Contents

- Windows Virtual Desktop
- Overview
  - What is Windows Virtual Desktop?
- Tutorials
  - 1. Create a tenant in Windows Virtual Desktop
  - 2. Create service principals and role assignments by using PowerShell
  - 3. Create a host pool by using the Azure Marketplace
  - 4. Manage app groups
  - 5. Create a host pool to validate service updates
  - 6. 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
  - Create a host pool and session hosts
    - Azure Resource Manager template
    - PowerShell
    - Deploy a Windows 7 virtual machine
    - Deploy a GPU-based session host
    - Expand an existing host pool
  - Create a profile container
    - Use a VM-based file share
    - Use Azure NetApp Files
  - 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
- Deploy the management tool
  - Azure Resource Manager template
  - PowerShell
- Use service diagnostics
  - Deploy the diagnostics tool
  - Use diagnostics with Log Analytics
- Publish built-in apps
- Set up MSIX app attach

Concepts

- 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 10 Enterprise multi-session FAQ
- Data locations

Troubleshoot

- Troubleshooting overview, feedback, and support
- Identify and diagnose issues
- Tenant and host pool creation
- Session host virtual machine configuration
- Troubleshoot Windows 7 virtual machines in Windows Virtual Desktop
- Troubleshoot the Windows Virtual Desktop management tool

- Remote Desktop service connections
- Remote Desktop client issues
- Windows Virtual Desktop PowerShell
- Diagnosing graphics performance issues

Reference

- Linux support
- PowerShell
- REST API
- Supported RDP file settings
- Network guidance
- Virtual machine sizing guidance

Resources

- Experience estimator
- Pricing calculator
- Learning path
- Tech Community support group
- UserVoice forum
- Microsoft 365 roadmap
- Azure Resource Manager templates

What is Windows Virtual Desktop?

2/19/2020 • 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 Office 365 ProPlus 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/nqfti3jltau

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 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 plan to add support for the following OSes, 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, F1, Business Windows E3, E5, A3, A5
Windows 7 Enterprise	Microsoft 365 E3, E5, A3, A5, F1, Business 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 with one of the following:
  - Azure AD Connect (for hybrid organizations)
  - Azure AD Domain Services (for hybrid or cloud organizations)
- An Azure subscription that contains a virtual network that either contains or is connected to the Windows Server Active Directory

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.

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

ADDRESS	OUTBOUND PORT	PURPOSE
*.wvd.microsoft.com	TCP port 443	Service traffic
*.blob.core.windows.net	TCP port 443	Agent, SXS stack updates, and Agent traffic

ADDRESS	OUTBOUND PORT	PURPOSE
*.core.windows.net	TCP port 443	Agent traffic
*.servicebus.windows.net	TCP port 443	Agent traffic
prod.warmpath.msftcloudes.com	TCP port 443	Agent traffic
catalogartifact.azureedge.net	TCP port 443	Azure Marketplace
kms.core.windows.net	TCP port 1688	Windows 10 activation

IMPORTANT

Opening these URLs is essential for a reliable Windows Virtual Desktop deployment. Blocking access to these URLs is unsupported and will affect service functionality. These URLs only correspond to Windows Virtual Desktop sites and resources, and don't include URLs for other services like Azure Active Directory.

NOTE

Windows Virtual Desktop currently doesn't have a list of IP address ranges that you can whitelist to allow network traffic. We only support whitelisting specific URLs at this time.

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** and navigate to **Windows logs** > **Application** > **WVD-Agent** and look for Event ID 3702.
3. Whitelist the URLs that you find under Event ID 3702. The URLs under Event ID 3702 are region-specific. You'll need to repeat the whitelisting process with the relevant URLs for each region you want to deploy your virtual machines in.

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.
- 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
- Web
- Mac
- iOS

- Android (Preview)

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 VIRTUAL DESKTOP AGENT UPDATES
Windows 10 multi-session, version 1903	Yes	Yes	Yes	Yes	Automatic
Windows 10 multi-session, version 1809	Yes	Yes	No	No	Automatic
Windows 10 Enterprise, version 1903	Yes	Yes	Yes	Yes	Automatic
Windows 10 Enterprise, version 1809	Yes	Yes	No	No	Automatic
Windows 7 Enterprise	Yes	Yes	No	No	Manual
Windows Server 2019	Yes	Yes	No	No	Automatic
Windows Server 2016	Yes	Yes	Yes	Yes	Automatic
Windows Server 2012 R2	Yes	Yes	No	No	Automatic

Next steps

To get started, you'll need to create a tenant. To learn more about how to create a tenant, continue to the tenant

creation tutorial.

Create a tenant in Windows Virtual Desktop

Tutorial: Create a tenant in Windows Virtual Desktop

2/14/2020 • 6 minutes to read • Edit Online

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
> ```

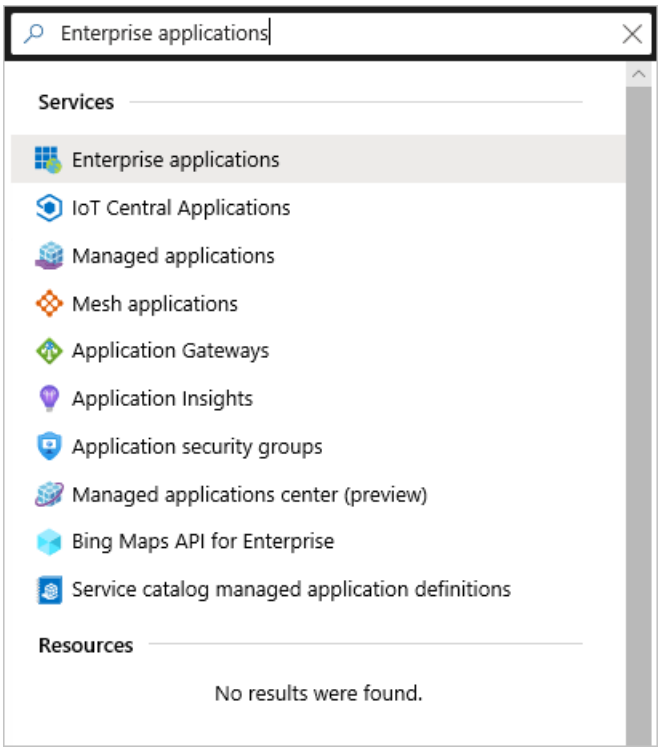
6. Sign in to the Windows Virtual Desktop consent page as global administrator, as you did in step 2.
7. Select **Accept**.

Assign the TenantCreator application role

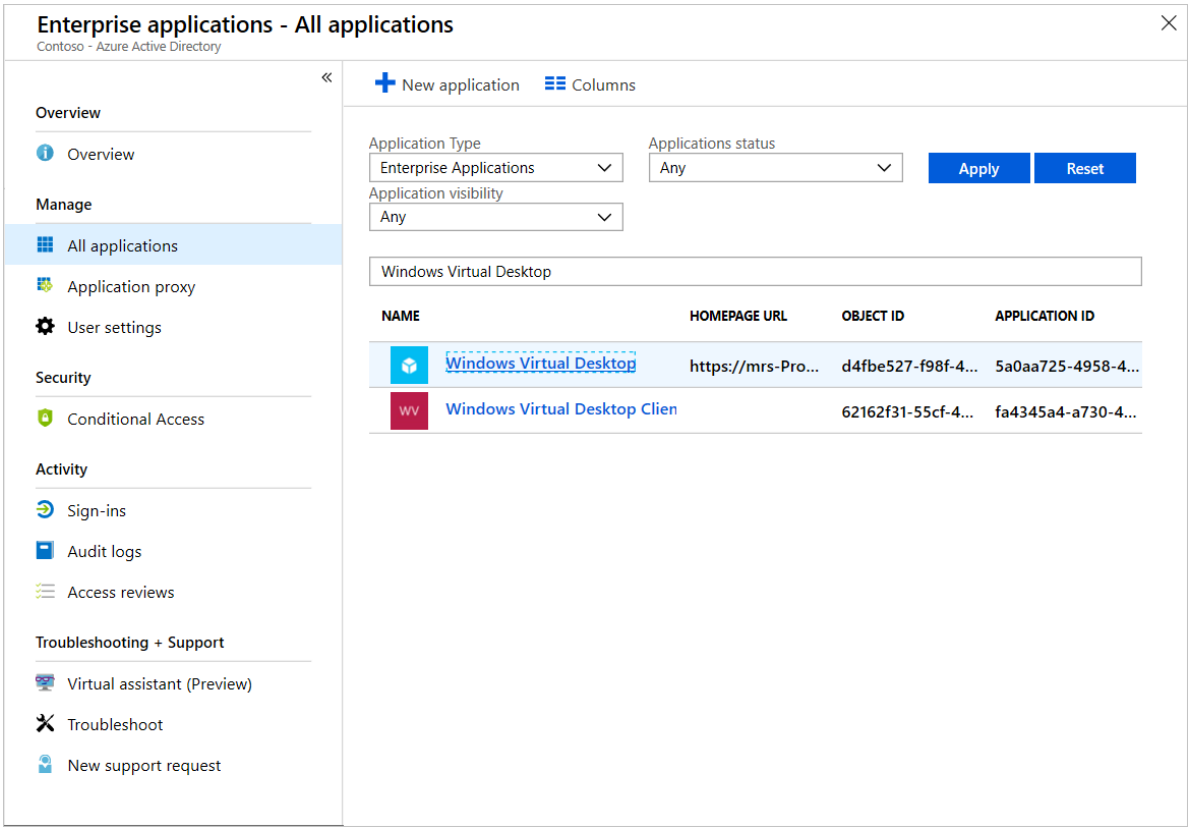
Assigning an Azure Active Directory user the TenantCreator application role allows that user to create a Windows Virtual Desktop tenant associated with the Azure Active Directory instance. You'll need to use your global administrator account to assign the TenantCreator role.

To assign the TenantCreator application role:

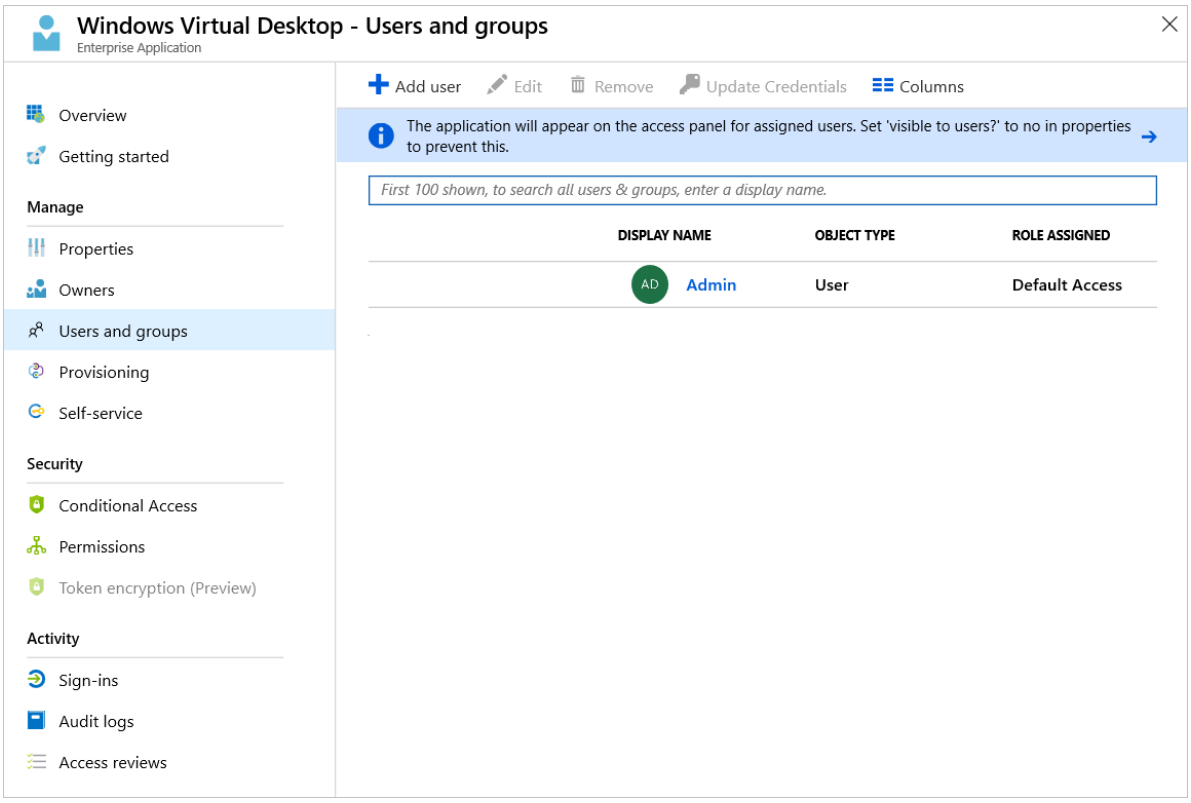
1. Go to theAzure portalto manage the TenantCreator application role. Search for and select **Enterprise applications**. If you're working with multiple Azure Active Directory tenants, it's a best practice to open a private browser session and copy and paste the URLs into the address bar.



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**.



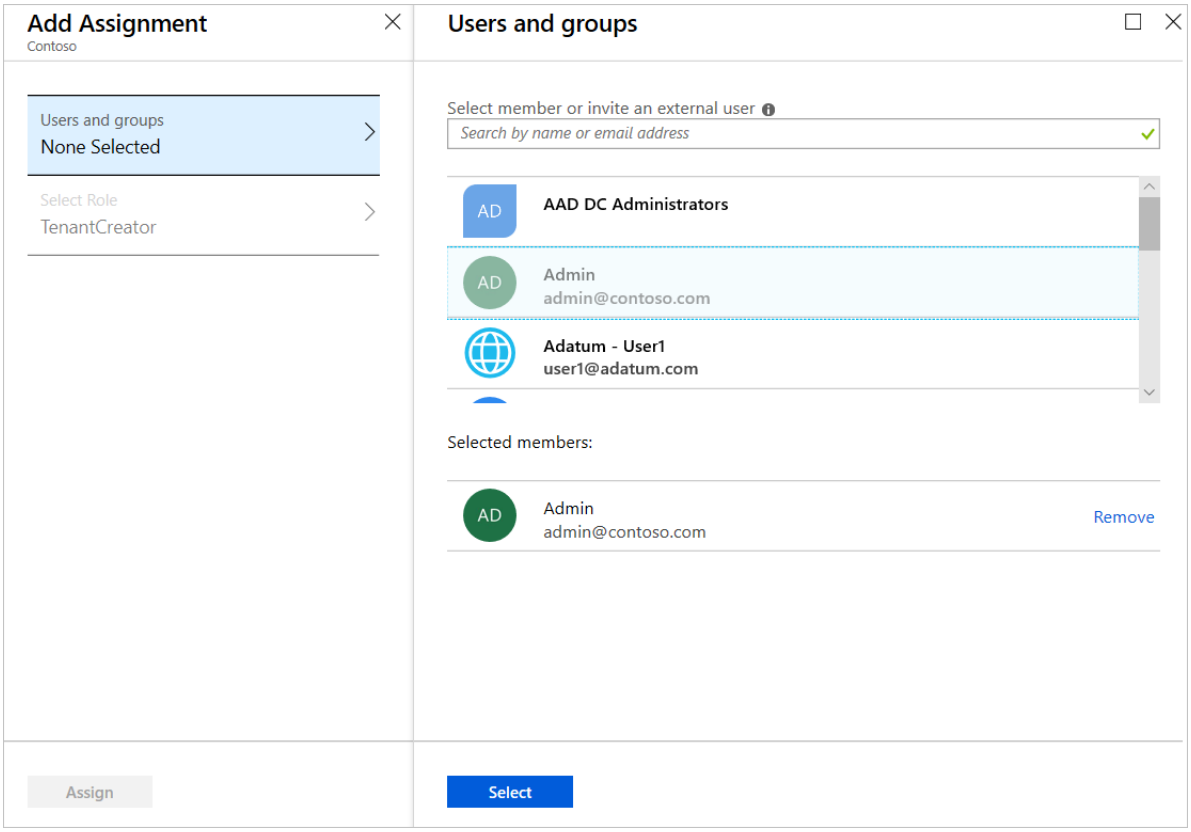
3. Select **Users and groups**. You might see that the administrator who granted consent to the application is already listed with the **Default Access** role assigned. This is not enough to create a Windows Virtual Desktop tenant. Continue following these instructions to add the **TenantCreator** role to a user.



4. Select **Add user**, and then select **Users and groups** in the **Add Assignment** blade.

5. Search for a user account that will create your Windows Virtual Desktop tenant. For simplicity, this can be the global administrator account.

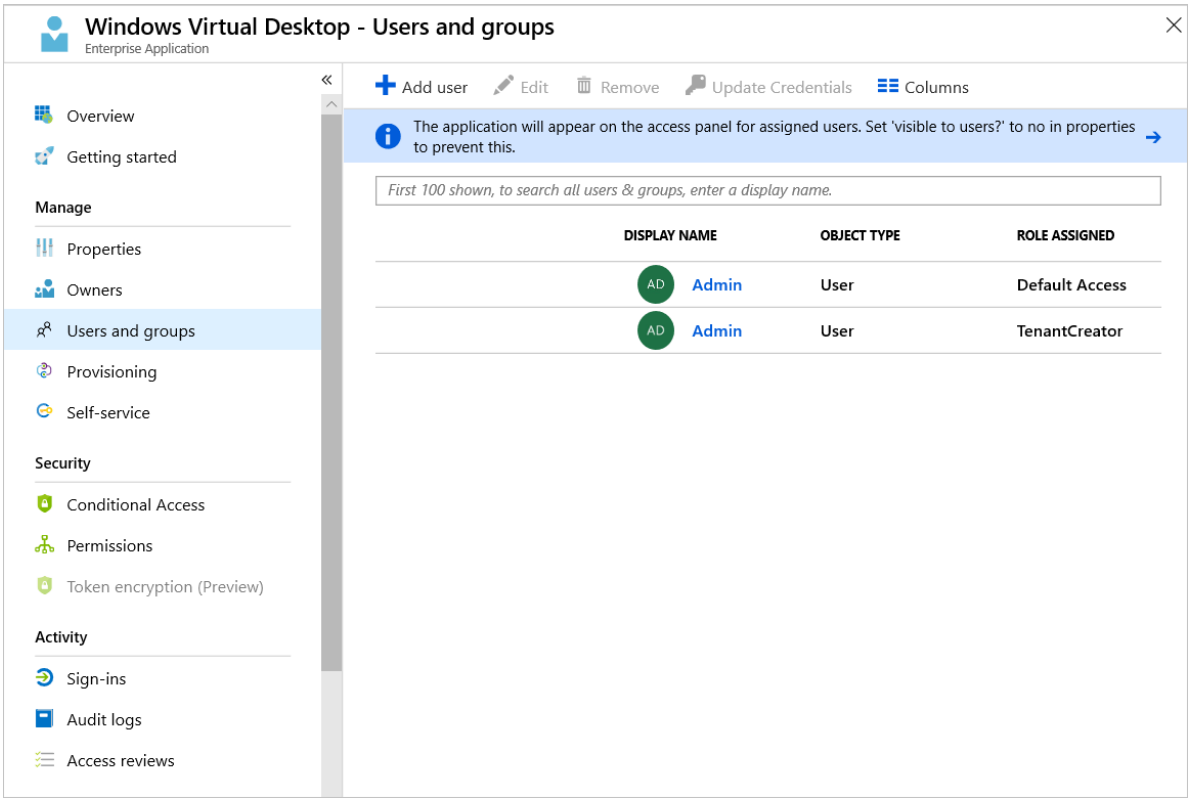
    - If you're using a Microsoft Identity Provider like contosoadmin@live.com or contosoadmin@outlook.com, you might not be able to sign in to Windows Virtual Desktop. We recommend using a domain-specific account like admin@contoso.com or admin@contoso.onmicrosoft.com instead.



> **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.

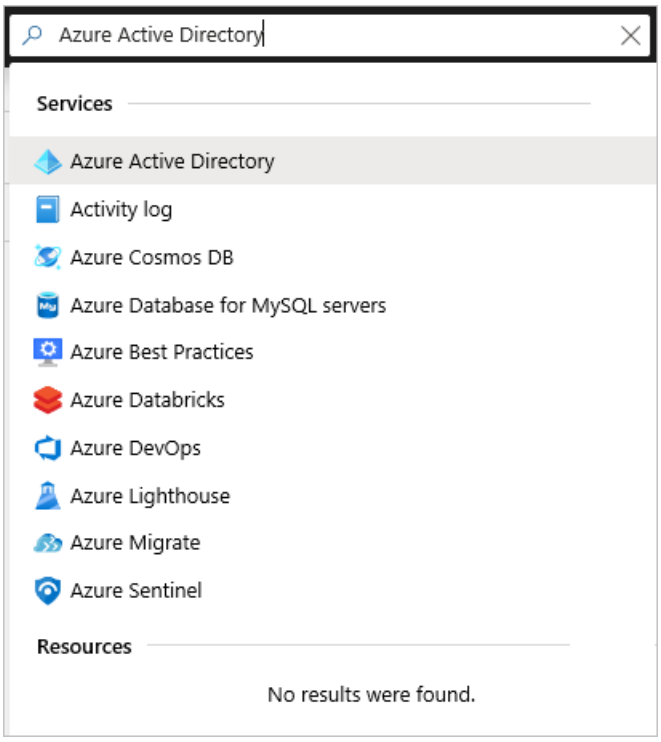


Before you continue on to create your Windows Virtual Desktop tenant, you need two pieces of information:

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

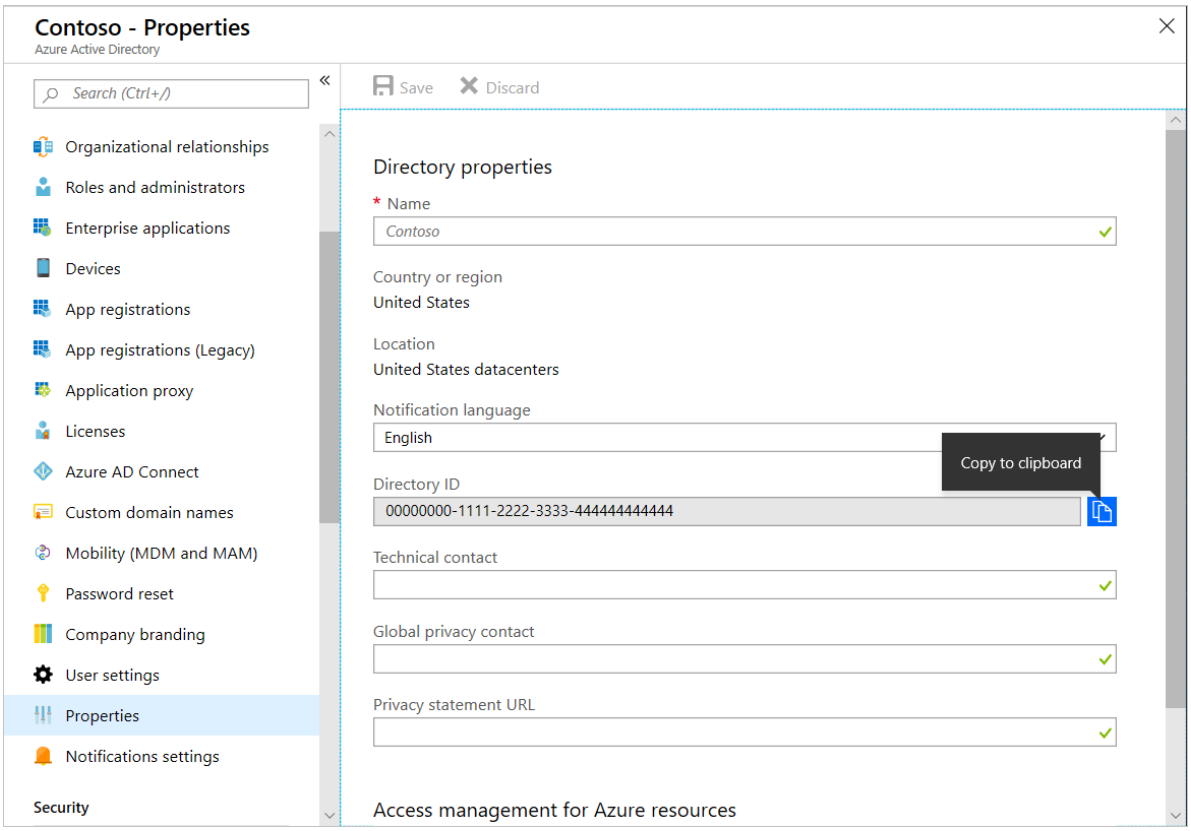
To find your Azure Active Directory tenant ID (or **Directory ID**):

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



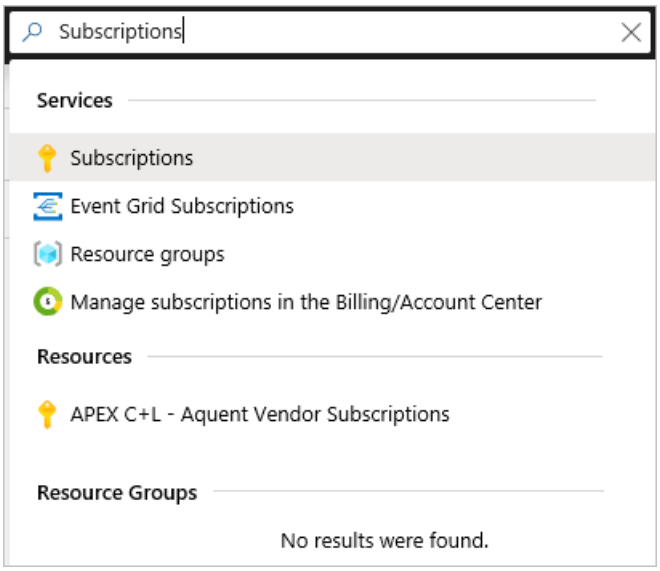
2. Scroll down until you find **Properties**, and then select it.

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.



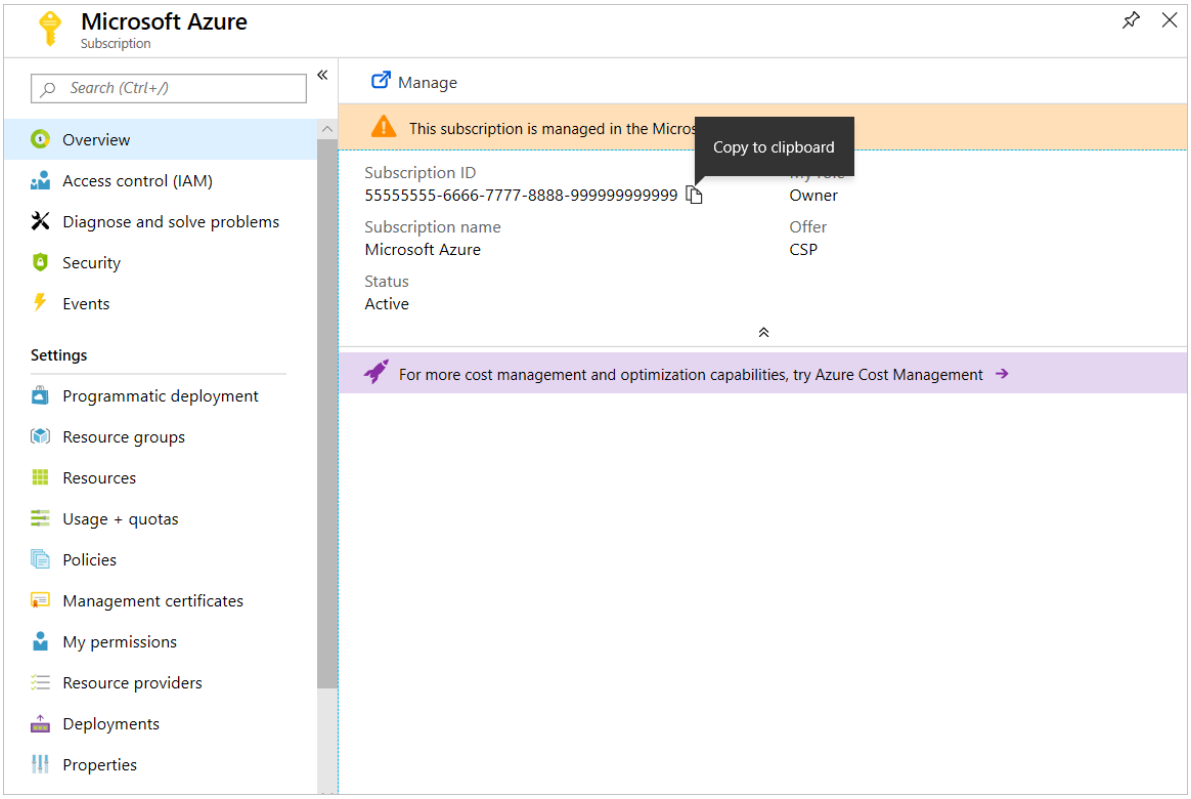
To find your Azure subscription ID:

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 service principals and role assignments by using PowerShell

2/14/2020 • 2 minutes to read • Edit Online

Service principals are identities that you can create in Azure Active Directory to assign roles and permissions for a specific purpose. In Windows Virtual Desktop, you can create a service principal to:

- Automate specific Windows Virtual Desktop management tasks.
- Use as credentials in place of MFA-required users when running any Azure Resource Manager template for Windows Virtual Desktop.

In this tutorial, learn how to:

- Create a service principal in Azure Active Directory.
- Create a role assignment in Windows Virtual Desktop.
- Sign in to Windows Virtual Desktop by using the service principal.

Prerequisites

Before you can create service principals and role assignments, you need to do three things:

1. Install the AzureAD module. To install the module, run PowerShell as an administrator and run the following cmdlet:

   ```
   Install-Module AzureAD
   ```

2. Download and import the Windows Virtual Desktop PowerShell module.

3. Follow all instructions in this article in the same PowerShell session. The process might not work if you interrupt your PowerShell session by closing the window and reopening it later.

Create a service principal in Azure Active Directory

After you've fulfilled the prerequisites in your PowerShell session, run the following PowerShell cmdlets to create a multitenant service principal in Azure.

```
Import-Module AzureAD
$aadContext = Connect-AzureAD
$svcPrincipal = New-AzureADApplication -AvailableToOtherTenants $true -DisplayName "Windows Virtual Desktop
Svc Principal"
$svcPrincipalCreds = New-AzureADApplicationPasswordCredential -ObjectId $svcPrincipal.ObjectId
```

View your credentials in PowerShell

Before you create the role assignment for your service principal, view your credentials and write them down for future reference. The password is especially important because you won't be able to retrieve it after you close this PowerShell session.

Here are the three credentials you should write down and the cmdlets you need to run to get them:

- Password:

```
$svcPrincipalCreds.Value
```

- Tenant ID:

```
$aadContext.TenantId.Guid
```

- Application ID:

```
$svcPrincipal.AppId
```

Create a role assignment in Windows Virtual Desktop Preview

Next, you need to create a role assignment so the service principal can sign in to Windows Virtual Desktop. Make sure to sign in with an account that has permissions to create role assignments.

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

Run the following PowerShell cmdlets to connect to Windows Virtual Desktop and display your tenants.

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

When you find the tenant name for the tenant you want to create a role assignment for, use that name in the following cmdlet:

```
$myTenantName = "<Windows Virtual Desktop Tenant Name>"
New-RdsRoleAssignment -RoleDefinitionName "RDS Owner" -ApplicationId $svcPrincipal.AppId -TenantName
$myTenantName
```

Sign in with the service principal

After you create a role assignment for the service principal, make sure the service principal can sign in to Windows Virtual Desktop by running the following cmdlet:

```
$creds = New-Object System.Management.Automation.PSCredential($svcPrincipal.AppId, (ConvertTo-SecureString
$svcPrincipalCreds.Value -AsPlainText -Force))
Add-RdsAccount -DeploymentUrl "https://rdbroker.wvd.microsoft.com" -Credential $creds -ServicePrincipal -
AadTenantId $aadContext.TenantId.Guid
```

After you've signed in, make sure everything works by testing a few Windows Virtual Desktop PowerShell cmdlets with the service principal.

Next steps

After you've created the service principal and assigned it a role in your Windows Virtual Desktop tenant, you can use it to create a host pool. To learn more about host pools, continue to the tutorial for creating a host pool in Windows Virtual Desktop.

Windows Virtual Desktop host pool tutorial

Tutorial: Create a host pool by using the Azure Marketplace

2/14/2020 • 5 minutes to read • Edit Online

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

This tutorial describes how to create a host pool within a Windows Virtual Desktop tenant by using a Microsoft Azure Marketplace offering. The tasks include:

- Create a host pool in Windows Virtual Desktop.
- Create a resource group with VMs in an Azure subscription.
- Join the VMs to the Active Directory domain.
- Register the VMs with Windows Virtual Desktop.

Before you begin, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

Sign in to Azure

Sign in to the Azure portal.

Run the Azure Marketplace offering to provision a new host pool

To run the Azure Marketplace offering to provision a new host pool:

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Enter **Windows Virtual Desktop** in the Marketplace search window.
3. Select **Windows Virtual Desktop - Provision a host pool**, and then select **Create**.

After that, follow the instructions in the next section to enter the information for the appropriate blades.

Basics

Here's what you do for the **Basics** blade:

1. Enter a name for the host pool that's unique within the Windows Virtual Desktop tenant.
2. Select the appropriate option for a personal desktop. If you select **Yes**, each user that connects to this host pool will be permanently assigned to a virtual machine.
3. Enter a comma-separated list of users who can sign in to the Windows Virtual Desktop clients and access a desktop after the Azure Marketplace offering finishes. For example, if you want to assign user1@contoso.com and user2@contoso.com access, enter "user1@contoso.com,user2@contoso.com."
4. Select **Create new** and provide a name for the new resource group.
5. For **Location**, select the same location as the virtual network that has connectivity to the Active Directory server.
6. Select **Next : Configure virtual machines >**.

> **IMPORTANT**
> If you're using a pure Azure Active Directory Domain Services and Azure Active Directory solution, make sure to deploy your host pool in the same region as your Azure Active Directory Domain Services to avoid domain-join and credential errors.

Configure virtual machines

For the **Configure virtual machines** blade:

1. Either accept the defaults or customize the number and size of the VMs.

   > **NOTE**
   > If the specific VM size you're looking for doesn't appear in the VM size selector, that's because we haven't onboarded it to the Azure Marketplace tool yet. To request a VM size, create a request or upvote an existing request in the Windows Virtual Desktop UserVoice forum.

2. Enter a prefix for the names of the virtual machines. For example, if you enter the name "prefix," the virtual machines will be called "prefix-0," "prefix-1," and so on.
3. Select **Next : Virtual machine settings**.

Virtual machine settings

For the **Virtual machine settings** blade:

> **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 a member of the AAD DC Administrators group.
>
> The account must also be part of the Azure AD DS managed domain or Azure AD tenant - accounts from external directories associated with your Azure AD tenant can't correctly authenticate during the domain-join process.

1. For **Image source**, select the source and enter the appropriate information for how to find it and how to store it. If you choose not to use managed disks, select the storage account that contains the .vhd file.
2. Enter the user principal name and password for the domain account that will join the VMs to the Active Directory domain. This same username and password will be created on the virtual machines as a local account. You can reset these local accounts later.
3. Select the virtual network that has connectivity to the Active Directory server, and then choose a subnet to host the virtual machines.
4. Select **Next: Windows Virtual Desktop information**.

Windows Virtual Desktop tenant information

For the **Windows Virtual Desktop tenant information** blade:

1. For **Windows Virtual Desktop tenant group name**, enter the name for the tenant group that contains your tenant. Leave it as the default unless you were provided a specific tenant group name.
2. For **Windows Virtual Desktop tenant name**, enter the name of the tenant where you'll be creating this host pool.
3. Specify the type of credentials that you want to use to authenticate as the Windows Virtual Desktop tenant RDS Owner. If you completed the Create service principals and role assignments with PowerShell tutorial, select **Service principal**. When **Azure AD tenant ID** appears, enter the ID for the Azure Active Directory instance that contains the service principal.

4. Enter the credentials for the tenant admin account. Only service principals with a password credential are supported.
5. Select **Next : Review + create**.

Complete setup and create the virtual machine

For the last two blades:

1. On the **Review and Create** blade, review the setup information. If you need to change something, go back to the appropriate blade and make your change before continuing. If the information looks right, select **OK**.
2. Select **Create** to deploy your host pool.

Depending on how many VMs you're creating, this process can take 30 minutes or more to complete.

(Optional) Assign additional users to the desktop application group

After the Azure Marketplace offering finishes, you can assign more users to the desktop application group before you start testing the full session desktops on your virtual machines. If you've already added default users in the Azure Marketplace offering and don't want to add more, you can skip this section.

To assign users to the desktop application group, you must first open a PowerShell window. After that, you'll need to enter the following two cmdlets.

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

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

Add users to the desktop application group by using this cmdlet:

```
Add-RdsAppGroupUser <tenantname> <hostpoolname> "Desktop Application Group" -UserPrincipalName <userupn>
```

The user's UPN should match the user's identity in Azure Active Directory (for example, user1@contoso.com). If you want to add multiple users, you must run this cmdlet for each user.

After you've completed these steps, users added to the desktop application group can sign in to Windows Virtual Desktop with supported Remote Desktop clients and see a resource for a session desktop.

Here are the current supported clients:

- Remote Desktop client for Windows 7 and Windows 10
- Windows Virtual Desktop web client

> **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.

Next steps

Now that you've made a host pool and assigned users to access its desktop, you can populate your host pool with RemoteApp programs. To learn more about how to manage apps in Windows Virtual Desktop, see this tutorial:

Manage app groups tutorial

Tutorial: Manage app groups for Windows Virtual Desktop

2/14/2020 • 2 minutes to read • Edit Online

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.

Before you begin, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

Create a RemoteApp group

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

   ```
   New-RdsAppGroup <tenantname> <hostpoolname> <appgroupname> -ResourceType "RemoteApp"
   ```

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-RdsAppGroup <tenantname> <hostpoolname>
   ```

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-RdsStartMenuApp <tenantname> <hostpoolname> <appgroupname>
   ```

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

   ```
   New-RdsRemoteApp <tenantname> <hostpoolname> <appgroupname> -Name <remoteappname> -AppAlias <appalias>
   ```

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

   ```
   New-RdsRemoteApp <tenantname> <hostpoolname> <appgroupname> -Name <remoteappname> -Filepath <filepath>
   -IconPath <iconpath> -IconIndex <iconindex>
   ```

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

```
Get-RdsRemoteApp <tenantname> <hostpoolname> <appgroupname>
```

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.

```
Add-RdsAppGroupUser <tenantname> <hostpoolname> <appgroupname> -UserPrincipalName <userupn>
```

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

2/14/2020 • 2 minutes to read • Edit Online

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.

Before you begin, download and import the Windows Virtual Desktop PowerShell module, if you haven't already. After that, run the following cmdlet to sign in to your account:

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

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 an Azure Resource Manager template
- 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 quotes by the values relevant to your session:

```
Add-RdsAccount -DeploymentUrl "https://rdbroker.wvd.microsoft.com"
Set-RdsHostPool -TenantName $myTenantName -Name "contosoHostPool" -ValidationEnv $true
```

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

```
Get-RdsHostPool -TenantName $myTenantName -Name "contosoHostPool"
```

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

```
TenantName          : contoso
TenantGroupName     : Default Tenant Group
HostPoolName        : contosoHostPool
FriendlyName        :
Description         :
Persistent          : False
CustomRdpProperty : use multimon:i:0;
MaxSessionLimit     : 10
LoadBalancerType : BreadthFirst
ValidationEnv       : True
Ring                :
```

Update schedule

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

Next steps

Now that you've created a validation host pool, you can learn how to deploy and connect to a management tool for managing Microsoft Virtual Desktop resources.

Deploy a management tool tutorial

Tutorial: Set up service alerts

2/14/2020 • 2 minutes to read • Edit Online

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.

Prerequisites

- Tutorial: Create a tenant in Windows Virtual Desktop
- Tutorial: Create service principals and role assignments with PowerShell
- Tutorial: Create a host pool with Azure Marketplace

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. Use 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

2/14/2020 • 2 minutes to read • Edit Online

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

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

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

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 feed

Get the list of managed resources available to you by subscribing to the feed provided by your admin. Subscribing makes the resources available on your local PC.

To subscribe to a feed:

1. Open the Windows Desktop client.
2. Select **Subscribe** on the main page to connect to the service and retrieve your resources.
3. Sign in with your user account when prompted.

After you successfully sign in, you should see a list of the resources you can access.

You can launch resources by one of two methods.

- From the client's main page, double-click a resource to launch it.
- Launch a resource as you normally would other apps from the Start Menu.
  - You can also search for the apps in the search bar.

Once subscribed to a feed, the content of the feed is updated 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 Windows Desktop client, check out Get started with the Windows Desktop

client.

Connect with the web client

11/4/2019 • 2 minutes to read • Edit Online

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	
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 Windows Virtual Desktop web client and sign in with your user account.

> **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.

Connect with the Android client

2/14/2020 • 2 minutes to read • Edit Online

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

NOTE

The ability to access Windows Virtual Desktop resources from the Android client is currently available in preview.

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.
   - 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 with the macOS client

2/14/2020 • 2 minutes to read • Edit Online

Applies to: macOS 10.12 or later

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 Feed** on the main page to connect to the service and retrieve your resources.
2. Enter the Feed URL. This can be a URL or email address:
   - If you use a URL, use the one your admin gave you. Normally, the URL is https://rdweb.wvd.microsoft.com.
   - 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 **Subscribe**.
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 with the iOS client

2/14/2020 • 2 minutes to read • Edit Online

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

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.
   - 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.

Create a host pool with an Azure Resource Manager template

2/14/2020 • 3 minutes to read • Edit Online

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

Follow this section's instructions to create a host pool for a Windows Virtual Desktop tenant with an Azure Resource Manager template provided by Microsoft. This article will tell you how to create a host pool in Windows Virtual Desktop, create a resource group with VMs in an Azure subscription, join those VMs to the AD domain, and register the VMs with Windows Virtual Desktop.

What you need to run the Azure Resource Manager template

Make sure you know the following things before running the Azure Resource Manager template:

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

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.

Run the Azure Resource Manager template for provisioning a new host pool

To start, go to this GitHub URL.

Deploy the template to Azure

If you're deploying in an Enterprise subscription, scroll down and select **Deploy to Azure**, then skip ahead fill out the parameters based on your image source.

If you're deploying in a Cloud Solution Provider subscription, follow these steps to deploy to Azure:

1. Scroll down and right-click **Deploy to Azure**, then select **Copy Link Location**.
2. Open a text editor like Notepad and paste the link there.
3. Right after "https://portal.azure.com/" and before the hashtag (#) enter an at sign (@) followed by the tenant domain name. Here's an example of the format you should use: https://portal.azure.com/@Contoso.onmicrosoft.com#create/.
4. Sign in to the Azure portal as a user with Admin/Contributor permissions to the Cloud Solution Provider subscription.
5. Paste the link you copied to the text editor into the address bar.

For guidance about which parameters you should enter for your scenario, see the Windows Virtual Desktop Readme file. The Readme is always updated with the latest changes.

Assign users to the desktop application group

After the GitHub Azure Resource Manager template completes, assign user access before you start testing the full session desktops on your virtual machines.

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

To assign users to the desktop application group, open a PowerShell window and run this cmdlet to sign in to the Windows Virtual Desktop environment:

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

After that, add users to the desktop application group with this cmdlet:

```
Add-RdsAppGroupUser <tenantname> <hostpoolname> "Desktop Application Group" -UserPrincipalName <userupn>
```

The user's UPN should match the user's identity in Azure Active Directory (for example, user1@contoso.com). If you want to add multiple users, you must run this cmdlet for each user.

After you've completed these steps, users added to the desktop application group can sign in to Windows Virtual Desktop with supported Remote Desktop clients and see a resource for a session desktop.

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 host pool with PowerShell

2/14/2020 • 4 minutes to read • Edit Online

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

Use your PowerShell client to create a host pool

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

Run the following cmdlet to sign in to the Windows Virtual Desktop environment

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

Next, run this cmdlet to create a new host pool in your Windows Virtual Desktop tenant:

```
New-RdsHostPool -TenantName <tenantname> -Name <hostpoolname>
```

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.

```
New-RdsRegistrationInfo -TenantName <tenantname> -HostPoolName <hostpoolname> -ExpirationHours <number of hours> | Select-Object -ExpandProperty Token | Out-File -FilePath <PathToRegFile>
```

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

```
Add-RdsAppGroupUser -TenantName <tenantname> -HostPoolName <hostpoolname> -AppGroupName "Desktop Application Group" -UserPrincipalName <userupn>
```

The **Add-RdsAppGroupUser** cmdlet doesn't support adding security groups and only adds one user at a time to the app group. If you want to add multiple users to the app group, rerun the cmdlet with the appropriate user principal names.

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 = (Export-RdsRegistrationInfo -TenantName <tenantname> -HostPoolName <hostpoolname>).Token
```

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.
   - Right-click the downloaded installer, select **Properties**, select **Unblock**, then select **OK**. This will allow your system to trust the installer.
   - Run the installer. When the installer asks you for the registration token, enter the value you got from

the **Export-RdsRegistrationInfo** cmdlet.

3. Download and install the Windows Virtual Desktop Agent Bootloader.
   - Download the Windows Virtual Desktop Agent Bootloader.
   - Right-click the downloaded installer, select **Properties**, select **Unblock**, then select **OK**. This will allow your system to trust the installer.
   - 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.

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

2/14/2020 • 2 minutes to read • Edit Online

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. After that, follow the instructions in Create host pools in Azure Marketplace to assign one or more users to the desktop application group.

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 do this, 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 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.

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

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

1/24/2020 • 4 minutes to read • Edit Online

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 Azure Marketplace.

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. Additionaly, 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 the hardware default graphics adapter 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. 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:

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. 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**.

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

   ```
   gpupdate.exe /force
   ```

4. 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 an 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-RdpCoreTS** > **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.
4. To determine if AVC 444 mode 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 a single session host 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 (coming soon) 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

2/23/2020 • 6 minutes to read • Edit Online

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 Marketplace offering
- GitHub Azure Resource Manager template
- 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 and Windows Virtual Desktop tenant administrator credentials
- Virtual network name and subnet name

The next three sections are three methods you can use to expand the host pool. You can do either with whichever deployment tool you're comfortable with.

> **NOTE**
> During the deployment phase, you'll see error messages for the previous session host VM resources if they're currently shut down. These errors happen because Azure can't run the PowerShell DSC extension to validate that the session host VMs are correctly registered to your existing host pool. You can safely ignore these errors, or you can avoid the errors by starting all session host VMs in the existing host pool before starting the deployment process.

Redeploy from Azure

If you've already created a host pool and session host VMs using the Azure Marketplace offering or GitHub Azure Resource Manager template, you can redeploy the same template from the Azure portal. Redeploying the template automatically reenters all the information you entered into the original template except for passwords.

Here's how to redeploy the Azure Resource Manager template to expand a host pool:

1. Sign in to the Azure portal.
2. From the search bar at the top of the Azure portal, search for **Resource groups** and select the item under **Services**.
3. Find and select the resource group you created when you made the host pool.
4. In the panel on the left side of the browser, select **Deployments**.
5. Select the appropriate deployment for your host pool creation process:
    - If you created the original host pool with the Azure Marketplace offering, select the deployment starting

with **rds.wvd-provision-host-pool**.

- If you created the original host pool with the GitHub Azure Resource Manager template, select the deployment named **Microsoft.Template**.

6. Select **Redeploy**.

> **NOTE**
> If the template doesn't automatically redeploy when you select **Redeploy**, select **Template** in the panel on the left side of your browser, then select **Deploy**.

7. Select the resource group that contains the current session host VMs in the existing host pool.

> **NOTE**
> If you see an error that tells you to select a different resource group even though the one you entered is correct, select another resource group, then select the original resource group.

8. Enter the following URL for the *_artifactsLocation*:
   `https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/`
9. Enter the new total number of session hosts you want into *Rdsh Number Of Instances*. For example, if you're expanding your host pool from five session hosts to eight, enter **8**.
10. Enter the same existing domain password that you used for the existing domain UPN. Don't change the username, because that will cause an error when you run the template.
11. Enter the same tenant admin password you used for the user or application ID you entered for *Tenant Admin Upn Or Application Id*. Once again, don't change the username.
12. Complete the submission to expand your host pool.

Run the Azure Marketplace offering

Follow the instructions in Create a host pool by using the Azure Marketplace until you reach Run the Azure Marketplace offering to provision a new host pool. When you get to that point, you'll need to enter the following information for each blade:

Basics

All values in this section should match what you provided when you first created the host pool and session host VMs, except for *Default desktop users*:

1. For *Subscription*, select the subscription where you first created the host pool.
2. For *Resource group*, select the same resource group where the existing host pool session host VMs are located.
3. For *Region*, select the same region where the existing host pool session host VMs are located.
4. For *Hostpool name*, enter the name of the existing host pool.
5. For *Desktop type*, select the desktop type that matches the existing host pool.
6. For *Default desktop users*, enter a comma-separated list of any additional users who you want to sign in to the Windows Virtual Desktop clients and access a desktop after the Azure Marketplace offering finishes. For example, if you want to assign user3@contoso.com and user4@contoso.com access, enter user3@contoso.com,user4@contoso.com.
7. Select **Next : Configure virtual machine**.

> **NOTE**
> Except for *Default desktop users*, all fields must match exactly what has been configure in the existing host pool. If there is a mismatch that will result in a new host pool.

Configure virtual machines

All parameter values in this section should match what you provided when you first created the host pool and session host VMs, except for the total number of VMs. The number of VMs you enter will be the number of VMs in your expanded host pool:

1. Select the VM size that matches the existing session host VMs.

   > **NOTE**
   > If the specific VM size you're looking for doesn't appear in the VM size selector, that's because we haven't onboarded it to the Azure Marketplace tool yet. To request a VM size, create a request or upvote an existing request in the Windows Virtual Desktop UserVoice forum.

2. Customize the *Usage Profile*, *Total users*, and *Number of virtual machines* parameters to select the total number of session hosts you would like to have in your host pool. For example, if you're expanding your host pool from five session hosts to eight, configure these options to get to 8 virtual machines.
3. Enter a prefix for the names of the virtual machines. For example, if you enter the name "prefix," the virtual machines will be called "prefix-0," "prefix-1," and so on.
4. Select **Next : Virtual machine settings**.

Virtual machine settings

All parameter values in this section should match what you provided when you first created the host pool and session host VMs:

1. For *Image source* and *Image OS version*, enter the same information that you provided when you first created the host pool.
2. For *AD domain join UPN* and the associated passwords, enter the same information that you provided when you first created the host pool to join the VMs to the Active Directory domain. These credentials will be used to create a local account on your virtual machines. You can reset these local accounts to change their credentials later.
3. For the virtual network information, select the same virtual network and subnet for where your existing host pool session host VMs are located.
4. Select **Next : Configure Windows Virtual Desktop information**.

Windows Virtual Desktop information

All parameter values in this section should match what you provided when you first created the host pool and session host VMs:

1. For *Windows Virtual Desktop tenant group name*, enter the name for the tenant group that contains your tenant. Leave it as the default unless you were provided a specific tenant group name.
2. For *Windows Virtual Desktop tenant name*, enter the name of the tenant where you'll be creating this host pool.
3. Specify the same credentials you used when you first created the host pool and session host VMs. If you are using a service principal, enter the ID of the Azure Active Directory instance where your service principal is located.
4. Select **Next : Review + create**.

Run the GitHub Azure Resource Manager template

Follow the instructions in Run the Azure Resource Manager template for provisioning a new host pool and provide all of the same parameter values except for the *Rdsh Number Of Instances*. Enter the number of session host VMs you want in the host pool after running the template. For example, if you're expanding your host pool from five session hosts to eight, enter **8**.

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

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

2/14/2020 • 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. 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 an FSLogix profile container for a host pool using Azure NetApp Files

2/14/2020 • 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:

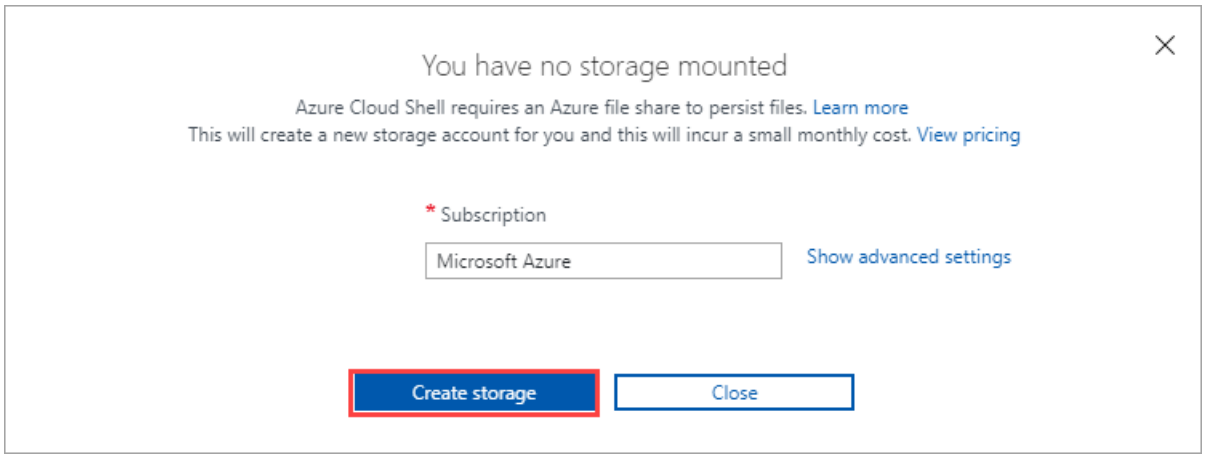
- Set up and configure Windows Virtual Desktop
- Provision a Windows Virtual Desktop host pool
- Enable your Azure NetApp Files subscription

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.

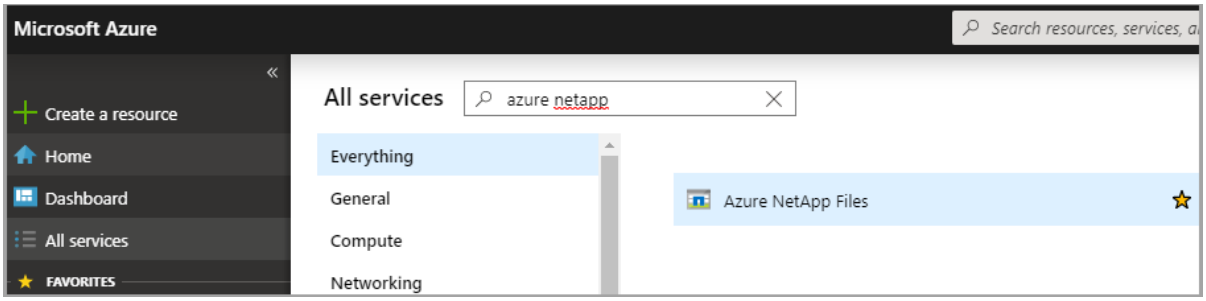


5. Once Azure Cloud Shell loads, run the following two cmdlets.

```
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** blade 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** blade 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.

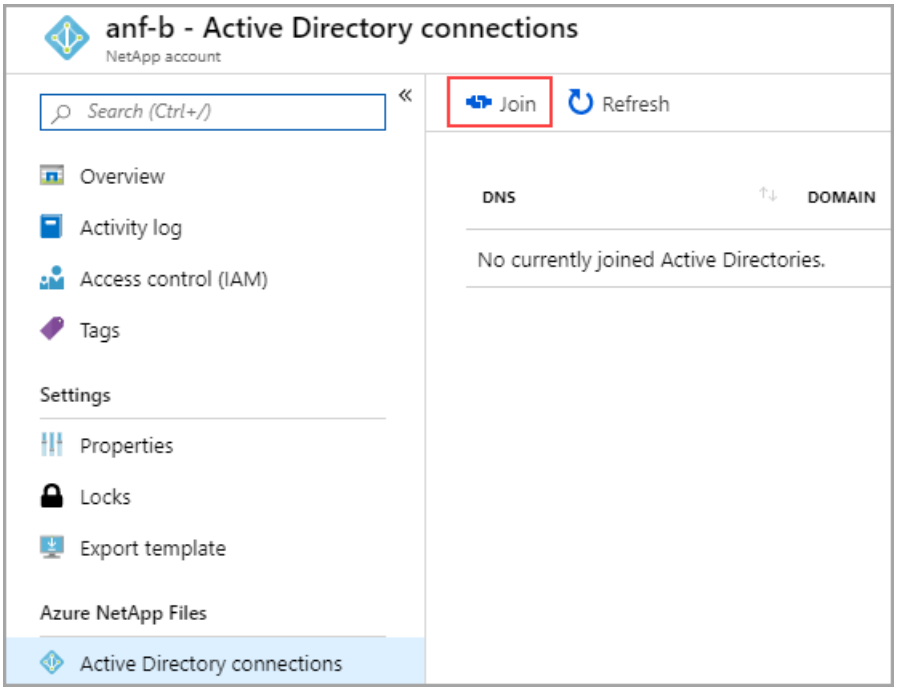
     > **NOTE**
     > The Premium setting provides the minimum throughput available for a Premium Service level, which is 256 MBps. You may need to adjust this throughput for a production environment. Final throughput is based on the relationship described in Throughput limits.

   - 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.

> **NOTE**
> It's best practice to confirm that the computer account you created in Join an Active Directory connection has appeared in your domain controller under **Computers** or **your enterprise's relevant OU**.

Create a new volume

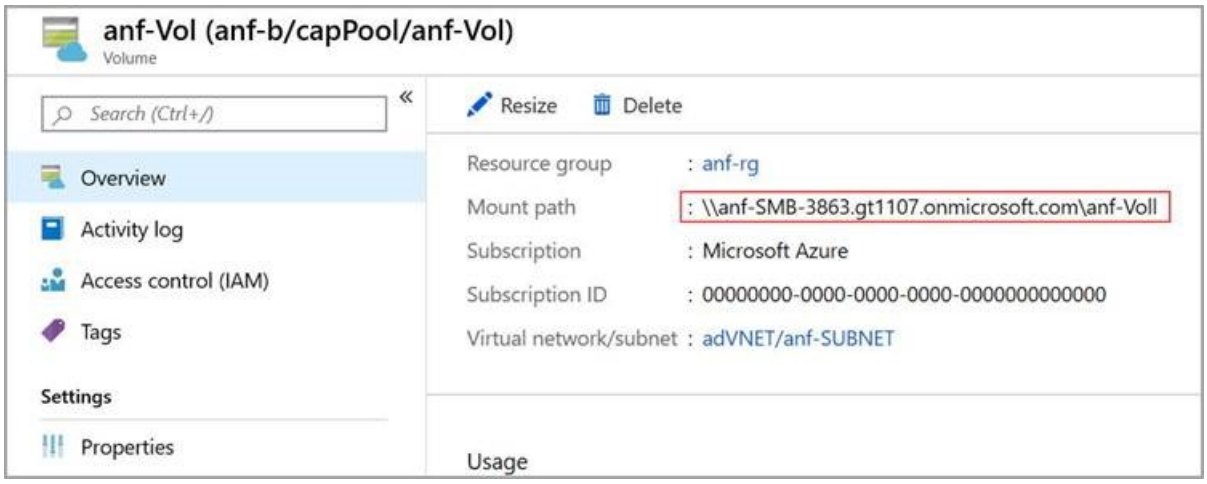
Next, you'll need to create a new volume.

1. Select **Volumes**, then select **Add volume**.
2. When the **Create a volume** blade opens, enter the following values:
   - For **Volume name**, enter a name for the new volume.
   - For **Capacity pool**, select the capacity pool you just created from the drop-down menu.
   - For **Quota (GiB)**, enter the volume size appropriate for your environment.
   - For **Virtual network**, select an existing virtual network that has connectivity to the domain controller from the drop-down menu.
   - Under **Subnet**, select **Create new**. Keep in mind that this subnet will be delegated to Azure NetApp Files.
3. Select **Next: Protocol >>** to open the Protocol tab and configure your volume access parameters.

Configure volume access parameters

After you create the volume, configure the volume access parameters.

1. Select **SMB** as the protocol type.
2. Under Configuration in the **Active Directory** drop-down menu, select the same directory that you originally connected in Join an Active Directory connection. Keep in mind that there's a limit of one Active Directory per subscription.
3. In the **Share name** text box, enter the name of the share used by the session host pool and its users.
4. Select **Review + create** at the bottom of the page. This opens the validation page. After your volume is validated successfully, select **Create**.
5. At this point, the new volume will start to deploy. Once deployment is complete, you can use the Azure NetApp Files share.
6. To see the mount path, select **Go to resource** and look for it in the Overview tab.



Configure FSLogix on session host virtual machines (VMs)

This section is based on Create a profile container for a host pool using a file share.

1. Download the FSLogix agent .zip file while you're still remoted in the session host VM.
2. Unzip the downloaded file.
3. In the file, go to **x64** > **Releases** and run **FSLogixAppsSetup.exe**. The installation menu will open.
4. If you have a product key, enter it in the Product Key text box.
5. Select the check box next to **I agree to the license terms and conditions**.
6. Select **Install**.
7. Navigate to **C:\Program Files\FSLogix\Apps** to confirm the agent installed.
8. From the Start menu, run **RegEdit** as administrator.
9. Navigate to **Computer\HKEY_LOCAL_MACHINE\software\FSLogix**.
10. Create a key named **Profiles**.
11. Create a value named **Enabled** with a **REG_DWORD** type set to a data value of **1**.
12. Create a value named **VHDLocations** with a **Multi-String** type and set its data value to the URI for the Azure NetApp Files share.
13. Create a value named **DeleteLocalProfileWhenVHDShouldApply** with a DWORD value of 1 to avoid problems with existing local profiles before you sign in.

> **WARNING**
> Be careful when creating the DeleteLocalProfileWhenVHDShouldApply value. When the FSLogix Profiles system determines a user should have an FSLogix profile, but a local profile already exists, Profile Container will permanently delete the local profile. The user will then be signed in with the new FSLogix profile.

Assign users to session host

1. Open **PowerShell ISE** as administrator and sign in to 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
```

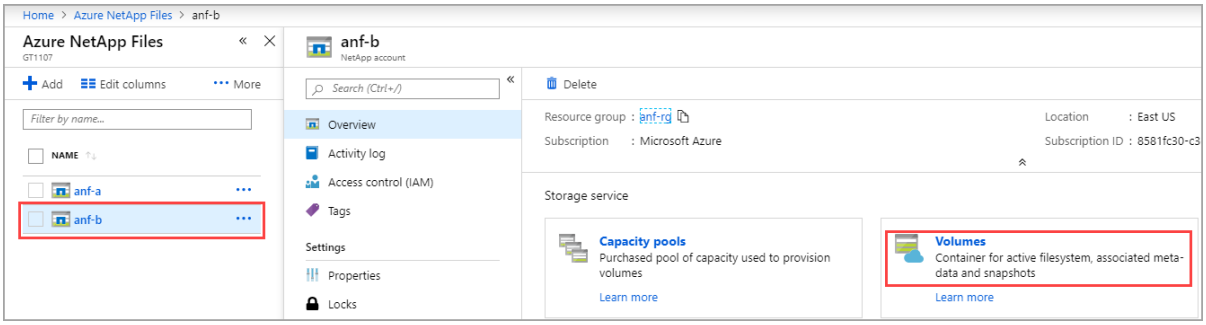
Make sure users can access the Azure NetApp File share

1. Open your internet browser and go to https://rdweb.wvd.microsoft.com/webclient/index.html.

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.

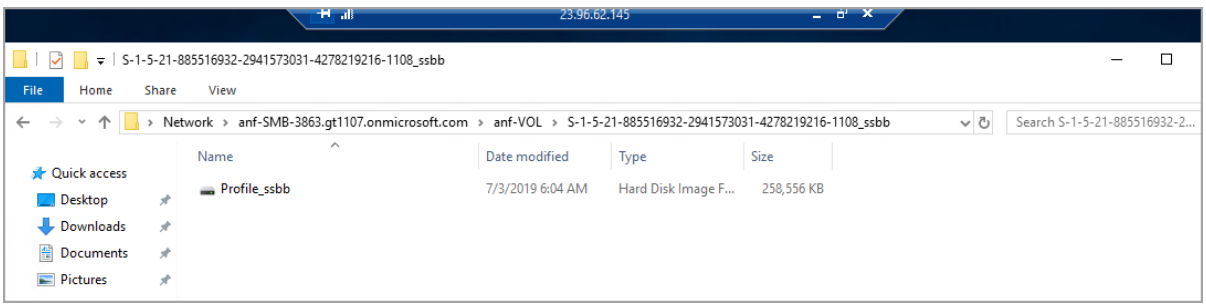
4. Open **Azure NetApp Files**, select your Azure NetApp Files account, and then select **Volumes**. Once the Volumes menu opens, select the corresponding volume.



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.

Customize Remote Desktop Protocol properties for a host pool

2/14/2020 • 2 minutes to read • Edit Online

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 using the **-CustomRdpProperty** parameter in the **Set-RdsHostPool** cmdlet.

See supported RDP file settings for a full list of supported properties and their default values.

First, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

Default RDP properties

By default, published RDP files contain the following properties:

RDP PROPERTIES	DESKTOPS	REMOTEAPPS
Multi-monitor mode	Enabled	N/A
Drive redirections enabled	Drives, clipboard, printers, COM ports, USB devices and smartcards	Drives, clipboard, and printers
Remote audio mode	Play locally	Play locally

Any custom properties you define for the host pool will override these defaults.

Add or edit a single custom RDP property

To add or edit a single custom RDP property, run the following PowerShell cmdlet:

```
Set-RdsHostPool -TenantName <tenantname> -Name <hostpoolname> -CustomRdpProperty "<property>"
```

```
PS C:\WINDOWS\system32> Set-RdsHostPool -TenantName          -Name hp0 -CustomRdpProperty
"audiocapturemode:i:1"


TenantName         :
TenantGroupName    : Default Tenant Group
HostPoolName       : HP0
FriendlyName       : HP0
Description        : Created through ARM template
Persistent         : False
CustomRdpProperty  : audiocapturemode:i:1;
MaxSessionLimit    : 999999
LoadBalancerType   : BreadthFirst
ValidationEnv      : False
Ring               :
```

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:

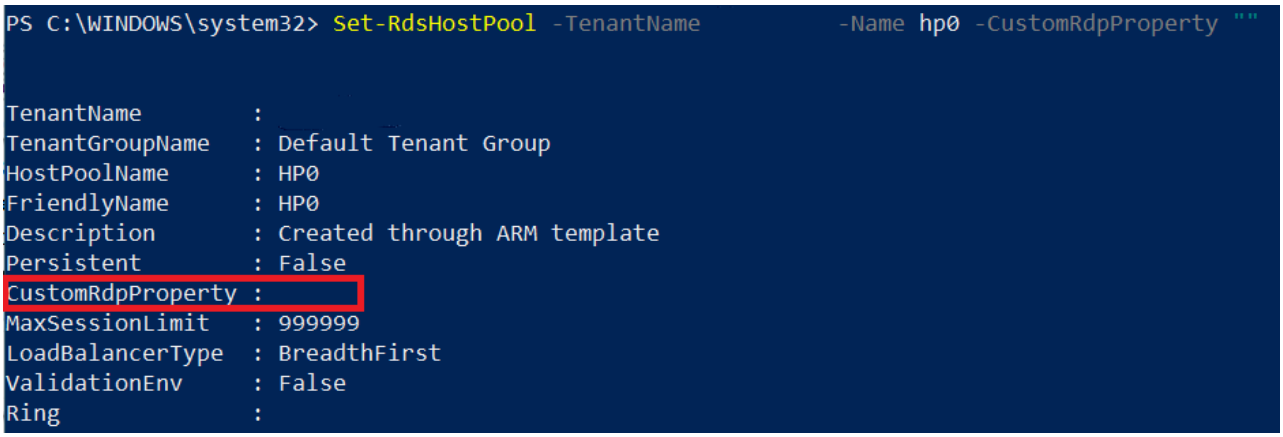
```
$properties="<property1>;<property2>;<property3>"
Set-RdsHostPool -TenantName <tenantname> -Name <hostpoolname> -CustomRdpProperty $properties
```

```
PS C:\WINDOWS\system32> $properties = "audiomode:i:0;audiocapturemode:i:1"
PS C:\WINDOWS\system32> Set-RdsHostPool -TenantName          -Name hp0 -CustomRdpProperty
$properties


TenantName         :
TenantGroupName    : Default Tenant Group
HostPoolName       : HP0
FriendlyName       : HP0
Description        : Created through ARM template
Persistent         : False
CustomRdpProperty  : audiomode:i:0;audiocapturemode:i:1;
MaxSessionLimit    : 999999
LoadBalancerType   : BreadthFirst
ValidationEnv      : False
Ring               :
```

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:

```
Set-RdsHostPool -TenantName <tenantname> -Name <hostpoolname> -CustomRdpProperty ""
```



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 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

Configure the Windows Virtual Desktop load-balancing method

2/14/2020 • 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.

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.

First, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

To configure a host pool to perform breadth-first load balancing without adjusting the maximum session limit, run the following PowerShell cmdlet:

```
Set-RdsHostPool <tenantname> <hostpoolname> -BreadthFirstLoadBalancer
```

To configure a host pool to perform breadth-first load balancing and to use a new maximum session limit, run the following PowerShell cmdlet:

```
Set-RdsHostPool <tenantname> <hostpoolname> -BreadthFirstLoadBalancer -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:

```
Set-RdsHostPool <tenantname> <hostpoolname> -DepthFirstLoadBalancer -MaxSessionLimit ###
```

Configure the personal desktop host pool assignment type

2/14/2020 • 2 minutes to read • Edit Online

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.

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.

Before you start, download and import the Windows Virtual Desktop PowerShell module if you haven't already.

> **NOTE**
> Make sure you've installed Windows Virtual Desktop PowerShell module version 1.0.1534.2001 or later before following these instructions.

After that, run the following cmdlet to sign in to your account:

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

To configure a host pool to automatically assign users to VMs, run the following PowerShell cmdlet:

```
Set-RdsHostPool <tenantname> <hostpoolname> -AssignmentType Automatic
```

To assign a user to the personal desktop host pool, run the following PowerShell cmdlet:

```
Add-RdsAppGroupUser <tenantname> <hostpoolname> "Desktop Application Group" -UserPrincipalName <userupn>
```

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:

```
Set-RdsHostPool <tenantname> <hostpoolname> -AssignmentType Direct
```

To assign a user to the personal desktop host pool, run the following PowerShell cmdlet:

```
Add-RdsAppGroupUser <tenantname> <hostpoolname> "Desktop Application Group" -UserPrincipalName <userupn>
```

To assign a user to a specific session host, run the following PowerShell cmdlet:

```
Set-RdsSessionHost <tenantname> <hostpoolname> -Name <sessionhostname> -AssignedUser <userupn>
```

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

Apply Windows license to session host virtual machines

2/13/2020 • 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 ot 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

2/14/2020 • 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.

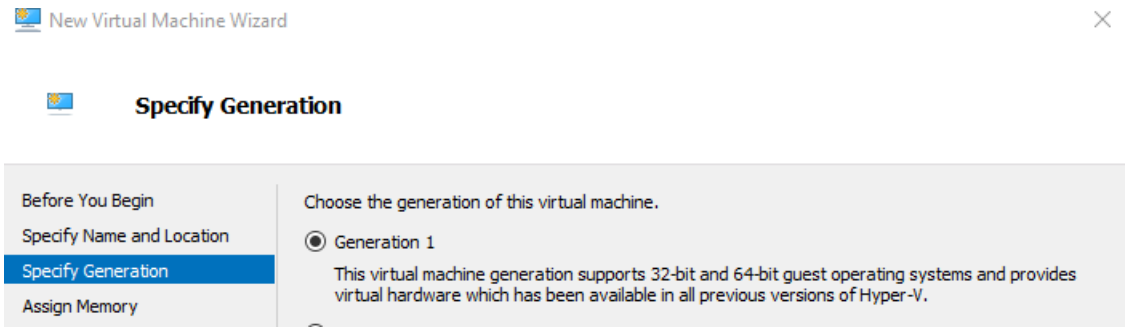
The second option is to create the image locally by downloading the image, provisioning a Hyper-V VM, and customizing it to suit your needs, which we cover in the following section.

Local image creation

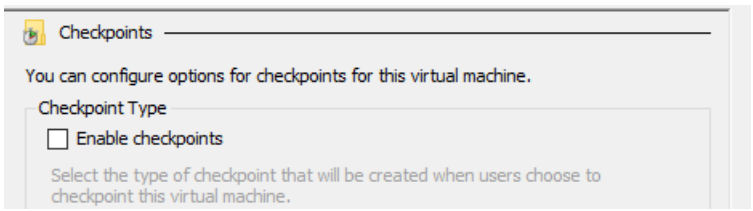
Once you've downloaded the image to a local location, open **Hyper-V Manager** to create a VM with the VHD you copied. The following instructions are a simple version, but you can find more detailed instructions in Create a virtual machine in Hyper-V.

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.



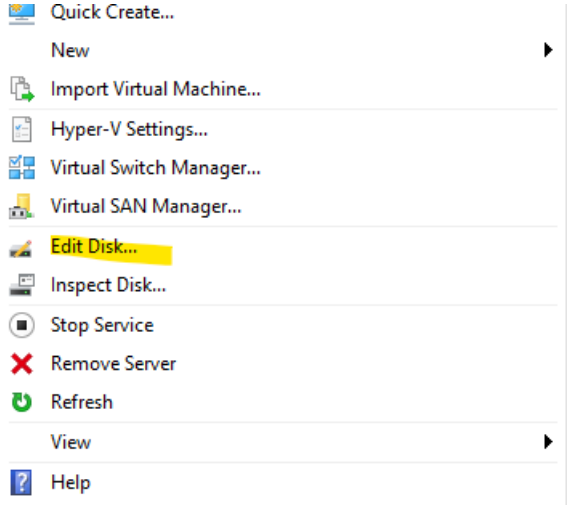
You can also run the following cmdlet in PowerShell to disable checkpoints.

```
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 Office 365 ProPlus 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:

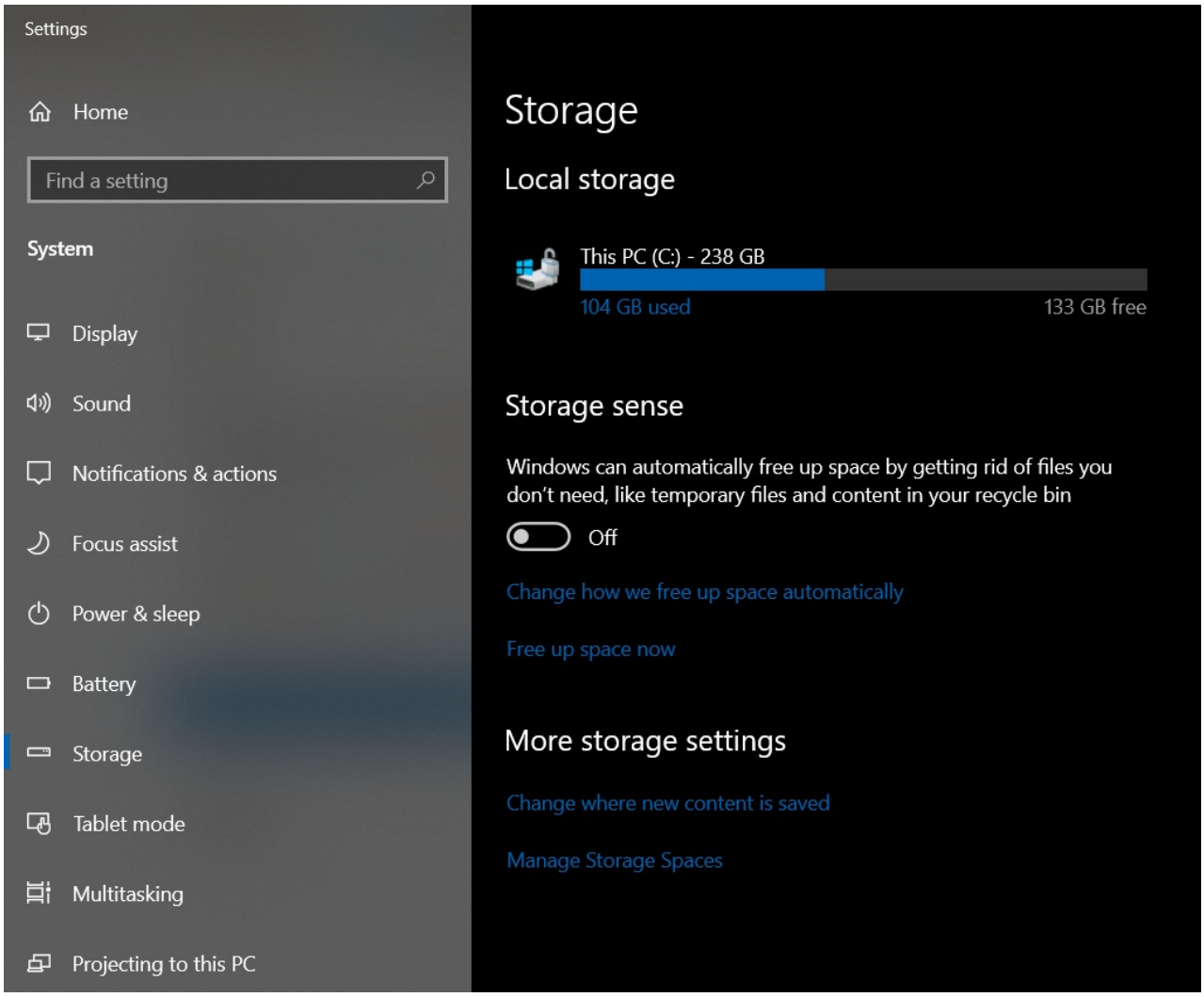
1. On the Active Directory server, open the **Group Policy Management Console**.
2. Expand your domain and Group Policy Objects.
3. Right-click the **Group Policy Object** that you created for the group policy settings and select **Edit**.
4. In the **Group Policy Management Editor**, navigate to **Computer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

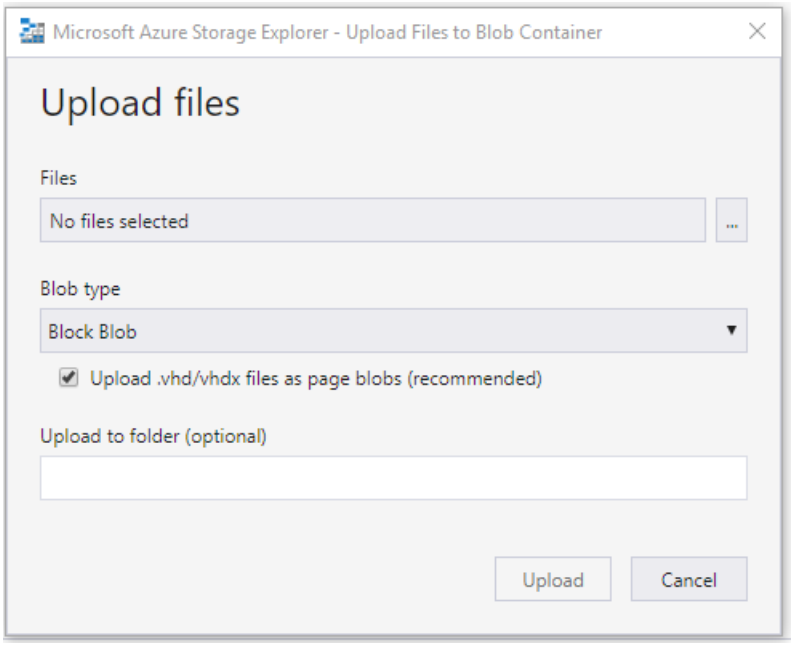
After preparing the image for upload, make sure the VM remains in the off or deallocated state.

Upload master image to a storage account in Azure

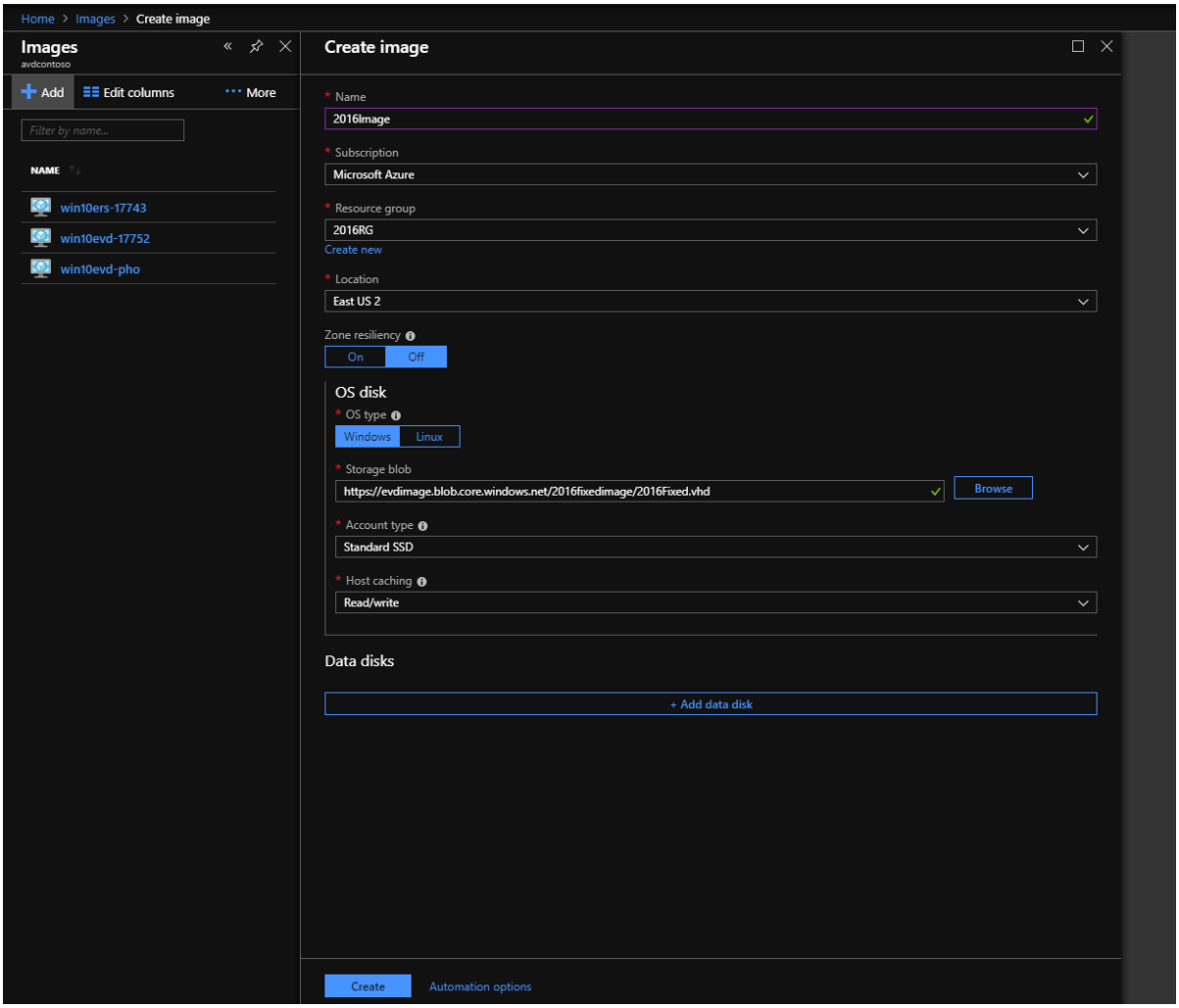
This section only applies when the master image was created locally.

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.

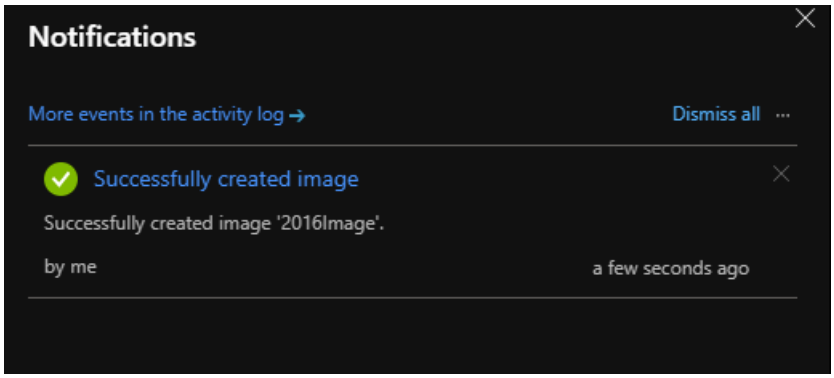
1. Convert the VM image (VHD) to Fixed if you haven’t already. If you don’t convert the image to Fixed, you can't successfully create the image.
2. Upload the VHD to a blob container in your storage account. You can upload quickly with the Storage Explorer tool. To learn more about the Storage Explorer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

2/14/2020 • 4 minutes to read • Edit Online

This article tells you how to install Office 365 ProPlus, 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 Office 365 ProPlus 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 Office 365 ProPlus.

Use the Office Deployment Tool to install Office. Windows 10 Enterprise multi-session only supports the following versions of Office:

- Office 365 ProPlus
- Office 365 Business that comes with a Microsoft 365 Business 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 channel and deliver updates from the monthly channel when they're executed.
- 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 release.

```
<Configuration>
  <Add OfficeClientEdition="64" Channel="Monthly">
    <Product ID="O365ProPlusRetail">
      <Language ID="en-US" />
      <Language ID="MatchOS" />
      <ExcludeApp ID="Groove" />
      <ExcludeApp ID="Lync" />
      <ExcludeApp ID="OneDrive" />
      <ExcludeApp ID="Teams" />
    </Product>
  </Add>
  <RemoveMSI/>
  <Updates Enabled="FALSE"/>
  <Display Level="None" AcceptEULA="TRUE" />
  <Logging Level=" Standard" Path="%temp%\WVDOfficeInstall" />
  <Property Name="FORCEAPPSHUTDOWN" Value="TRUE"/>
  <Property Name="SharedComputerLicensing" Value="1"/>
</Configuration>
```

> **NOTE**
>
> The Office team recommends using 64-bit install for the **OfficeClientEdition** parameter.

After installing Office, you can update the default Office behavior. Run the following commands individually or in a batch file to update the behavior.

```
rem Mount the default user registry hive
reg load HKU\TempDefault C:\Users\Default\NTUSER.DAT
rem Must be executed with default registry hive mounted.
reg add HKU\TempDefault\SOFTWARE\Policies\Microsoft\office\16.0\common /v InsiderSlabBehavior /t REG_DWORD /d
2 /f
rem Set Outlook's Cached Exchange Mode behavior
rem Must be executed with default registry hive mounted.
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v enable /t REG_DWORD
/d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v syncwindowsetting /t
REG_DWORD /d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v
CalendarSyncWindowSetting /t REG_DWORD /d 1 /f
reg add "HKU\TempDefault\software\policies\microsoft\office\16.0\outlook\cached mode" /v
CalendarSyncWindowSettingMonths  /t REG_DWORD /d 1 /f
rem Unmount the default user registry hive
reg unload HKU\TempDefault

rem Set the Office Update UI behavior.
reg add HKLM\SOFTWARE\Policies\Microsoft\office\16.0\common\officeupdate /v hideupdatenotifications /t
REG_DWORD /d 1 /f
reg add HKLM\SOFTWARE\Policies\Microsoft\office\16.0\common\officeupdate /v hideenabledisableupdates /t
REG_DWORD /d 1 /f
```

Install OneDrive in per-machine mode

OneDrive is normally installed per-user. In this environment, it should be installed per-machine.

Here's how to install OneDrive in per-machine mode:

1. First, create a location to stage the OneDrive installer. A local disk folder or [\\unc] (file://unc) location is fine.
2. Download OneDriveSetup.exe to your staged location with this link: https://aka.ms/OneDriveWVD-Installer
3. If you installed office with OneDrive by omitting **<ExcludeApp ID="OneDrive" />**, uninstall any existing OneDrive per-user installations from an elevated command prompt by running the following command:

   ```
   "[staged location]\OneDriveSetup.exe" /uninstall
   ```

4. Run this command from an elevated command prompt to set the **AllUsersInstall** registry value:

   ```
   REG ADD "HKLM\Software\Microsoft\OneDrive" /v "AllUsersInstall" /t REG_DWORD /d 1 /reg:64
   ```

5. Run this command to install OneDrive in per-machine mode:

   ```
   Run "[staged location]\OneDriveSetup.exe" /allusers
   ```

6. Run this command to configure OneDrive to start at sign in for all users:

   ```
   REG ADD "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v OneDrive /t REG_SZ /d "C:\Program Files
   (x86)\Microsoft OneDrive\OneDrive.exe /background" /f
   ```

7. Enable **Silently configure user account** by running the following command.

   ```
   REG ADD "HKLM\SOFTWARE\Policies\Microsoft\OneDrive" /v "SilentAccountConfig" /t REG_DWORD /d 1 /f
   ```

8. Redirect and move Windows known folders to OneDrive by running the following command.

```
REG ADD "HKLM\SOFTWARE\Policies\Microsoft\OneDrive" /v "KFMSilentOptIn" /t REG_SZ /d "<your-AzureAdTenantId>" /f
```

Teams and Skype

Windows Virtual Desktop doesn't support Skype for Business and Teams.

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

2/14/2020 • 10 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 Azure Automation and Azure Logic Apps that will automatically scale session host virtual machines in your Windows Virtual Desktop environment. To learn how to use the scaling tool, skip ahead to Prerequisites.

How the scaling tool works

The scaling tool provides a low-cost automation option for customers who want to optimize their session host VM costs.

You can use the scaling tool to:

- Schedule VMs to start and stop based on Peak and Off-Peak business hours.
- Scale out VMs based on number of sessions per CPU core.
- Scale in VMs during Off-Peak hours, leaving the minimum number of session host VMs running.

The scaling tool uses a combination of Azure Automation PowerShell runbooks, webhooks, and Azure Logic Apps to function. When the tool runs, Azure Logic Apps 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 **createazurelogicapp.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 Set-RdsHostPool to configure the *MaxSessionLimit* parameter accordingly.

During the off-peak usage time, the job determines which session host VMs should shut down based on the *MinimumNumberOfRDSH* parameter. The job will set the session host VMs to drain mode to prevent new sessions connecting to the hosts. If you set the *LimitSecondsToForceLogOffUser* parameter to a non-zero positive value, the script will 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 script will shut down the VM.

If you set the *LimitSecondsToForceLogOffUser* parameter to zero, the job will allow 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** > **Terminal Services** > **Terminal Server** > **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 are no active sessions, the job will shut down the session host VM.

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 virtual machines 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 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 Apps.

> **NOTE**
> The scaling tool controls the load balancing mode of the host pool it is scaling. It sets it to breadth-first load balancing 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 tenant and 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 an Azure Automation account

First, you'll need an Azure Automation account to run the PowerShell runbook. Here's how to set up your account:

1. Open Windows PowerShell as an administrator.
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 that you want to deploy the scaling tool on.

3. Run the following cmdlet to download the script for creating the Azure Automation account:

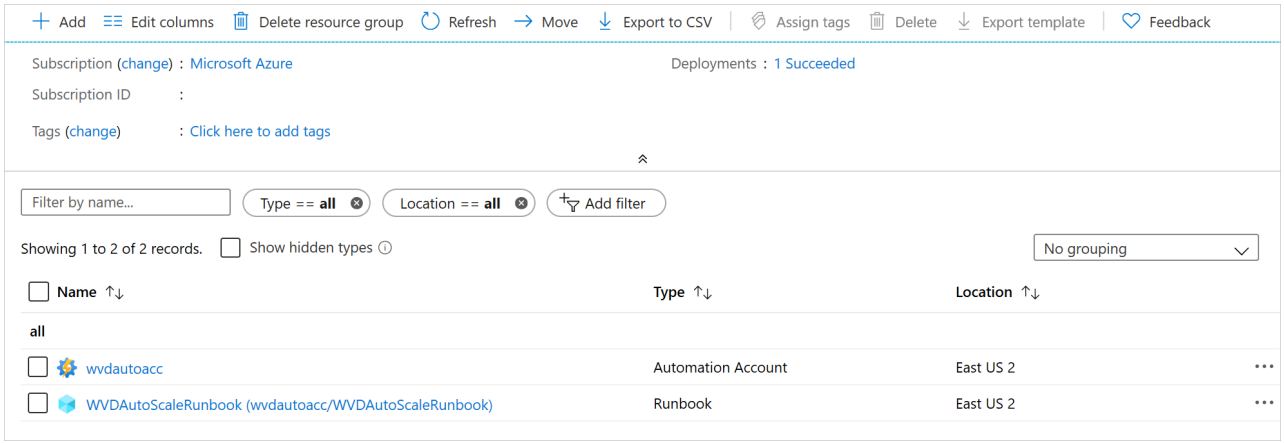
   ```
   Invoke-WebRequest -Uri "https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/wvd-scaling-script/createazureautomationaccount.ps1" -OutFile "your local machine path\createazureautomationaccount.ps1"
   ```

4. Run the following cmdlet to execute the script and create the Azure Automation account:

```
.\createazureautomationaccount.ps1 -SubscriptionID <azuresubscriptionid> -ResourceGroupName
<resourcegroupname> -AutomationAccountName <name of automation account> -Location "Azure region for
deployment"
```

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 apps.

After you've set up your Azure Automation account, sign in to your Azure subscription and check to make sure your Azure Automation account and the relevant runbook have appeared in your specified resource group, as shown in the following image:



To check if your webhook is where it should be, go to the Resources list on the left side of your screen and select **Webhook**.

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 to access your Azure resources.

An Azure Automation Run As account provides authentication for managing resources in Azure with the 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, the 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 authentication, see Limiting 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 by following the next section's instructions.

To create a Run As account in your Azure 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 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 Automation account. The connection asset holds the application ID, tenant ID, subscription ID, and

certificate thumbprint. Remember the application ID, because you'll use it later.

Create a role assignment in Windows Virtual Desktop

Next, you need to create a role assignment so that AzureRunAsConnection can interact with Windows Virtual Desktop. Make sure to use PowerShell to sign in with an account that has permissions to create role assignments.

First, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. Run the following PowerShell cmdlets to connect to Windows Virtual Desktop and display your tenants.

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

Get-RdsTenant
```

When you find the tenant with the host pools you want to scale, follow the instructions in Create an Azure Automation account and use the tenant name you got from the previous cmdlet in the following cmdlet to create the role assignment:

```
New-RdsRoleAssignment -RoleDefinitionName "RDS Contributor" -ApplicationId <applicationid> -TenantName
<tenantname>
```

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.

1. Open Windows PowerShell as an Administrator
2. Run the following cmdlet to sign in to your Azure Account.

   ```
   Login-AzAccount
   ```

3. Run the following cmdlet to download the createazurelogicapp.ps1 script file on your local machine.

   ```
   Invoke-WebRequest -Uri "https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/wvd-
   scaling-script/createazurelogicapp.ps1" -OutFile "your local machine path\ createazurelogicapp.ps1"
   ```

4. Run the following cmdlet to sign into Windows Virtual Desktop with an account that has RDS Owner or RDS Contributor permissions.

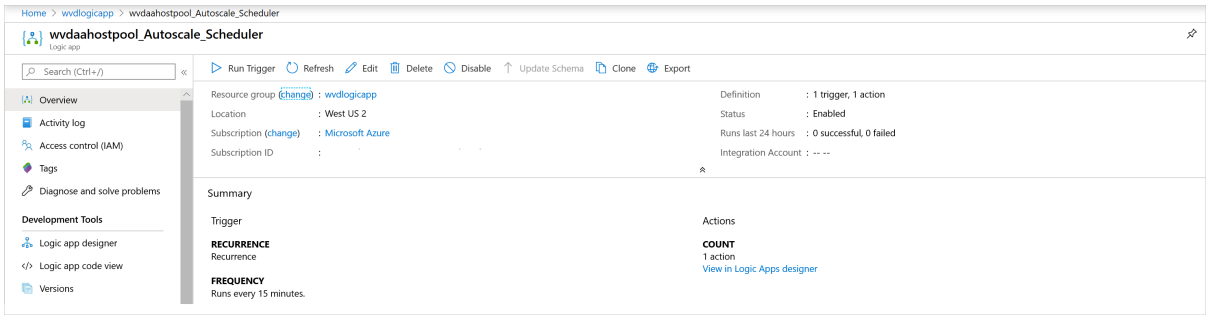
   ```
   Add-RdsAccount -DeploymentUrl "https://rdbroker.wvd.microsoft.com"
   ```

5. Run the following PowerShell script to create the Azure Logic app and execution schedule.

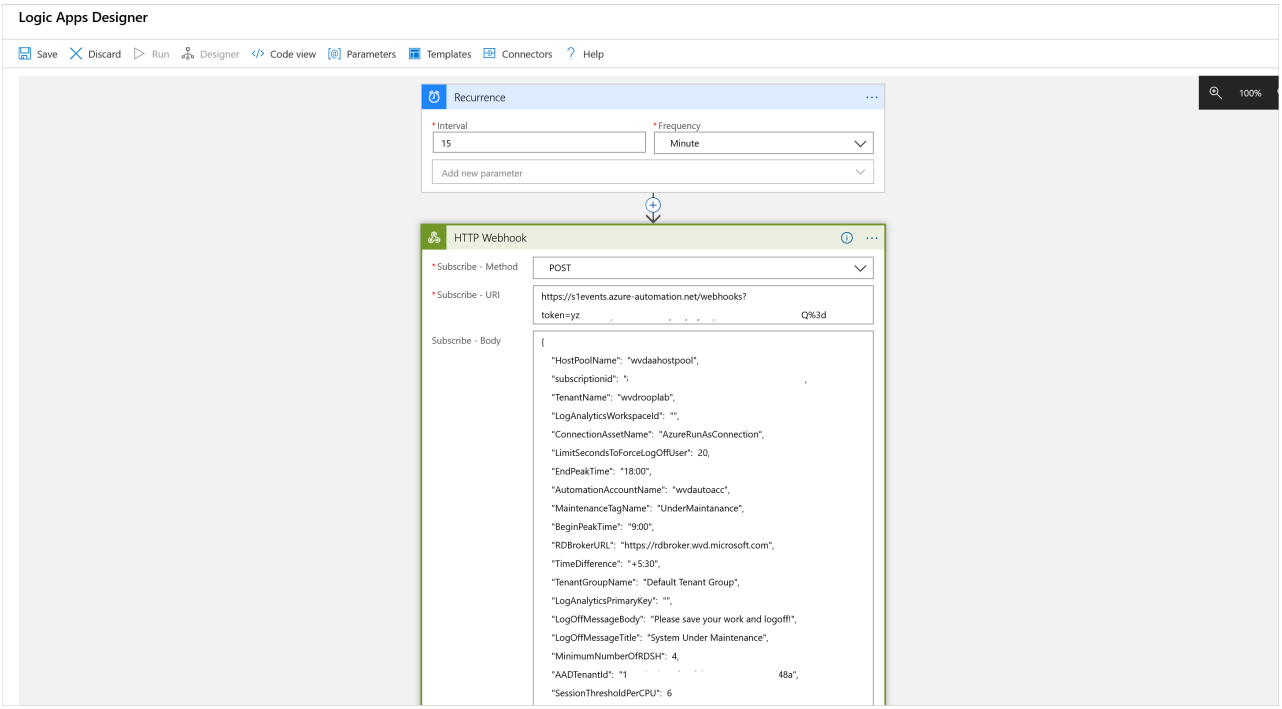
   ```
   $resourceGroupName = Read-Host -Prompt "Enter the name of the resource group for the new Azure Logic
   App"

   $aadTenantId = Read-Host -Prompt "Enter your Azure AD tenant ID"

   $subscriptionId = Read-Host -Prompt "Enter your Azure Subscription ID"

   $tenantName = Read-Host -Prompt "Enter the name of your WVD tenant"

   $hostPoolName = Read-Host -Prompt "Enter the name of the host pool you'd like to scale"

   $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"

$limitSecondsToForceLogOffUser = Read-Host -Prompt "Enter the number of seconds to wait before
automatically signing out users. If set to 0, users will be signed out immediately"

$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"

$location = Read-Host -Prompt "Enter the name of the Azure region where you will be creating the logic
app"

$connectionAssetName = Read-Host -Prompt "Enter the name of the Azure RunAs connection asset"

$webHookURI = Read-Host -Prompt "Enter the URI of the WebHook returned by when you created the Azure
Automation Account"

$automationAccountName = Read-Host -Prompt "Enter the name of the Azure Automation Account"

$maintenanceTagName = Read-Host -Prompt "Enter the name of the Tag associated with VMs you don’t want to
be managed by this scaling tool"

.\createazurelogicapp.ps1 -ResourceGroupName $resourceGroupName `
  -AADTenantID $aadTenantId `
  -SubscriptionID $subscriptionId `
  -TenantName $tenantName `
  -HostPoolName $hostPoolName `
  -RecurrenceInterval $recurrenceInterval `
  -BeginPeakTime $beginPeakTime `
  -EndPeakTime $endPeakTime `
  -TimeDifference $timeDifference `
  -SessionThresholdPerCPU $sessionThresholdPerCPU `
  -MinimumNumberOfRDSH $minimumNumberOfRdsh `
  -LimitSecondsToForceLogOffUser $limitSecondsToForceLogOffUser `
  -LogOffMessageTitle $logOffMessageTitle `
  -LogOffMessageBody $logOffMessageBody `
  -Location $location `
  -ConnectionAssetName $connectionAssetName `
  -WebHookURI $webHookURI `
  -AutomationAccountName $automationAccountName `
  -MaintenanceTagName $maintenanceTagName
```

After you run the script, the 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 Autoscale scheduler and select **Edit** to go to the Logic Apps Designer.



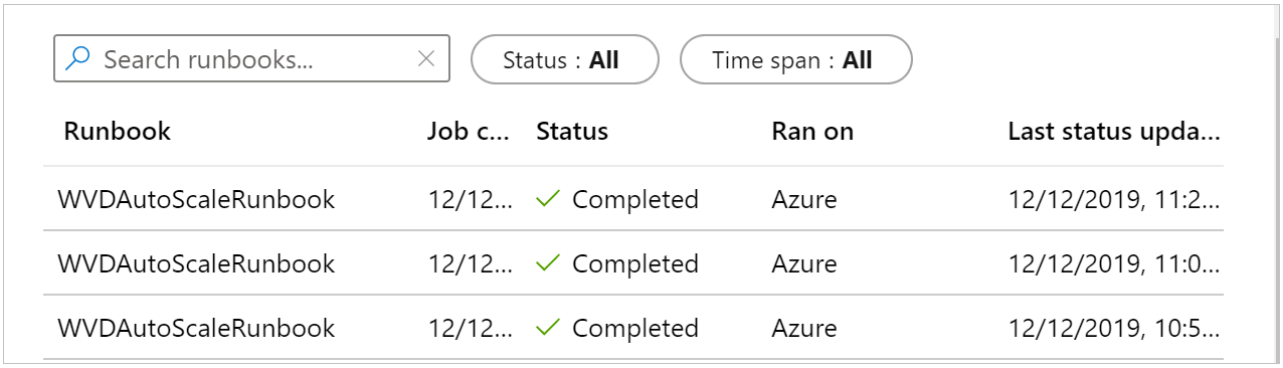
Manage your scaling tool

Now that you've created your scaling tool, you can access its output. This section describes a few features you might find helpful.

View job status

You can view a summarized status of all runbook jobs or view a more in-depth status of a specific runbook job in the Azure portal.

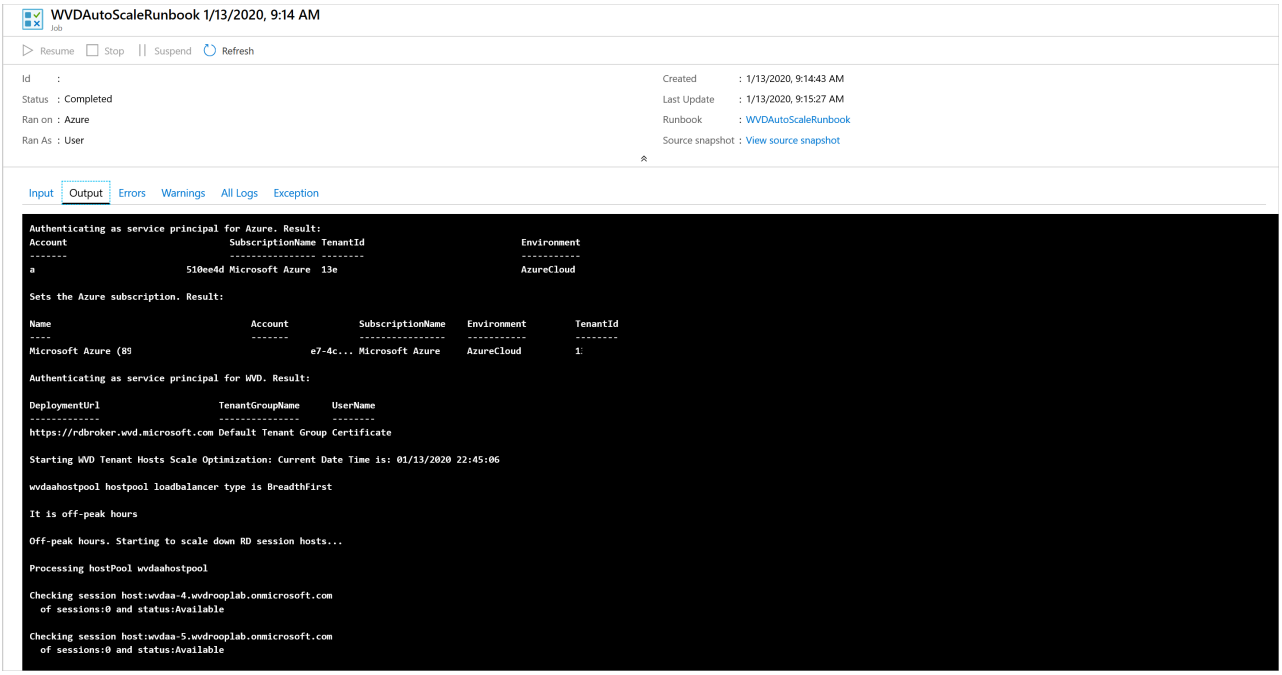
On the right of your selected Automation account, under "Job Statistics," you can view a list of summaries of all runbook jobs. Opening the **Jobs** page on the left side of the window shows current job statuses, start times, and completion times.



View logs and scaling tool output

You can view the logs of scale-out and scale-in operations by opening your runbook and selecting the name of your job.

Navigate to the runbook (the default name is WVDAutoScaleRunbook) 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.



Customize feed for Windows Virtual Desktop users

2/14/2020 • 2 minutes to read • Edit Online

You can customize the feed so the RemoteApp and remote desktop resources appear in a recognizable way for your users.

First, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

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-RdsRemoteApp -TenantName <tenantname> -HostPoolName <hostpoolname> -AppGroupName <appgroupname>
```

```
PS C:\WINDOWS\system32> Get-RdsRemoteApp -TenantName          -HostPoolName hp0 -AppGroupName hp0-win10ms-apps


TenantGroupName     : Default Tenant Group
TenantName          :
HostPoolName        : hp0
AppGroupName        : hp0-win10ms-apps
RemoteAppName       : onedrive
FilePath            : C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe
AppAlias            :
CommandLineSetting  : DoNotAllow
Description         :
FriendlyName        : onedrive
IconIndex           : 0
IconPath            : C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe
RequiredCommandLine :
ShowInWebFeed       : True
```

To assign a friendly name to a RemoteApp, run the following PowerShell cmdlet:

```
Set-RdsRemoteApp -TenantName <tenantname> -HostPoolName <hostpoolname> -AppGroupName <appgroupname> -Name
<existingappname> -FriendlyName <newfriendlyname>
```

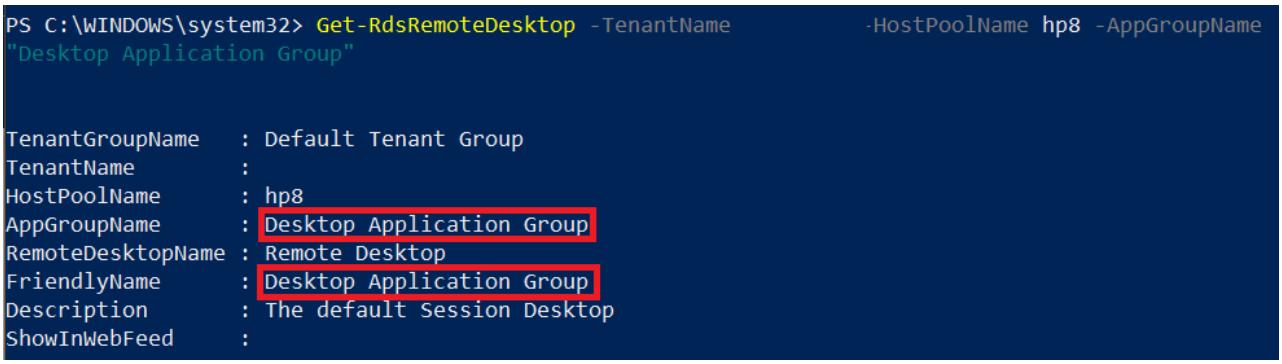
```
PS C:\WINDOWS\system32> Set-RdsRemoteApp -TenantName          -HostPoolName hp0 -AppGroupName hp0-win10ms-apps -Name
onedrive -FriendlyName "onedrive-on-hp0"


TenantGroupName     : Default Tenant Group
TenantName          :
HostPoolName        : hp0
AppGroupName        : hp0-win10ms-apps
RemoteAppName       : onedrive
FilePath            : C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe
AppAlias            :
CommandLineSetting  : DoNotAllow
Description         :
FriendlyName        : onedrive-on-hp0
IconIndex           : 0
IconPath            : C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe
RequiredCommandLine :
ShowInWebFeed       : True
```

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.

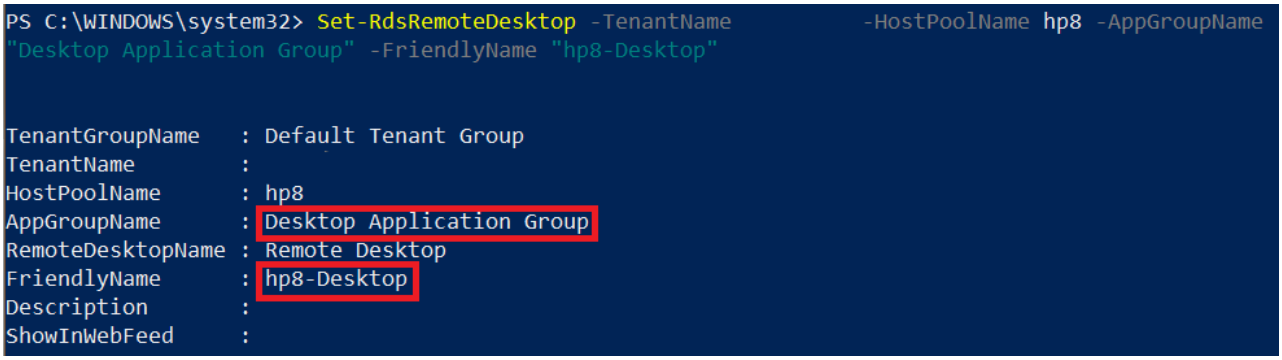
To retrieve the remote desktop resource, run the following PowerShell cmdlet:

```
Get-RdsRemoteDesktop -TenantName <tenantname> -HostPoolName <hostpoolname> -AppGroupName <appgroupname>
```



To assign a friendly name to the remote desktop resource, run the following PowerShell cmdlet:

```
Set-RdsRemoteDesktop -TenantName <tenantname> -HostPoolName <hostpoolname> -AppGroupName <appgroupname> -
FriendlyName <newfriendlyname>
```



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 from Windows 10 or Windows 7
- Connect from a web browser

Deploy a management tool with an Azure Resource Manager template

2/14/2020 • 5 minutes to read • Edit Online

The instructions in this article will tell you how to deploy the UI by using an Azure Resource Manager template.

Important considerations

Since the app requires consent to interact with Windows Virtual Desktop, this tool doesn't support Business-to-Business (B2B) scenarios. Each Azure Active Directory (AAD) tenant's subscription will need its own separate deployment of the management tool.

This management tool is a sample. Microsoft will provide important security and quality updates. The source code is available in GitHub. Customers and partners are encouraged to customize the tool to fit their business needs.

To following browsers are compatible with the management tool:

- Google Chrome 68 or later
- Microsoft Edge 40.15063 or later
- Mozilla Firefox 52.0 or later
- Safari 10 or later (macOS only)

What you need to deploy the management tool

Before deploying the management tool, you'll need an Azure Active Directory (Azure AD) user to create an app registration and deploy the management UI. This user must:

- Have Azure Multi-Factor Authentication (MFA) disabled
- Have permission to create resources in your Azure subscription
- Have permission to create an Azure AD application. Follow these steps to check if your user has the required permissions by following the instructions in Required permissions.

After you deploy and configure the management tool, we recommend you ask a user to launch the management UI to make sure everything works. The user who launches the management UI must have a role assignment that lets them view or edit the Windows Virtual Desktop tenant.

Deploy the management tool

Before you start, ensure the server and client apps have consent by visiting the Windows Virtual Desktop Consent Page for the Azure Active Directory (AAD) represented.

Follow these instructions to deploy the Azure Resource Management template:

1. Go to the GitHub Azure RDS-Templates page.
2. Deploy the template to Azure.
   - If you're deploying in an Enterprise subscription, scroll down and select **Deploy to Azure**.
   - If you're deploying in a Cloud Solution Provider subscription, follow these instructions to deploy to Azure:
     a. Scroll down and right-click **Deploy to Azure**, then select **Copy Link Location**.
     b. Open a text editor like Notepad and paste the link there.

c. Right after https://portal.azure.com/ and before the hashtag (#), enter an at sign (@) followed by the tenant domain name. Here's an example of the format: https://portal.azure.com/@Contoso.onmicrosoft.com#create/.

d. Sign in to the Azure portal as a user with Admin/Contributor permissions to the Cloud Solution Provider subscription.

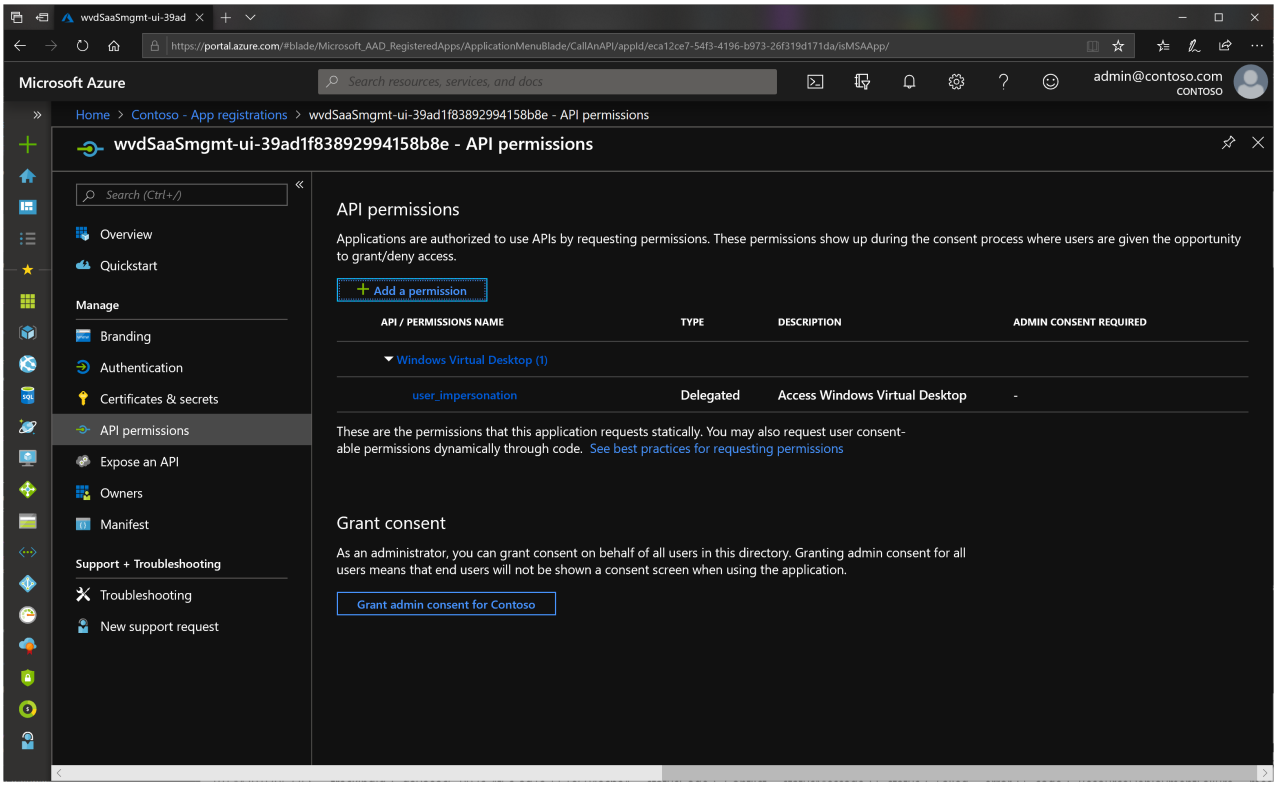
e. Paste the link you copied to the text editor into the address bar.

3. When entering the parameters, do the following:
   - For the **isServicePrincipal** parameter, select **false**.
   - For the credentials, enter your Azure AD credentials with multi-factor authentication disabled. These credentials will be used to create the Azure AD application and Azure resources. To learn more, see the What you need to deploy the management tool.
   - For the **applicationName**, use a unique name for your app that will be registered in your Azure Active Directory. This name will also be used for the web app URL. For example, you can use a name like "Apr3UX."
4. Once you provide the parameters, accept the terms and conditions and select **Purchase**.

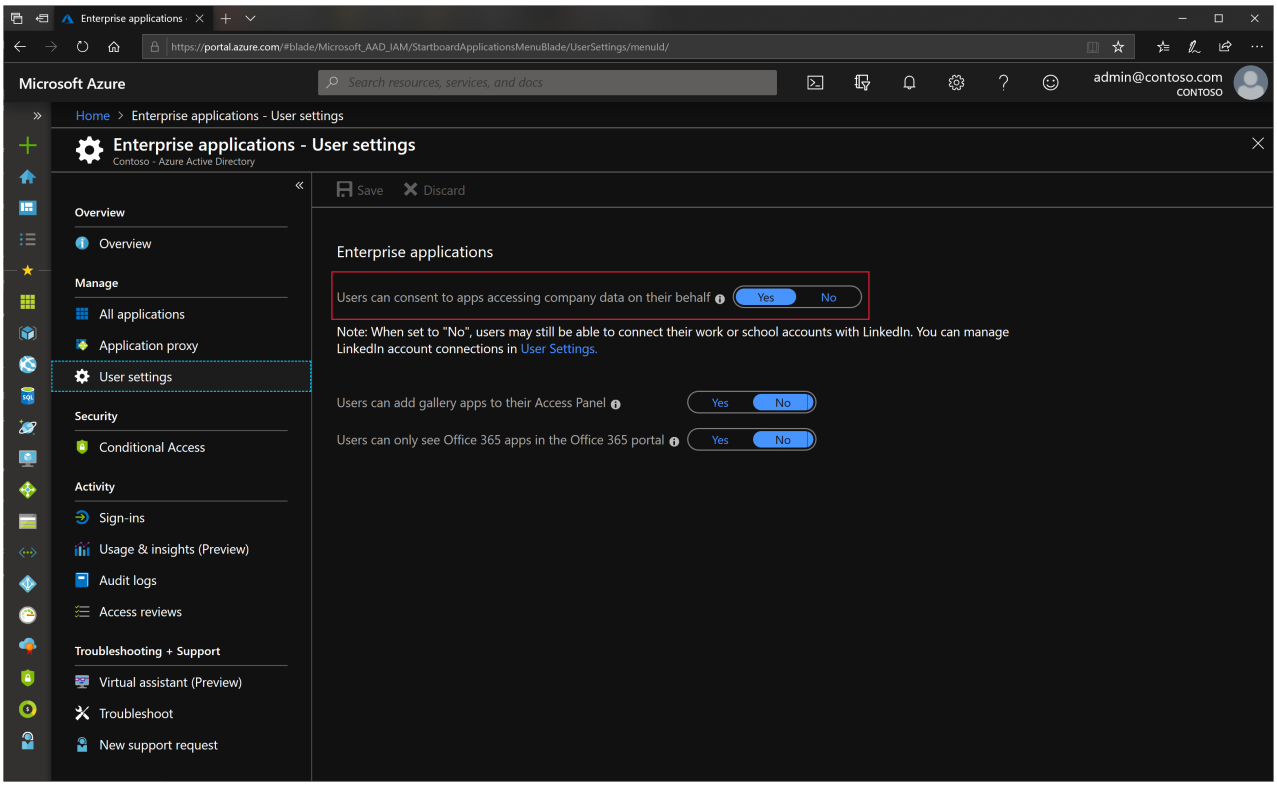
Provide consent for the management tool

After the GitHub Azure Resource Manager template completes, you'll find a resource group containing two app services along with one app service plan in the Azure portal.

Before you sign in and use the management tool, you must provide consent for the new Azure AD application associated with the management tool. Providing consent lets the management tool make Windows Virtual Desktop management calls on behalf of the user currently signed in to the tool.



To determine which user you can use to sign in to the tool, go to your Azure Active Directory user settings page and take note of the value for **Users can consent to apps accessing company data on their behalf**.



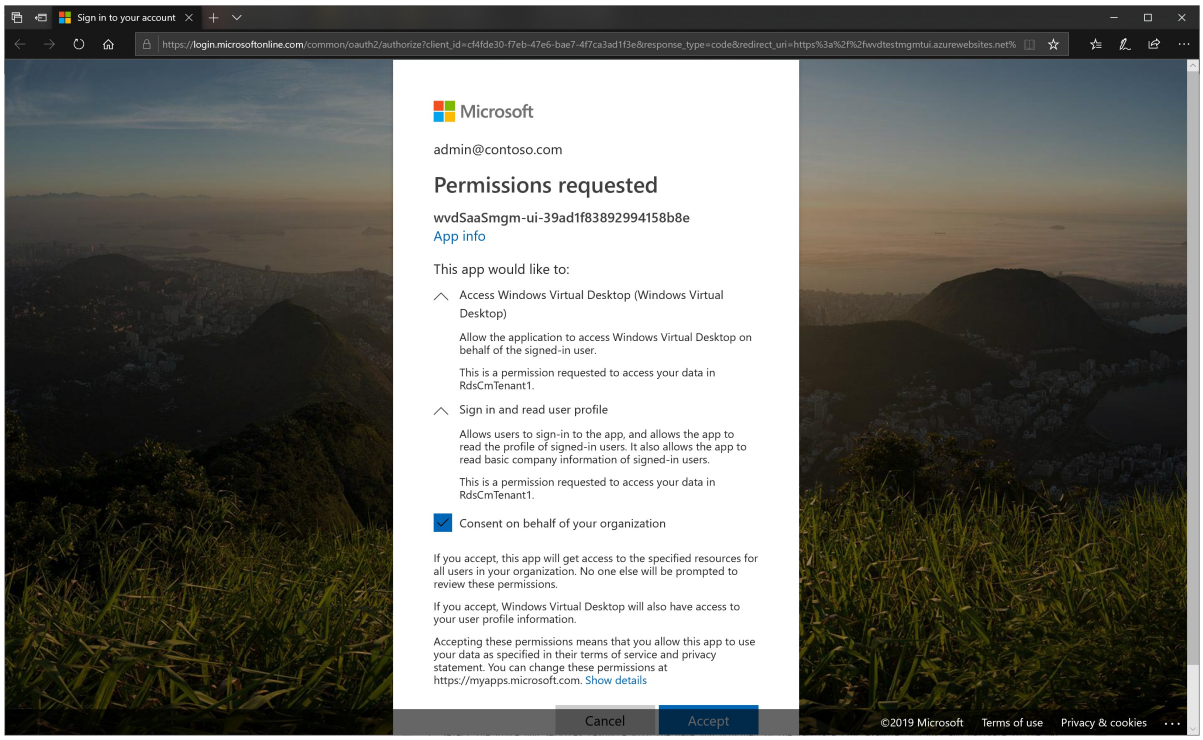
- If the value is set to **Yes**, you can sign in with any user account in the Azure Active Directory and provide consent for that user only. However, if you sign in to the management tool with a different user later, you must perform the same consent again.
- If the value is set to **No**, you must sign in as a Global Administrator in the Azure Active Directory and provide admin consent for all users in the directory. No other users will face a consent prompt.

Once you decide which user you'll use to provide consent, follow these instructions to provide consent to the tool:

1. Go to your Azure resources, select the Azure App Services resource with the name you provided in the template (for example, Apr3UX)and navigate to the URL associated with it; for example, https://rdmimgmtweb-210520190304.azurewebsites.net.

2. Sign in using the appropriate Azure Active Directory user account.

3. If you authenticated with a Global Administrator, you can now select the checkbox to **Consent on behalf of your organization**. Select **Accept** to provide consent.



This will now take you to the management tool.

Use the management tool

After providing consent for the organization or for a specified user, you can access the management tool at any time.

Follow these instructions to launch the tool:

1. Select the Azure App Services resource with the name you provided in the template (for example, Apr3UX)and navigate to the URL associated with it; for example, https://rdmimgmtweb-210520190304.azurewebsites.net.
2. Sign in using your Windows Virtual Desktop credentials.
3. When prompted to choose a Tenant Group, select **Default Tenant Group** from the drop-down list.
4. When you select **Default Tenant Group**, a menu should appear on the left side of your window. In this menu, find the name of your tenant group and select it.

> **NOTE**
> If you have a custom tenant group, enter the name manually instead of choosing from the drop-down list.

Report issues

If you come across any issues with the management tool or other Windows Virtual Desktop tools, follow the directions in Azure Resource Manager templates for Remote Desktop Services to report them on GitHub.

Next steps

Now that you've learned how to deploy and connect to the management tool, you can learn how to use Azure Service help to monitor service issues and health advisories. To learn more, see our Set up service alerts tutorial.

Deploy a management tool with PowerShell

2/14/2020 • 5 minutes to read • Edit Online

This article will show you how to deploy the management tool using PowerShell.

Important considerations

Each Azure Active Directory (Azure AD) tenant's subscription needs its own separate deployment of the management tool. This tool doesn't support Azure AD Business-to-Business (B2B) scenarios.

This management tool is a sample. Microsoft will provide important security and quality updates. The source code is available in GitHub. Whether you're a customer or partner, we encourage you to customize the tool to satisfy your business needs.

The following browsers are compatible with the management tool:

- Google Chrome 68 or later
- Microsoft Edge 40.15063 or later
- Mozilla Firefox 52.0 or later
- Safari 10 or later (macOS only)

What you need to deploy the management tool

Before deploying the management tool, you'll need an Azure Active Directory (Azure AD) user to create an app registration and deploy the management UI. This user must:

- Have permission to create resources in your Azure subscription
- Have permission to create an Azure AD application. Follow these steps to check if your user has the required permissions by following the instructions in Required permissions.

In order to successfully deploy and configure the management tool, you first need to download the following PowerShell scripts from the RDS-Templates GitHub repo and save them to the same folder on your local machine.

- createWvdMgmtUxAppRegistration.ps1
- updateWvdMgmtUxApiUrl.ps1

After you deploy and configure the management tool, we recommend you ask a user to launch the management UI to make sure everything works. The user who launches the management UI must have a role assignment that lets them view or edit the Windows Virtual Desktop tenant.

Set up PowerShell

Get started by signing in to both the Az and Azure AD PowerShell modules. Here's how to sign in:

1. Open PowerShell as an Administrator and navigate to the directory where you saved the PowerShell scripts.
2. Sign in to Azure with an account that has Owner or Contributor permissions on the Azure subscription you plan to use to create the management tool by running the following cmdlet:

   ```
   Login-AzAccount
   ```

3. Run the following cmdlet to sign in to Azure AD with the same account you used for the Az PowerShell

module:

```
Connect-AzureAD
```

4. After that, navigate to the folder where you saved the two PowerShell scripts from the RDS-Templates GitHub repo.

Keep the PowerShell window you used to sign in open to run additional PowerShell cmdlets while signed in.

Create an Azure Active Directory app registration

Run the following commands to create the app registration with required API permissions:

```
$appName = Read-Host -Prompt "Enter a unique name for the management tool's app registration. The name can't
contain spaces or special characters."
$subscriptionId = Read-Host -Prompt "Enter the Azure subscription ID where you will be deploying the management
tool."

.\createWvdMgmtUxAppRegistration.ps1 -AppName $appName -SubscriptionId $subscriptionId
```

Now that you've completed the Azure AD app registration, you can deploy the management tool.

Deploy the management tool

Run the following PowerShell commands to deploy the management tool and associate it with the service principal you just created:

```
$resourceGroupName = Read-Host -Prompt "Enter the Resource Group name"
$location = Read-Host -Prompt "Enter the location (i.e. centralus)"
$templateParameters = @{}
$templateParameters.Add('isServicePrincipal', $true)
$templateParameters.Add('azureAdminUserPrincipalNameOrApplicationId', $ServicePrincipalCredentials.UserName)
$templateParameters.Add('azureAdminPassword', $servicePrincipalCredentials.Password)
$templateParameters.Add('applicationName', $appName)

Get-AzSubscription -SubscriptionId $subscriptionId | Select-AzSubscription
New-AzResourceGroup -Name $resourceGroupName -Location $location
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName `
    -TemplateUri "https://raw.githubusercontent.com/Azure/RDS-Templates/master/wvd-templates/wvd-management-
ux/deploy/mainTemplate.json" `
    -TemplateParameterObject $templateParameters `
    -Verbose
```

After you've created the web app, you must add a redirect URI to the Azure AD application to successfully sign in users.

Set the Redirect URI

Run the following PowerShell commands to retrieve the web app URL and set it as the authentication redirect URI (also called a reply URL):

```
$webApp = Get-AzWebApp -ResourceGroupName $resourceGroupName -Name $appName
$redirectUri = "https://" + $webApp.DefaultHostName + "/"
Get-AzureADApplication -All $true | where { $_.AppId -match $servicePrincipalCredentials.UserName } | Set-
AzureADApplication -ReplyUrls $redirectUri
```

Now that you've added a redirect URI, you next need to update the API URL so the management tool can interact

with the API-backend service.

Update the API URL for the web application

Run the following script to update the API URL configuration in the web application front end:

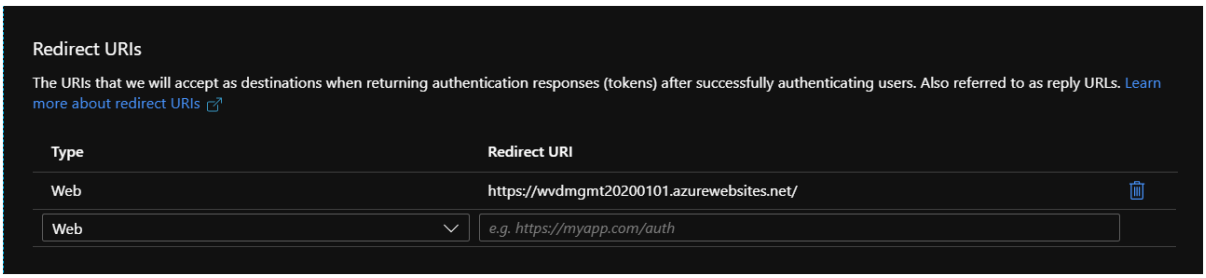
```
.\updateWvdMgmtUxApiUrl.ps1 -AppName $appName -SubscriptionId $subscriptionId
```

Now that you've fully configured the management tool web app, it's time to verify the Azure AD application and provide consent.

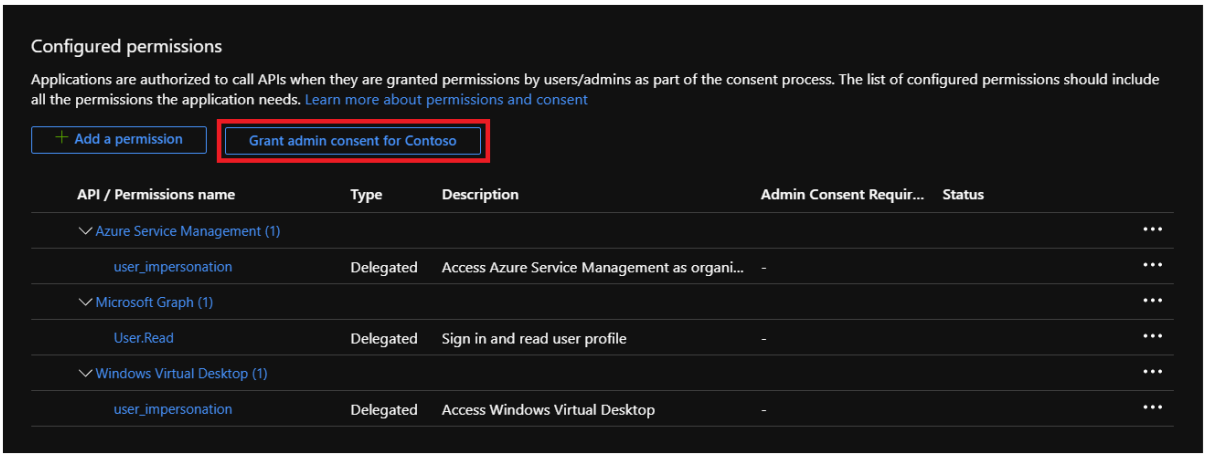
Verify the Azure AD application and provide consent

To verify the Azure AD application configuration and provide consent:

1. Open your internet browser and sign in to the Azure portal with your administrative account.
2. From the search bar at the top of the Azure portal, search for **App registrations** and select the item under **Services**.
3. Select **All applications** and search the unique app name you provided for the PowerShell script in Create an Azure Active Directory app registration.
4. In the panel on the left side of the browser, select **Authentication** and make sure the redirect URI is the same as the web app URL for the management tool, as shown in the following image.



5. In the left panel, select **API permissions** to confirm that permissions were added. If you're a global admin, select the **Grant admin consent for** `tenantname` button and follow the dialog prompts to provide admin consent for your organization.



You can now start using the management tool.

Use the management tool

Now that you've set up the management tool at any time, you can launch it anytime, anywhere. Here's how to

launch the tool:

1. Open the URL of the web app in a web browser. If you don't remember the URL, you can sign in to Azure, find the app service you deployed for the management tool, and then select the URL.
2. Sign in using your Windows Virtual Desktop credentials.

   > **NOTE**
   > If you didn't grant admin consent while configuring the management tool, each user who signs in will need to provide their own user consent in order to use the tool.

3. When prompted to choose a tenant group, select **Default Tenant Group** from the drop-down list.
4. When you select **Default Tenant Group**, a menu should appear on the left side of your window. In this menu, find the name of your tenant group and select it.

   > **NOTE**
   > If you have a custom tenant group, enter the name manually instead of choosing from the drop-down list.

Report issues

If you come across any issues with the management tool or other Windows Virtual Desktop tools, follow the directions in Azure Resource Manager templates for Remote Desktop Services to report them on GitHub.

Next steps

Now that you've learned how to deploy and connect to the management tool, you can learn how to use Azure Service Health to monitor service issues and health advisories. To learn more, see our Set up service alerts tutorial.

Deploy the diagnostics tool

2/14/2020 • 9 minutes to read • Edit Online

Here's what the diagnostics tool for Windows Virtual Desktop can do for you:

- Look up diagnostic activities (management, connection, or feed) for a single user over a period of one week.
- Gather session host information for connection activities from your Log Analytics workspace.
- Review virtual machine (VM) performance details for a particular host.
- See which users are signed in to the session host.
- Send message to active users on a specific session host.
- Sign users out of a session host.

Prerequisites

You need to create an Azure Active Directory App Registration and a Log Analytics workspace before you can deploy the Azure Resource Manager template for the tool. You or the administrator need these permissions to do that:

- Owner of the Azure subscription
- Permission to create resources in your Azure subscription
- Permission to create an Azure AD app
- RDS Owner or Contributor rights

You also need to install these two PowerShell modules before you get started:

- Azure PowerShell module
- Azure AD module

Make sure you have your Subscription ID ready for when you sign in.

After you have everything in order, you can create the Azure AD app registration.

Create an Azure Active Directory app registration

This section will show you how to use PowerShell to create the Azure Active Directory app with a service principal and get API permissions for it.

> **NOTE**
> The API permissions are Windows Virtual Desktop, Log Analytics and Microsoft Graph API permissions are added to the Azure Active Directory Application.

1. Open PowerShell as an Administrator.
2. Sign in to Azure with an account that has Owner or Contributor permissions on the Azure subscription you would like to use for the diagnostics tool:

   ```
   Login-AzAccount
   ```

3. Sign in to Azure AD with the same account:

```
Connect-AzureAD
```

4. Go to the RDS-Templates GitHub repo and run the **CreateADAppRegistrationforDiagnostics.ps1** script in PowerShell.
5. When the script asks you to name your app, enter a unique app name.

After the script successfully runs, it should show the following things in its output:

- A message that confirms your app now has a service principal role assignment.
- Your Client ID and Client Secret Key that you'll need for when you deploy the diagnostics tool.

Now that you've registered your app, it's time to configure your Log Analytics workspace.

Configure your Log Analytics workspace

For the best possible experience, we recommend you configure your Log Analytics workspace with the following performance counters that allow you to derive statements of the user experience in a remote session. For a list of recommended counters with suggested thresholds, see Windows performance counter thresholds.

Create an Azure Log Analytics workspace using PowerShell

You can run a PowerShell script to create a Log Analytics workspace and configure the recommended Windows performance counters to monitor user experience and app performance.

> **NOTE**
> If you already have an existing Log Analytics workspace that you made without the PowerShell script that you want to use, skip ahead to Validate the script results in the Azure portal.

To run the PowerShell script:

1. Open PowerShell as an admin.
2. Go to the RDS-Templates GitHub repo and run the **CreateLogAnalyticsWorkspaceforDiagnostics.ps1** script in PowerShell.
3. Enter the following values for the parameters:
   - For **ResourceGroupName**, enter the name for the resource group.
   - For **LogAnalyticsWorkspaceName**, enter a unique name for your Log Analytics workspace.
   - For **Location**, enter the Azure region you're using.
   - Enter the **Azure Subscription ID**, which you can find in the Azure portal under **Subscriptions**.
4. Enter the credentials of a user with delegated admin access.
5. Sign in to the Azure portal with the same user's credentials.
6. Write down or memorize the LogAnalyticsWorkspace ID for later.
7. If you set up the Log Analytics workspace with the PowerShell script, then your performance counters should already be configured and you can skip ahead to Validate the script results in the Azure portal. Otherwise, continue to the next section.

Configure Windows performance counters in your existing Log Analytics workspace

This section is for users who want to use an existing Azure Log Analytics workspace created without the PowerShell script in the previous section. If you haven't used the script, then you must configure the recommended Windows performance counters manually.

Here's how to manually configure the recommended performance counters:

1. Open your internet browser and sign in to the Azure portal with your administrative account.
2. Next, go to **Log Analytics workspaces** to review the configured Windows Performance Counters.
3. In the **Settings** section, select **Advanced settings**.
4. After that, navigate to **Data** > **Windows Performance Counters** and add the following counters:
   - LogicalDisk(*)\%Free Space
   - LogicalDisk(C:)\Avg. Disk Queue Length
   - Memory(*)\Available Mbytes
   - Processor Information(*)\Processor Time
   - User Input Delay per Session(*)\Max Input Delay

Learn more about the performance counters at Windows and Linux performance data sources in Azure Monitor.

> **NOTE**
> Any additional counters you configure won't show up in the diagnostics tool itself. To make it appear in the diagnostics tool, you need to configure the tool's config file. Instructions for how to do this with advanced administration will be available in GitHub at a later date.

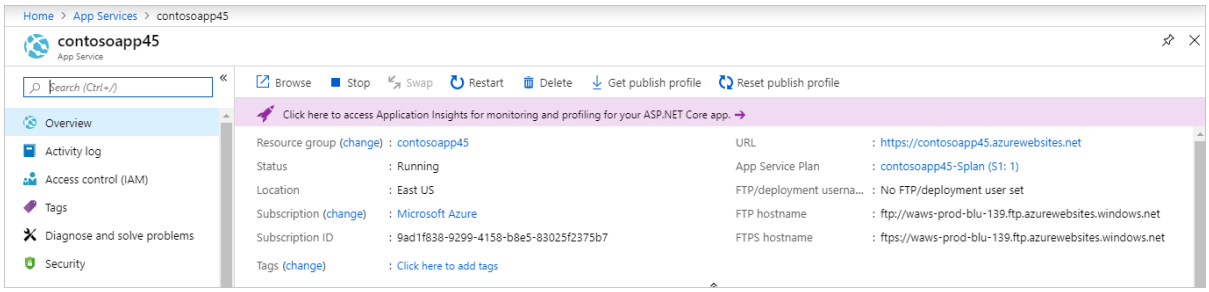
Validate the script results in the Azure portal

Before you continue deploying the diagnostics tool, we recommend that you verify that your Azure Active Directory application has API permissions and your Log Analytics workspace has the preconfigured Windows performance counters.

Review your app registration

To make sure your app registration has API permissions:

1. Open a browser and connect to the Azure portal with your administrative account.
2. Go to **App registrations** and look for your Azure AD App registration.



Review your Log Analytics workspace

To make sure your Log Analytics workspace has the preconfigured Windows performance counters:

1. In the Azure portal, go to **Log Analytics workspaces** to review the configured Windows Performance Counters.
2. Under **Settings**, select **Advanced settings**.
3. After that, go to **Data** > **Windows Performance Counters**.
4. Make sure the following counters are preconfigured:
   - LogicalDisk(*)\%Free Space: Displays the amount of free space of the total usable space on the disk as a

percentage.

- LogicalDisk(C:)\Avg. Disk Queue Length: The length of disk transfer request for your C drive. The value shouldn't exceed 2 for more than a short period of time.
- Memory(*)\Available Mbytes: The available memory for the system in megabytes.
- Processor Information(*)\Processor Time: the percentage of elapsed time that the processor spends to execute a non-Idle thread.
- User Input Delay per Session(*)\Max Input Delay

Connect to VMs in your Log Analytics workspace

In order to be able to view the health of VMs, you'll need to enable the Log Analytics connection. Follow these steps to connect your VMs:

1. Open a browser and sign in to the Azure portal with your administrative account.
2. Go to your Log Analytics Workspace.
3. In the left panel, under Workspace Data Sources, select **virtual machines**.
4. Select the name of the VM you want to connect to.
5. Select **Connect**.

Deploy the diagnostics tool

To deploy the Azure Resource Management template for the diagnostics tool:

1. Go to the GitHub Azure RDS-Templates page.
2. Deploy the template to Azure and follow the instructions in the template. Make sure you have the following information available:
   - Client-Id
   - Client-Secret
   - Log Analytics workspace ID
3. Once the input parameters are provided, accept the terms and conditions, then select **Purchase**.

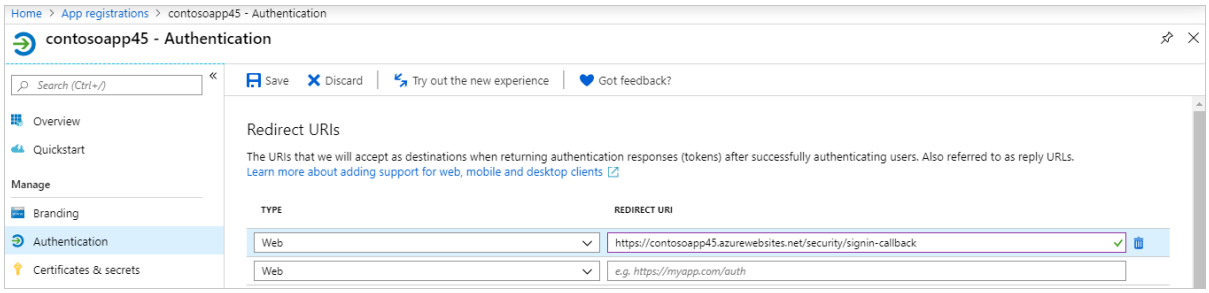
The deployment will take 2–3 minutes. After successful deployment, go to the resource group and make sure the web app and app service plan resources are there.

After that, you need to set the Redirect URI.

Set the Redirect URI

To set the Redirect URI:

1. In the Azure portal, go to **App Services** and locate the application you created.
2. Go to the overview page and copy the URL you find there.
3. Navigate to **app registrations** and select the app you want to deploy.
4. In the left panel, under Manage section, select **Authentication**.
5. Enter the desired Redirect URI into the **Redirect URI** text box, then select **Save** in the top-left corner of the menu.
6. Select **Web** in the drop-down menu under Type.
7. Enter the URL from the app overview page and add **/security/signin-callback** to the end of it. For example: `https://<yourappname>.azurewebsites.net/security/signin-callback`.



8. Now, go to your Azure resources, select the Azure App Services resource with the name you provided in the template and navigate to the URL associated with it. (For example, if the app name you used in the template was `contosoapp45`, then your associated URL is https://contosoapp45.azurewebsites.net).
9. Sign in using the appropriate Azure Active Directory user account.
10. Select **Accept**.

Distribute the diagnostics tool

Before you make the diagnostics tool available to your users, make sure they have the following permissions:

- Users need read access for log analytics. For more information, see Get started with roles, permissions, and security with Azure Monitor.
- Users also need read access for the Windows Virtual Desktop tenant (RDS Reader role). For more information, see Delegated access in Windows Virtual Desktop.

You also need to give your users the following information:

- The app's URL
- The names of the tenant group individual tenant they can access.

Use the diagnostics tool

After you've signed in to your account using the information you've received from your organization, have the UPN ready for the user you want to query activities for. A search will give you all activities under the specified activity type that happened within the last week.

How to read activity search results

Activities are sorted by timestamp, with the latest activity first. If the results return an error, first check to see if it's a service error. For service errors, create a support ticket with the activity information to help us debug the issue. All other error types can usually be solved by the user or administrator. For a list of the most common error scenarios and how to solve them, see Identify and diagnose issues.

> **NOTE**
> Service errors are called "external errors" in the linked documentation. This will be changed when we update the PowerShell reference.

Connection activities might have more than one error. You can expand the activity type to see any other errors the user has come across. Select the name of the error code to open up a dialog to see more information about it.

Investigate the session host

In the search results, find and select the session host you want information about.

You can analyze session host health:

- Based on a predefined threshold, you can retrieve the session host health information that Log Analytics

queries.

- When there's no activity or the session host isn't connected to Log Analytics, the information won't be available.

You can also interact with users on the session host:

- You can either sign out or send a message to signed in users.
- The user you originally searched for is selected by default, but you can also select additional users to send messages or sign out multiple users at once.

Windows performance counter thresholds

- LogicalDisk(*)\%Free Space:
  - Displays the percentage of the total usable space on the logical disk that is free.
  - Threshold: Less than 20% is marked as unhealthy.
- LogicalDisk(C:)\Avg. Disk Queue Length:
  - Represents storage system conditions.
  - Threshold: Higher than 5 is marked as unhealthy.
- Memory(*)\Available Mbytes:
  - The available memory for the system.
  - Threshold: Less than 500 megabytes marked as unhealthy.
- Processor Information(*)\Processor Time:
  - Threshold: Higher than 80% is marked as unhealthy.
- User Input Delay per Session(*)\Max Input Delay:
  - Threshold: Higher than 2000 ms is marked as unhealthy.

Next steps

- Learn how to monitor activity logs at Use diagnostics with Log Analytics.
- Read about common error scenarios and how to fix them at Identify and diagnose issues.

Use Log Analytics for the diagnostics feature

12/18/2019 • 3 minutes to read • Edit Online

Windows Virtual Desktop offers a diagnostics feature that allows the administrator to identify issues through a single interface. This feature logs diagnostics information whenever someone assigned Windows Virtual Desktop role uses the service. Each log contains information about which Windows Virtual Desktop role was involved in the activity, any error messages that appear during the session, tenant information, and user information. The diagnostics feature creates activity logs for both user and administrative actions. Each activity log falls under three main categories:

- Feed subscription activities: when a user tries to connect to their feed through Microsoft Remote Desktop applications.
- Connection activities: when a user tries to connect to a desktop or RemoteApp through Microsoft Remote Desktop applications.
- Management activities: when an administrator performs management operations on the system, such as creating host pools, assigning users to app groups, and creating role assignments.

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.

Why you should use Log Analytics

We recommend you use Log Analytics to analyze diagnostics data in the Azure client that goes beyond single-user troubleshooting. As you can pull in VM performance counters into Log Analytics you have one tool to gather information for your deployment.

Before you get started

Before you can use Log Analytics with the diagnostics feature, you'll need to create a workspace.

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.

Push diagnostics data to your workspace

You can push diagnostics data from your Windows Virtual Desktop tenant into the Log Analytics for your workspace. You can set up this feature right away when you first create your tenant by linking your workspace to your tenant, or you can set it up later with an existing tenant.

To link your tenant to your Log Analytics workspace while you're setting up your new tenant, run the following cmdlet to sign in to Windows Virtual Desktop with your TenantCreator user account:

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

If you're going to link an existing tenant instead of a new tenant, run this cmdlet instead:

```
Set-RdsTenant -Name <TenantName> -AzureSubscriptionId <SubscriptionID> -LogAnalyticsWorkspaceId <String> -
LogAnalyticsPrimaryKey <String>
```

You'll need to run these cmdlets for every tenant you want to link to Log Analytics.

> **NOTE**
> If you don't want to link the Log Analytics workspace when you create a tenant, run the `New-RdsTenant` cmdlet instead.

Cadence for sending diagnostic events

Diagnostic events are sent to Log Analytics when completed.

Example queries

The following example queries show how the diagnostics feature generates a report for the most frequent activities in your system:

This first example shows connection activities initiated by users with supported remote desktop clients:

```
WVDActivityV1_CL

| where Type_s == "Connection"

| join kind=leftouter (

 WVDErrorV1_CL

 | summarize Errors = makelist(pack('Time', Time_t, 'Code', ErrorCode_s , 'CodeSymbolic', ErrorCodeSymbolic_s,
'Message', ErrorMessage_s, 'ReportedBy', ReportedBy_s , 'Internal', ErrorInternal_s )) by ActivityId_g

 ) on $left.Id_g == $right.ActivityId_g

| join kind=leftouter (

WVDCheckpointV1_CL

 | summarize Checkpoints = makelist(pack('Time', Time_t, 'ReportedBy', ReportedBy_s, 'Name', Name_s,
'Parameters', Parameters_s) ) by ActivityId_g

 ) on $left.Id_g == $right.ActivityId_g

|project-away ActivityId_g, ActivityId_g1
```

This next example query shows management activities by admins on tenants:

```
WVDActivityV1_CL

| where Type_s == "Management"

| join kind=leftouter (

 WVDErrorV1_CL

 | summarize Errors = makelist(pack('Time', Time_t, 'Code', ErrorCode_s , 'CodeSymbolic', ErrorCodeSymbolic_s,
'Message', ErrorMessage_s, 'ReportedBy', ReportedBy_s , 'Internal', ErrorInternal_s )) by ActivityId_g

 ) on $left.Id_g == $right.ActivityId_g

| join kind=leftouter (

WVDCheckpointV1_CL

 | summarize Checkpoints = makelist(pack('Time', Time_t, 'ReportedBy', ReportedBy_s, 'Name', Name_s,
'Parameters', Parameters_s) ) by ActivityId_g

 ) on $left.Id_g == $right.ActivityId_g

|project-away ActivityId_g, ActivityId_g1
```

Stop sending data to Log Analytics

To stop sending data from an existing tenant to Log Analytics, run the following cmdlet and set empty strings:

```
Set-RdsTenant -Name <TenantName> -AzureSubscriptionId <SubscriptionID> -LogAnalyticsWorkspaceId <String> -
LogAnalyticsPrimaryKey <String>
```

You'll need to run this cmdlet for every tenant you want to stop sending data from.

Next steps

To review common error scenarios that the diagnostics feature can identify for you, see Identify and diagnose issues.

Publish built-in apps in Windows Virtual Desktop

2/20/2020 • 2 minutes to read • Edit Online

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-RdsRemoteApp <tenantname> <hostpoolname> <appgroupname> -Name <remoteappname> -FriendlyName
<remoteappname> -FilePath "shell:appsFolder\<PackageFamilyName>!App"
```

NOTE

Windows Virtual Desktop only supports publishing apps with install locations that begin with `C:\Program Files\Windows Apps`.

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-RdsRemoteApp <tenantname> <hostpoolname> <appgroupname> -Name <remoteappname> -FriendlyName <remoteappname>
-FilePath "shell:Appsfolder\Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge"
```

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

2/19/2020 • 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 it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see Supplemental Terms of Use for Microsoft Azure Previews.

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. For information, see Create a tenant in Windows Virtual Desktop.
- 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'll use for the MSIX app. To get the OS image:

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 19035** 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

Before you get started, 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
```

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 **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 **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 the PowerShell script

Before you update the PowerShell scripts, make sure you have the volume GUID of the volume in the VHD. To get the volume GUID:

1. Open the network share where the VHD is located inside the VM where you'll run the script.
2. Right-click the VHD and select **Mount**. This will mount the VHD to a drive letter.
3. After you mount the VHD, the **File Explorer** window will open. Capture the parent folder and update the **$parentFolder** variable

   > **NOTE**
   > If you don't see a parent folder, that means the MSIX wasn't expanded properly. Redo the previous section and try again.

4. Open the parent folder. If correctly expanded, you'll see a folder with the same name as the package. Update the **$packageName** variable to match the name of this folder.

   For example, `VSCodeUserSetup-x64-1.38.1_1.38.1.0_x64__8wekyb3d8bbwe`.

5. Open a command prompt and enter **mountvol**. This command will display a list of volumes and their

GUIDs. Copy the GUID of the volume where the drive letter matches the drive you mounted your VHD to in step 2.

For example, in this example output for the mountvol command, if you mounted your VHD to Drive C, you'll want to copy the value above `C:\`:

```
Possible values for VolumeName along with current mount points are:

\\?\Volume{a12b3456-0000-0000-0000-10000000000}\
*** NO MOUNT POINTS ***

\\?\Volume{c78d9012-0000-0000-0000-20000000000}\
    E:\

\\?\Volume{d34e5678-0000-0000-0000-30000000000}\
    C:\
```

6. Update the **$volumeGuid** variable with the volume GUID you just copied.
7. Open an Admin PowerShell prompt and update the following PowerShell script with the variables that apply to your environment.

```
#MSIX app attach staging sample

#region variables

$vhdSrc="<path to vhd>"

$packageName = "<package name>"

$parentFolder = "<package parent folder>"

$parentFolder = "\" + $parentFolder + "\"

$volumeGuid = "<vol guid>"

$msixJunction = "C:\temp\AppAttach\"

#endregion

#region mountvhd

try

{

Mount-Diskimage -ImagePath $vhdSrc -NoDriveLetter -Access ReadOnly

Write-Host ("Mounting of " + $vhdSrc + " was completed!") -BackgroundColor Green

}

catch

{

Write-Host ("Mounting of " + $vhdSrc + " has failed!") -BackgroundColor Red

}

#endregion

#region makelink
```

```
$msixDest = "\\?\Volume{" + $volumeGuid + "}\"

if (!(Test-Path $msixJunction))

{

md $msixJunction

}

$msixJunction = $msixJunction + $packageName

cmd.exe /c mklink /j $msixJunction $msixDest

#endregion

#region stage

[Windows.Management.Deployment.PackageManager,Windows.Management.Deployment,ContentType=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, or leave feedback for the MSIX app and packaging tool at the MSIX app attach feedback hub and the MSIX packaging tool

feedback hub.

Windows Virtual Desktop environment

9/30/2019 • 3 minutes to read • Edit Online

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.

Tenants

The Windows Virtual Desktop tenant is the primary interface for managing your Windows Virtual Desktop environment. Each Windows Virtual Desktop tenant must be associated with the Azure Active Directory containing the users who will sign in to the environment. From the Windows Virtual Desktop tenant, you can begin creating host pools to run your users' workloads.

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't be assigned to both a desktop app group and a RemoteApp app group in the same host pool.
- 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.

Tenant groups

In Windows Virtual Desktop, the Windows Virtual Desktop tenant is where most of the setup and configuration happens. The Windows Virtual Desktop tenant contains the host pools, app groups, and app group user assignments. However, there may be certain situations where you need to manage multiple Windows Virtual Desktop tenants at once, particularly if you're a Cloud Service Provider (CSP) or a hosting partner. In these situations, you can use a custom Windows Virtual Desktop tenant group to place each of the customers' Windows Virtual Desktop tenants and centrally manage access. However, if you're only managing a single Windows Virtual Desktop tenant, the tenant group concept doesn't apply and you can continue to operate and manage your tenant that exists in the default tenant group.

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 tenant, see Create a tenant in Windows Virtual Desktop.

To learn how to connect to Windows Virtual Desktop, see one of the following articles:

- Connect from Windows 10 or Windows 7
- Connect from a web browser

Determine user connection latency in Windows Virtual Desktop

10/30/2019 • 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

2/14/2020 • 2 minutes to read • Edit Online

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
  - Service principals
- Role definition
  - Built-in roles
- Scope
  - Tenant groups
  - Tenants
  - Host pools
  - App groups

Built-in roles

Delegated access in Windows Virtual Desktop has several built-in role definitions you can assign to users and service principals.

- An RDS Owner can manage everything, including access to resources.
- An RDS Contributor can manage everything but access to resources.
- An RDS Reader can view everything but can't make any changes.
- An RDS Operator can view diagnostic activities.

PowerShell cmdlets for role assignments

You can run the following cmdlets to create, view, and remove role assignments:

- **Get-RdsRoleAssignment** displays a list of role assignments.
- **New-RdsRoleAssignment** creates a new role assignment.
- **Remove-RdsRoleAssignment** deletes role assignments.

Accepted parameters

You can modify the basic three cmdlets with the following parameters:

- **AadTenantId**: specifies the Azure Active Directory tenant ID from which the service principal is a member.
- **AppGroupName**: name of the Remote Desktop app group.
- **Diagnostics**: indicates the diagnostics scope. (Must be paired with either the **Infrastructure** or **Tenant** parameters.)
- **HostPoolName**: name of the Remote Desktop host pool.
- **Infrastructure**: indicates the infrastructure scope.

- **RoleDefinitionName**: name of the Remote Desktop Services role-based access control role assigned to the user, group, or app. (For example, Remote Desktop Services Owner, Remote Desktop Services Reader, and so on.)
- **ServerPrincipleName**: name of the Azure Active Directory application.
- **SignInName**: the user's email address or user principal name.
- **TenantName**: name of the Remote Desktop tenant.

Next steps

For a more complete list of PowerShell cmdlets each role can use, see the PowerShell reference.

For guidelines for how to set up a Windows Virtual Desktop environment, see Windows Virtual Desktop environment.

Host pool load-balancing methods

11/4/2019 • 2 minutes to read • Edit Online

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

2/21/2020 • 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	
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

2/14/2020 • 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.

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
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
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	Azure 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

2/13/2020 • 16 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 gives organizations the ability to connect their own instance of Microsoft Azure to the simple, intuitive Horizon Cloud control plane, creating a secure, comprehensive, cloud-hosted solution for delivering virtualized Windows applications and desktops. With the release of Windows Virtual Desktop, VMware has partnered with Microsoft to extend the functionality of Windows Virtual Desktop to customers using Horizon Cloud on Microsoft Azure. This functionality will be available in Tech Preview by the end of 2019.

As an approved provider, the benefits of Windows Virtual Desktop will be extended to VMware Horizon Cloud on Microsoft Azure customers, including Windows 10 Enterprise for multi-session, Windows 7 with free Extended Security Updates for up to three years, and FSLogix. Additionally, customers will receive the enterprise-class features of Horizon Cloud, such as broad endpoint support; global, cloud-optimized brokering; flexible desktop options and configurations; user-environment management; and support for hybrid environments.

- Register for the Tech Preview of Horizon Cloud on Microsoft Azure.
- Go to the partner website.

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.

Cloudhouse

Cloudhouse™
Compatibility · Containers

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

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.

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.

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.

- 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 Office 365, ormoving 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, savingmoney on servers, managing usersmore 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.

- 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.

- 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

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 ensure 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.

- 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.

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 uses 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 10 Enterprise multi-session FAQ

2/20/2020 • 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 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 Office Pro Plus, go to the Azure portal and search for Microsoft Windows 10 + Office 365 ProPlus.

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 life-cycle policy as Windows 10 Enterprise, which means the spring release is supported for 18 months and the fall 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 Preview documentation
- Visit our Windows Virtual Desktop TechCommunity
- Set up your Windows Virtual Desktop deployment with the Windows Virtual Desktop tutorials

Data locations for Windows Virtual Desktop

9/27/2019 • 2 minutes to read • Edit Online

Windows Virtual Desktop is currently available for all geographical locations. Initially, service metadata can only be stored in the United States (US) geography. 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 located in the United States. The stored metadata is encrypted at rest, and geo-redundant mirrors are maintained within the United States. All customer data, such as app settings and user data, resides in the location the customer chooses and isn't managed by the service.

Service metadata is replicated in the United States for disaster recovery purposes.

Troubleshooting overview, feedback, and support

2/14/2020 • 4 minutes to read • Edit Online

This article provides an overview of the issues you may encounter when setting up a Windows Virtual Desktop tenant environment and provides ways to resolve the issues.

Provide feedback

Visit the Windows Virtual Desktop Tech Community to discuss the Windows Virtual Desktop service with the product team and active community members.

Escalation tracks

Use the following table to identify and resolve issues you may encounter when setting up a tenant environment using Remote Desktop client. Once your tenant's set up, you can use our new Diagnostics service to identify issues for common scenarios.

> **NOTE**
> We have a Tech Community forum which you can visit to discuss your issues with the product team and active community members. Visit the Windows Virtual Desktop Tech Community to start a discussion.

ISSUE	SUGGESTED SOLUTION
Creating a Windows Virtual Desktop tenant	If there's an Azure outage, open an Azure support request; otherwise open an Azure support request, select **Windows Virtual Desktop** for the service, select **Deployment** for the problem type, then select **Issues creating a Windows Virtual Desktop tenant** for the problem subtype.
Accessing Marketplace templates in Azure portal	If there's an Azure outage, open an Azure support request. Azure Marketplace Windows Virtual Desktop templates are freely available.
Accessing Azure Resource Manager templates from GitHub	See the Creating Windows Virtual Desktop session host VMs section of Tenant and host pool creation. If the problem is still unresolved, contact the GitHub support team. If the error occurs after accessing the template in GitHub, contact Azure Support.
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 **Virtual Machine running Windows** for the service. For issues with the Azure Resource Manager templates that are provided with Windows Virtual Desktop, see Creating Windows Virtual Desktop tenant section of Tenant and host pool creation.

ISSUE	SUGGESTED SOLUTION
Managing Windows Virtual Desktop session host environment from the Azure portal	Open an Azure support request. 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 tenant using PowerShell** for the problem subtype.
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	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. If it's a network issue, your users need to contact their network administrator.
Connected but no feed	Troubleshoot using the User connects but nothing is displayed (no feed) section of Windows Virtual Desktop service connections. 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.
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 when using Windows Virtual Desktop tools on GitHub (Azure Resource Manager templates, diagnostics tool, management tool)	See Azure Resource Manager Templates for Remote Desktop Services to report issues.

Next steps

- To troubleshoot issues while creating a tenant and host pool in a Windows Virtual Desktop environment, see

Tenant 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 errors during deployment, see View deployment operations.

Identify and diagnose issues

2/14/2020 • 7 minutes to read • Edit Online

Windows Virtual Desktop offers a diagnostics feature that allows the administrator to identify issues through a single interface. The Windows Virtual Desktop roles log a diagnostic activity whenever a user interacts with the system. Each log contains relevant information such as the Windows Virtual Desktop roles involved in the transaction, error messages, tenant information, and user information. Diagnostic activities are created by both end-user and administrative actions, and can be categorized into three main buckets:

- Feed subscription activities: the end-user triggers these activities whenever they try to connect to their feed through Microsoft Remote Desktop applications.
- Connection activities: the end-user triggers these activities whenever they try to connect to a desktop or RemoteApp through Microsoft Remote Desktop applications.
- Management activities: the administrator triggers these activities whenever they perform management operations on the system, such as creating host pools, assigning users to app groups, and creating role assignments.

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.

To get started, download and import the Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already. After that, run the following cmdlet to sign in to your account:

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

Diagnose issues with PowerShell

Windows Virtual Desktop Diagnostics uses just one PowerShell cmdlet but contains many optional parameters to help narrow down and isolate issues. The following sections list the cmdlets you can run to diagnose issues. Most filters can be applied together. Values listed in brackets, such as `<tenantName>`, should be replaced with the values that apply to your situation.

Retrieve diagnostic activities in your tenant

You can retrieve diagnostic activities by entering the **Get-RdsDiagnosticActivities** cmdlet. The following example cmdlet will return a list of diagnostic activities, sorted from most to least recent.

```
Get-RdsDiagnosticActivities -TenantName <tenantName>
```

Like other Windows Virtual Desktop PowerShell cmdlets, you must use the **-TenantName** parameter to specify the name of the tenant you want to use for your query. The tenant name is applicable for almost all diagnostic activity queries.

Retrieve detailed diagnostic activities

The **-Detailed** parameter provides additional details for each diagnostic activity returned. The format for each activity varies depending on its activity type. The **-Detailed** parameter can be added to any **Get-RdsDiagnosticActivities** query, as shown in the following example.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -Detailed
```

Retrieve a specific diagnostic activity by activity ID

The **-ActivityId** parameter returns a specific diagnostic activity if it exists, as shown in the following example cmdlet.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -ActivityId <ActivityIdGuid>
```

View error messages for a failed activity by activity ID

To view the error messages for a failed activity, you must run the cmdlet with the **-Detailed** parameter. You can view the list of errors by running the **Select-Object** cmdlet.

```
Get-RdsDiagnosticActivities -TenantName <tenantname> -ActivityId <ActivityGuid> -Detailed | Select-Object -
ExpandProperty Errors
```

Filter diagnostic activities by user

The **-UserName** parameter returns a list of diagnostic activities initiated by the specified user, as shown in the following example cmdlet.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -UserName <UserUPN>
```

The **-UserName** parameter can also be combined with other optional filtering parameters.

Filter diagnostic activities by time

You can filter the returned diagnostic activity list with the **-StartTime** and **-EndTime** parameters. The **-StartTime** parameter will return a diagnostic activity list starting from a specific date, as shown in the following example.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -StartTime "08/01/2018"
```

The **-EndTime** parameter can be added to a cmdlet with the **-StartTime** parameter to specify a specific period of time you want to receive results for. The following example cmdlet will return a list of diagnostic activities from between August 1 and August 10.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -StartTime "08/01/2018" -EndTime "08/10/2018"
```

The **-StartTime** and **-EndTime** parameters can also be combined with other optional filtering parameters.

Filter diagnostic activities by activity type

You can also filter diagnostic activities by activity type with the **-ActivityType** parameter. The following cmdlet will return a list of end-user connections:

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -ActivityType Connection
```

The following cmdlet will return a list of administrator management tasks:

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -ActivityType Management
```

The **Get-RdsDiagnosticActivities** cmdlet doesn't currently support specifying Feed as the ActivityType.

Filter diagnostic activities by outcome

You can filter the returned diagnostic activity list by outcome with the **-Outcome** parameter. The following example cmdlet will return a list of successful diagnostic activities.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -Outcome Success
```

The following example cmdlet will return a list of failed diagnostic activities.

```
Get-RdsDiagnosticActivities -TenantName <tenantName> -Outcome Failure
```

The **-Outcome** parameter can also be combined with other optional filtering parameters.

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 tenant administrator and need to be resolved as a support issue. When providing feedback through the Windows Virtual Desktop Tech Community, include the activity ID and approximate time frame of when the issue occurred.
- External Issue: relate to scenarios which can be mitigated by the system administrator. 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.

External management error codes

NUMERIC CODE	ERROR CODE	SUGGESTED SOLUTION
3	UnauthorizedAccess	The user who tried to run the administrative PowerShell cmdlet either doesn't have permissions to do so or mistyped their username.
1000	TenantNotFound	The tenant name you entered doesn't match any existing tenants. Review the tenant name for typos and try again.
1006	TenantCannotBeRemovedHasSessionHostPools	You can't delete a tenant as long it contains objects. Delete the session host pools first, then try again.
2000	HostPoolNotFound	The host pool name you entered doesn't match any existing host pools. Review the host pool name for typos and try again.
2005	HostPoolCannotBeRemovedHasApplicationGroups	You can't delete a host pool as long as it contains objects. Remove all app groups in the host pool first.

NUMERIC CODE	ERROR CODE	SUGGESTED SOLUTION
2004	HostPoolCannotBeRemovedHasSessionHosts	Remove all sessions hosts first before deleting the session host pool.
5001	SessionHostNotFound	The session host you queried might be offline. Check the host pool's status.
5008	SessionHostUserSessionsExist	You must sign out all users on the session host before executing your intended management activity.
6000	AppGroupNotFound	The app group name you entered doesn't match any existing app groups. Review the app group name for typos and try again.
6022	RemoteAppNotFound	The RemoteApp name you entered doesn't match any RemoteApps. Review RemoteApp name for typos and try again.
6010	PublishedItemsExist	The name of the resource you're trying to publish is the same as a resource that already exists. Change the resource name and try again.
7002	NameNotValidWhiteSpace	Don’t use white space in the name.
8000	InvalidAuthorizationRoleScope	The role name you entered doesn't match any existing role names. Review the role name for typos and try again.
8001	UserNotFound	The user name you entered doesn't match any existing user names. Review the name for typos and try again.
8005	UserNotFoundInAAD	The user name you entered doesn't match any existing user names. Review the name for typos and try again.
8008	TenantConsentRequired	Follow the instructions here to provide consent for your tenant.

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.

NUMERIC CODE	ERROR CODE	SUGGESTED SOLUTION
-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	The application the user tried to access is either incorrectly installed or set to an incorrect path.
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.

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.

Tenant and host pool creation

2/14/2020 • 12 minutes to read • Edit Online

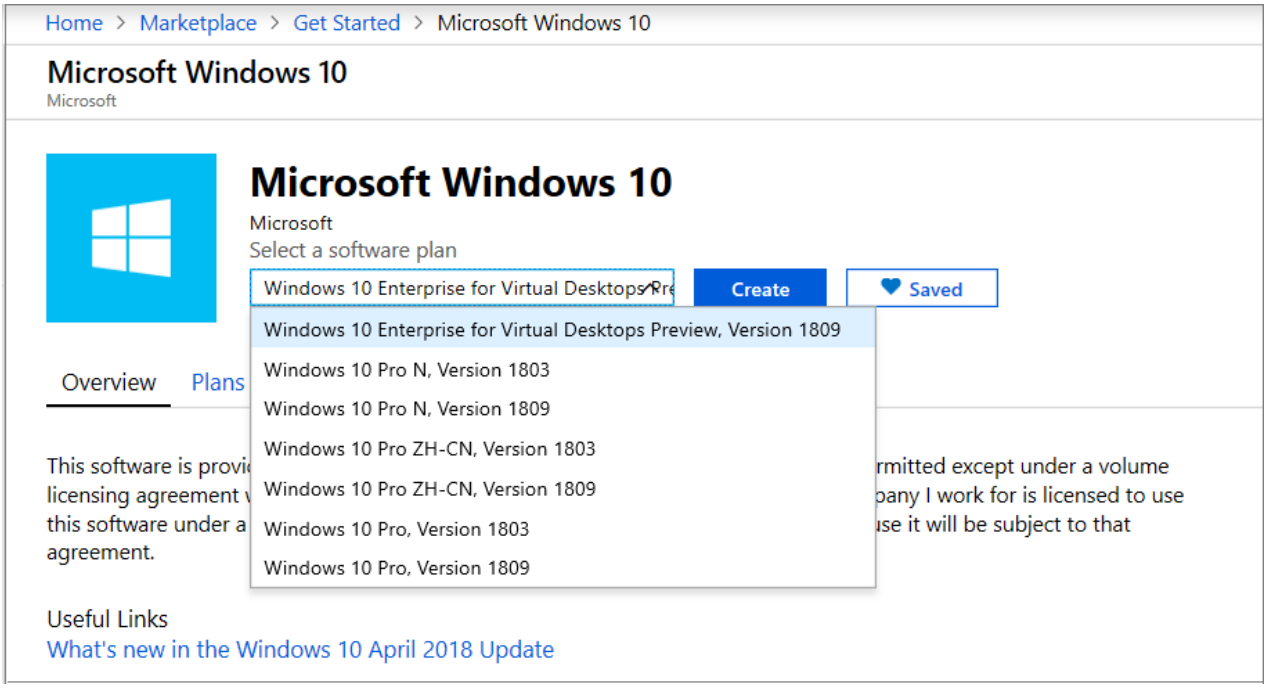
This article covers issues during the initial setup of the Windows Virtual Desktop tenant and the related session host pool infrastructure.

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

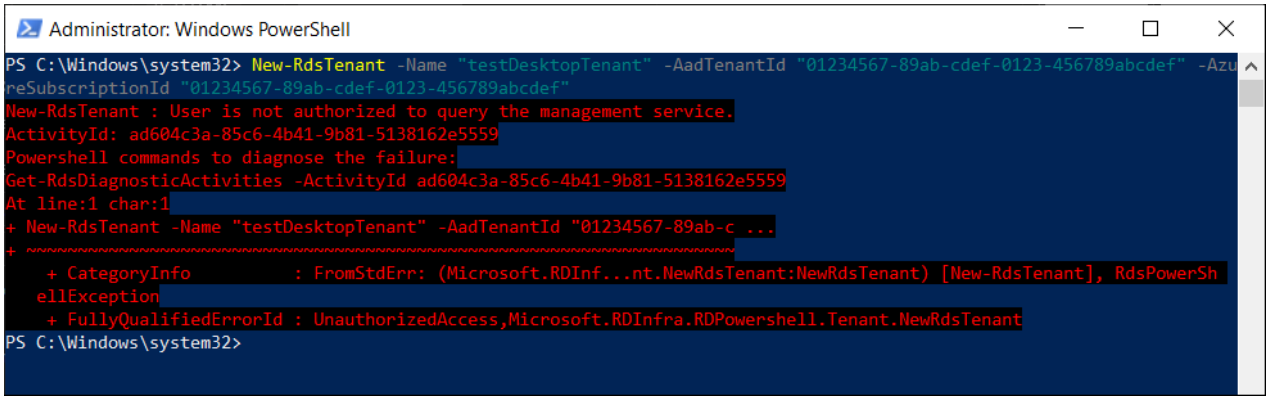
To use the Windows 10 Enterprise multi-session image, go to the Azure Marketplace, select **Get Started** > **Microsoft Windows 10** > and Windows 10 Enterprise for Virtual Desktops, Version 1809.



Creating Windows Virtual Desktop tenant

This section covers potential issues when creating the Windows Virtual Desktop tenant.

Error: The user isn't authorized to query the management service

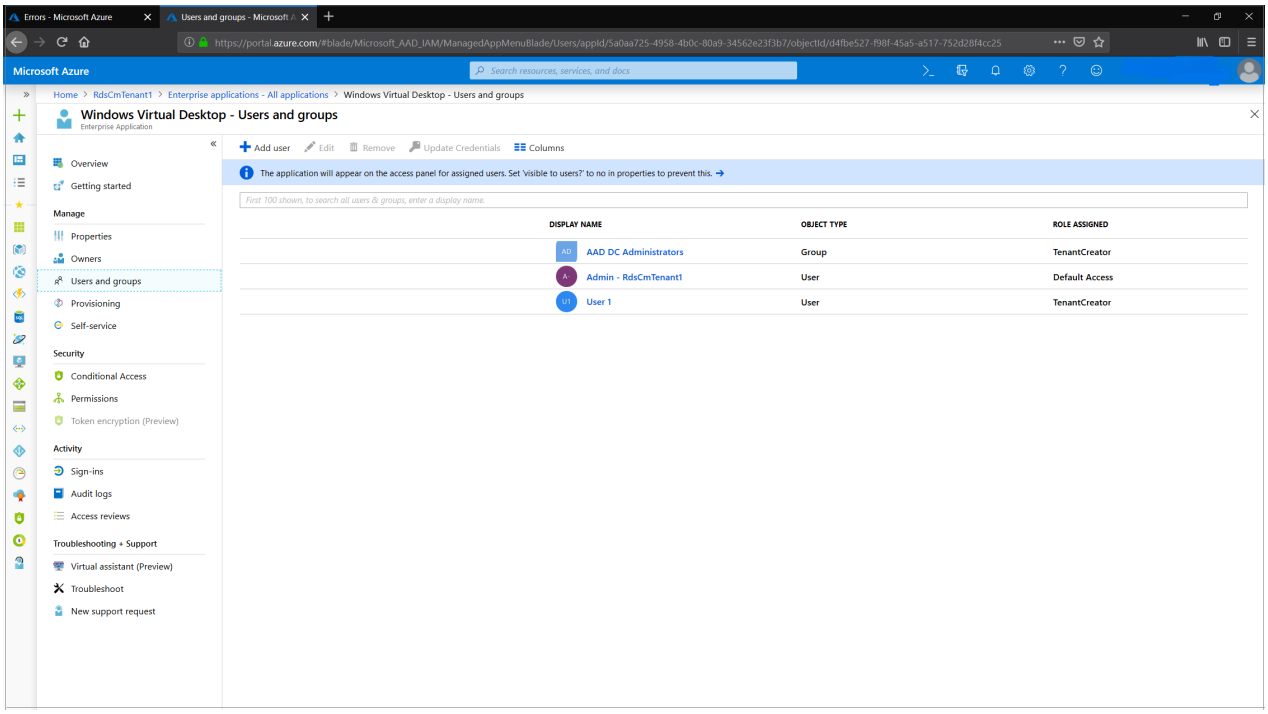


Example of raw error:

```
   New-RdsTenant : User isn't authorized to query the management service.
   ActivityId: ad604c3a-85c6-4b41-9b81-5138162e5559
   Powershell commands to diagnose the failure:
   Get-RdsDiagnosticActivities -ActivityId ad604c3a-85c6-4b41-9b81-5138162e5559
   At line:1 char:1
   + New-RdsTenant -Name "testDesktopTenant" -AadTenantId "01234567-89ab-c ...
   + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       + CategoryInfo          : FromStdErr: (Microsoft.RDInf...nt.NewRdsTenant:NewRdsTenant) [New-
RdsTenant], RdsPowerSh
      ellException
       + FullyQualifiedErrorId : UnauthorizedAccess,Microsoft.RDInfra.RDPowershell.Tenant.NewRdsTenant
```

Cause: The user who's signed in hasn't been assigned the TenantCreator role in their Azure Active Directory.

Fix: Follow the instructions in Assign the TenantCreator application role to a user in your Azure Active Directory tenant. After following the instructions, you'll have a user assigned to the TenantCreator role.



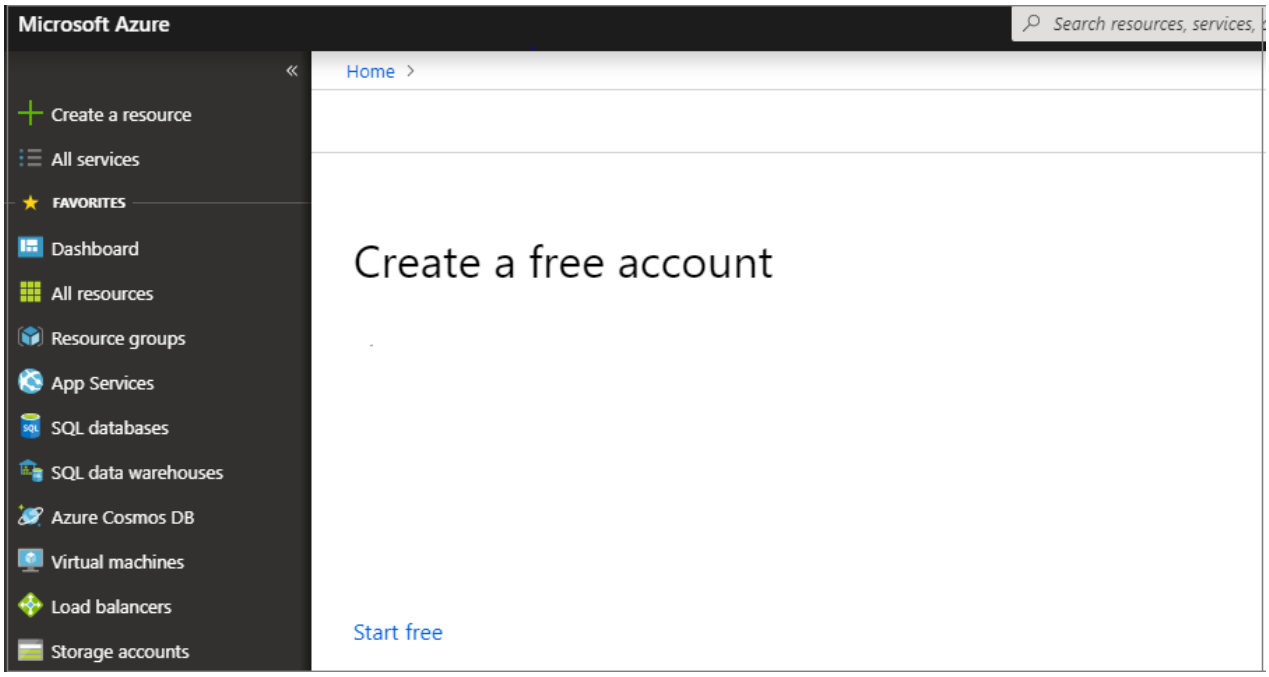
Creating Windows Virtual Desktop session host VMs

Session host VMs can be created in several ways, but the Windows Virtual Desktop team only supports VM provisioning issues related to the Azure Marketplace offering. For more information, see Issues using Windows Virtual Desktop - Provision a host pool Azure Marketplace offering.

Issues using Windows Virtual Desktop – Provision a host pool Azure Marketplace offering

The Windows Virtual Desktop – Provision a host pool template is available from the Azure Marketplace.

Error: When using the link from GitHub, the message "Create a free account" appears



Cause 1: There aren't active subscriptions in the account used to sign in to Azure or the account used doesn't have permissions to view the subscriptions.

Fix 1: Sign in with an account that has contributor access (at a minimum) to the subscription where session host VMs are going to be deployed.

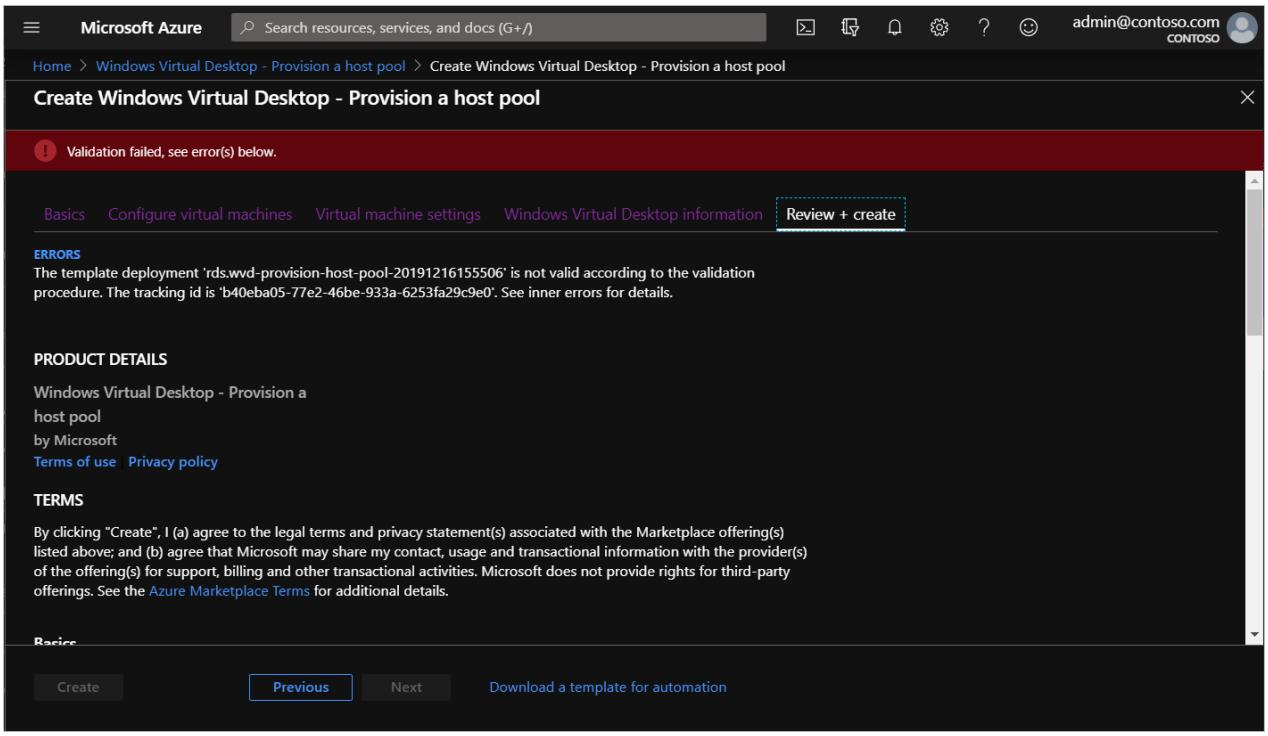
Cause 2: The subscription being used is part of a Microsoft Cloud Service Provider (CSP) tenant.

Fix 2: Go to the GitHub location for **Create and provision new Windows Virtual Desktop host pool** and follow these instructions:

1. Right-click on **Deploy to Azure** and select **Copy link address**.
2. Open **Notepad** and paste the link.
3. Before the # character, insert the CSP end customer tenant name.
4. Open the new link in a browser and the Azure portal will load the template.

```
Example: https://portal.azure.com/<CSP end customer tenant name>
#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%
2FRDS-Templates%2Fmaster%2Fwvd-
templates%2FCreate%20and%20provision%20WVD%20host%20pool%2FmainTemplate.json
```

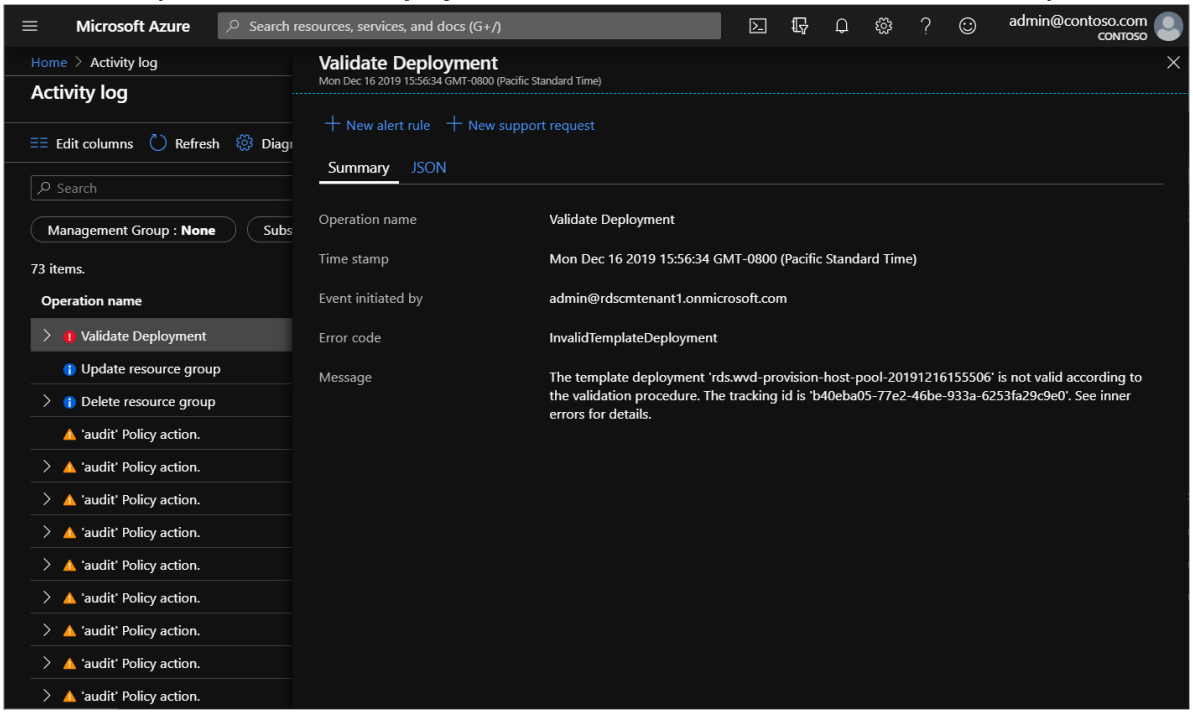
Error: You receive "template deployment is not valid" error



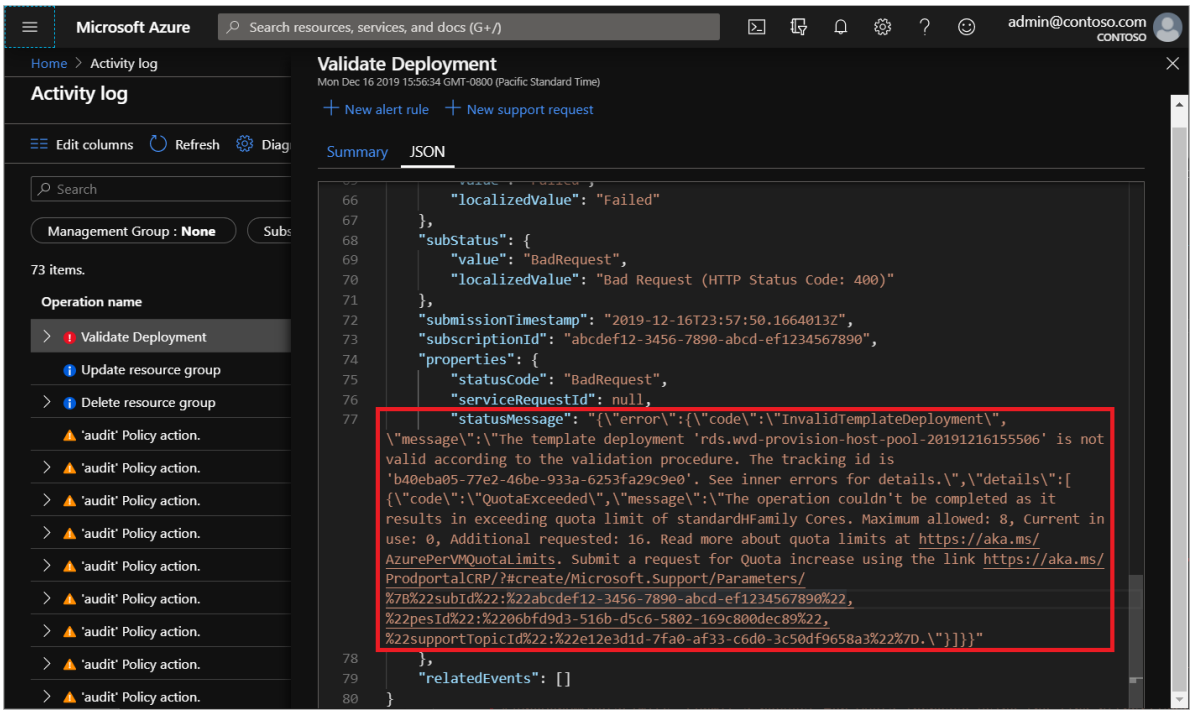
Before taking specific action, you'll need to check the activity log to see the detailed error for the failed deployment validation.

To view the error in the activity log:

1. Exit the current Azure Marketplace deployment offering.
2. In the top search bar, search for and select **Activity Log**.
3. Find an activity named **Validate Deployment** that has a status of **Failed** and select the activity.



4. Select JSON, then scroll down to the bottom of the screen until you see the "statusMessage" field.



If your operation template goes over the quota limit, you can do one of the following things to fix it:

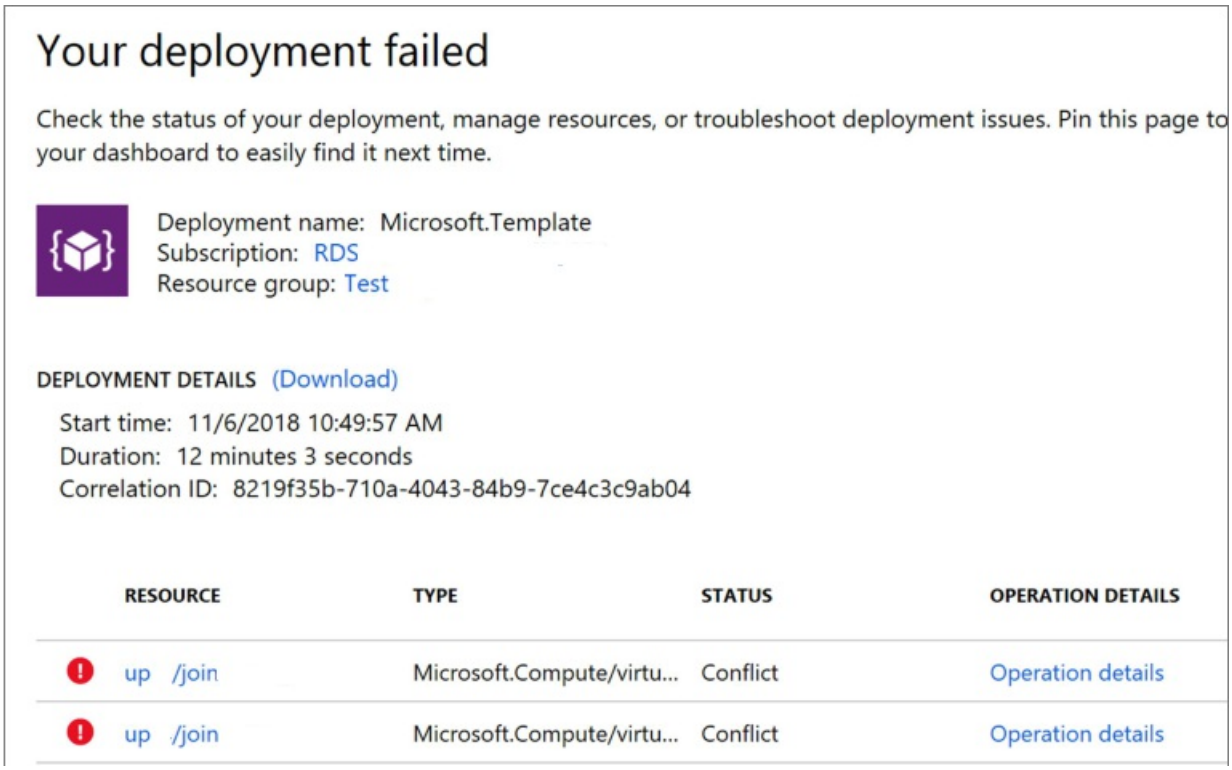
- Run the Azure Marketplace with the parameters you used the first time, but this time use fewer VMs and VM cores.
- Open the link you see in the **statusMessage** field in a browser to submit a request to increase the quota for your Azure subscription for the specified VM SKU.

Azure Resource Manager template and PowerShell Desired State Configuration (DSC) errors

Follow these instructions to troubleshoot unsuccessful deployments of Azure Resource Manager templates and PowerShell DSC.

1. Review errors in the deployment using View deployment operations with Azure Resource Manager.
2. If there are no errors in the deployment, review errors in the activity log using View activity logs to audit actions on resources.
3. Once the error is identified, use the error message and the resources in Troubleshoot common Azure deployment errors with Azure Resource Manager to address the issue.
4. Delete any resources created during the previous deployment and retry deploying the template again.

Error: Your deployment failed....<hostname>/joindomain



Example of raw error:

```
 {"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list
deployment operations for details.
 Please see https://aka.ms/arm-debug for usage details.","details":[{"code":"Conflict","message":"{\r\n
\"status\": \"Failed\",\r\n \"error\":
 {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n \"message\": \"The resource operation completed with
terminal provisioning state 'Failed'.
 \",\r\n \"details\": [\r\n {\r\n \"code\": \"VMExtensionProvisioningError\",\r\n \"message\": \"VM has
reported a failure when processing
 extension 'joindomain'. Error message: \\\"Exception(s) occurred while joining Domain
'diamondsg.onmicrosoft.com'\\\".\"\r\n }\r\n ]\r\n }\r\n}"}]}
```

Cause 1: Credentials provided for joining VMs to the domain are incorrect.

Fix 1: See the "Incorrect credentials" error for VMs are not joined to the domain in Session host VM configuration.

Cause 2: Domain name doesn't resolve.

Fix 2: See Error: Domain name doesn't resolve in Session host VM configuration.

Cause 3: Your virtual network (VNET) DNS configuration is set to **Default**.

To fix this, do the following things:

1. Open the Azure Portal and go to the **Virtual networks** blade.
2. Find your VNET, then select **DNS servers**.
3. The DNS servers menu should appear on the right side of your screen. On that menu, select **Custom**.
4. Make sure the DNS servers listed under Custom match your domain controller or Active Directory domain. If you don't see your DNS server, you can add it by entering its value into the **Add DNS server** field.

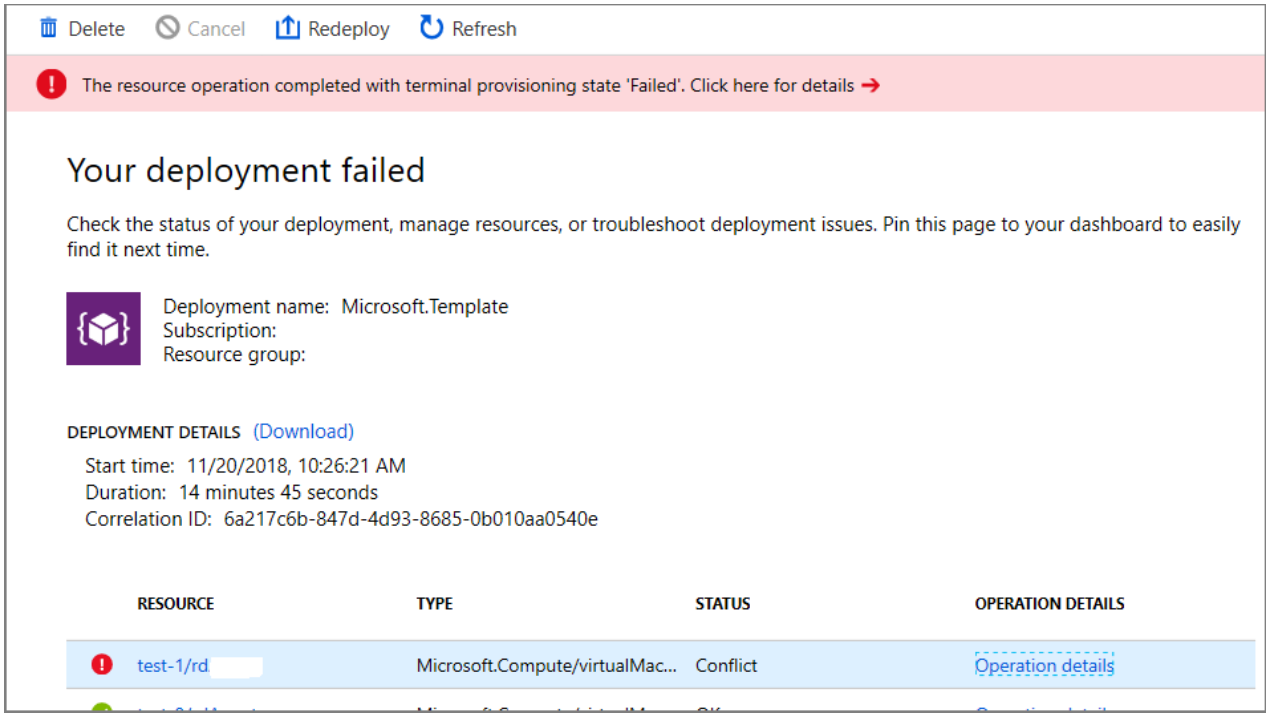
Error: Your deployment failed...\Unauthorized

```
{"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage details.","details":
[{"code":"Unauthorized","message":"{\r\n \"Code\": \"Unauthorized\",\r\n \"Message\": \"The scale operation
is not allowed for this subscription in this region. Try selecting different region or scale option.\",\r\n
\"Target\": null,\r\n \"Details\": [\r\n {\r\n \"Message\": \"The scale operation is not allowed for this
subscription in this region. Try selecting different region or scale option.\"\r\n },\r\n {\r\n \"Code\":
\"Unauthorized\"\r\n },\r\n {\r\n \"ErrorEntity\": {\r\n \"ExtendedCode\": \"52020\",\r\n
\"MessageTemplate\": \"The scale operation is not allowed for this subscription in this region. Try selecting
different region or scale option.\",\r\n \"Parameters\": [\r\n \"default\"\r\n ],\r\n \"Code\":
\"Unauthorized\",\r\n \"Message\": \"The scale operation is not allowed for this subscription in this region.
Try selecting different region or scale option.\"\r\n }\r\n }\r\n ],\r\n \"Innererror\": null\r\n}"}]}
```


Cause: The subscription you're using is a type that can't access required features in the region where the customer is trying to deploy. For example, MSDN, Free, or Education subscriptions can show this error.

Fix: Change your subscription type or region to one that can access the required features.

Error: VMExtensionProvisioningError

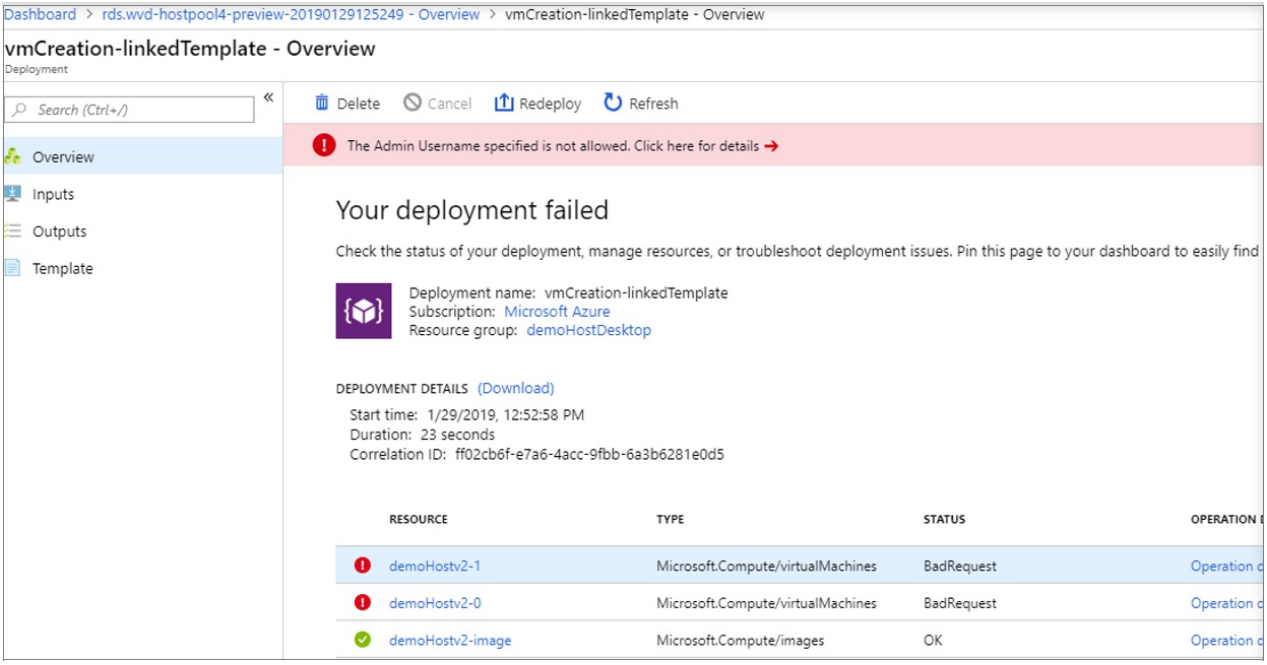


Cause 1: Transient error with the Windows Virtual Desktop environment.

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



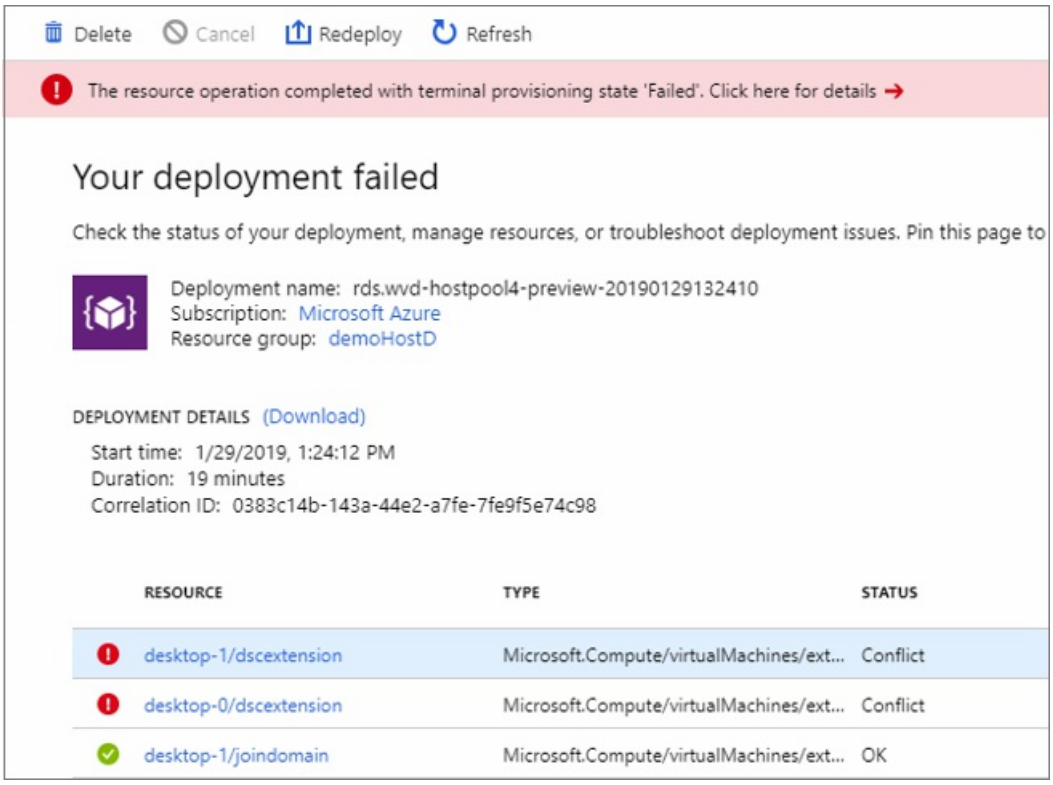
Example of raw error:

```
 { "id": "/subscriptions/EXAMPLE/resourceGroups/demoHostDesktop/providers/Microsoft.
  Resources/deployments/vmCreation-linkedTemplate/operations/EXAMPLE", "operationId": "EXAMPLE",
"properties": { "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" } }, "targetResource": { "id":
"/subscriptions/EXAMPLE
 /resourceGroups/demoHostDesktop/providers/Microsoft.Compute/virtualMachines/demo", "resourceType":
"Microsoft.Compute/virtualMachines", "resourceName": "demo" } }}
```

Cause: Password provided contains forbidden substrings (admin, administrator, root).

Fix: Update username or use different users.

Error: VM has reported a failure when processing extension



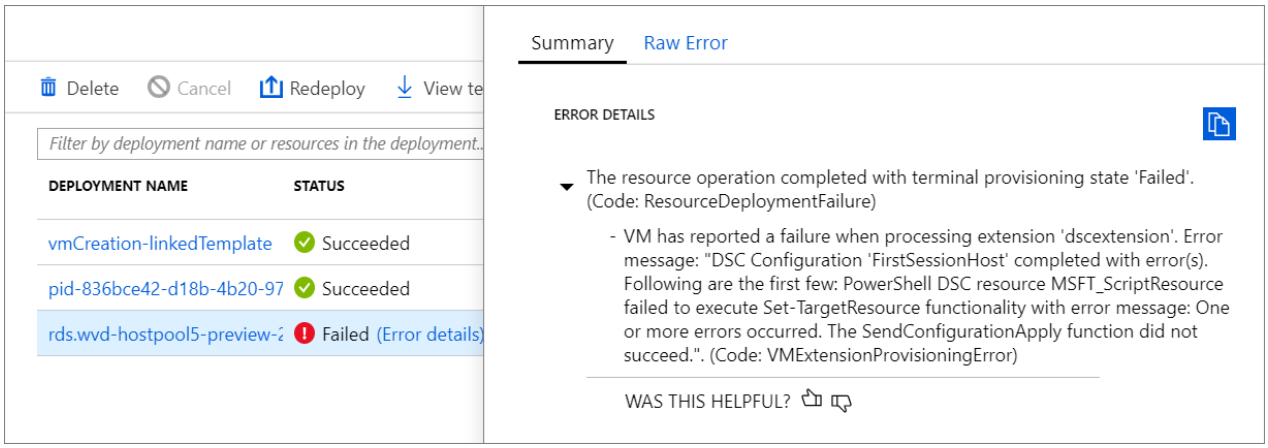
Example of raw error:

```
{ "id": "/subscriptions/EXAMPLE/resourceGroups/demoHostD/providers/Microsoft.Resources/deployments/
 rds.wvd-provision-host-pool-20190129132410/operations/5A0757AC9E7205D2", "operationId": "5A0757AC9E7205D2",
"properties":
 { "provisioningOperation": "Create", "provisioningState": "Failed", "timestamp": "2019-01-
29T21:43:05.1416423Z",
 "duration": "PT7M56.8150879S", "trackingId": "43c4f71f-557c-4abd-80c3-01f545375455", "statusCode":
"Conflict",
 "statusMessage": { "status": "Failed", "error": { "code": "ResourceDeploymentFailure", "message":
 "The resource operation completed with terminal provisioning state 'Failed'.", "details": [ { "code":
 "VMExtensionProvisioningError", "message": "VM has reported a failure when processing extension
'dscextension'.
 Error message: \"DSC Configuration 'SessionHost' completed with error(s). Following are the first few:
 PowerShell DSC resource MSFT_ScriptResource failed to execute Set-TargetResource functionality with error
message:
 One or more errors occurred. The SendConfigurationApply function did not succeed.\"." } ] } },
"targetResource":
 { "id": "/subscriptions/EXAMPLE/resourceGroups/demoHostD/providers/Microsoft.
 Compute/virtualMachines/desktop-1/extensions/dscextension",
 "resourceType": "Microsoft.Compute/virtualMachines/extensions", "resourceName": "desktop-1/dscextension" }
}}
```

Cause: PowerShell DSC extension was not able to get admin access on the VM.

Fix: Confirm username and password have administrative access on the virtual machine and run the Azure Resource Manager template again.

Error: DeploymentFailed – PowerShell DSC Configuration 'FirstSessionHost' completed with Error(s)



Example of raw error:


```
{
    "code": "DeploymentFailed",
   "message": "At least one resource deployment operation failed. Please list
 deployment operations for details. 4 Please see https://aka.ms/arm-debug for usage details.",
 "details": [
        { "code": "Conflict",
        "message": "{\r\n \"status\": \"Failed\",\r\n \"error\": {\r\n \"code\":
        \"ResourceDeploymentFailure\",\r\n \"message\": \"The resource
        operation completed with terminal provisioning state 'Failed'.\",\r\n
        \"details\": [\r\n {\r\n \"code\":
       \"VMExtensionProvisioningError\",\r\n \"message\": \"VM has
             reported a failure when processing extension 'dscextension'.
             Error message: \\\"DSC Configuration 'FirstSessionHost'
             completed with error(s). Following are the first few:
             PowerShell DSC resource MSFT ScriptResource failed to
             execute Set-TargetResource functionality with error message:
             One or more errors occurred. The SendConfigurationApply
             function did not succeed.\\\".\"\r\n }\r\n ]\r\n }\r\n}"  }
```


Cause: PowerShell DSC extension was not able to get admin access on the VM.

Fix: Confirm username and password provided have administrative access on the virtual machine and run the Azure Resource Manager template again.

Error: DeploymentFailed – InvalidResourceReference

Example of raw error:


```
{"code":"DeploymentFailed","message":"At least one resource deployment operation
failed. Please list deployment operations for details. Please see https://aka.ms/arm-
debug for usage details.","details":[{"code":"Conflict","message":"{\r\n \"status\":
\"Failed\",\r\n \"error\": {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n
\"message\": \"The resource operation completed with terminal provisioning state
'Failed'.\",\r\n \"details\": [\r\n {\r\n \"code\": \"DeploymentFailed\",\r\n
\"message\": \"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage
details.\",\r\n \"details\": [\r\n {\r\n \"code\": \"BadRequest\",\r\n \"message\":
\"{\\r\\n \\\"error\\\": {\\r\\n \\\"code\\\": \\\"InvalidResourceReference\\\",\\r\\n
\\\"message\\\": \\\"Resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/virtualNetworks/wvd-vnet/subnets/default
referenced by resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/networkInterfaces/erd. Please make sure that
the referenced resource exists, and that both resources are in the same
region.\\\",\\r\\n\\\"details\\\": []\\r\\n }\\r\\n}\"\r\n }\r\n ]\r\n }\r\n ]\r\n }\r\n}"}]}
```


Cause: Part of the resource group name is used for certain resources being created by the template. Due to the

name matching existing resources, the template may select an existing resource from a different group.

Fix: When running the Azure Resource Manager template to deploy session host VMs, make the first two characters unique for your subscription resource group name.

Error: DeploymentFailed – InvalidResourceReference

Example of raw error:

```
{"code":"DeploymentFailed","message":"At least one resource deployment operation
failed. Please list deployment operations for details. Please see https://aka.ms/arm-
debug for usage details.","details":[{"code":"Conflict","message":"{\r\n \"status\":
\"Failed\",\r\n \"error\": {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n
\"message\": \"The resource operation completed with terminal provisioning state
'Failed'.\",\r\n \"details\": [\r\n {\r\n \"code\": \"DeploymentFailed\",\r\n
\"message\": \"At least one resource deployment operation failed. Please list
deployment operations for details. Please see https://aka.ms/arm-debug for usage
details.\",\r\n \"details\": [\r\n {\r\n \"code\": \"BadRequest\",\r\n \"message\":
\"{\\r\\n \\\"error\\\": {\\r\\n \\\"code\\\": \\\"InvalidResourceReference\\\",\\r\\n
\\\"message\\\": \\\"Resource /subscriptions/EXAMPLE/resourceGroups/ernani-wvd-
demo/providers/Microsoft.Network/virtualNetworks/wvd-vnet/subnets/default
referenced by resource
/subscriptions/EXAMPLE/resourceGroups/DEMO/providers/Microsoft.Network/networkInterfaces
/EXAMPLE was not found. Please make sure that the referenced resource exists, and that both
resources are in the same region.\\\",\\r\\n \\\"details\\\": []\\r\\n }\\r\\n}\"\r\n
}\r\n ]\r\n }\r\n ]\r\n }\r\n\
```

Cause: This error is because the NIC created with the Azure Resource Manager template has the same name as another NIC already in the VNET.

Fix: Use a different host prefix.

Error: DeploymentFailed – Error downloading

Example of raw error:

```
\\\"The DSC Extension failed to execute: Error downloading
https://catalogartifact.azureedge.net/publicartifacts/rds.wvd-provision-host-pool-
2dec7a4d-006c-4cc0-965a-02bbe438d6ff-prod
/Artifacts/DSC/Configuration.zip after 29 attempts: The remote name could not be
resolved: 'catalogartifact.azureedge.net'.\\nMore information about the failure can
be found in the logs located under
'C:\\\\WindowsAzure\\\\Logs\\\\Plugins\\\\Microsoft.Powershell.DSC\\\\2.77.0.0' on
the VM.\\\"
```

Cause: This error is due to a static route, firewall rule, or NSG blocking the download of the zip file tied to the Azure Resource Manager template.

Fix: Remove blocking static route, firewall rule, or NSG. Optionally, open the Azure Resource Manager template json file in a text editor, take the link to zip file, and download the resource to an allowed location.

Error: The user isn't authorized to query the management service

Example of raw error:

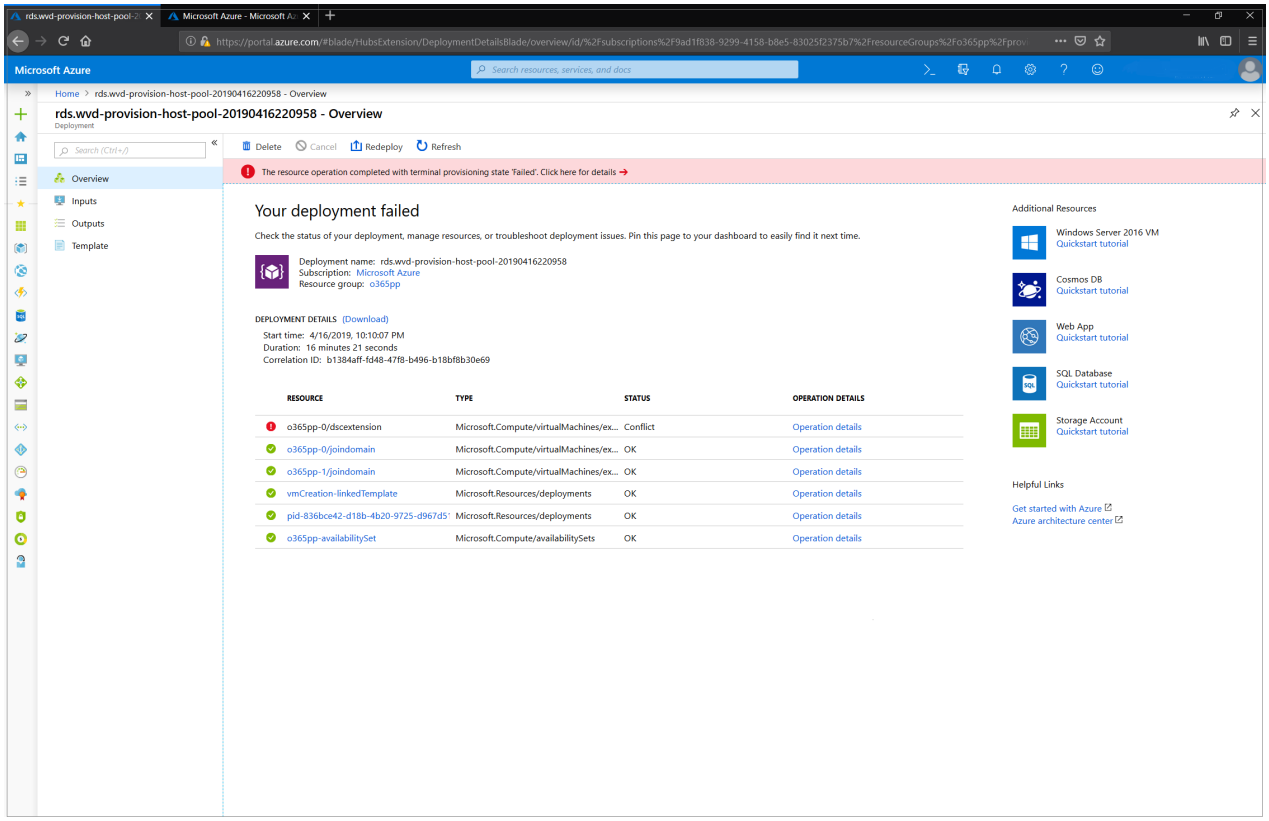
```
"response": { "content": { "startTime": "2019-04-01T17:45:33.3454563+00:00", "endTime": "2019-04-
01T17:48:52.4392099+00:00",
"status": "Failed", "error": { "code": "VMExtensionProvisioningError", "message": "VM has reported a failure
when processing
extension 'dscextension'. Error message: \"DSC Configuration 'FirstSessionHost' completed with error(s).
Following are the first few: PowerShell DSC resource MSFT_ScriptResource failed to execute Set-TargetResource
 functionality with error message: User is not authorized to query the management service.
\nActivityId: 1b4f2b37-59e9-411e-9d95-4f7ccd481233\nPowershell commands to diagnose the failure:
\nGet-RdsDiagnosticActivities -ActivityId 1b4f2b37-59e9-411e-9d95-4f7ccd481233\n
The SendConfigurationApply function did not succeed.\"." }, "name": "2c3272ec-d25b-47e5-8d70-a7493e9dc473" }
} }}
```

Cause: The specified Windows Virtual Desktop tenant admin doesn't have a valid role assignment.

Fix: The user who created the Windows Virtual Desktop tenant needs to sign in to Windows Virtual Desktop PowerShell and assign the attempted user a role assignment. If you're running the GitHub Azure Resource Manager template parameters, follow these instructions using PowerShell commands:

```
Add-RdsAccount -DeploymentUrl "https://rdbroker.wvd.microsoft.com"
New-RdsRoleAssignment -TenantName <Windows Virtual Desktop tenant name> -RoleDefinitionName "RDS Contributor"
-SignInName <UPN>
```

Error: User requires Azure Multi-Factor Authentication (MFA)



Example of raw error:

```
"message": "{\r\n  \"status\": \"Failed\",\r\n  \"error\": {\r\n    \"code\":
\"ResourceDeploymentFailure\",\r\n    \"message\": \"The resource operation completed with terminal
provisioning state 'Failed'.\",\r\n    \"details\": [\r\n      {\r\n        \"code\":
\"VMExtensionProvisioningError\",\r\n        \"message\": \"VM has reported a failure when processing
extension 'dscextension'. Error message: \\\"DSC Configuration 'FirstSessionHost' completed with error(s).
Following are the first few: PowerShell DSC resource MSFT_ScriptResource  failed to execute Set-
TargetResource functionality with error message: One or more errors occurred.  The SendConfigurationApply
function did not succeed.\\\".\"\r\n      }\r\n    ]\r\n  }\r\n}"
```

Cause: The specified Windows Virtual Desktop tenant admin requires Azure Multi-Factor Authentication (MFA) to sign in.

Fix: Create a service principal and assign it a role for your Windows Virtual Desktop tenant by following the steps in Tutorial: Create service principals and role assignments with PowerShell. After verifying that you can sign in to Windows Virtual Desktop with the service principal, rerun the Azure Marketplace offering or the GitHub Azure Resource Manager template, depending on which method you're using. Follow the instructions below to enter the correct parameters for your method.

If you're running the Azure Marketplace offering, provide values for the following parameters to properly authenticate to Windows Virtual Desktop:

- Windows Virtual Desktop tenant RDS Owner: Service principal
- Application ID: The application identification of the new service principal you created
- Password/Confirm Password: The password secret you generated for the service principal
- Azure AD Tenant ID: The Azure AD Tenant ID of the service principal you created

If you're running the GitHub Azure Resource Manager template, provide values for the following parameters to properly authenticate to Windows Virtual Desktop:

- Tenant Admin user principal name (UPN) or Application ID: The application identification of the new service principal you created
- Tenant Admin Password: The password secret you generated for the service principal
- IsServicePrincipal: **true**
- AadTenantId: The Azure AD Tenant ID of the service principal you created

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

2/14/2020 • 12 minutes to read • Edit Online

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 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 command line on 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 Resource Manager **Create and provision Windows Virtual Desktop host pool** 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 that the tenant name is accurate and the tenant exists in Windows Virtual Desktop.
- Confirm the account has at least RDS Contributor permissions.

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 tenant to create

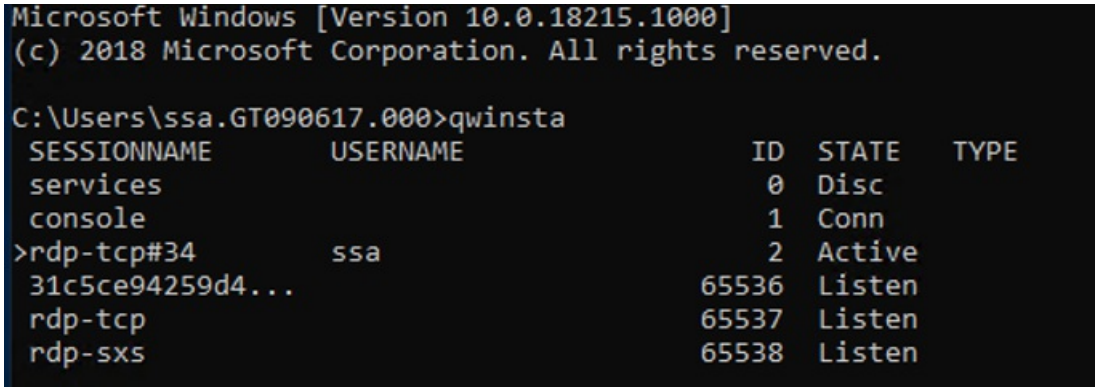
host pools.

- 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 applicable to the Windows Virtual Desktop Agent and the token.

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



Cause: The agent isn't able to update itself to a new version.

Fix: Follow these instructions to manually update the agent.

1. Download a new version of the agent on the session host VM.
2. Launch Task Manager and, in the Service Tab, stop the RDAgentBootLoader service.
3. Run the installer for the new version of the 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 or has been generated with expiration value of 999999.

Fix: Follow these instructions to fix the agent registry error.

1. If there's already a registration token, remove it with Remove-RDSRegistrationInfo.
2. Generate new token with Rds-NewRegistrationInfo.
3. Confirm that the -ExpriationHours parameter is set to 72 (max value is 99999).

Error: Windows Virtual Desktop agent isn't reporting a heartbeat when running Get-RdsSessionHost

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

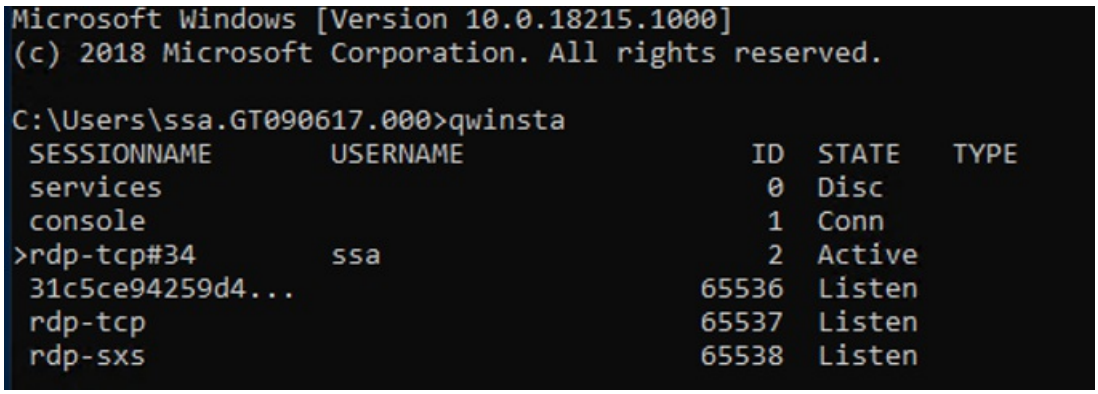
The Windows Virtual Desktop side-by-side stack is automatically installed with Windows Server 2019. Use Microsoft Installer (MSI) to install the side-by-side stack on Microsoft Windows Server 2016 or Windows Server 2012 R2. For Microsoft Windows 10, the Windows Virtual Desktop side-by-side stack is enabled 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 Resource Manager **Create and provision new Windows Virtual Desktop host pool** 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.

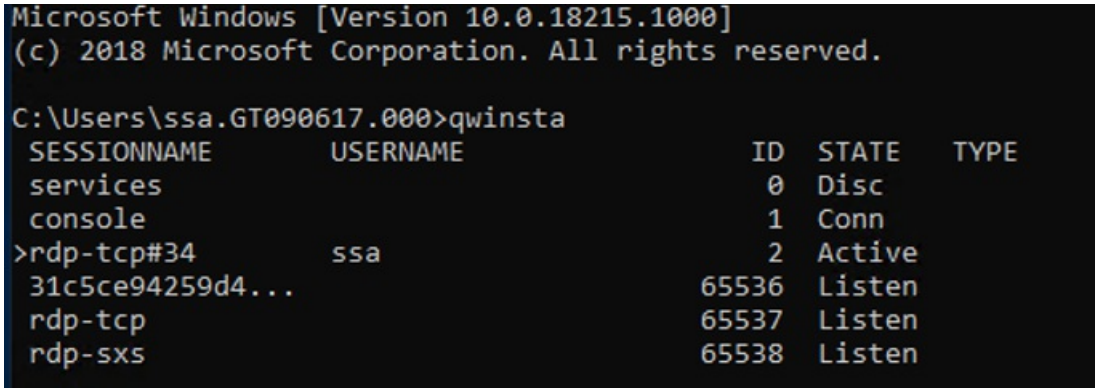


Examine the registry entries listed below and confirm that their values match. If registry keys are missing or values are mismatched, follow the instructions in Create a host pool with PowerShell on how to reinstall the side-

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



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. Download and import The Windows Virtual Desktop PowerShell module to use in your PowerShell session if you haven't already, then run this cmdlet to sign in to your account:

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

3. 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:

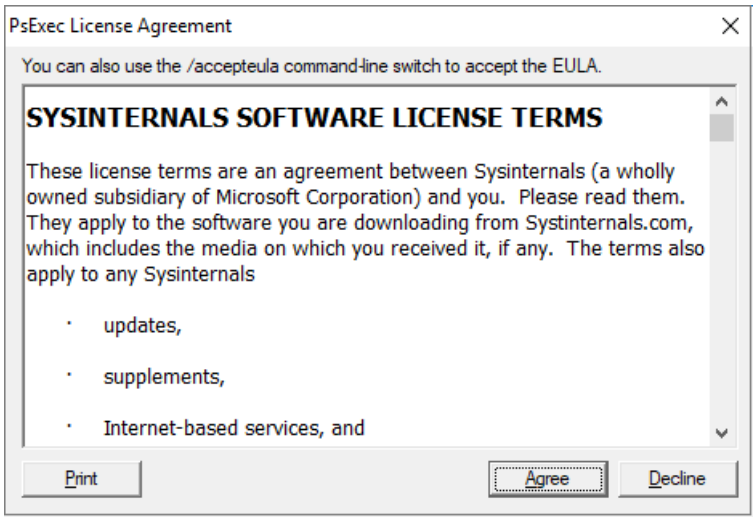
1. Connect with standard Remote Desktop Protocol (RDP) to the VM from where fix will be applied.
2. Download PsExec from https://docs.microsoft.com/sysinternals/downloads/psexec.

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
```

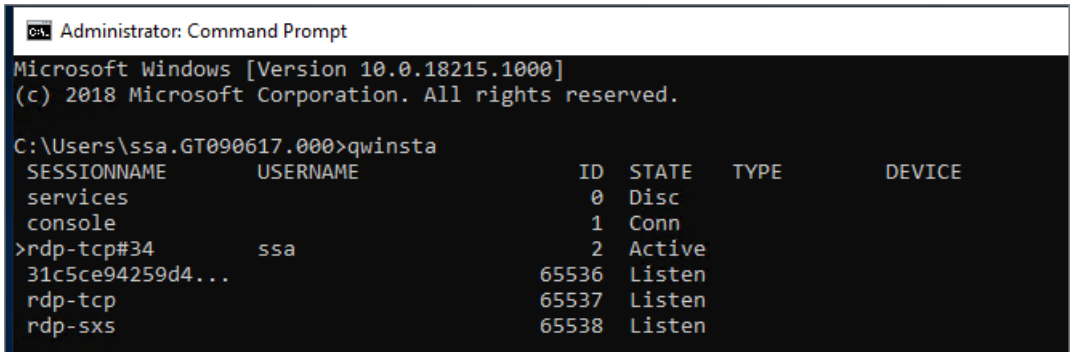
> **NOTE**
> VMname is the machine name of the VM with the malfunctioning side-by-side stack.

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

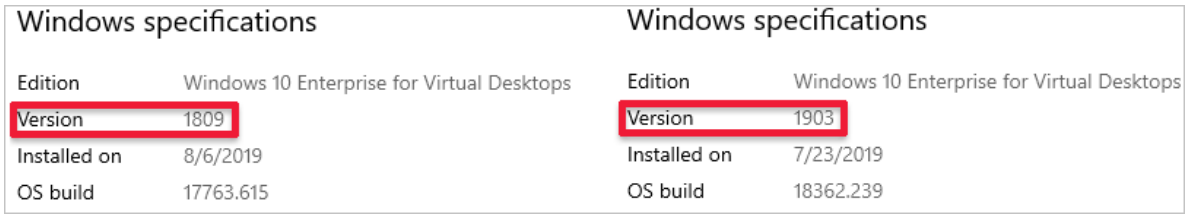
Check the group policy setting by opening the Group Policy Editor in the VM and navigating to **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Licensing** > **Set the Remote Desktop licensing mode**. If the group policy setting 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.



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.

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 tenant and host pool in a Windows Virtual Desktop environment, see Tenant 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.

Troubleshoot Windows 7 virtual machines in Windows Virtual Desktop

11/4/2019 • 2 minutes to read • Edit Online

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

Known issues

Windows 7 on Windows Virtual Desktops doesn't support the following features:

- Virtualized applications (RemoteApps)
- Time zone redirection
- Automatic DPI scaling

Windows Virtual Desktop can only virtualize full desktops for Windows 7.

While Automatic DPI scaling isn't supported, you can manually change the resolution on your virtual machine by right-clicking the icon in the Remote Desktop client and selecting **Resolution**.

Error: Can't access the Remote Desktop User group

If Windows Virtual Desktop can't find you or your users' credentials in the Remote Desktop User group, you may see one of the following error messages:

- "This user is not a member of the Remote Desktop User group"
- "You must be granted permissions to sign in through Remote Desktop Services"

To fix this error, add the user to the Remote Desktop User group:

1. Open the Azure portal.
2. Select the virtual machine you saw the error message on.
3. Select **Run a command**.
4. Run the following command with `<username>` replaced by the name of the user you want to add:

```
net localgroup "Remote Desktop Users" <username> /add
```

Troubleshoot the Windows Virtual Desktop management tool

11/10/2019 • 2 minutes to read • Edit Online

This article describes issues that can occur while deploying the Windows Virtual Desktop management tool and how to fix them.

Error: Management tool services configured but automated setup fails

When you successfully set up services for the management tool but automated setup fails, you'll see this error message:

```
{"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 }\r\n}"}]}
```

This usually means one of the following two things:

- The user has owner permissions on their subscription and global admin at tenant level, but they can't sign in to Azure.
- The user's account settings have multi-factor authentication enabled.

To fix this:

1. Make sure the user you created for the Azure Active Directory User Principal Name has the "Contributor" subscription level.
2. Sign in to <portal.azure.com> with the UPN account to check the account settings and make sure multi-factor authentication isn't on. If it's turned on, turn it off.
3. Visit the Windows Virtual Desktop Consent page and make sure the server and client apps have consent.
4. Review the Deploy a management tool tutorial if the issue continues and redeploy the tool.

Error: Job with specified ID already exists

If your user sees the error message "Job with specified ID already exists," it's because they didn't provide a unique name in the "Application name" parameter when deploying the template.

To fix this, redeploy the management tool with the "Application name" parameter filled.

Delayed consent prompt when opening management tool

When you deploy the management tool, the consent prompt might not open right away. This means the Azure Web app service is taking longer than usual to load. The prompt should appear after Azure Web is done loading.

The user can't deploy the management tool in the East US region

If a customer sets the region to East US, they can't deploy the management tool.

To fix this, deploy the management tool in a different region. Redeploying the tool in a different region shouldn't affect user experience.

Next steps

- Learn about escalation tracks at Troubleshooting overview, feedback, and support.
- Learn how to report issues with Windows Virtual Desktop tools at ARM Templates for Remote Desktop Services.
- For an overview on troubleshooting Windows Virtual Desktop and the escalation tracks, see Troubleshooting overview, feedback, and support.
- To learn how to deploy the management tool, see Deploy a management tool.

Windows Virtual Desktop service connections

1/14/2020 • 2 minutes to read • Edit Online

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.

Confirm that the user reporting the issues has been assigned to application groups by using this command line:

```
Get-RdsAppGroupUser <tenantname> <hostpoolname> <appgroupname>
```

Confirm that the user is logging in with the correct credentials.

If the web client is being used, confirm that there are no cached credentials issues.

Windows 10 Enterprise multi-session virtual machines don't respond

If a virtual machine isn't responsive and you can't access it through RDP, you'll need to troubleshoot it with the diagnostics feature by checking the host status.

To check the host status, run this cmdlet:

```
Get-RdsSessionHost -TenantName $TenantName -HostPoolName $HostPool | ft SessionHostName, LastHeartBeat,
AllowNewSession, Status
```

If the host status is `NoHeartBeat`, that means the VM isn't responding and the agent can't communicate with the Windows Virtual Desktop service.

```
SessionHostName          LastHeartBeat     AllowNewSession    Status
---------------          -------------     ---------------    ------
WVDHost1.contoso.com     21-Nov-19 5:21:35            True  Available
WVDHost2.contoso.com     21-Nov-19 5:21:35            True  Available
WVDHost3.contoso.com     21-Nov-19 5:21:35            True  NoHeartBeat
WVDHost4.contoso.com     21-Nov-19 5:21:35            True  NoHeartBeat
WVDHost5.contoso.com     21-Nov-19 5:21:35            True  NoHeartBeat
```

There are a few things you can do to fix the NoHeartBeat status.

Update FSLogix

If your FSLogix isn't up to date, especially if it's version 2.9.7205.27375 of frxdrvvt.sys, it could cause a deadlock. Make sure to update FSLogix to the latest version.

Disable BgTaskRegistrationMaintenanceTask

If updating FSLogix doesn't work, the issue might be that a BiSrv component is exhausting system resources during a weekly maintenance task. Temporarily disable the maintenance task by disabling the BgTaskRegistrationMaintenanceTask with one of these two methods:

- Go to the Start menu and search for **Task Scheduler**. Navigate to **Task Scheduler Library** > **Microsoft** > **Windows** > **BrokerInfrastructure**. Look for a task named **BgTaskRegistrationMaintenanceTask**. When you find it, right-click it and select **Disable** from the drop-down menu.
- Open a command-line menu as administrator and run the following command:

```
schtasks /change /tn "\Microsoft\Windows\BrokerInfrastructure\BgTaskRegistrationMaintenanceTask"
/disable
```

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 tenant and host pool in a Windows Virtual Desktop environment, see Tenant 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

1/14/2020 • 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

Use the following PowerShell cmdlets to clean up out-of-band (OOB) client registries.

```
Remove-ItemProperty 'HKCU:\Software\Microsoft\Terminal Server Client\Default' - Name FeedURLs

#Remove RdClientRadc registry key
Remove-Item 'HKCU:\Software\Microsoft\RdClientRadc' -Recurse

#Remove all files under %appdata%\RdClientRadc
Remove-Item C:\Users\pavithir\AppData\Roaming\RdClientRadc\* -Recurse
```

Navigate to **%AppData%\RdClientRadc** and delete all content.

Uninstall and reinstall Remote Desktop client for Windows 7 and Windows 10.

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.

Opening another site fails

This is usually caused by network connection problems or a network outage. We recommend you contact network support.

Nslookup cannot resolve the name

This is usually caused by 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.

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 details, see How to delete cookie files in Internet Explorer.
4. Clear browser cache. For more details, 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 tenant and host pool in a Windows Virtual Desktop environment, see Tenant 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

2/14/2020 • 3 minutes to read • Edit Online

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: Add-RdsAppGroupUser command -- The specified UserPrincipalName is already assigned to a RemoteApp app group in the specified Host Pool

```
Add-RdsAppGroupUser -TenantName <TenantName> -HostPoolName <HostPoolName> -AppGroupName 'Desktop Application
Group' -UserPrincipalName <UserName>
```

Cause: The username used has been already assigned to an app group of a different type. Users can't be assigned to both a remote desktop and remote app group under the same session host pool.

Fix: If user needs both remote apps and remote desktop, create different host pools or grant user access to the remote desktop, which will permit the use of any application on the session host VM.

Error: Add-RdsAppGroupUser command -- The specified UserPrincipalName doesn't exist in the Azure Active Directory associated with the Remote Desktop tenant

```
Add-RdsAppGroupUser -TenantName <TenantName> -HostPoolName <HostPoolName> -AppGroupName "Desktop Application
Group" -UserPrincipalName <UserPrincipalName>
```

Cause: The user specified by the -UserPrincipalName cannot be found in the Azure Active Directory tied to the Windows Virtual Desktop tenant.

Fix: Confirm the items in the following list.

- The user is synched to Azure Active Directory.
- The user isn't tied to business to consumer (B2C) or business-to-business (B2B) commerce.
- The Windows Virtual Desktop tenant is tied to correct Azure Active Directory.

Error: Get-RdsDiagnosticActivities -- User isn't authorized to query the management service

```
Get-RdsDiagnosticActivities -ActivityId <ActivityId>
```

Cause: -TenantName parameter

Fix: Issue Get-RdsDiagnosticActivities with -TenantName <TenantName>.

Error: Get-RdsDiagnosticActivities -- the user isn't authorized to query the management service

```
Get-RdsDiagnosticActivities -Deployment -username <username>
```

Cause: Using -Deployment switch.

Fix: -Deployment switch can be used only by deployment administrators. These administrators are usually members of the Remote Desktop Services/Windows Virtual Desktop team. Replace the -Deployment switch with -TenantName <TenantName>.

Error: New-RdsRoleAssignment -- the user isn't authorized to query the management service

Cause 1: The account being used doesn't have Remote Desktop Services Owner permissions on the tenant.

Fix 1: A user with Remote Desktop Services owner permissions needs to execute the role assignment.

Cause 2: The account being used has Remote Desktop Services owner permissions but isn't part of the tenant'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.

> **NOTE**
> New-RdsRoleAssignment cannot give permissions to a user that doesn't exist in the Azure Active Directory (AD).

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 tenant and host pool in a Windows Virtual Desktop environment, see Tenant 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 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.

Diagnose graphics performance issues in Remote Desktop

2/14/2020 • 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.

Linux support

2/27/2020 • 2 minutes to read • Edit Online

You can use the Linux SDK for Windows Virtual Desktop to build a standalone Windows Virtual Desktop client. You can also use it to enable Windows Virtual Desktop support on your client application. This quick guide will explain what the Linux SDK is and how to start using it.

What is the Linux SDK?

You can use the 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.

> **NOTE**
> The SDK is currently in development. We'll update this document with instructions to access the SDK when it is available.

Supported Linux distributions

The SDK is compatible with most operating systems based on Ubuntu 18.04 or later. If you have a different Linux distribution, we can work with you to figure out how to best support your needs.

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.

Get started with the Linux SDK

Before you can develop a Linux client for Windows Virtual Desktop, you need to do the following things:

1. Build and deploy a Windows Virtual Desktop environment for testing or production use.
2. Test the available first-party clients to familiarize yourself with the Windows Virtual Desktop user experience.

Next steps

Check out our documentation for the following clients:

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