open the app folder and confirm package content is present.
4. Unmount the VHD.

Configure Windows Virtual Desktop infrastructure

By design, a single MSIX expanded package (the VHD you created in the previous section) can be shared between multiple session host VMs as the VHDs are attached in read-only mode.

Before you start, make sure your network share meets these requirements:

- The share is SMB compatible.
- The VMs that are part of the session host pool have NTFS permissions to the share.

Set up an MSIX app attach share

In your Windows Virtual Desktop environment, create a network share and move the package there.

NOTE

The best practice for creating MSIX network shares is to set up the network share with NTFS read-only permissions.

Install certificates

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, as shown in the following image. 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**.

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{a12b3456-0000-0000-0000-100000000000}\  
*** NO MOUNT POINTS ***
```

```
\\?\Volume{c78d9012-0000-0000-0000-200000000000}\  
E:\
```

```
\\?\Volume{d34e5678-0000-0000-0000-300000000000}\  
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=WindowsRuntime]
| 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 # needed if we do the pbisigned.vhd
$path = ([System.Uri]$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.

```
#MSIX app attach de staging sample

#region variables
$packageName = "<package name>"
$msixJunction = "C:\temp\AppAttach\"
#endregion

#region deregister
Remove-AppxPackage -AllUsers -Package $packageName
cd $msixJunction
rmdir $packageName -Force -Verbose
#endregion
```

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.

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 [Windows Virtual Desktop TechCommunity](#).

You can also leave feedback for Windows Virtual Desktop at the [Windows Virtual Desktop feedback hub](#).

Use Microsoft Teams on Windows Virtual desktop

8/25/2020 • 6 minutes to read • [Edit Online](#)

IMPORTANT

Media optimization for Teams is not supported for Microsoft 365 Government environments.

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 Windows 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 Windows 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 Windows 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 on the host using per-machine installation. Media optimization for Microsoft Teams requires Teams desktop app version 1.3.00.4461 or later.

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**.
2. Create the following value for the Teams key:

NAME	TYPE	DATA/VALUE
IsWVDEnvironment	DWORD	1

Install the Teams WebSocket Service

Install the latest [WebSocket 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
1.0.2006.11001	07/28/2020
0.11.0	05/29/2020

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 Windows 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 64-bit operating system and to the Program Files folder on a 32-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> /! *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. Select your user profile image, then select **About**.
2. Select **Version**.

If media optimizations loaded, the banner will show you **WVD Media optimized**. If the banner shows you **WVD Media not connected**, quit the Teams app and try again.

3. 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** 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.

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 an endpoint is not supported.

Calls and meetings

- The Teams desktop client in Windows Virtual Desktop environments doesn't support live events. For now, we recommend you join 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 Windows 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.

For Teams known issues that aren't related to virtualized environments, see [Support Teams in your organization](#)

UserVoice site

Provide feedback for Microsoft Teams on Windows Virtual Desktop on the Teams [UserVoice site](#).

Collect Teams logs

If you encounter issues with the Teams desktop app in your Windows 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 Multi-Factor Authentication for Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

If you're visiting this page from the Windows Virtual Desktop (classic) documentation, make sure to [return to the Windows Virtual Desktop \(classic\) documentation](#) once you're finished.

The Windows client for Windows Virtual Desktop is an excellent option for integrating Windows Virtual Desktop with your local machine. However, when you configure your Windows 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 MFA. 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**, 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'll need to make sure the client keeps asking for Azure Multi-Factor Authentication (MFA) credentials. This article will show you how to configure the Conditional Access policy for Windows 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 MFA 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 [Windows Virtual Desktop web client](#).

Create a Conditional Access policy

Here's how to create a Conditional Access policy that requires multi-factor authentication when connecting to Windows 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 groups of apps based on which version of Windows Virtual Desktop you're using.
 - If you're using Windows Virtual Desktop (classic), choose these two apps:
 - **Windows Virtual Desktop** (App ID 5a0aa725-4958-4b0c-80a9-34562e23f3b7)
 - **Windows Virtual Desktop Client** (App ID fa4345a4-a730-4230-84a8-7d9651b86739)
 - If you're using Windows Virtual Desktop, choose these two apps instead:
 - **Windows Virtual Desktop** (App ID 9cdead84-a844-4324-93f2-b2e6bb768d07)
 - **Windows Virtual Desktop Client** (App ID a85cf173-4192-42f8-81fa-777a763e6e2c)

IMPORTANT

The Windows Virtual Desktop Client apps are used for the web client. However, don't select the app called Windows 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 MFA.

10. Once you've selected your app, choose **Select**, and then select **Done**.

New

×

Info

Try out the new configuration experience. Click to enable the preview.

→

Name *

wvdrdsdemopolicy ✓

Assignments

Users and groups ⓘ >

Specific users included

Cloud apps or actions ⓘ >

No cloud apps or actions sele...

Conditions ⓘ >

0 conditions selected

Access controls

Grant ⓘ >

0 controls selected

Session ⓘ >

0 controls selected

Enable policy

Report-only On Off

Create

Cloud apps or actions

□ ×

Select what this policy applies to

Cloud apps User actions

Include Exclude

None

All cloud apps

Select apps

Select

Windows Virtual Desktop >

WV Windows Virtual Desktop ...

9cdead84-a844-4324-93f2-b2e6bb768...

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.

- Under **Access controls** > **Grant**, select **Grant access**, **Require multi-factor authentication**, and then **Select**.
- Under **Access controls** > **Session**, select **Sign-in frequency**, set the value to **1** and the unit to **Hours**, and then select **Select**.
- Confirm your settings and set **Enable policy** to **On**.
- Select **Create** to enable your policy.

Next steps

- [Learn more about Conditional Access policies](#)
- [Learn more about user sign in frequency](#)

Configure Microsoft Endpoint Configuration Manager

8/25/2020 • 2 minutes to read • [Edit Online](#)

This article explains how to configure Microsoft Endpoint Configuration Manager to automatically apply updates to a Windows 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.

Configure the software update point

To receive updates for Windows 10 Enterprise multi-session, you need to enable Windows Server, version 1903 and later as a product within Microsoft Endpoint Configuration Manager. This product setting also applies if you use the Windows Server Update Service to deploy updates to your systems.

To receive updates:

1. Open Microsoft Endpoint Configuration Manager and select **Sites**.
2. Select **Configure Site Components**.
3. Select **Software Update Point** in the drop-down menu.
4. Select the **Products** tab.
5. Select the check box that says **Windows Server, version 1903 and later**.
6. Go to **Software Library > Overview > Software Updates > All Software Updates** and select **Synchronize Software Updates**.
7. Check the wsyncmgr.log file in **Program Files > Microsoft Configuration Manager > Logs** to make sure your changes were saved. It may take a few minutes to synchronize the 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**.

Add language packs to a Windows 10 multi-session image

8/25/2020 • 6 minutes to read • [Edit Online](#)

Windows 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 and Feature on Demand (FOD) Disk 1 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 Language Pack ISO](#)
 - FOD Disk 1 ISO:
 - [Windows 10, version 1903 or 1909 FOD Disk 1 ISO](#)
 - [Windows 10, version 2004 FOD Disk 1 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:

1. On an Azure VM, download the Windows 10 Multi-Language ISO and FODs 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.

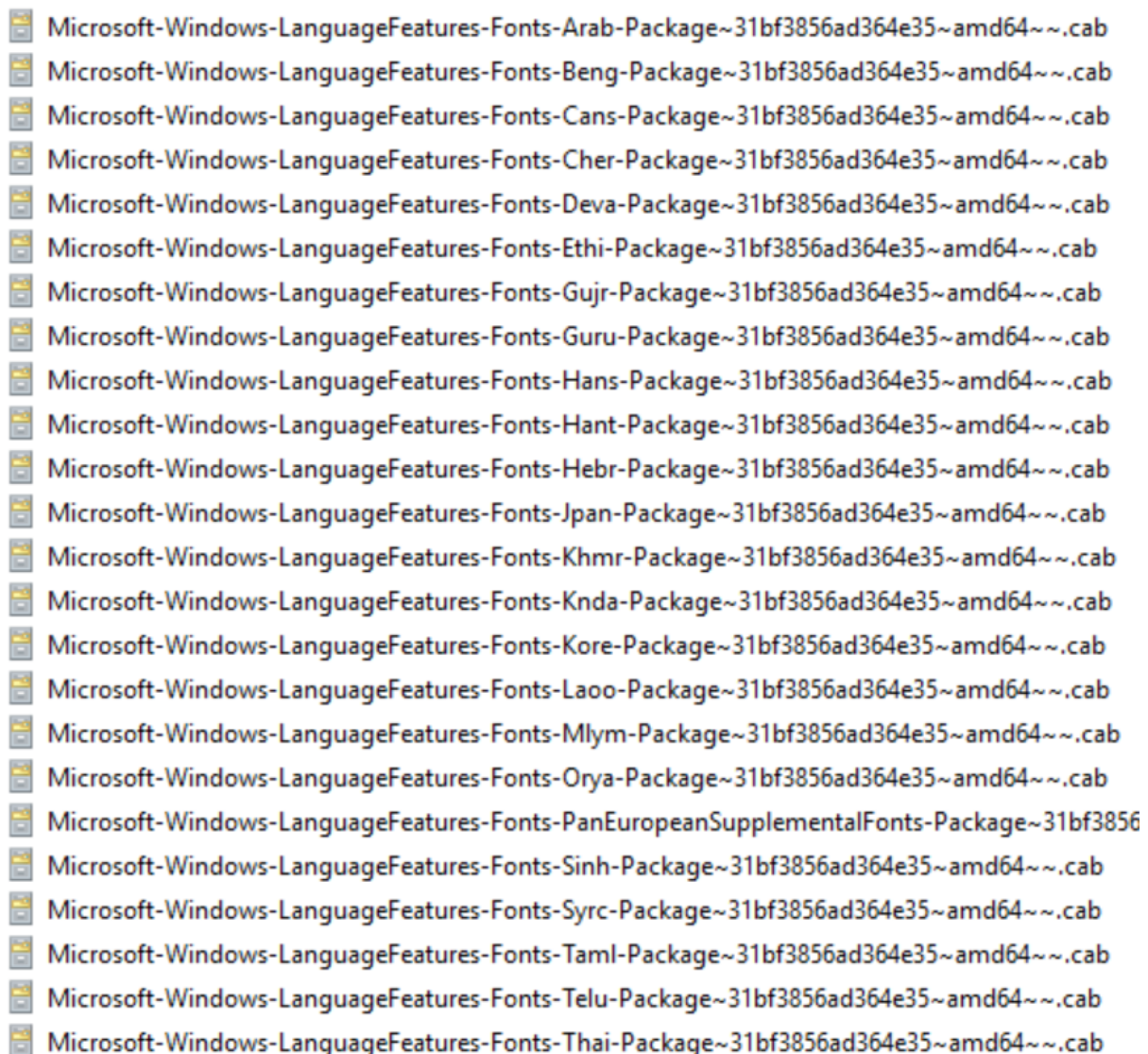
4. Go to the **FOD ISO file**, copy all of its content, then paste it into the file share.

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."



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 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 and FOD 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
```



```

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

```


IMPORTANT

Windows 10 Enterprise versions 1903 and 1909 don't require the `Microsoft-Windows-Client-Language-Pack_x64_<language-code>.cab` package file.

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.

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. Shut down 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 Windows 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, 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 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 Windows 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](#).

Safe URL list

8/25/2020 • 2 minutes to read • [Edit Online](#)

You'll need to unblock certain URLs so your Windows Virtual Desktop deployment works properly. This article lists these URLs so you know which ones are safe.

Virtual machines

The Azure virtual machines you create for Windows Virtual Desktop must have access to the following URLs:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*.wvd.microsoft.com	443	Service traffic	WindowsVirtualDesktop
mrsglobalsteus2prod.blob.core.windows.net	443	Agent and SXS stack updates	AzureCloud
*.core.windows.net	443	Agent traffic	AzureCloud
*.servicebus.windows.net	443	Agent traffic	AzureCloud
gcs.prod.monitoring.core.windows.net	443	Agent traffic	AzureCloud
catalogartifact.azureedge.net	443	Azure Marketplace	AzureCloud
kms.core.windows.net	1688	Windows activation	Internet
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

Windows Virtual Desktop now supports the FQDN tag. For more information, see [Use Azure Firewall to protect Windows 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 Windows Virtual Desktop sites and resources. They don't include URLs for other services like Azure Active Directory.

The following table lists optional URLs that your Azure virtual machines can have access to:

ADDRESS	OUTBOUND TCP PORT	PURPOSE	SERVICE TAG
*.microsoftonline.com	443	Authentication to Microsoft Online Services	None
*.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	None
*.sfx.ms	443	Updates for OneDrive client software	None
*.digicert.com	443	Certificate revocation check	None

NOTE

Windows 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.

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 Windows Virtual Desktop host pool.
2. Open **Event viewer**, then go to **Windows logs > Application > WVD-Agent** and look for Event ID 3701.
3. Whitelist 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)
*.wvd.microsoft.com	443	Service traffic	All
*.servicebus.windows.net	443	Troubleshooting data	All
go.microsoft.com	443	Microsoft FWLinks	All
aka.ms	443	Microsoft URL shortener	All
docs.microsoft.com	443	Documentation	All
privacy.microsoft.com	443	Privacy statement	All

ADDRESS	OUTBOUND TCP PORT	PURPOSE	CLIENT(S)
query.prod.cms.rt.microsoft.com	443	Client updates	Windows Desktop

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 on the [Office 365 URLs and IP address ranges](#).

Windows Virtual Desktop environment

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows 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 Windows Virtual Desktop environment.

Host pools

A host pool is a collection of Azure virtual machines that register to Windows Virtual Desktop as session hosts when you run the Windows 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 Windows 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 Windows Virtual Desktop. Each Windows 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 Windows Virtual Desktop deployment with any of the Windows Virtual Desktop clients.

Next steps

Learn more about delegated access and how to assign roles to users at [Delegated Access in Windows Virtual Desktop](#).

To learn how to set up your Windows Virtual Desktop host pool, see [Create a host pool with the Azure portal](#).

To learn how to connect to Windows 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](#)

Determine user connection latency in Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

Windows 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. Windows Virtual Desktop services will continuously roll out to new geographies to improve latency.

The [Windows 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 Windows Virtual Desktop rolls out to new areas.

Azure Traffic Manager

Windows Virtual Desktop uses the Azure Traffic Manager, which checks the location of the user's DNS server to find the nearest Windows Virtual Desktop service instance. We recommend admins review the location of the user's DNS server before choosing the location for the VMs.

Next steps

- To check the best location for optimal latency, see the [Windows Virtual Desktop Experience Estimator tool](#).
- For pricing plans, see [Windows Virtual Desktop pricing](#).
- To get started with your Windows Virtual Desktop deployment, check out [our tutorial](#).

Delegated access in Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows 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 Windows 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](#).

Windows 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 Windows Virtual Desktop PowerShell module if you haven't already.

Windows 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 Windows Virtual Desktop environment, see [Windows Virtual Desktop environment](#).

Host pool load-balancing methods

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows Virtual Desktop supports two load-balancing methods. Each method determines which session host will host a user's session when they connect to a resource in a host pool.

The following load-balancing methods are available in Windows 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 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 methods share the following behaviors no matter which host pool they're in:

- If a user already has a session in the host pool and is reconnecting to that session, 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`.
- 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 method

The breadth-first load-balancing method allows you to distribute user connections to optimize for this scenario. This method is ideal for organizations that want to provide the best experience for users connecting to their pooled virtual desktop environment.

The breadth-first method first queries session hosts that allow new connections. The method then selects the session host with the least number of sessions. If there is a tie, the method selects the first session host in the query.

Depth-first load-balancing method

The depth-first load-balancing method allows you to saturate one session host at a time to optimize for this scenario. This method 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 method first queries session hosts that allow new connections and haven't gone over their maximum session limit. The method then selects the session host with highest number of sessions. If there's a tie, the method selects the first session host in the query.

FSLogix profile containers and Azure files

8/25/2020 • 5 minutes to read • [Edit Online](#)

The Windows Virtual Desktop service recommends FSLogix profile containers as a user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Windows 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 Windows 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. UPD with S2D is the solution we recommend for Windows Virtual Desktop.

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. [OneDrive for Business and FSLogix best practices](#) describes 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 \(AD DS\)](#). By addressing both cost and administrative overhead, Azure Files with Azure AD DS Authentication is a premium solution for user profiles in the Windows Virtual Desktop service.

Best practices for Windows Virtual Desktop

Windows Virtual Desktop offers full control over size, type, and count of VMs that are being used by customers. For more information, see [What is Windows Virtual Desktop?](#).

To ensure your Windows 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 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 Windows Virtual Desktop environment.

- To start building out your desktop virtualization solution, see [Create a tenant in Windows Virtual Desktop](#).
- To create a host pool within your Windows 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 Windows Virtual Desktop](#).
- To access your Windows Virtual Desktop resources from a web browser, see [Connect to Windows Virtual Desktop](#).

Storage options for FSLogix profile containers in Windows Virtual Desktop

8/25/2020 • 2 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 Windows Virtual Desktop FSLogix user profile containers. We recommend storing FSLogix profile containers on Azure Files for most of our customers.

Windows Virtual Desktop offers FSLogix profile containers as the recommended user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Windows 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 Windows 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	Locally redundant	Locally redundant/zone-redundant/geo-redundant
Tiers and performance	Standard Premium Up to max 100k IOPS per share with 5 GBps per share at about 3 ms latency	Standard Premium Ultra Up to 320k (16K) IOPS with 4.5 GBps per volume at about 1 ms latency	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	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 2.1/3. and 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 [Windows Virtual Desktop pricing](#) for information about our pricing plans.

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 Windows 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 Windows Virtual Desktop solution at [Create a tenant in Windows Virtual Desktop](#).

Windows Virtual Desktop partner integrations

8/25/2020 • 22 minutes to read • [Edit Online](#)

This article lists approved partner providers and independent software vendors for Windows Virtual Desktop.

Citrix



Citrix is an approved provider that offers enterprises centralized hybrid management of virtual apps and desktops workloads in Azure, side by side with on-premises deployments. Citrix Workspace with the Virtual Apps and Desktops service allows users to access apps and desktops from any device, leveraging the advanced Citrix HDX protocol to deliver a high definition experience from anywhere.

Citrix extends the value of Windows Virtual Desktop with robust enterprise tools to improve user density and performance, provision workloads on demand, and simplify image and application management. IT can optimize costs with intelligent scaling tools, while delivering an incredible user experience that's field-tested against the toughest applications across industries. Additionally, Citrix Managed Desktops is a Windows Virtual Desktop-enabled desktops-as-a-service program that provides a simple, cloud-based management solution for delivering virtual apps and desktops to any device.

- [Go to the partner website.](#)

VMware



VMware Horizon Cloud on Microsoft Azure is a native cloud service that lets organizations quickly deploy remote desktops and applications from their existing Microsoft Azure subscriptions while leveraging all the features of VMware Horizon. Horizon Cloud uses the Horizon Control Plane to provide a single management interface for all Horizon environments, on-premises or in the cloud. This enables hybrid desktop virtualization and lets customers move their workloads to Azure at their own pace.

As a Windows Virtual Desktop approved provider, VMware can help customers that want to use Windows Virtual Desktop while still enjoying the additional functionality that comes with VMware Horizon, such as integrated and easy-to-use power management, cloud-based monitoring, and the Blast Extreme protocol. These features adapt to changing network conditions on the fly to provide a consistently excellent user experience. VMware Horizon Cloud also comes with VMware App Volumes and Dynamic Environment Manager, which add advanced application and user environment management capabilities that work with MSIX app attach and FSLogix.

- [Go to the partner website.](#)
- [Read VMware Horizon Cloud technical documentation.](#)

10ZiG



10ZiG Technology, with cutting-edge Thin and Zero Client hardware and software, is a longstanding partner with

Microsoft and a dedicated Microsoft Azure and Windows Virtual Desktop partner. 10ZiG Windows 10 IoT-based Thin Clients are powerful, reliable, and affordable endpoints for all Windows Virtual Desktop multi-users. 10ZiG Manager Software provides exceptional management and deployment without license limitations at no additional cost. The 10ZiG Tech Team, Advance Warranty Program, and no-hassle demos are a one-stop Windows Virtual Desktop multi-session support solution in the cloud.

10ZiG's world-market leadership in Thin and Zero Client endpoint devices and management software for virtual desktops is exemplified by how they work for their customers. Its Thin Client hardware comes with thoughtfully constructed benefit features and options designed to ensure customers receive the right Client devices based on their needs. 10ZiG customizes its devices to fit into customer environments with Windows-based and Linux-based Clients that provide the best possible performance in virtual desktops, both inside and outside the cloud.

- [Go to the partner website.](#)

Automai



You can use Automai's robotic automation platform to test key business processes in a Windows Virtual Desktop environment before your deployment goes live.

With Automai's ScenarioBuilder tool and GUI-based workflow engine, IT teams can record real end-user workflows and automatically translate them into scripts. Automai then uses bots running processes from individual desktops to emulate end-user activity in a simulation and report the results. This greatly simplifies testing processes so that IT admins can stress-test even the most complex scenarios.

Once you're ready for launch, you can use all the workflow scripts you created for load testing to continuously monitor performance in production. Automai's bots can do more than just availability monitoring. The bots can also test end-user workflows from key locations, taking screenshots and collecting error reports in real time. This leads to a more proactive than reactive approach to bug fixes for Windows Virtual Desktop applications.

Automai lets you use the same scripts for performance testing, functional testing, performance monitoring, and even robotic process automation, all on one platform.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Cloudhouse



Cloudhouse is a Windows Virtual Desktop value-added services provider that offers customers a turnkey application migration service that can move all applications, including ones that are incompatible with modern Windows operating systems, to the Windows Virtual Desktop environment, allowing customers to truly leverage multi-session Windows 10.

By leveraging proven Cloudhouse containerization technology, the Cloudhouse service takes all applications, including ones designed for Windows XP, Windows 7, or Windows 8, and deploys them to a modern Windows Virtual Desktop without needing to change code or impact user experience. Cloudhouse further adds to the value of Windows Virtual Desktop by isolating applications from the underlying operating system, allowing Windows Servicing updates to be rolled out without affecting the containerized application.

- [Go to the partner website.](#)

CloudJumper



CloudJumper is a Windows Virtual Desktop value-added services provider that equips solution providers and enterprise IT with software to provision and manage Windows Virtual Desktop environments holistically. With CloudJumper software, IT can manage every layer of a Windows Virtual Desktop deployment. Delivery of workloads and applications is automated, ensuring that users can quickly access their desktop anywhere on any device.

CloudJumper's software, Cloud Workspace Management Suite extends the value of Windows Virtual Desktop by simplifying deployment and ongoing administration tasks in Azure. From a single pane of glass, IT can provision, manage, and optimize infrastructure for user workspaces. CloudJumper's Simple Script Triggering Engine integrates with IT service platforms to automate tasks involved in provisioning Windows Virtual Desktop. Additionally, CloudJumper APIs allow further extensibility and integration with other enterprise systems like ServiceNow and BMC Ready.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

ControlUp



ControlUp is a Windows Virtual Desktop value-added services provider that enables IT teams to monitor, troubleshoot, analyze, and directly remediate problems in their on-premises, hybrid cloud, and cloud infrastructure in real time from a single console. ControlUp's analytics and management platform also allows IT to proactively automate fixes for a rapidly growing set of use cases.

When used with Windows Virtual Desktop, ControlUp provides additional capabilities to optimize Windows Virtual Desktop environments and the end-user experience. From the ControlUp console, IT gets end-user environment visibility to effectively monitor and troubleshoot performance issues. An intuitive dashboard provides insights and analytics for virtual desktop deployments, as well as options for automated reporting enriched with community benchmarks. ControlUp can manage multiple data sources and types, organizing them in high-performance data sets aggregated across compute, storage, and Windows Virtual Desktop infrastructure, allowing granular visibility from a single pane of glass.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Dell



Dell Technologies' thin clients are optimized to access Microsoft Azure and Windows Virtual Desktop services. Capable of meeting the needs from collaborative knowledge workers up to graphics-intensive power users, Wyse thin clients deliver a high-quality computing experience to take full advantage of the growing number of apps and content. Ideal for space-constrained environments, Wyse thin clients adapt to the way people work with versatile form factors and a wide array of choices for mounting options.

Wyse thin clients are designed with security in mind with limited attack surface, support for security compliance

standards, and advanced multi-factor authentication solutions. Deploy highly secure thin clients with Dell-exclusive Wyse ThinOS, or experience secure access to cloud applications and virtual workspaces from hardened Wyse ThinLinux with a commercial proven design and Windows 10 IoT Enterprise with Dell-added security features. With secure, HTTPS-based communications and active directory authentication for role-based administration, Wyse Management Suite keeps Wyse endpoints always up to date, and the mobile app for WMS Pro allows IT to view critical alerts and send real-time commands with one tap at any time.

- [Go to the partner website.](#)

deviceTRUST



deviceTRUST is a Windows Virtual Desktop value-added services provider that contextualizes the corporate enterprise. It allows users the freedom to access their Windows Virtual Desktop from any location, on any device, over any network, while giving IT departments the information and control they need to meet their governance requirements.

deviceTRUST extends the value of Windows Virtual Desktop with their contextual security technology. deviceTRUST enables conditional access for a secure Windows Virtual Desktop access, conditional application access within Windows Virtual Desktop and to apply conditional Windows Virtual Desktop policies without any additional infrastructure. Using deviceTRUST enables a mobile, flexible workspace that meets all security, compliance, and regulatory requirements.

- [Go to the partner website.](#)

Ekran System



Ekran System is a Windows Virtual Desktop value-add partner that lets IT teams monitor all remote user activity on Microsoft Azure virtual machines. With Ekran System, you can record on-screen activity for every user session in published applications or virtual desktops while collecting a wide range of context-rich metadata, such as application names, active window titles, visited URLs, and keystrokes. Advanced features offer in-depth visibility and quick incident response times, making Ekran System an efficient insider threat management and compliance solution.

The unique floating endpoint licensing of Ekran System clients is automated to support dynamically changing virtual desktops. Ekran System lets you automatically unassign licenses from deleted non-persistent virtual desktops and remove them from your database. Ekran System seamlessly integrates with Azure Active Directory and Azure Sentinel.

- [See the joint solution brief.](#)
- [Go to the partner page.](#)

FabulaTech



FabulaTech seamlessly integrates with Windows Virtual Desktop clients. Once installed, FabulaTech software automatically starts working when you establish a connection with a remote desktop.

When a user signs in to their virtual desktop, the FabulaTech software creates a virtual device. For example, you can create a virtual webcam, scanner, or fingerprint reader. Any apps running in a remote session can access the virtual device as if it was a physical device. You can configure the virtual device in Windows Virtual Desktop with the System Tray Icon menu, which means you can also use this solution on thin clients. On top of that, all communication happens over the existing remote desktop connection, which means the firewall is set up for you. Everything works right out of the box.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Flexxible IT



Flexxible IT is a Windows Virtual Desktop value-add partner that offers organizations the ability to rapidly scale, monitor, and efficiently manage Windows Virtual Desktop and Citrix Workspace infrastructure. Flexxible|SUITE allows IT admins to intelligently provision and manage Windows Virtual Desktop workloads on-premises and hosted in Azure.

Flexxible IT's technology extends the value of both native Windows Virtual Desktop and Citrix Workspace by automating common processes to simplify infrastructure configuration, desktop provisioning, and day-to-day management. With no need for complex PowerShell scripts or time-consuming manual processes, SUITE provides scalable desktop deployment, extensive monitoring and reporting, and secure delegated management. These features allow you to focus on delivering enhanced levels of service and a quality Windows Virtual Desktop experience for your users.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

HP



HP Thin Client is an approved and verified partner of Microsoft's Azure and Windows Virtual Desktop services. HP Thin Clients with Windows 10 IoT Enterprise offer out-of-box support for Azure-based workloads and Windows Virtual Desktop hosted desktops. The hardware and OS are optimized to provide a best-in-class experience that effectively delivers remote workloads while reducing the OS footprint, hardware, and maintenance costs.

As HP looked at industry trends, customer challenges, and the solutions virtualization offered during the development process, they were inspired to invent the ideal cloud endpoint using a four-pillar value proposition: design, manageability, security, and versatility. Every HP Thin Client is purpose-built with IT decision makers in mind. HP Thin Clients are long-lasting, secure, easy to deploy and manage, and powerful so you can effortlessly transition to VDI or cloud computing. HP's versatile portfolio gives you the freedom to choose the modern endpoint solution that's right for you.

- [Go to the partner website.](#)

IGEL



IGEL is an approved and verified partner of Microsoft Azure and Windows Virtual Desktop services. IGEL offers IGEL OS, the next-gen edge OS for cloud workspaces designed to access virtual apps, desktops, and cloud workspaces from one or more user devices with a lightweight, simple, and secure Linux-based endpoint. A platform-independent software solution, IGEL OS and its server-based management and control software, IGEL Universal Management Suite (UMS), comprise an endpoint management and control solution that frees enterprises to take full advantage of Azure-based cloud instances and Windows Virtual Desktop desktops, including economical multi-session Windows Virtual Desktop, while reducing endpoint hardware and endpoint device management and operations costs.

IGEL OS supports all popular virtual apps, desktops, and cloud workspace client protocols from Citrix, Microsoft, and VMware. It includes integrated technologies from 85 peripheral, interface, and protocol partners to help organizations quickly adopt Windows Virtual Desktop services into their own unique user environments. IGEL OS is a read-only, modular endpoint OS, which helps protect it from tampering. It now also includes a complete "chain of trust" that verifies the integrity of all key major processes running on the endpoint, from the endpoint hardware (some selected models) or UEFI process all the way to the Azure cloud and Windows Virtual Desktop services. With IGEL OS, enterprises can subscribe to Windows Virtual Desktop from the Azure cloud with full confidence in the integrity, security, and manageability of their users' endpoint devices.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Ivanti



Ivanti User Workspace Manager is a Windows Virtual Desktop value-added service that eases desktop deployment and management by separating user data from the desktop for seamless portability. With Ivanti, users can deliver complex projects like migrating to Windows 10, adopting Microsoft 365, or moving services to the cloud faster.

When used with Windows Virtual Desktop, Ivanti User Workspace Manager provides simple contextual management of the user desktop experience, eliminating long sign-in times and eradicating group policy nightmares. Ivanti User Workspace Manager out-of-the-box templates simplify installation for users through agents and the existing console. Ivanti User Workspace Manager delivers responsive, secure desktops that users love, saving money on servers, managing users more effectively, and reducing endpoint security risk.

- [Go to the partner website.](#)

Lakeside Software



Lakeside Software is a Windows Virtual Desktop value-added services provider that equips IT teams with software for monitoring performance and assessing Azure migration readiness of user workloads. With this software, IT gains clearer visibility into application usage and resource consumption to streamline the migration process. Lakeside Software collects data at every workspace to create a comprehensive report on user environments, enabling quick troubleshooting and optimization of assets.

Lakeside Software's digital experience monitoring solution, SysTrack, can help provide a great user experience by tracking performance and identifying ideal workloads for migration. SysTrack works to extend the value of Windows Virtual Desktop through right-sizing assessments and continuous monitoring of user environments.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Liquidware



Liquidware is a Windows Virtual Desktop value-added services provider that delivers software that manages and optimizes Windows Virtual Desktop deployment. The Liquidware Essentials suite provides application delivery through layering, user environment management, and key user experience visibility and diagnostics. With solutions for assessing migration readiness and analyzing usage metrics, Liquidware provides a seamless virtual desktop experience for end users.

Liquidware Essentials extends the value of Windows Virtual Desktop by efficiently harvesting user profiles and gathering key user data to streamline migration of user environments to Azure. Additionally, Liquidware Essentials simplifies image management by unifying user profiles and layering apps based on configurable rights management settings.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Liquit



Liquit application aggregation and delivery software enables enterprises and service providers to connect to and combine with all workspace back-ends (Citrix, VMWare, Windows Virtual Desktop, RDP, and Legacy) and deliver a customized and consistent customer experience, regardless of where the customer's applications reside. When a customer publishes the smart icon, Liquit decides where to start the application based on the customer's location, device, and profile rights.

As a certified integration partner, Liquit helps accelerate transition to the cloud without a rip-and-replace delay. Windows Virtual Desktop can easily connect to an existing environment, create a workspace, and deliver the desktop. You can then take your time migrating off of old platforms and make changes on the back-end without your users noticing. Gain a consistent end-user experience, flexible infrastructure, and maintain control of your applications no matter where they are.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Login VSI



Login VSI is a Windows Virtual Desktop value-added services provider and Microsoft partner delivering software for application performance testing in Windows Virtual Desktop environments. Customers moving their on-premises business services to Windows Virtual Desktop use Login VSI Enterprise Edition to evaluate and maintain

optimal performance, scalability, and availability of Windows 10 Enterprise multi-session, Windows 10 Enterprise, and Windows 7 enabled with their business critical applications.

- [Go to the partner website.](#)

Nerdio



Nerdio is an Azure IT automation platform that makes it easy to deploy and manage Windows Virtual Desktop. Nerdio provides the knowledge and technology to deploy, price, package, manage, and optimize customers' Azure deployments—with Windows Virtual Desktop front-and-center.

Nerdio extends the value of Windows Virtual Desktop by making it easy to provision Azure resources and streamline deployment. With Nerdio for Azure, IT can automatically deploy and manage a complete Azure environment, including Windows Virtual Desktop, in under two hours.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Numecent



Numecent is a Windows Virtual Desktop value-added services provider that significantly reduces the total operating costs through rapid onboarding and migration of complicated or incompatible Windows apps in Windows Virtual Desktop environments. Numecent also minimizes the amount of configuration that users need to do, reduces application updates, and simplifies complex processes. Because Numecent Cloudpaging supports more applications seamlessly than any other application delivery tool, it reduces time and IT workloads in environments with a diverse set of applications.

When used with Windows Virtual Desktop, Cloudpaging further reduces costs by completing software asset lifecycle from deployment to upgrading, metering, and removing applications. Cloudpaging simplifies image management by dynamically provisioning apps as needed in real time to the Windows Virtual Desktop deployments. Cloudpaging helps applications run without administration or intervention through the periodic Windows 10 updates. Cloudpaging also reduces the licensing cost of expensive applications by enabling more efficient deployment and usage of these applications.

- [Go to the partner website.](#)

PolicyPak



PolicyPak Software is a Windows Virtual Desktop partner that performs total settings management for applications, desktop, browsers, Java, and security settings. PolicyPak keeps your desktop, system, and security settings in compliance. PolicyPak enhances the value of Windows Virtual Desktop by adding a suite of components to enhance Windows' built-in administration. Use your existing Active Directory Group Policy and/or Windows Intune to deliver PolicyPak's settings and increase administrators' ability to manage their Windows 10 machines.

The top use cases for PolicyPak are to remove local admin rights and overcome UAC prompts, block ransomware, manage multiple browsers, manage Internet Explorer's Enterprise and Compatibility modes, reduce the number of GPOs, manage Windows 10 File Associations, manage Windows 10 Start Menu and Taskbar, and manage Windows

10 Features and Optional features.

- [See the joint solution brief.](#)
- [Go to partner website.](#)

PrinterLogic



PrinterLogic is a Windows Virtual Desktop value-added service provider platform that empowers IT professionals to eliminate all print servers and deliver a highly available serverless printing infrastructure. PrinterLogic extends the value of Windows Virtual Desktop and Azure by making it easy to manage centrally and deploy printer objects to any printer or endpoint OS.

Available as SaaS or as a web stack in your own private cloud, the PrinterLogic platform ensures users always have the right printers they need in their virtual sessions based on user ID, device name, or location. This functionality is complemented by a full suite of enterprise print management features such as print tracking and reporting, mobile printing, and secure badge release printing.

- [Go to partner website.](#)

Printix



Printix is a Windows Virtual Desktop value-added service provider that automates user connection to office printing resources. As the missing piece in your customer Azure migration, Printix is the most cost-effective service available to remove infrastructure and IT tasks associated with supporting and optimizing print workflow for every user, regardless of location.

Printing is a fundamental task in just about every office and small business environment. In order to take full advantage of Windows Virtual Desktop and provide a great user experience, it's essential to ensure your users can connect to printers with minimum effort and maximum reliability. With Printix, you can get the most out of Windows Virtual Desktop through single sign-on (SSO), silent configuration, regular updates, and continuous monitoring of your print environment.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

RDPSoft



RDPSoft is a Windows Virtual Desktop partner that provides powerful and inexpensive monitoring, management, and reporting solutions. Their Remote Desktop Commander offerings allow IT professionals to gain insight into the health, performance, user activity, licensing, and security of their Windows Virtual Desktop deployments.

RDPSoft's Remote Desktop Commander solutions enhance Windows Virtual Desktop administration. Premium Management features simplify delegation of Windows Virtual Desktop management tasks to support desk staff by providing remote assistance, user session, and process management. At the same time, the Remote Desktop Commander Suite collects rich metrics about per-user performance and load, user activity and auditing, Windows Virtual Desktop connection quality (latency and bandwidth), licensing, and security into a central Azure SQL Database instance for review. With RDPSoft, rich historic reporting and comprehensive dashboards are just a click away.

- [Go to the partner website.](#)

Rimo3



Rimo3 enhances the Windows Virtual Desktop experience by accelerating deployment and improving ongoing change management. Rimo3 equips IT teams with the knowledge they need to support your application portfolio for Azure migration readiness of application workloads. Users can onboard applications and test them in their target Windows Virtual Desktop workspace quickly and painlessly. Users can also proactively understand the impact of any changes as their organization rolls out new applications and updates. Finally, IT admins can leverage Rimo3's Intelligent Smoke Testing capability and automate functionality testing without interrupting user sessions.

Rimo3 offers an easy-to-use, scalable, automated application testing platform. This platform includes capabilities for all three application testing fundamentals: compatibility, functionality, and performance. It allows organizations, leaders, and teams to improve business continuity, adopt change faster, and optimize user experience.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

sepago



sepago was founded in 2002 by four friends in Cologne. Today, sepago is an IT management consultancy with a steadily increasing number of sepagists, with locations throughout Germany in Cologne, Munich, and Hamburg. sepago are experts on automated application provisioning, virtualization, cloud solutions, and IT security. sepago supports medium-sized and large companies on their way to digital transformation and ensures that users can work securely and efficiently.

sepago's innovation and development lab builds smart solutions using big data and AI technologies. These solutions focus on improving the business, user experience, and administrations of partner products like Windows Virtual Desktop.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

ThinPrint



ThinPrint is a Windows Virtual Desktop value-added services provider that delivers simple and secure cloud printing from Windows Virtual Desktop. With its services and software, existing print infrastructure can be utilized to print documents from the cloud. ThinPrint enables connection to both local and network printers, making it easy for users to print while at the office or working remotely.

ThinPrint's ezeep solution extends the value of Windows Virtual Desktop by enabling the connection to existing enterprise print infrastructure. ezeep gives users control over printing in the enterprise no matter where they are. Using ezeep, users can bridge the gap between Windows Virtual Desktop and printing hardware.

- [See the joint solution brief.](#)
- [Go to partner website.](#)

Tricerat



Tricerat offers a superior print management solution for Windows Virtual Desktop and other desktop platforms. Tricerat software has robust functionality, offering a better experience for both users and administrators. Administrators gain efficiencies through complete driver management, simplified deployment of print queues, and consistent management across hybrid platforms. User experience improves with shorter sign-in times, intelligent print queues based on user, device, and network location, and self-service options for quick printer selection.

With Tricerat, printing is seamless in Windows Virtual Desktop and beyond. Tricerat software allows administrators to easily connect on-premises printers to the cloud, extending enterprise print management from traditional environments to new, modern workspaces.

- [Go to the partner website.](#)

vast limits



vast limits, the uberAgent company, provides visibility in Windows Virtual Desktop deployments. It creates software for enterprise IT because it knows how IT professionals think and which tools they need. Its products help IT pros be more efficient by giving them exactly what they need to get their jobs done; no more, no less.

uberAgent is a monitoring and analytics product designed for end-user computing that doesn't just collect data—it gives customers the information that matters. uberAgent has its own metrics, covering key aspects of user experience, application performance, and endpoint security, telling you everything you need to know about your Windows Virtual Desktop VMs without affecting your systems' user density. uberAgent simplifies troubleshooting, helps with sizing, and provides rich information vital for information security.

- [Go to the partner website.](#)

Workspace 365



Workspace 365 unites all your information (business data, documents, communication and micro apps) and provides access to any local, web, or hosted application in one workspace. It automatically adapts to your role, location, device, browser, and more to provide a personalized workspace. Users get a simplified and consistent experience, no matter what technology lies below the surface. You can integrate your current solutions, such as RDP, Citrix and legacy applications, and move them to Windows Virtual Desktop while maintaining the same user experience. Furthermore, you can integrate all your file locations, such as SharePoint, OneDrive, Teams, and file servers, in one document management app.

With Workspace 365, IT admins can make Windows Virtual Desktop-enabled applications available to people based on permissions. The admin can then add those applications to a shared application group. When the Windows Virtual Desktop application is visible in Workspace 365, users can open it from their workspace without having to sign in again.

- [See the joint solution brief.](#)
- [Go to the partner website.](#)

Workspot



Workspot is a Windows Virtual Desktop value-added services provider that equips enterprises with high-performance desktops and workstations in Azure. With Workspot, infrastructure provisioning is automated, which means users can access their Windows Virtual Desktop environment from anywhere around the world with high availability.

Workspot extends the value of Windows Virtual Desktop by simplifying the provisioning process of cloud desktop infrastructure. With Workspot, resources can be easily scaled up and down to meet the needs of different users and use cases. Workspot can optimize deployments for high-performance GPU workstations necessary for CAD and engineering users, as well as Windows applications and Windows 10 desktops for all business users.

- [See the joint solution brief.](#)
- [Go to partner website.](#)

Next steps

- [Learn more about Windows Virtual Desktop.](#)
- [Create a tenant in Windows Virtual Desktop.](#)

Windows Virtual Desktop FAQ

8/25/2020 • 5 minutes to read • [Edit Online](#)

This article answers frequently asked questions and explains best practices for Windows Virtual Desktop.

What are the minimum admin permissions I need to manage objects?

If you want to create host pools and other objects, you must be assigned the Contributor role on the subscription or resource group you're working with.

You must be assigned the User Access Admin role on an app group to publish app groups to users or user groups.

To restrict an admin to only manage user sessions, such as sending messages to users, signing out users, and so on, you can create custom roles. For example:

```
"actions": [
  "Microsoft.Resources/deployments/operations/read",
  "Microsoft.Resources/tags/read",
  "Microsoft.Authorization/roleAssignments/read",
  "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/read",
  "Microsoft.DesktopVirtualization/hostpools/sessionhosts/read",
  "Microsoft.DesktopVirtualization/hostpools/sessionhosts/write",
  "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/write",
  "Microsoft.DesktopVirtualization/hostpools/sessionhosts/usersessions/delete"
],
"notActions": [],
"dataActions": [],
"notDataActions": []
}
```

Does Windows Virtual Desktop support split Azure Active Directory models?

When a user is assigned to an app group, the service does a simple Azure role assignment. As a result, the user's Azure Active Directory (AD) and the app group's Azure AD must be in the same location. All service objects, such as host pools, app groups, and workspaces, also must be in the same Azure AD as the user.

You can create virtual machines (VMs) in a different Azure AD as long as you sync the Active Directory with the user's Azure AD in the same virtual network (VNET).

What are location restrictions?

All service resources have a location associated with them. A host pool's location determines which geography the service metadata for the host pool is stored in. An app group can't exist without a host pool. If you add apps to a RemoteApp app group, you'll also need a session host to determine the start menu apps. For any app group action, you'll also need a related data access on the host pool. To make sure data isn't being transferred between multiple locations, the app group's location should be the same as the host pool's.

Workspaces also must be in the same location as their app groups. Whenever the workspace updates, the related app group updates along with it. Like with app groups, the service requires that all workspaces are associated with app groups created in the same location.

How do you expand an object's properties in PowerShell?

When you run a PowerShell cmdlet, you only see the resource name and location.

For example:

```
Get-AzWvdHostPool -Name 0224hp -ResourceGroupName 0224rg
```

Location	Name	Type
westus	0224hp	Microsoft.DesktopVirtualization/hostpools

To see all of a resource's properties, add either `format-list` or `fl` to the end of the cmdlet.

For example:

```
Get-AzWvdHostPool -Name 0224hp -ResourceGroupName 0224rg |fl
```

To see specific properties, add the specific property names after `format-list` or `fl`.

For example:

```
Get-AzWvdHostPool -Name demohp -ResourceGroupName 0414rg |fl CustomRdpProperty
```

```
CustomRdpProperty :  
audiocapturemode:i:0;audiomode:i:0;drivestoredirect:s;;redirectclipboard:i:1;redirectcomports:i:0;redirectprint  
ers:i:1;redirectsmartcards:i:1;screen modeid:i:2;
```

Does Windows Virtual Desktop support guest users?

Windows Virtual Desktop doesn't support Azure AD guest user accounts. For example, let's say a group of guest users have Microsoft 365 E3 Per-user, Windows E3 Per-user, or WIN VDA licenses in their own company, but are guest users in a different company's Azure AD. The other company would manage the guest users' user objects in both Azure AD and Active Directory like local accounts.

You can't use your own licenses for the benefit of a third party. Also, Windows Virtual Desktop doesn't currently support Microsoft Account (MSA).

Why don't I see the client IP address in the WVDConnections table?

We don't currently have a reliable way to collect the web client's IP addresses, so we don't include that value in the table.

How does Windows Virtual Desktop handle backups?

There are multiple options in Azure for handling backup. You can use Azure backup, Site Recovery, and snapshots.

Does Windows Virtual Desktop support third-party collaboration apps?

Windows Virtual Desktop is currently optimized for Teams. Microsoft currently doesn't support third-party collaboration apps like Zoom. Third-party organizations are responsible for giving compatibility guidelines to their customers. Windows Virtual Desktop also doesn't support Skype for Business.

Can I change from pooled to personal host pools?

Once you create a host pool, you can't change its type. However, you can move any VMs you register to a host pool to a different type of host pool.

What's the largest profile size FSLogix can handle?

Limitations or quotas in FSLogix depend on the storage fabric used to store user profile VHD(X) files.

The following table gives an example of how many resources an FSLogix profile needs to support each user. Requirements can vary widely depending on the user, applications, and activity on each profile.

RESOURCE	REQUIREMENT
Steady state IOPS	10
Sign in/sign out IOPS	50

The example in this table is of a single user, but can be used to estimate requirements for the total number of users in your environment. For example, you'd need around 1,000 IOPS for 100 users, and around 5,000 IOPS during sign-in and sign-out.

Is there a scale limit for host pools created in the Azure portal?

These factors can affect scale limit for host pools:

- The Azure template is limited to 800 objects. To learn more, see [Azure subscription and service limits, quotas, and constraints](#). Each VM also creates about six objects, so that means you can create around 132 VMs each time you run the template.
- There are restrictions on how many cores you can create per region and per subscription. For example, if you have an Enterprise Agreement subscription, you can create 350 cores. You'll need to divide 350 by either the default number of cores per VM or your own core limit to determine how many VMs you can create each time you run the template. Learn more at [Virtual Machines limits - Azure Resource Manager](#).
- The VM prefix name and the number of VMs is fewer than 15 characters. To learn more, see [Naming rules and restrictions for Azure resources](#).

Can I manage Windows Virtual Desktop environments with Azure Lighthouse?

Azure Lighthouse doesn't fully support managing Windows Virtual Desktop environments. Since Lighthouse doesn't currently support cross-Azure AD tenant user management, Lighthouse customers still need to sign in to the Azure AD that customers use to manage users.

You also can't use CSP sandbox subscriptions with the Windows Virtual Desktop service. To learn more, see [Integration sandbox account](#).

Finally, if you enabled the resource provider from the CSP owner account, the CSP customer accounts won't be able to modify the resource provider.

Windows 10 Enterprise multi-session FAQ

8/25/2020 • 5 minutes to read • [Edit Online](#)

This article answers frequently asked questions and explains best practices for Windows 10 Enterprise multi-session.

What is Windows 10 Enterprise multi-session?

Windows 10 Enterprise multi-session, formerly known as Windows 10 Enterprise for Virtual Desktops (EVD), is a new Remote Desktop Session Host that allows multiple concurrent interactive sessions. Previously, only Windows Server could do this. This capability gives users a familiar Windows 10 experience while IT can benefit from the cost advantages of multi-session and use existing per-user Windows licensing instead of RDS Client Access Licenses (CALs). For more information about licenses and pricing, see [Windows Virtual Desktop pricing](#).

How many users can simultaneously have an interactive session on Windows 10 Enterprise multi-session?

How many interactive sessions that can be active at the same time relies on your system's hardware resources (vCPU, memory, disk, and vGPU), how your users use their apps while signed in to a session, and how heavy your system's workload is. We suggest you validate your system's performance to understand how many users you can have on Windows 10 Enterprise multi-session. To learn more, see [Windows Virtual Desktop pricing](#).

Why does my application report Windows 10 Enterprise multi-session as a Server operating system?

Windows 10 Enterprise multi-session is a virtual edition of Windows 10 Enterprise. One of the differences is that this operating system (OS) reports the [ProductType](#) as having a value of 3, the same value as Windows Server. This property keeps the OS compatible with existing RDSH management tooling, RDSH multi-session-aware applications, and mostly low-level system performance optimizations for RDSH environments. Some application installers can block installation on Windows 10 multi-session depending on whether they detect the ProductType is set to Client. If your app won't install, contact your application vendor for an updated version.

Can I run Windows 10 Enterprise multi-session on-premises?

Windows 10 Enterprise multi-session can't run in on-premises production environments because it's optimized for the Windows Virtual Desktop service for Azure. It's against the licensing agreement to run Windows 10 Enterprise multi-session outside of Azure for production purposes. Windows 10 Enterprise multi-session won't activate against on-premises Key Management Services (KMS).

How do I customize the Windows 10 Enterprise multi-session image for my organization?

You can start a virtual machine (VM) in Azure with Windows 10 Windows 10 Enterprise multi-session and customize it by installing LOB applications, sysprep/generalize, and then create an image using the Azure portal.

To get started, create a VM in Azure with Windows 10 Enterprise multi-session. Instead of starting the VM in Azure, you can download the VHD directly. After that, you'll be able to use the VHD you downloaded to create a new Generation 1 VM on a Windows 10 PC with Hyper-V enabled.

Customize the image to your needs by installing LOB applications and sysprep the image. When you're done customizing, upload the image to Azure with the VHD inside. After that, get Windows Virtual Desktop from the Azure Marketplace and use it to deploy a new host pool with the customized image.

How do I manage Windows 10 Enterprise multi-session after deployment?

You can use any supported configuration tool, but we recommend Configuration Manager version 1906 because it supports Windows 10 Enterprise multi-session. We're currently working on Microsoft Intune support.

Can Windows 10 Enterprise multi-session be Azure Active Directory (AD)-joined?

Windows 10 Enterprise multi-session is currently supported to be hybrid Azure AD-joined. After Windows 10 Enterprise multi-session is domain-joined, use the existing Group Policy Object to enable Azure AD registration. For more information, see [Plan your hybrid Azure Active Directory join implementation](#).

Where can I find the Windows 10 Enterprise multi-session image?

Windows 10 Enterprise multi-session is in the Azure gallery. To find it, navigate to the Azure portal and search for the Windows 10 Enterprise for Virtual Desktops release. For an image integrated with Microsoft 365 Apps for enterprise, go to the Azure portal and search for **Microsoft Windows 10 + Microsoft 365 Apps for enterprise**.

Which Windows 10 Enterprise multi-session image should I use?

The Azure gallery has several releases, including Windows 10 Enterprise multi-session, version 1809, and Windows 10 Enterprise multi-session, version 1903. We recommend using the latest version for improved performance and reliability.

Which Windows 10 Enterprise multi-session versions are supported?

Windows 10 Enterprise multi-session, versions 1809 and later are supported and are available in the Azure gallery. These releases follow the same support lifecycle policy as Windows 10 Enterprise, which means the March release is supported for 18 months and the September release for 30 months.

Which profile management solution should I use for Windows 10 Enterprise multi-session?

We recommend you use FSLogix profile containers when you configure Windows 10 Enterprise in non-persistent environments or other scenarios that need a centrally stored profile. FSLogix ensures the user profile is available and up-to-date for every user session. We also recommend you use your FSLogix profile container to store a user profile in any SMB share with appropriate permissions, but you can store user profiles in Azure page blob storage if necessary. Windows Virtual Desktop users can use FSLogix at no additional cost.

For more information about how to configure an FSLogix profile container, see [Configure the FSLogix profile container](#).

Which license do I need to access Windows 10 Enterprise multi-session?

For a full list of applicable licenses, see [Windows Virtual Desktop pricing](#).

Why do my apps disappear after I sign out?

This happens because you're using Windows 10 Enterprise multi-session with a profile management solution like FSLogix. Your admin or profile solution configured your system to delete user profiles when users sign out. This configuration means that when your system deletes your user profile after you sign out, it also removes any apps you installed during your session. If you want to keep the apps you installed, you'll need to ask your admin to provision these apps for all users in your Windows Virtual Desktop environment.

How do I make sure apps don't disappear when users sign out?

Most virtualized environments are configured by default to prevent users from installing additional apps to their profiles. If you want to make sure an app doesn't disappear when your user signs out of Windows Virtual Desktop, you have to provision that app for all user profiles in your environment. For more information about provisioning apps, check out these resources:

- [Publish built-in apps in Windows Virtual Desktop](#)
- [DISM app package servicing command-line options](#)
- [Add-AppxProvisionedPackage](#)

How do I make sure users don't download and install apps from the Microsoft Store?

You can disable the Microsoft Store app to make sure users don't download extra apps beyond the apps you've already provisioned for them.

To disable the Store app:

1. Create a new Group Policy.
2. Select **Computer Configuration > Administrative Templates > Windows Components**.
3. Select **Store**.
4. Select **Store Application**.
5. Select **Disabled**, then select **OK**.
6. Select **Apply**.

Next steps

To learn more about Windows Virtual Desktop and Windows 10 Enterprise multi-session:

- Read our [Windows Virtual Desktop documentation](#)
- Visit our [Windows Virtual Desktop TechCommunity](#)
- Set up your Windows Virtual Desktop deployment with the [Windows Virtual Desktop tutorials](#)

What is MSIX app attach?

8/25/2020 • 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 Windows Virtual Desktop. This article will describe what MSIX app attach is and what it can do for you.

Application delivery options in Windows Virtual Desktop

You can deliver apps in Windows 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 Windows 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

8/25/2020 • 3 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 support SMB 3 or later. Applications get staged from this MSIX share without having to move application files to the system drive.

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

Expanding MSIX package is a multi-step process. It takes the MSIX file and put 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 Windows Virtual Desktop, uploads happen once per MSIX share. Once you upload a package, all host pools in the same subscription can reference it.

Publish an MSIX package

In Windows Virtual Desktop, publishing an MSIX package links it to a remote app or desktop.

Assign an MSIX package

In Windows 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 Windows 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 Windows 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 than VHD files. CIM also consumes less CPU and memory than VHD.

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](#).

MSIX app attach FAQ

8/25/2020 • 2 minutes to read • [Edit Online](#)

This article answers frequently asked questions about MSIX app attach for Windows Virtual Desktop.

Does MSIX app attach use FSLogix?

MSIX app attach doesn't use FSLogix. However, app attach and FSLogix are designed to work together to provide a seamless user experience.

Can I use MSIX app attach outside of Windows Virtual Desktop?

Yes, MSIX app attach is a feature that's included with Windows 10 Enterprise and can be used outside of Windows Virtual Desktop. However, there's no management plane for MSIX app attach outside of Windows Virtual Desktop.

How do I get an MSIX package?

Your software vendor will give you an MSIX package. You can also convert non-MSIX packages to MSIX. Learn more at [How to move your existing installers to MSIX](#).

Which operating systems support MSIX app attach?

Windows 10 Enterprise and Windows 10 Enterprise Multi-session.

Next steps

If you want to learn more about MSIX app attach, check out our [overview glossary](#). Otherwise, get started with [Set up app attach](#).

Data and metadata locations for Windows Virtual Desktop

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows 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 the [Azure datacenter map](#).

NOTE

Microsoft doesn't control or limit the regions where you or your users can access your user and app-specific data.

IMPORTANT

Windows Virtual Desktop stores global metadata information like tenant names, host pool names, app group names, and user principal names in a datacenter. Whenever a customer creates a service object, they must enter a location for the service object. The location they enter determines where the metadata for the object will be stored. The customer will choose an Azure region and the metadata will be stored in the related geography. For a list of all Azure regions and related geographies, see [Azure geographies](#).

At the moment, we only support storing metadata in the United States (US) Azure geography. The stored metadata is encrypted at rest, and geo-redundant mirrors are maintained within the geography. All customer data, such as app settings and user data, resides in the location the customer chooses and isn't managed by the service. More geographies will become available as the service grows.

Service metadata is replicated within the Azure geography for disaster recovery purposes.

Troubleshooting overview, feedback, and support for Windows Virtual Desktop

8/25/2020 • 4 minutes to read • [Edit Online](#)

IMPORTANT

This content applies to Windows Virtual Desktop with Azure Resource Manager Windows Virtual Desktop objects. If you're using Windows 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 a Windows Virtual Desktop environment and provides ways to resolve the issues.

Report issues

To report issues or suggest features for Windows Virtual Desktop with Azure Resource Manager integration, visit the [Windows 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 Windows Virtual Desktop aren't being used	Open an Azure support request , then select Windows Virtual Desktop for the service. For issues with the Azure Resource Manager templates that are provided with Windows Virtual Desktop, see Azure Resource Manager template errors section of Host pool creation .

ISSUE	SUGGESTED SOLUTION
Managing Windows Virtual Desktop session host environment from the Azure portal	<p>Open an Azure support request.</p> <p>For management issues when using Remote Desktop Services/Windows Virtual Desktop PowerShell, see Windows Virtual Desktop PowerShell or open an Azure support request, select Windows Virtual Desktop for the service, select Configuration and management for the problem type, then select Issues configuring environment using PowerShell for the problem subtype.</p>
Managing Windows Virtual Desktop configuration tied to host pools and application groups (app groups)	See Windows Virtual Desktop PowerShell , or open an Azure support request , select Windows 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 Windows Virtual Desktop for the service, select FSLogix for the problem type, then select the appropriate problem subtype.
Remote desktop clients malfunction on start	<p>See Troubleshoot the Remote Desktop client and if that doesn't resolve the issue, Open an Azure support request, select Windows Virtual Desktop for the service, then select Remote Desktop clients for the problem type.</p> <p>If it's a network issue, your users need to contact their network administrator.</p>
Connected but no feed	<p>Troubleshoot using the User connects but nothing is displayed (no feed) section of Windows Virtual Desktop service connections.</p> <p>If your users have been assigned to an app group, open an Azure support request, select Windows Virtual Desktop for the service, then select Remote Desktop Clients for the problem type.</p>
Feed discovery problems due to the network	Your users need to contact their network administrator.
Connecting clients	See Windows 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	Verify that your third-party provider supports Windows Virtual Desktop scenarios and approach them regarding any known issues.

ISSUE	SUGGESTED SOLUTION
Issues using Log Analytics for Windows Virtual Desktop	<p>For issues with the diagnostics schema, open an Azure support request.</p> <p>For queries, visualization, or other issues in Log Analytics, select the appropriate problem type under Log Analytics.</p>
Issues using M365 apps	Contact the M365 admin center with one of the M365 admin center help options .

Next steps

- To troubleshoot issues while creating a host pool in a Windows Virtual Desktop environment, see [host pool creation](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues with Windows Virtual Desktop client connections, see [Windows Virtual Desktop service connections](#).
- To troubleshoot issues with Remote Desktop clients, see [Troubleshoot the Remote Desktop client](#)
- To troubleshoot issues when using PowerShell with Windows Virtual Desktop, see [Windows Virtual Desktop PowerShell](#).
- To learn more about the service, see [Windows 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](#).

Identify and diagnose Windows Virtual Desktop issues

8/25/2020 • 4 minutes to read • [Edit Online](#)

IMPORTANT

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

Windows Virtual Desktop offers a diagnostics feature that allows the administrator to identify issues through a single interface. To learn more about the diagnostic capabilities of Windows Virtual Desktop, see [Use Log Analytics for the diagnostics feature](#).

Connections that don't reach Windows Virtual Desktop won't show up in diagnostics results because the diagnostics role service itself is part of Windows Virtual Desktop. Windows Virtual Desktop connection issues can happen when the end-user is experiencing network connectivity issues.

Common error scenarios

Error scenarios are categorized in internal to the service and external to Windows Virtual Desktop.

- Internal Issue: specifies scenarios that can't be mitigated by the customer and need to be resolved as a support issue. When providing feedback through the [Windows Virtual Desktop Tech Community](#), include the correlation ID and approximate time frame of when the issue occurred.
- External Issue: relate to scenarios that can be mitigated by the customer. These are external to Windows Virtual Desktop.

The following table lists common errors your admins might run into.

NOTE

This list includes most common errors and is updated on a regular cadence. To ensure you have the most up-to-date information, be sure to check back on this article at least once a month.

Management errors

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's unavailable, drain mode can't be changed.

ERROR MESSAGE	SUGGESTED SOLUTION
Failed to disconnect user sessions	Couldn't disconnect the user from the VM. Check the VM status. If the VM's unavailable, the user session can't be disconnected. 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.

External connection error codes

NUMERIC CODE	ERROR CODE	SUGGESTED SOLUTION
-2147467259	ConnectionFailedAdTrustedRelationshipFailure	The session host is not correctly joined to the Active Directory.
-2146233088	ConnectionFailedUserHasValidSessionButRdshIsUnhealthy	The connections failed because the session host is unavailable. Check the session host's health.
-2146233088	ConnectionFailedClientDisconnect	If you see this error frequently, make sure the user's computer is connected to the network.
-2146233088	ConnectionFailedNoHealthyRdshAvailable	The session the host user tried to connect to isn't healthy. Debug the virtual machine.
-2146233088	ConnectionFailedUserNotAuthorized	The user doesn't have permission to access the published app or desktop. The error might appear after the admin removed published resources. Ask the user to refresh the feed in the Remote Desktop application.
2	FileNotFound	<p>The application the user tried to access is either incorrectly installed or set to an incorrect path.</p> <p>When publishing new apps while the user has an active session, the user won't be able to access this app. The session must be shut down and restarted before the user can access the app.</p>

NUMERIC CODE	ERROR CODE	SUGGESTED SOLUTION
3	InvalidCredentials	The username or password the user entered doesn't match any existing usernames or passwords. Review the credentials for typos and try again.
8	ConnectionBroken	The connection between Client and Gateway or Server dropped. No action needed unless it happens unexpectedly.
14	UnexpectedNetworkDisconnect	The connection to the network dropped. Ask the user to connect again.
24	ReverseConnectFailed	The host virtual machine has no direct line of sight to RD Gateway. Ensure the Gateway IP address can be resolved.

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).

Cause: 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.

Fix: 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 learn more about roles within Windows Virtual Desktop, see [Windows Virtual Desktop environment](#).

To see a list of available PowerShell cmdlets for Windows Virtual Desktop, see the [PowerShell reference](#).

Host pool creation

8/25/2020 • 8 minutes to read • [Edit Online](#)

IMPORTANT

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

This article covers issues during the initial setup of the Windows Virtual Desktop tenant and the related session host pool infrastructure.

Provide feedback

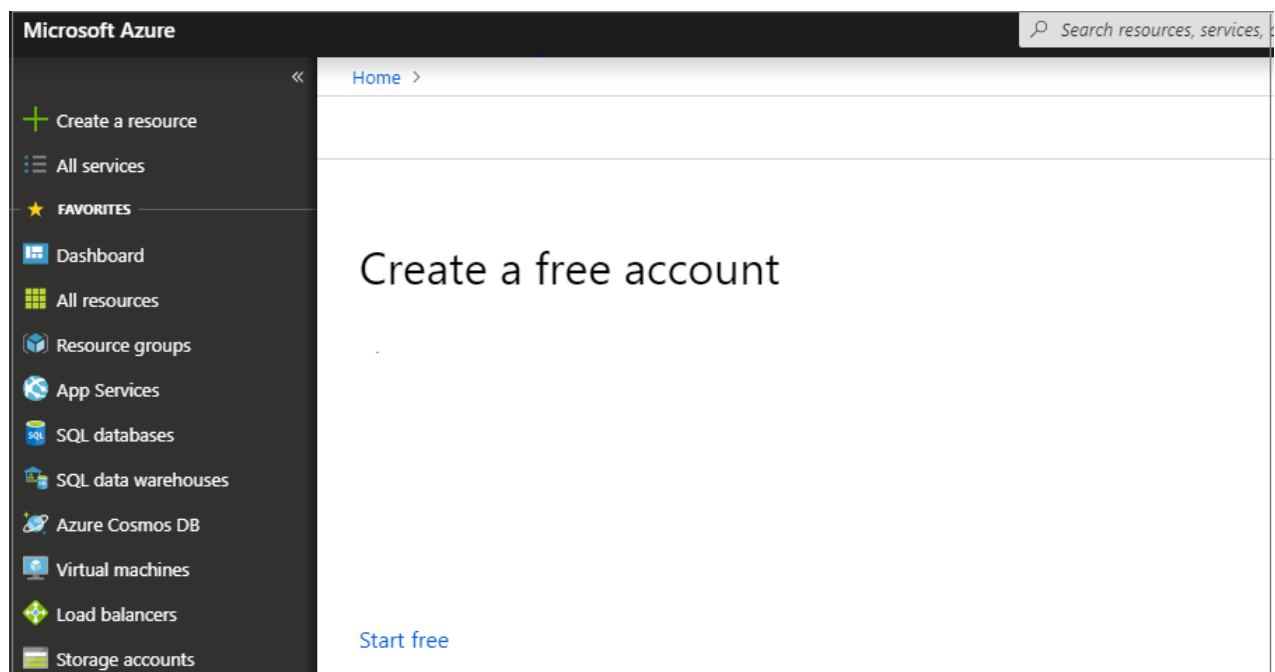
Visit the [Windows Virtual Desktop Tech Community](#) to discuss the Windows Virtual Desktop service with the product team and active community members.

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.

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

 The resource operation completed with terminal provisioning state 'Failed'. Click here for details →

Your deployment failed



Deployment name: vmCreation-linkedTemplate-56284822-e435-4...




Subscription: Microsoft Azure

Resource group: 0226RG

Start time: 2/26/2020, 11:29:11 AM

Correlation ID: aae12551-2578-44a6-8f05-443c162909a2

Deployment details [\(Download\)](#)

Resource	Type	Status	Operation details
 pdwindows-1/joindomain	Microsoft.Compute/virtual...	Conflict	Operation details
 pdwindows-0/joindomain	Microsoft.Compute/virtual...	Conflict	Operation details
 pdwindows-0	Microsoft.Compute/virtual...	OK	Operation details

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\r\n\"status\": \"Failed\", \r\n\r\n\"error\":  
\r\n\r\n\"code\": \"ResourceDeploymentFailure\", \r\n\r\n\"message\": \"The resource operation completed with terminal provisioning state 'Failed'.  
\r\n\r\n\"details\": [\r\n\r\n {\r\n\r\n \"code\": \"VMExtensionProvisioningError\", \r\n\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 ] \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      \"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  \"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

[Delete](#) [Cancel](#) [Redeploy](#) [Refresh](#)

 The resource operation completed with terminal provisioning state 'Failed'. Click here for details →

Your deployment failed


Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.



Deployment name: Microsoft.Template
Subscription:
Resource group:

DEPLOYMENT DETAILS [\(Download\)](#)

Start time: 11/20/2018, 10:26:21 AM
Duration: 14 minutes 45 seconds
Correlation ID: 6a217c6b-847d-4d93-8685-0b010aa0540e

RESOURCE	TYPE	STATUS	OPERATION DETAILS
 test-1/rd	Microsoft.Compute/virtualMac...	Conflict	Operation details

Cause 1: Transient error with the Windows Virtual Desktop environment.

Cause 2: Transient error with connection.

Fix: Confirm Windows 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

Dashboard > rds.wvd-hostpool4-preview-20190129125249 - Overview > vmCreation-linkedTemplate - Overview

vmCreation-linkedTemplate - Overview

Deployment

Search (Ctrl+ /) « Delete Cancel Redeploy Refresh

The Admin Username specified is not allowed. Click here for details →

Your deployment failed

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find

Deployment name: vmCreation-linkedTemplate
Subscription: [Microsoft Azure](#)
Resource group: [demoHostDesktop](#)

DEPLOYMENT DETAILS (Download)
Start time: 1/29/2019, 12:52:58 PM
Duration: 23 seconds
Correlation ID: ff02cb6f-e7a6-4acc-9fbb-6a3b6281e0d5

RESOURCE	TYPE	STATUS	OPERATION
demoHostv2-1	Microsoft.Compute/virtualMachines	BadRequest	Operation c
demoHostv2-0	Microsoft.Compute/virtualMachines	BadRequest	Operation c
demoHostv2-image	Microsoft.Compute/images	OK	Operation c

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

Delete Cancel Redeploy Refresh

The resource operation completed with terminal provisioning state 'Failed'. Click here for details →

Your deployment failed

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to

Deployment name: rds.wvd-hostpool4-preview-20190129132410
Subscription: [Microsoft Azure](#)
Resource group: [demoHostD](#)

DEPLOYMENT DETAILS (Download)
Start time: 1/29/2019, 1:24:12 PM
Duration: 19 minutes
Correlation ID: 0383c14b-143a-44e2-a7fe-7fe9f5e74c98

RESOURCE	TYPE	STATUS
desktop-1/dscextension	Microsoft.Compute/virtualMachines/ext...	Conflict
desktop-0/dscextension	Microsoft.Compute/virtualMachines/ext...	Conflict
desktop-1/joindomain	Microsoft.Compute/virtualMachines/ext...	OK

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
'dsccextension'.
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)

DEPLOYMENT NAME	STATUS
vmCreation-linkedTemplate	✓ Succeeded
pid-836bce42-d18b-4b20-97	✓ Succeeded
rds.wvd-hostpool5-preview-2	✗ Failed (Error details)

Summary [Raw Error](#)

ERROR DETAILS

▼ The resource operation completed with terminal provisioning state 'Failed'. (Code: ResourceDeploymentFailure)

- VM has reported a failure when processing extension 'dsccextension'. 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.". (Code: VMExtensionProvisioningError)

WAS THIS HELPFUL?

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 'dsccextension'. 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  } }"
    }
  ]
}
```

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\\n\\\"error\\\": {\\r\\n\\n\\\"code\\\": \\\"InvalidResourceReference\\\",\\r\\n\\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\\n\\\"details\\\": [\\r\\n\\n}\\r\\n\\\"\\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\": \"{\\\"error\\\": {\\\"code\\\": \\\"InvalidResourceReference\\\", \\\"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.\\\"}, \\\"details\\\": []}\\\"}\\\"\", \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 Windows Virtual Desktop and the escalation tracks, see [Troubleshooting overview, feedback, and support](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues with Windows Virtual Desktop client connections, see [Windows Virtual Desktop service connections](#).
- To troubleshoot issues with Remote Desktop clients, see [Troubleshoot the Remote Desktop client](#)
- To troubleshoot issues when using PowerShell with Windows Virtual Desktop, see [Windows Virtual Desktop PowerShell](#).
- To learn more about the service, see [Windows 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

8/25/2020 • 13 minutes to read • [Edit Online](#)

IMPORTANT

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

Use this article to troubleshoot issues you're having when configuring the Windows Virtual Desktop session host virtual machines (VMs).

Provide feedback

Visit the [Windows Virtual Desktop Tech Community](#) to discuss the Windows Virtual Desktop service with the product team and active community members.

VMs are not 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 multi-factor 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 do not 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](#).

Windows Virtual Desktop Agent and Windows Virtual Desktop Boot Loader are not installed

The recommended way to provision VMs is using the Azure portal creation template. The template automatically installs the Windows Virtual Desktop Agent and Windows 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 **Windows Virtual Desktop Agent** and **Windows Virtual Desktop Agent Boot Loader** are not 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 was not 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: Windows Virtual Desktop Agent and Windows 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 Windows 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 Windows 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 Windows Virtual Desktop.

Fix: Confirm the items in the following list.

- Manually register the VMs with the Windows Virtual Desktop service.
- Confirm account used for connecting to Windows Virtual Desktop has permissions on the Azure subscription or resource group to create host pools.
- Confirm account doesn't have MFA.

Windows Virtual Desktop Agent is not registering with the Windows Virtual Desktop service

When the Windows 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 Windows Virtual Desktop Agent and the token.

Error: The status filed in Get-AzWvdSessionHost cmdlet shows status as Unavailable

```
Microsoft Windows [Version 10.0.18215.1000]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ssa.GT090617.000>qwinsta
SESSIONNAME      USERNAME          ID  STATE  TYPE
services         0  Disc
console          1  Conn
>rdp-tcp#34      ssa              2  Active
31c5ce94259d4... 65536 Listen
rdp-tcp          65537 Listen
rdp-sxs          65538 Listen
```

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 Windows 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: Windows 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 -ExpirationTime parameter is set to 3 days.

Error: Windows 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 Windows Virtual Desktop side-by-side stack

The Windows Virtual Desktop side-by-side stack is automatically installed with Windows Server 2019. 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 Windows 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 Windows 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.

```
Microsoft Windows [Version 10.0.18215.1000]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ssa.GT090617.000>qwinsta
SESSIONNAME      USERNAME          ID   STATE   TYPE
services         0               Disc
console          1               Conn
>rdp-tcp#34      ssa              2     Active
31c5ce94259d4... 65536           Listen
rdp-tcp          65537           Listen
rdp-sxs          65538           Listen
```

Examine the registry entries listed below and confirm that their values match. If registry keys are missing or values are mismatched, 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

```
Microsoft Windows [Version 10.0.18215.1000]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ssa.GT090617.000>qwinsta
SESSIONNAME      USERNAME          ID   STATE   TYPE
services         0               Disc
console          1               Conn
>rdp-tcp#34       ssa              2   Active
31c5ce94259d4... 65536          Listen
rdp-tcp          65537          Listen
rdp-sxs          65538          Listen
```

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 a Windows 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 Windows Virtual Desktop side-by-side stack. Once you uninstall the side-by-side stack, go to "Register the VM with the Windows 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.

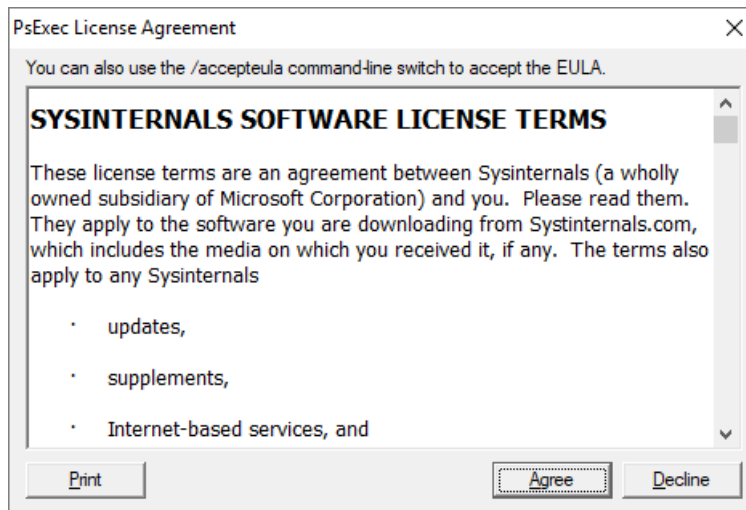
5. Navigate to folder where PsExec was unzipped.
6. From command prompt, use the following command:

```
psexec.exe \\<VMname> cmd
```

NOTE

VMname is the machine name of the VM with the malfunctioning side-by-side stack.

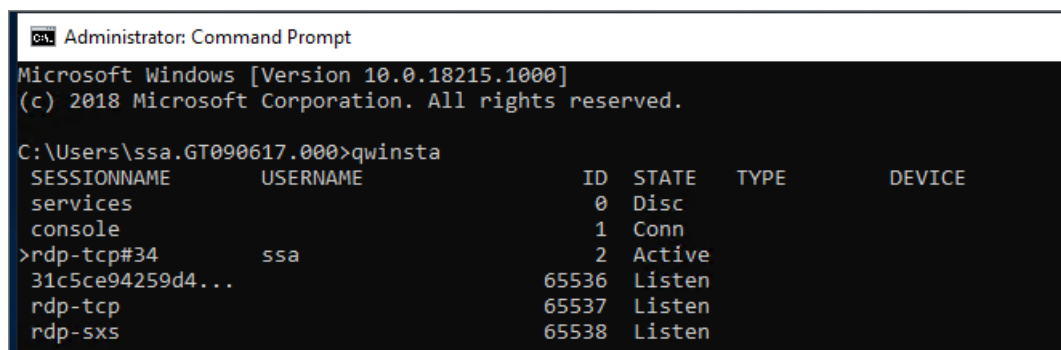
7. Accept the PsExec License Agreement by clicking Agree.



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 Windows 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 is not 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, this means the image doesn't have the latest Windows updates installed or that you are 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

Windows 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.

Windows specifications		Windows specifications	
Edition	Windows 10 Enterprise for Virtual Desktops	Edition	Windows 10 Enterprise for Virtual Desktops
Version	1809	Version	1903
Installed on	8/6/2019	Installed on	7/23/2019
OS build	17763.615	OS build	18362.239

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](#).

Next steps

- For an overview on troubleshooting Windows Virtual Desktop and the escalation tracks, see [Troubleshooting overview, feedback, and support](#).
- To troubleshoot issues while creating a host pool in a Windows Virtual Desktop environment, see [Environment and host pool creation](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues with Windows Virtual Desktop client connections, see [Windows Virtual Desktop service](#)

[connections](#).

- To troubleshoot issues with Remote Desktop clients, see [Troubleshoot the Remote Desktop client](#)
- To troubleshoot issues when using PowerShell with Windows Virtual Desktop, see [Windows Virtual Desktop PowerShell](#).
- To learn more about the service, see [Windows 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](#).

Windows Virtual Desktop service connections

8/25/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

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

Use this article to resolve issues with Windows Virtual Desktop client connections.

Provide feedback

You can give us feedback and discuss the Windows Virtual Desktop Service with the product team and other active community members at the [Windows 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. Windows Virtual Desktop doesn't support Azure AD distribution groups.

Next steps

- For an overview on troubleshooting Windows Virtual Desktop and the escalation tracks, see [Troubleshooting overview, feedback, and support](#).
- To troubleshoot issues while creating a Windows Virtual Desktop environment and host pool in a Windows Virtual Desktop environment, see [Environment and host pool creation](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues when using PowerShell with Windows Virtual Desktop, see [Windows Virtual Desktop PowerShell](#).
- To go through a troubleshoot tutorial, see [Tutorial: Troubleshoot Resource Manager template deployments](#).

Troubleshoot the Remote Desktop client

8/25/2020 • 2 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 Windows Virtual Desktop, you should either sign out or

use a private browser window.

If you're using Windows 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 Windows 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.

Next steps

- For an overview on troubleshooting Windows Virtual Desktop and the escalation tracks, see [Troubleshooting overview, feedback, and support](#).
- To troubleshoot issues while creating a Windows Virtual Desktop environment and host pool in a Windows Virtual Desktop environment, see [Environment and host pool creation](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues when using PowerShell with Windows Virtual Desktop, see [Windows Virtual Desktop PowerShell](#).
- To go through a troubleshoot tutorial, see [Tutorial: Troubleshoot Resource Manager template deployments](#).

Windows Virtual Desktop PowerShell

8/25/2020 • 3 minutes to read • [Edit Online](#)

IMPORTANT

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

Use this article to resolve errors and issues when using PowerShell with Windows Virtual Desktop. For more information on Remote Desktop Services PowerShell, see [Windows Virtual Desktop PowerShell](#).

Provide feedback

Visit the [Windows Virtual Desktop Tech Community](#) to discuss the Windows Virtual Desktop service with the product team and active community members.

PowerShell commands used during Windows Virtual Desktop setup

This section lists PowerShell commands that are typically used while setting up Windows 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 Windows 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 Windows 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: Windows 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 Windows Virtual Desktop and the escalation tracks, see [Troubleshooting overview, feedback, and support](#).
- To troubleshoot issues while setting up your Windows Virtual Desktop environment and host pools, see [Environment and host pool creation](#).
- To troubleshoot issues while configuring a virtual machine (VM) in Windows Virtual Desktop, see [Session host virtual machine configuration](#).
- To troubleshoot issues with Windows Virtual Desktop client connections, see [Windows Virtual Desktop service connections](#).
- To troubleshoot issues with Remote Desktop clients, see [Troubleshoot the Remote Desktop client](#).
- To learn more about the service, see [Windows 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](#).

Diagnose graphics performance issues in Remote Desktop

8/25/2020 • 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 Windows 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](#).

Security best practices

8/25/2020 • 7 minutes to read • [Edit Online](#)

Windows Virtual Desktop is a managed virtual desktop service that includes many security capabilities for keeping your organization safe. In a Windows 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' Windows 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 Windows 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 Windows 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

Windows Virtual Desktop is a service under Azure. To maximize the safety of your Windows Virtual Desktop

deployment, you should make sure to secure the surrounding Azure infrastructure and management plane as well. To secure your infrastructure, consider how Windows 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 Azure Security Center

We recommend you enable Azure Security Center Standard for subscriptions, virtual machines, key vaults, and storage accounts.

With Azure Security Center Standard, you can:

- Manage vulnerabilities.
- Assess compliance with common frameworks like PCI.
- Strengthen the overall security of your environment.

To learn more, see [Onboard your Azure subscription to Security Center Standard](#).

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 Azure Security Center](#).

Windows Virtual Desktop security best practices

Windows 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 Windows Virtual Desktop improves the security of your entire deployment. To learn more, see [Enable Azure Multi-Factor Authentication for Windows Virtual Desktop](#).

Enable Conditional Access

Enabling [Conditional Access](#) lets you manage risks before you grant users access to your Windows 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 Windows Virtual Desktop. Some examples of key audit logs are:

- [Azure Activity Log](#)
- [Azure Active Directory Activity Log](#)
- [Azure Active Directory](#)
- [Session hosts](#)
- [Windows 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 a subset of the remote machine exposed by the application.

Monitor usage with Azure Monitor

Monitor your Windows Virtual Desktop service's usage and availability with [Azure Monitor](#). Consider creating [service health alerts](#) for the Windows 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 Windows 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 [Azure Security Center](#) 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. Azure Security Center 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 Azure Security Center 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 Windows 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 Windows 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 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.

Next steps

To learn how to enable multi-factor authentication, see [Set up multi-factor authentication](#).


Linux support

8/25/2020 • 2 minutes to read • [Edit Online](#)

You can access Windows Virtual Desktop resources from your Linux devices with the following supported clients, provided by our Linux thin client partners. We are working with a number of partners to enable supported Windows Virtual Desktop clients on more Linux-based operating systems and devices. If you would like Windows Virtual Desktop support on a Linux platform that is not listed here, please let us know on our [UserVoice page](#).

Connect with your Linux device

The following partners have approved Windows Virtual Desktop clients for Linux devices.

PARTNER	PARTNER DOCUMENTATION	PARTNER SUPPORT
	IGEL client documentation	IGEL support

What is the Linux SDK?

Linux thin client partners can use the Windows Virtual Desktop Linux SDK APIs to retrieve resource feeds, connect to desktop or remote application sessions, and use many of the redirections that our first-party clients support. The SDK is compatible with most operating systems based on Ubuntu 18.04 or later.

Feature support

The SDK supports multiple connections to desktop and remote application sessions. The following redirections are supported:

REDIRECTION	SUPPORTED
Keyboard	✓
Mouse	✓
Audio in	✓
Audio out	✓
Clipboard (text)	✓
Clipboard (image)	✓
Clipboard (file)	✓
Smartcard	✓
Drive/folder	✓

The SDK also supports multiple monitor display configurations, as long as the monitors you select for your session are contiguous.

We'll update this document as we add support for new features and redirections. If you want to suggest new features and other improvements, visit our [UserVoice page](#).

Next steps

Check out our documentation for the following clients:

- [Windows Desktop client](#)
- [Web client](#)
- [Android client](#)
- [macOS client](#)
- [iOS client](#)