Microsoft Cloud Workshop

Windows Virtual Desktop on Azure Lab

Instructor Guide - Hands-on lab

October 2019

Contents

Windows Virtual Desktops Airlift Trainers Guide

Abstract

Understand the Windows Virtual Desktop Infrastructure and how-to setup a working WVD environment end-to-end in a typical Enterprise model.

- Deploy an AD Infrastructure in Azure
- Deploy a Windows Virtual Desktop Environment
- Publish Windows Virtual Desktop(s)
- Publish remoteapps
- Configure User Profiles with FSLogix
- Configure Monitoring and Security
- Understand Master Images

Overview

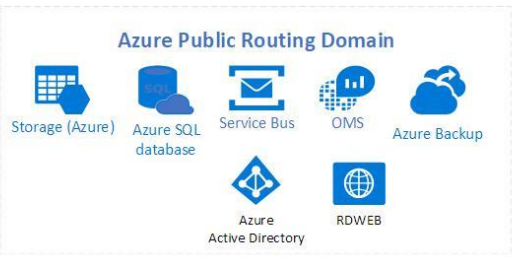
In this lab, attendees will deploy the Windows Virtual Desktop (WVD) solution . Exclusively available as an Azure cloud service, Windows Virtual Desktop allows you to choose a flexible end user virtualized application or desktop delivery model that best aligns with your enterprise Azure cloud strategy. WVD simplifies the IT model to virtualize and deploy modern and legacy desktop app experiences with unified management—without needing to host, install, configure and manage components such as diagnostics, networking, connection brokering, and gateway. WVD brings together Microsoft Office 365 and Azure to provide users with the only multi-session Windows 10 experience with exceptional scale and reduced IT costs while empowering today's modern digital workspace.

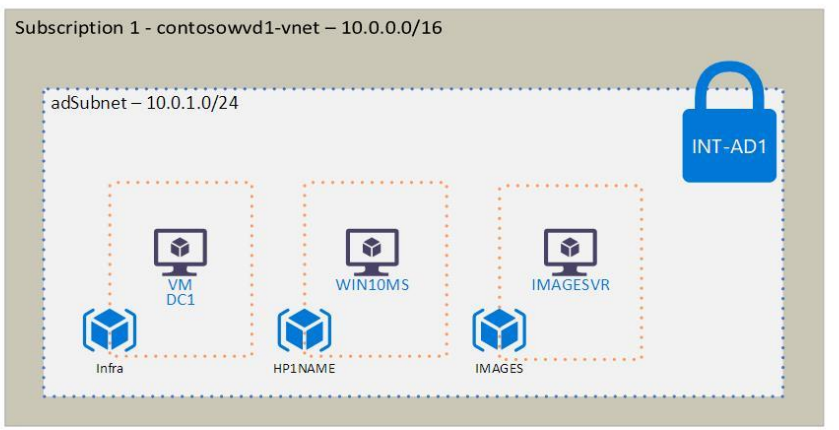
Helpful Hints for this Lab

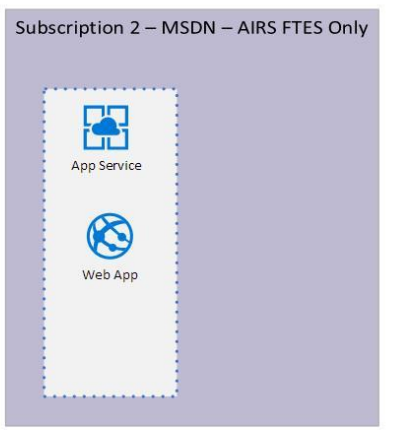
- Take your time, there is a lot of information to digest and go through
- Make sure to keep track of what user accounts you are using and where you are using them
- Regions and locations, make sure to stay consistent as much as possible
- Terminology is important

Solution architecture

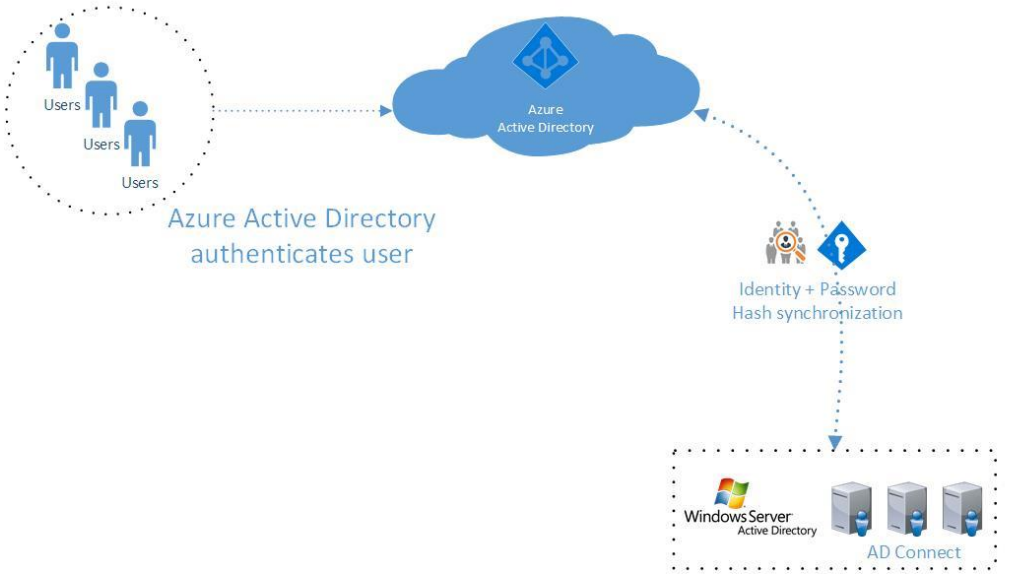
At the end of the lab, attendees will have deployed an Azure Active Directory Tenant, Azure infrastructure, Active Directory, Active Directory Domain Controller using an ARM template from GitHub, Windows Virtual Desktop Tenant(s), Host Pool(s) and session host(s).







Most customers will have only a single subscription where everything falls into. MS FTEs we need to have another off shoot because of internal permissions.



Terminology

WVD – Windows Virtual Desktop

GA – Azure Active Directory Global Administrator

ARM – Azure Resource Manager

Sub – Subscription

AAD – Azure Active Directory **"ilovewvd.onmicrosoft.com"**

AD – Active Directory **"ilovewvd.com"**

Tenant – Azure AD Tenant Name **"ilovewvd.onmicrosoft.com"**

Custom Domain – Your purchased domain from a registrar such as Go-Daddy **"ilovewvd.com"**

AAD Domain – Azure Active Directory **"ilovewvd.onmicrosoft.com"**

Domain Admin – custom domain \ AD account that has rights to add servers to your domain etc.. See below

Accounts needed for deployment of the entire WVD solution

Azure AD Global Admin	• Grant Azure Active Directory permissions to the Windows Virtual Desktop service • Assign the Tenant Creator Application role to a user in your Azure Active Directory • submit WVD applications (there are two server and client) • Consent to WVD applications
Active Directory Domain/Account Admin	• Domain Join WVD Session Hosts • Needed to create security groups • Needed to create test users • Needed to add production users to security groups – although this is irrelevant at this point as WVD does not support groups in anyway – this is on the roadmap
WVD Tenant Creator	• This is an AAD account that DOES NOT have MFA enabled that will create the WVD resources in a subscription. This account should have contributor role access to the subscription hosting the WVD resources. • Creates WVD tenant • Creates WVD host pools • Creates WVD Application Groups – These last two tasks could also be done by another account that has RDS permissions but is also NOT a tenant creator – and is arguably best practices (New-RdsRoleAssignment -RoleDefinitionName "RDS Owner")
WVD RDS Owner	• Creates WVD tenant • Creates WVD host pools • Creates WVD Application Groups • Assigns Users • All WVD Administrative duties

Windows Virtual Desktop Requirements

The following outlines the prerequisites needed to successfully deploy and access a Windows Virtual Desktop environment. It is important for everyone attending this training to complete this pre-work prior to arriving to the class. You should have prior knowledge and experience using Azure Services as well as understanding the fundamentals of Active Directory, Authentication, Infrastructure, PowerShell and networking.

- **Personal Laptop** eg Microsoft Surface device. If you are using a non-windows endpoint device, you may also wish to provision a Windows VM in Azure before the class. This Windows VM will need to be remotely accessible to use with PowerShell.
    - Updated Azure PowerShell modules installed on your Windows laptop or Windows VM in Azure
    - WVD PowerShell Cmdlets installed on your Windows laptop or Windows VM in Azure
- **Azure Subscription**
    - Enough Quota Cores to build 4 - 4 core servers
    - MSA Account for example: bobsmith@gmail.com, johnsmith@live.com You will need to have access to the email of these accounts for verification purposes
    - Access to the Azure AD Global Admin account for your new or existing Azure AD Tenant
    - Owner rights on all Azure Subscription(s)

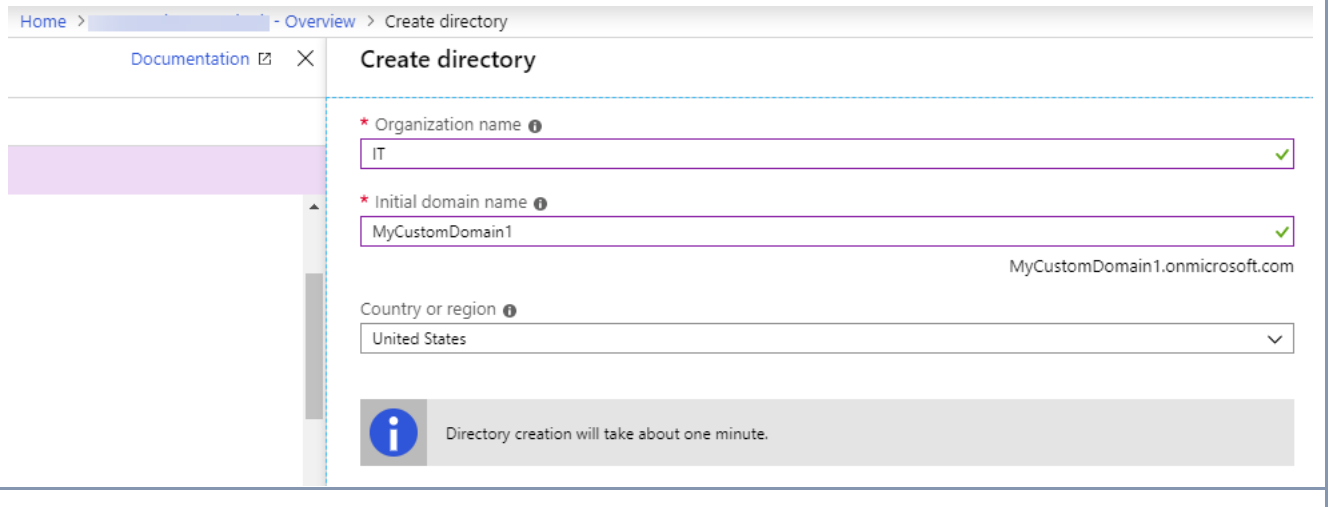
- **Custom Domain Name** i.e. www.MyADDomain.com registered via a registrar like www.godaddy.com you own or have access to the registrar for DNS record edits. This domain is used for creating your AD Infrastructure in Azure and will be used by WVD for permissions to applications and desktop publishing. **All DNS records should be created prior to the lab to ensure proper time for replication.**

Exercise 1: AAD tenant in Azure Subscription

A new Azure AD tenant should be created that matches your custom domain from your DNS registrar.

Task 1: Create AAD tenant

1.	From your Azure subscription navigate to **Azure Active Directory->Create Directory**

2.	Create your new AAD tenant. Click **Create**. ***Note: Do not use more than 14 characters for the Initial domain name.** 
3.	Once your AAD tenant is created, go ahead and create a new Azure AD user (JOHNSMITH@MyCustomDomain1.ONMICROSOFT.COM) and add the Global Admin role to this account. You will eventually add this account to the WVD RDS Owners group later in this Lab.

Exercise 2: Custom domain to Azure Active Directory

Before starting this step, you must have a paid for public pre-registered Custom domain. In addition, you must have full control to add dns entries for the domain. This can be done at any DNS provider for example godaddy.com.

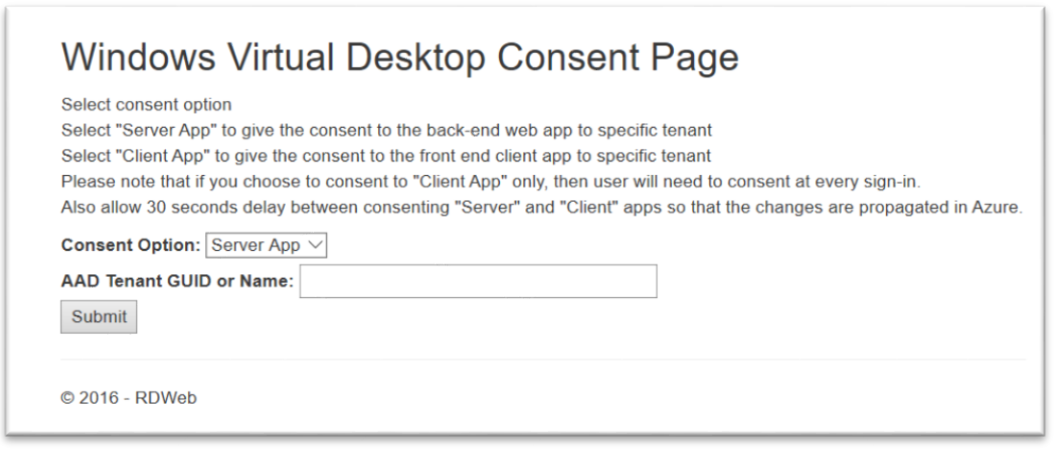
Task 1: Connecting the custom domain in Azure

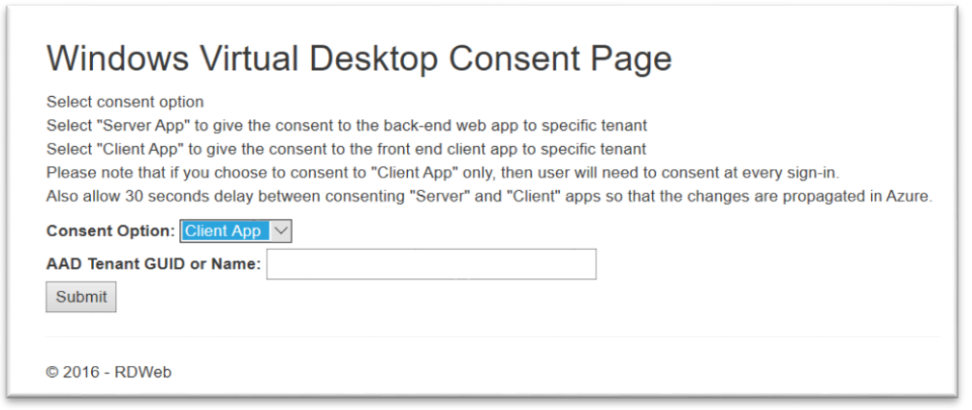
1.	Open the AAD portal and navigate to **custom domain** names section.
2.	Click **+Add Custom Domain Name.**
3.	**Copy** the TXT record. Connect to to your custom domain registrar to create the TXT record and then click Verify in the AAD Portal. Please refer to the link below if needed. https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/add-custom-domain#add-your-custom-domain-name-to-azure-ad
4.	From the Microsoft Azure portal, navigate to the subscription where the WVD machines will be deployed to. **Add** a **new IAM user** (Your Global Admin Account from the tenant created above) to your subscription as an owner. Make sure to check the email of the Global Admin account as you will need to validate this action.
5.	Locate and **copy** the **AAD ID** as you will need this ID in the next step.

Exercise 3: Grant consent for WVD service

Creating a tenant in Windows Virtual Desktop Preview is the first step toward building your desktop virtualization solution. A tenant is a group of one or more host pools. With a tenant, you can build host pools, create app groups, assign users, and make connections through the service.

Task 1: Enabling WVD consent to Azure AD

1.	Browse to the URL: https://rdweb.wvd.microsoft.com. Login with the Global Admin Account for your new AAD tenant.
2.	There are two steps that need to take place. First select from the dropdown, **server** and then paste your AAD ID in the field box.  Click **Submit**. Login with the Azure AD Global Admin Account from your new tenant you created earlier.
3.	Click **Accept** on the permissions request page
4.	WAIT One Minute before moving to next step
5.	Navigate back to https://rdweb.wvd.microsoft.com

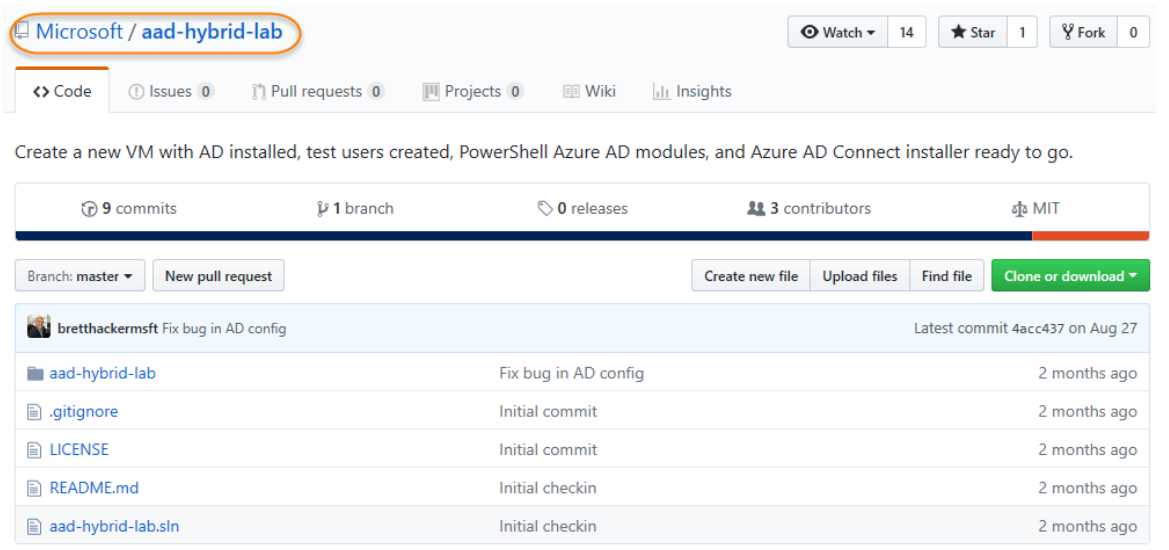
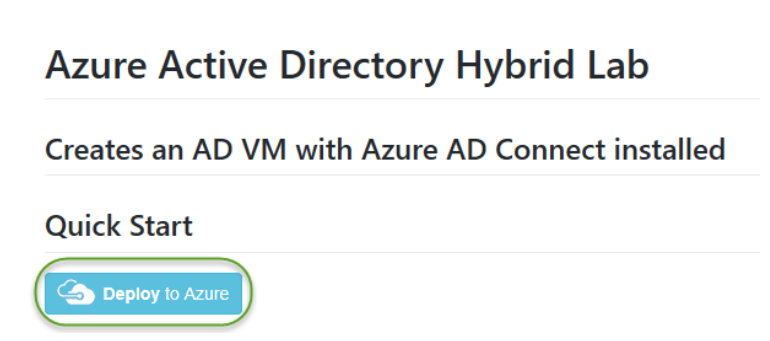
6.	Next, From the drop down select **Client App** from the drop-down box and add the same AAD ID again then **submit**. 
7.	Login with the same Azure AD Global Admin Account you created earlier
8.	Click **Accept** on the permissions request page. Once this is complete you can close out of your browser session.

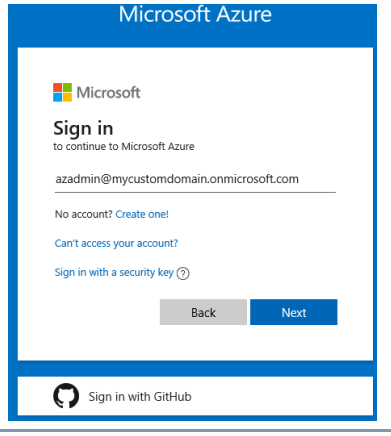
Exercise 4: Deploying Azure infrastructure and AD

In this exercise, attendees will leverage their Microsoft Azure Subscription account. In this subscription account, atendees will deploy a [AAD hybrid lab ARM template from GitHub] that will provision the following:

- Virtual Network
- 1 subnet
- 1 Network Security Groups
    - AD - permits AD traffic, RDP incoming to network; limits DMZ access.
    - **Note**: Bastion service can applied later on to remove the PIP and provide better security.
- Public IP Address
- Active Directory VM
- DSC installs AD
- The Azure vNet is updated with a custom DNS entry pointing to the DC
- Test users are created in the local AD by passing in an array. There is an array sample set as the default value in the deployment template.
- Azure Active Directory Connect is installed and available to configure.

Task 1: Deploying the ARM template into Azure subscription

1.	***Tip:*** Because internally at MS we have different tenant directories as noted above, ARM templates from GitHub can sometimes get in a loop as to which tenant they should deploy the resource to. To avoid confsion and errors **Open a private browser window**. In your browser paste in the link for the ARM guide below. https://github.com/Microsoft/aad-hybrid-lab  Note: This ARM template was developed by the MS National CSA team Bret Hacker. Extra kudos to him for simplifying the lab!
2.	Read through the notes for ARM template prior to deploying. Click the **Deploy to Azure** button in the Git Hub page to open the template in Azure. 

3.	Please sign in with your MS Alias that is an owner in your Azure subscription. All others please make use your subscription owner account. Click **Next**. 

4. Fill in the ARM template parameters with the information as follows:

Subscription: **Your Azure subscription where VMs will get deployed to**
Resource Group: Create new RG called **Infra**

Location: **East US \ East US 2**

Admin Username: **ADadmin**

Admin Password: **WVD@zureL@b2019!**

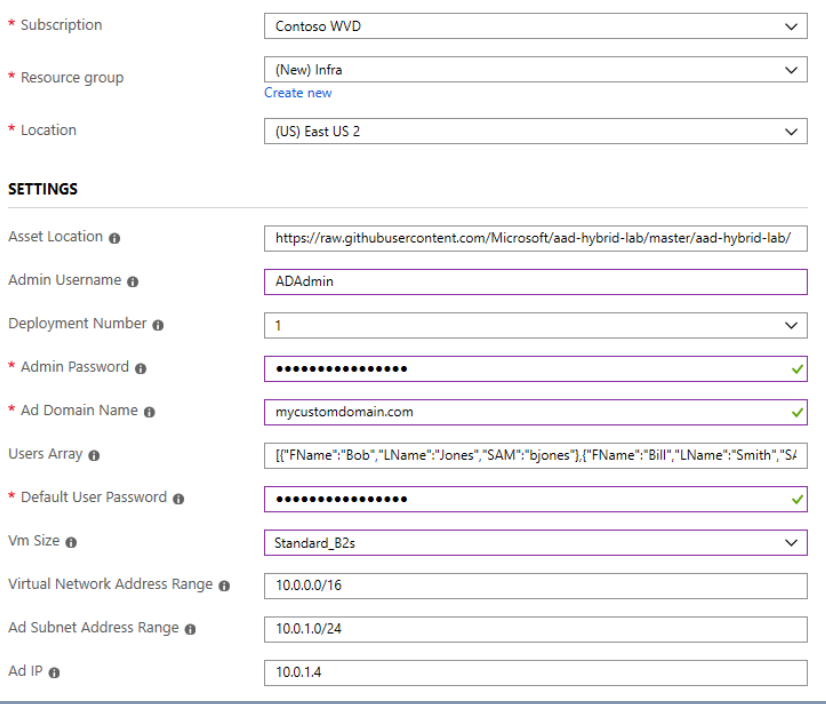
Ad Domain Name: **MyADDomain.com.** ***This CAN match your publicly registered domain for simplicity.***

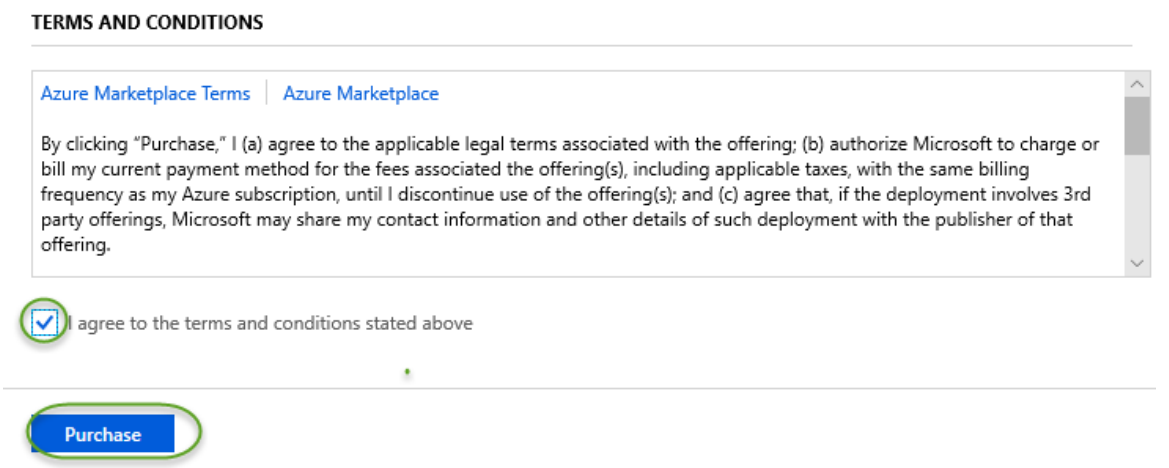
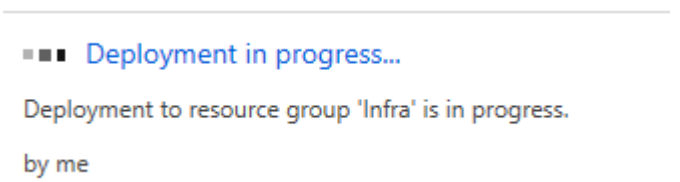
Note: Your AD domain name can also be the same as your custom public domain name, however if the name is more than 15 characters in length it will fail in the deployment. You must shorten your domain name to not exceed the max character length.

Default User Password: **WVD@zureL@b2019!**

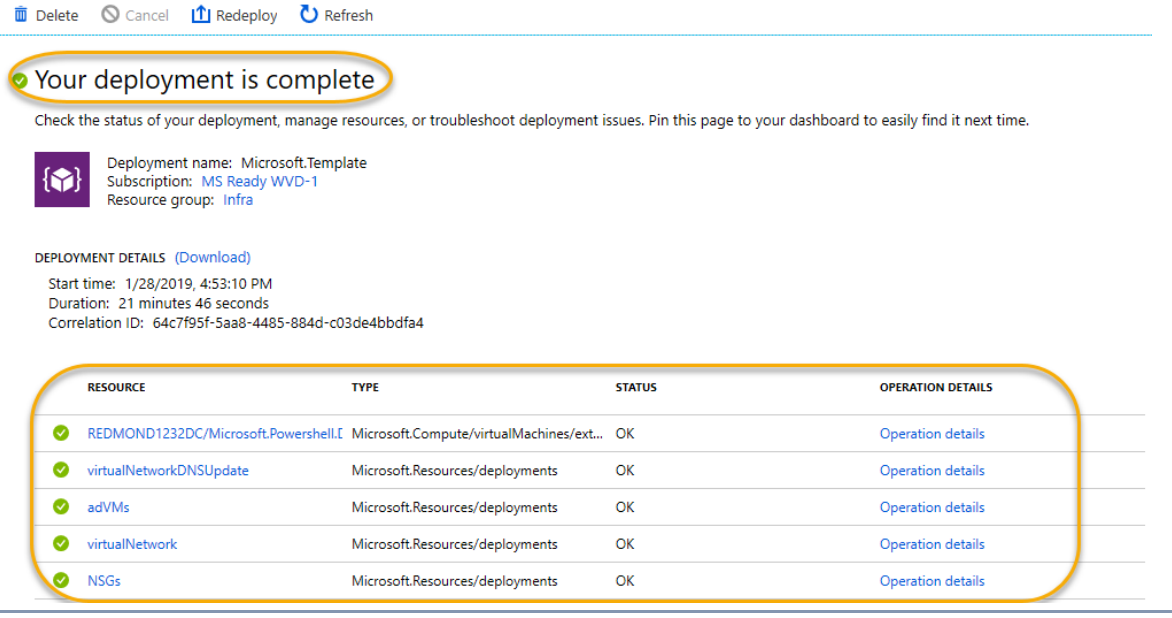
Vm Size: **Standard_B2ms (this can be any instance that is cost effective)**

Accept all the defaults for the other fields



5.	Agree to the Terms and conditions and click **Purchase**. 
6.	The ARM template will deploy. On average in takes around 30 mts to deploy the vm and components in Azure.
7.	Check the deployment progress by clicking on the **notifications** icon. This is important to keep an eye on to see how the deployment progress of each component of the template is going.  Click on **Deployment in Progress…** 
8.	Note: While automation make things simpler and repeatable, sometimes it can also fail. If at any time during the ARM template deployment there is a failure, please delete the Resource Group then try the ARM template again.

9. Once the ARM template is done being deployed the status should change to complete. At this point the domain controller is ready for RDP connectivity.

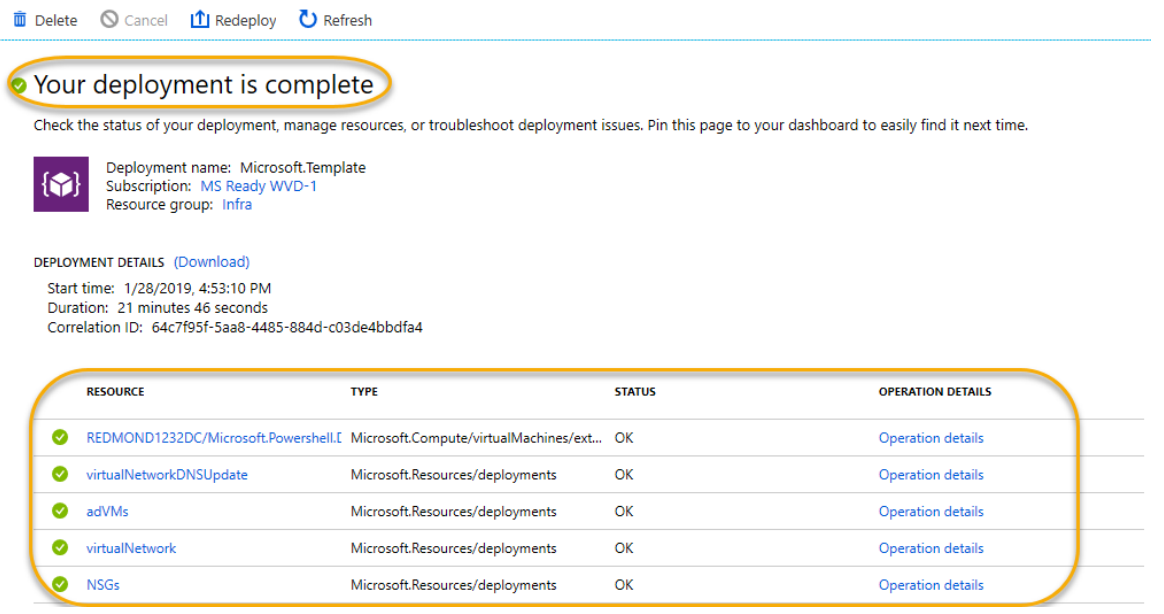


Exercise 5: Azure AD Connect with Active Directory

All the session host VMs (RDSH or Windows 10) in the Windows Virtual Desktop tenant environment must be domain-joined and the AD domain must be synchronized with the tenant's Azure AD. In this exercise we will RDP into the domain controller that was built from the ARM template earlier and configure Azure Active Directory Connect to synchronize the Active Directory accounts into Azure.

Note: PIP RDP access to a domain controller is not a best practice and is only done to simplify this lab. Better security practices such a removing the PIP, JIT, and Bastion service should be applied later to enhance security. **This lab should not be deployed into a production environment.**

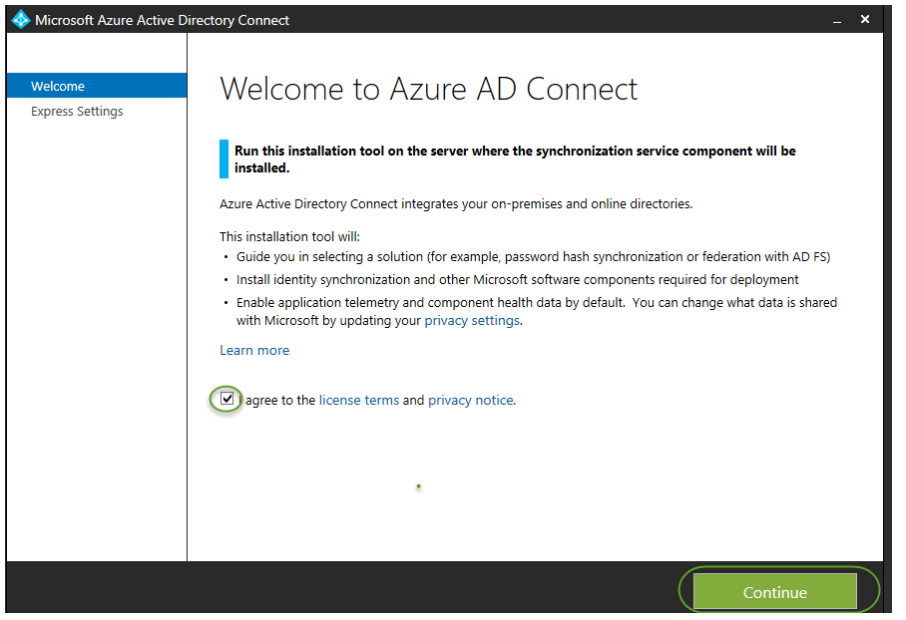
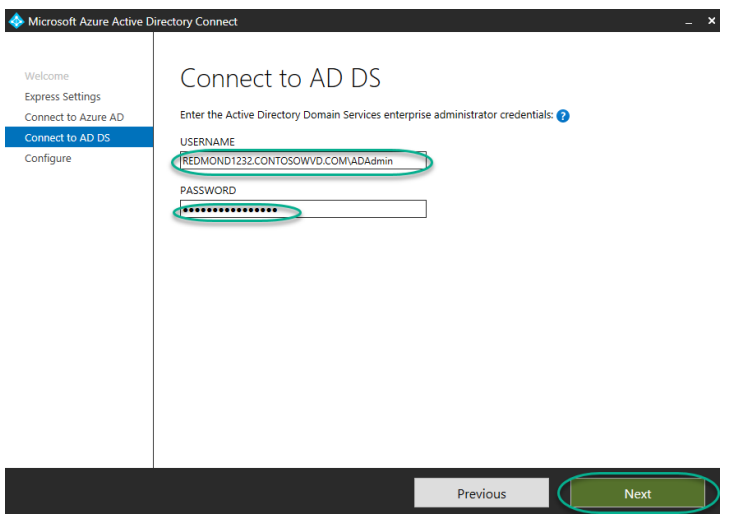
Task 1: Configuring Azure AD Connect

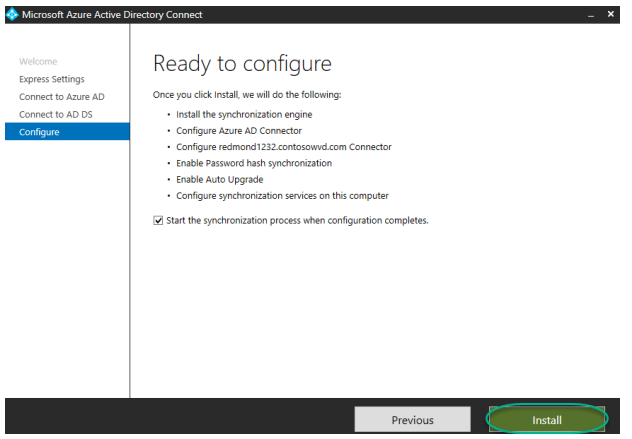
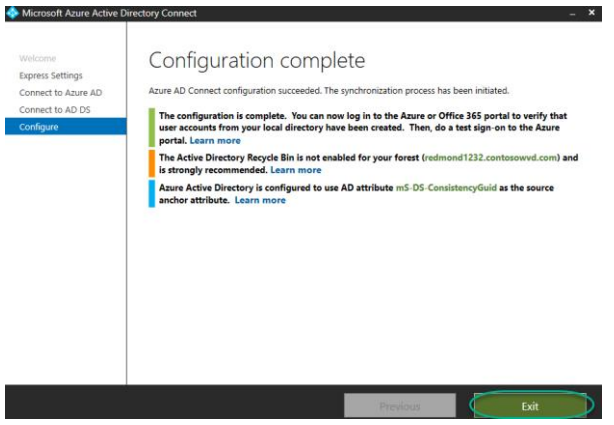
Step	Action
1.	Verify that the ARM template deployed the domain controller successfully before continuing. 
2.	In the Azure Blade click on **Resource Groups.** Click **Infra**. In the overview click **AdPubIP1**. Locate the public IP address of the machine and copy that into your buffer.
3.	On your local machine open RUN and type **MSTSC**
4.	In the RDP window paste in the public IP address copied earlier. Click **Connect**.
5.	When prompted login with the AD domain UPN credentials **adadmin@MyADDomain.com** and a password of **WVD@zureL@b2019!** Note: This is the Active Directory account from the ARM template not the Azure AD Global Admin account. You may also wish to type the username and password in as copy and paste may include an unnecessary space which will cause the authentication to fail. Click **Yes** to the RDP certificate message.

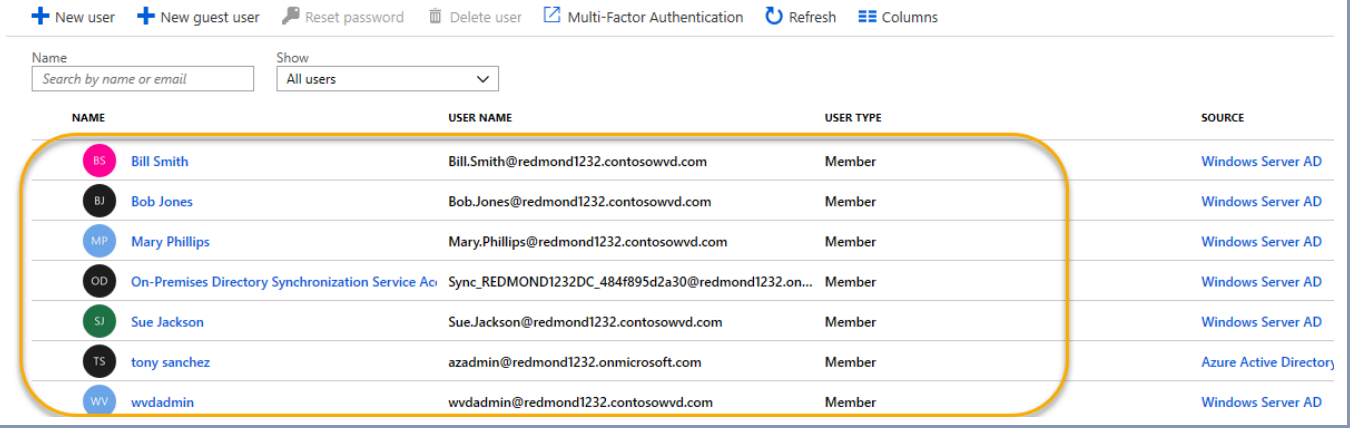
Step	Action

6.	Once connected to the domain controller **open PowerShell ISE. Copy** the code below and **paste that into ISE. Press F5 or the green run script button.** This will disable IE Sec for administrators and simplify a future lab exercise that requires outbound internet access.
	
	`function Disable-IEESC`
	`{`
	`$AdminKey = "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}"`
	`Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 0`
	`Stop-Process -Name Explorer`
	`Write-Host "IE Enhanced Security Configuration (ESC) has been disabled." -ForegroundColor Green`
	`}`
	`Disable-IEESC`
	
	The command should run successfully**.**

7.	By default, AAD Connect does not synchronize the built-in domain administrator account ADAdmin@MyADDomain.com.This is because the system account has the attribute isCriticalSystemObject set to true. While it is possible to modify this, it is not a best practice to do so. **Optional**: In order to have an account that has domain admin rights that will be synchronized to AAD, you will need to add a **new** account in the Active Directory domain before Azure AD Connect synchronization.
8.	**Create a new user and give Domain Admin permissions to this account.** This account will be used during the Host Pool creation process for the servers to be domain joined. ***Note:*** Domain Admin permissions simplify the lab however, any Active Directory account the has the following permissions will suffice. This can be done using Active Directory Delegate Control. • Create computer objects • Delete computer objects
9.	**Minimize** the PowerShell ISE window.
10.	On the domain controller desktop an icon should display Azure AD Connect. Click on **Azure AD Connect**.

Step	Action
11.	Agree to the terms the click **Continue**. 
12.	Click **Use express settings**.
13.	The required components will install.
14.	Enter in your Azure AD Global Admin credentials for example **azadmin@MyAADdomain.onmicrosoft.com** and the correct password. Click **Next**.
15.	Enter in the Active Directory credential that was used in the ARM template deployment for the domain controller **MyADDomain.com\ADadmin** and a password of: **WVD@zureL@b2019!** then click **Next**. Note: If copying and pasting the password in, please ensure that there are no spaces as that will cause the verification to fail. 

Step	Action
16.	Click **Install** to start the configuration and synchronization.  After a few minutes the Azure AD Connect installation is complete. Click **Exit**. 
17.	Wait a few minutes to ensure AD Sync has synchronized the AD accounts into the Azure AD tenant directory. After the allotted time has passed, minimize the RDP session to the domain controller.
18.	Maximize the Edge browser and in the Azure portal switch directories to AAD GA tenant **MyAADdomain.onmicrosoft.com**

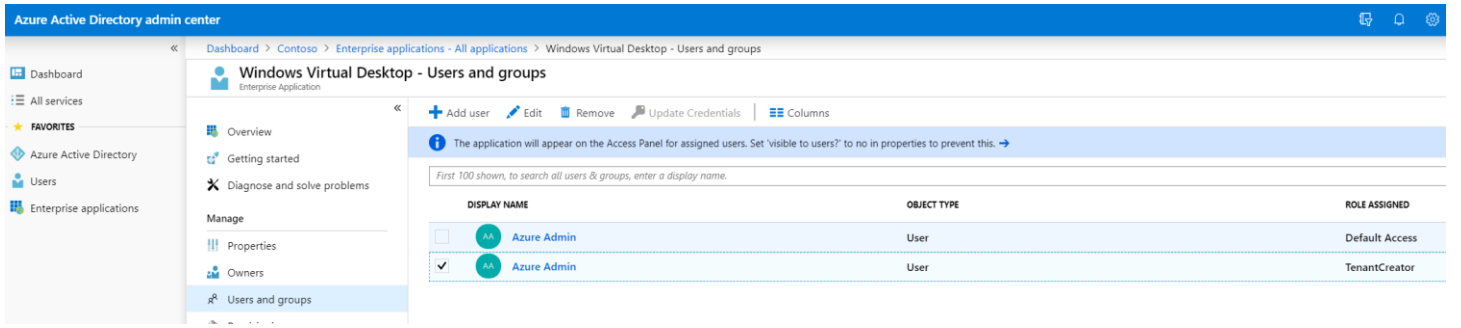
Step	Action
19.	To verify that the accounts are replicated from the Active Directory VM, open the Azure portal and navigate to the **Azure Active Directory Blade**. Click on **Users**. The objects should now appear like below. Note: It can take up to 15 minutes for the Active Directory objects to be synchronized to AAD tenant. 

Exercise 6: Assign the “TenantCreator” role to a user account

Task 1: WVD Tenant Creator:

1. Log in to the Microsoft Azure Portal.
2. Navigate to **Azure Active Directory** from the left menu.
3. In the Middle Pane, click on **Enterprise applications**.
4. Search for and select **Windows Virtual Desktop**.
5. In the Middle Pane, select **Users and groups**.
6. Select **Add user**, select **Users and groups**, and search for the user to whom you want to grant permissions to perform the Windows Virtual Desktop tenant creation. You should use the account created in the Azure AD section and the one you applied Global Admin rights to and should mimic this format, myaccountname@MyAADdomain.onmicrosoft.com
7. Select the user and hit **Select,** followed by **Assign**.

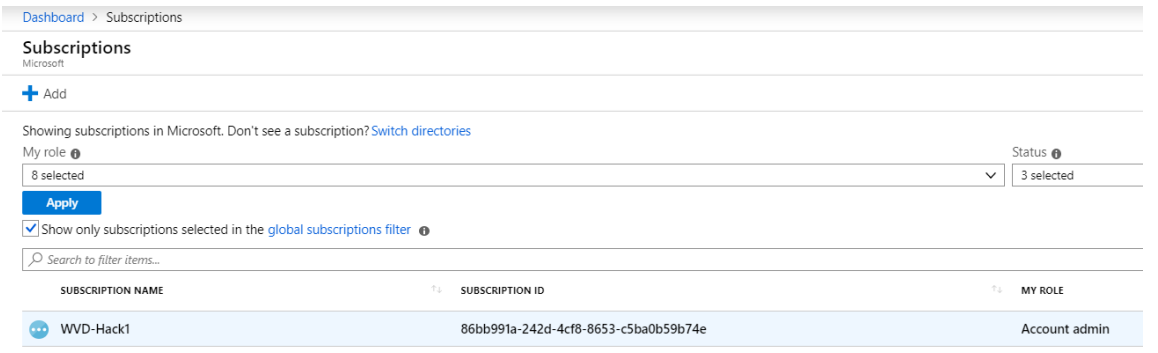
Your user should now have the role of “TenantCreator.”



Exercise 7: Create a Windows Virtual Desktop Tenant

Now that you have a user with the right permissions to create a Windows Virtual Desktop tenant, let’s go ahead and create it. During this step, you will need two IDs:

- Your Azure AD tenant ID (again).
- Your Azure subscription ID, which can be found by visiting the Microsoft Azure Portal and doing a keyword search for “Subscriptions.” Select **Subscriptions** from the search results and your subscription ID, assuming you have an active subscription, will be displayed below.



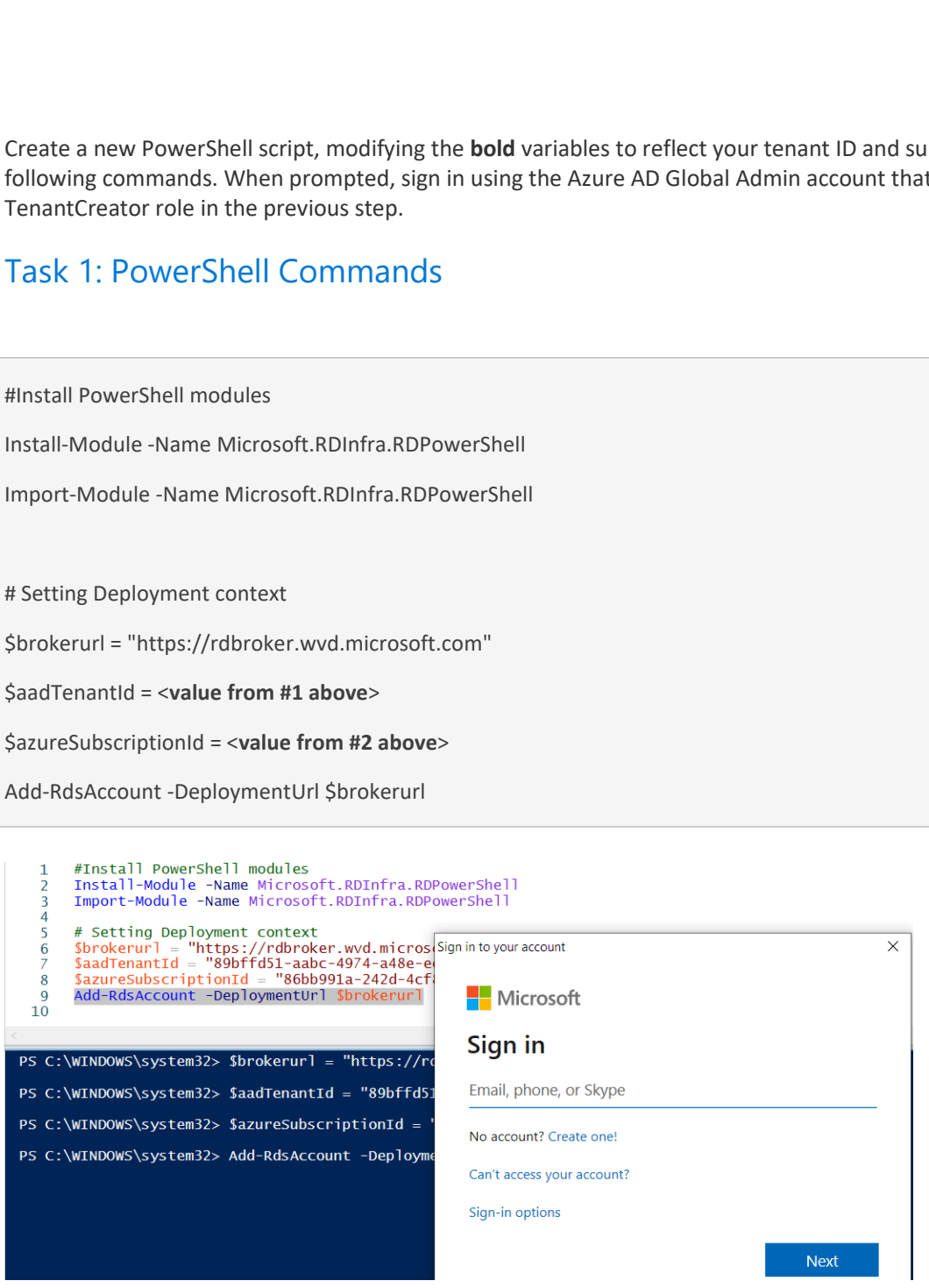
Once you have these two IDs, you can create the Windows Virtual Desktop tenant.

Note: Before proceeding, make sure you import the Windows Virtual Desktop cmdlets for Windows PowerShell as described in the prerequisites section above. If you haven’t completed this step yet, see these instructions.

Create a new PowerShell script, modifying the **bold** variables to reflect your tenant ID and subscription ID, and execute the following commands. When prompted, sign in using the Azure AD Global Admin account that was assigned to the TenantCreator role in the previous step.

Task 1: PowerShell Commands

```
#Install PowerShell modules

Install-Module -Name Microsoft.RDInfra.RDPowerShell

Import-Module -Name Microsoft.RDInfra.RDPowerShell


# Setting Deployment context

$brokerurl = "https://rdbroker.wvd.microsoft.com"

$aadTenantId = <value from #1 above>

$azureSubscriptionId = <value from #2 above>

Add-RdsAccount -DeploymentUrl $brokerurl
```



Next, use the following command to create the Windows Virtual Desktop tenant.

```
New-RdsTenant -Name <yourWVDTenantNamehere> -AadTenantId $aadTenantId -AzureSubscriptionId $azureSubscriptionId
```

```
PS C:\WINDOWS\system32> New-RdsTenant -Name "yourtenantnamehere" -AadTenantId
$aadTenantId -AzureSubscriptionId a5915b41-c517-4f34-9dcc-5019fcc13


TenantGroupName         : Default Tenant Group
AadTenantId             : bad69d79-ae9e-4e94-9157-966cbd2d9xxx
TenantName              : yourtenantnamehere
Description             :
FriendlyName            :
SsoAdfsAuthority        :
SsoClientId             :
SsoClientSecret         :
AzureSubscriptionId     : a5915b41-c517-4f34-9dcc-5019fcc11xxx
LogAnalyticsWorkspaceId :
LogAnalyticsPrimaryKey  :
```

Add your Azure AD Global Admin account and your AD Domain Admin equivalent account to the WVD RDS Owners group by typing the below into the PowerShell session. Please see the beginning of this document for Account Guidance for the reason of this step.

New-RdsRoleAssignment -SignInName myaccount@MyAADdomain.onmicrosoft.com -RoleDefinitionName "RDS Contributor" -Tenan tName "yourtenantnamehere" -AadTenantId bad69d79-xxxx-4e94-9157-966cbd2d9933

New-RdsRoleAssignment -SignInName admin@MyADDomain.com -RoleDefinitionName "RDS Contributor" -TenantName "yourtenan tnamehere" -AadTenantId bad69d79-xxxx-4e94-9157-966cbd2d9933

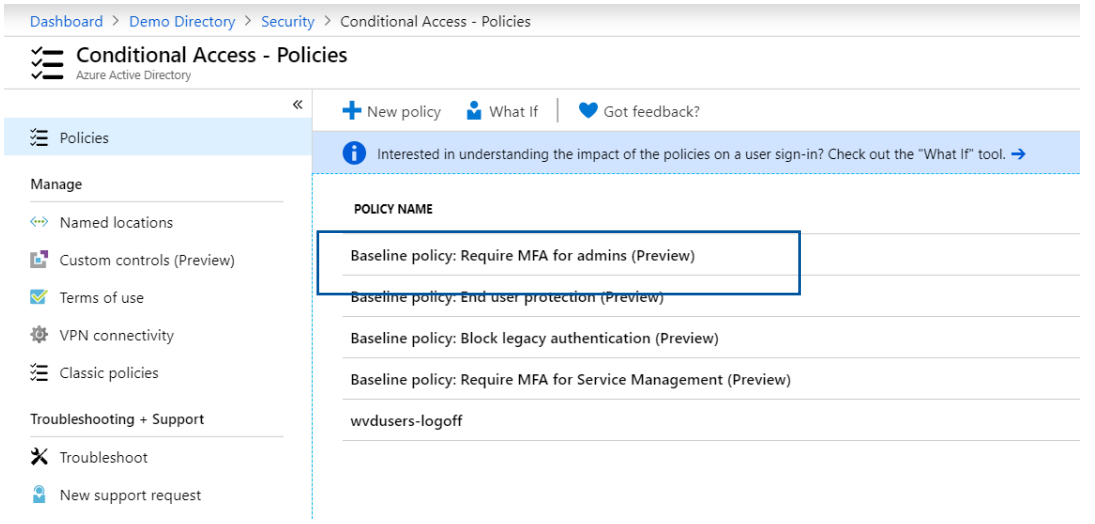
Example below:

```
PS C:\WINDOWS\system32> New-RdsRoleAssignment -SignInName jjenner@mswvd.com -RoleDefinitionName
"RDS Contributor" -TenantName "jojennertest" -AadTenantId bad69d79-ae9e-4e94-9157-966cbd2d9933


RoleAssignmentId   : bd46a08f-9c3f-404c-f80f-08d7406fc30a
Scope              : /Default Tenant Group/jojennertest
TenantGroupName    : Default Tenant Group
TenantName         : jojennertest
DisplayName        :
SignInName         : jjenner@mswvd.com
GroupObjectId      :
AADTenantId        : bad69d79-ae9e-4xx4-9xx7-966cbd2d9933
AppId              :
RoleDefinitionName : RDS Contributor
RoleDefinitionId   : f5dc85e1-b94d-48f0-f5dc-08d623dd1cc4
ObjectId           : 4ded2eed-xxxx-xxxx-d49c-08d740738918
ObjectType         : User
Item               :
```

NOTE: MFA

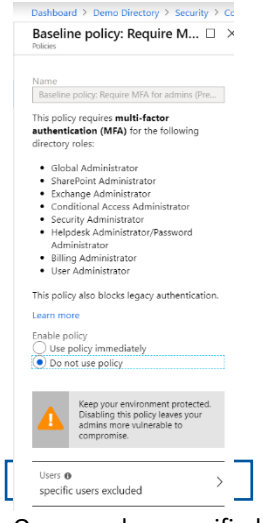
*Note: For MFA security, if you have MFA setup or wish to use it for WVD, you can follow these instructions

From within your AAD portal, Browse to your directory, Security, Conditional Access – Policies and match the image below.



Select the first option in the list ‘Baseline Policy: Require MFA for admins (Preview).

Default radio button below, change your selection to ‘Use Policy immediately’. Select the users you want to exclude from the policy, like the GA account which is required for the consenting to the WVD service. For the users that you want to have MFA, you will need to at this point login to the Azure Portal as a user and create the authentication process using your phone and SMS or Email options.



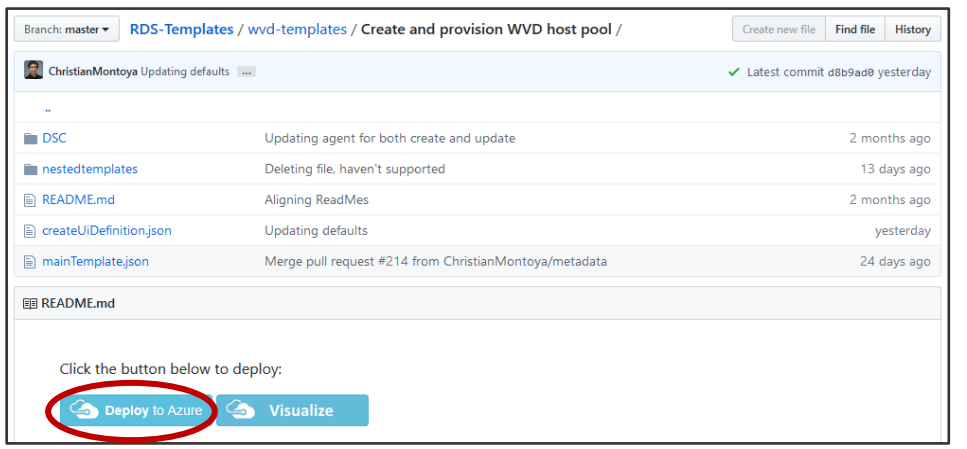
Once you have verified the user account, you can now login to the WVD Browser or RDClient using MFA.

Exercise 8: Deploy Windows Virtual Desktop Host Pool – GitHub

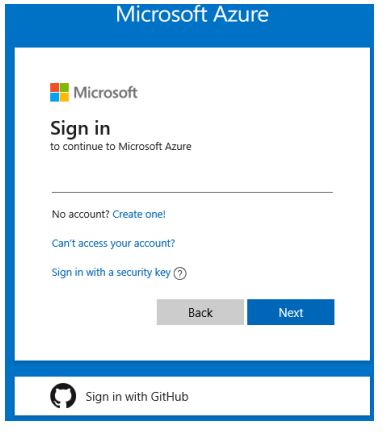
A host pool is a collection of VMs that offer a similar service, such as a full desktop experience. When users connect, they can access a virtual desktop session on any of the hosts in the pool. Host Pools can be provision with several different methods. In this section we'll focus on deploying a Host Pool via a GitHub Template.

Task 1: Configure basic settings

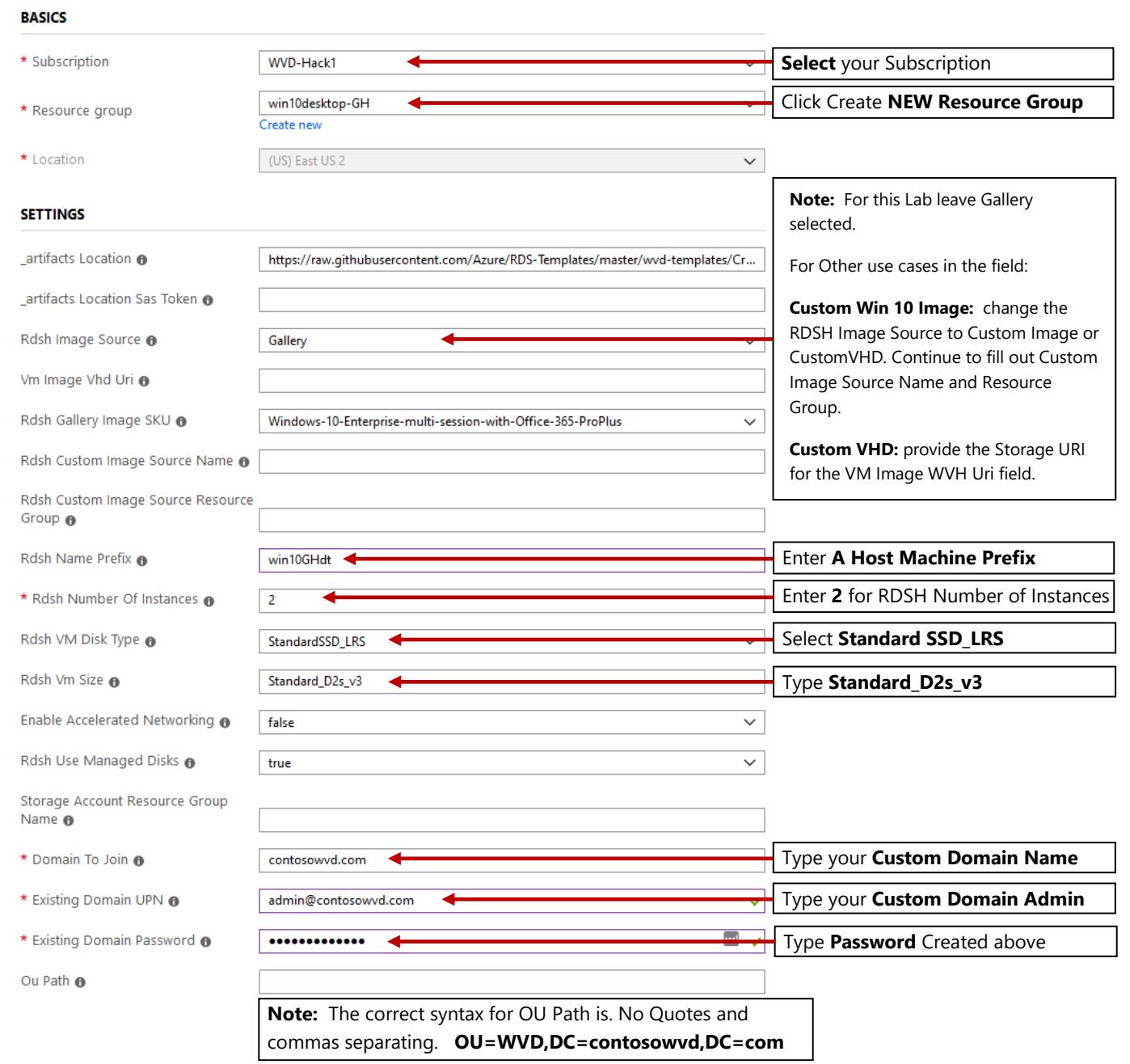
To get started, visit the Microsoft WVD GitHub Page. Select **Deploy to Azure**. Log into the Azure Portal with your **Azure AD GA account**.



Please sign in using your Azure subscription owner account. Click **Next**.



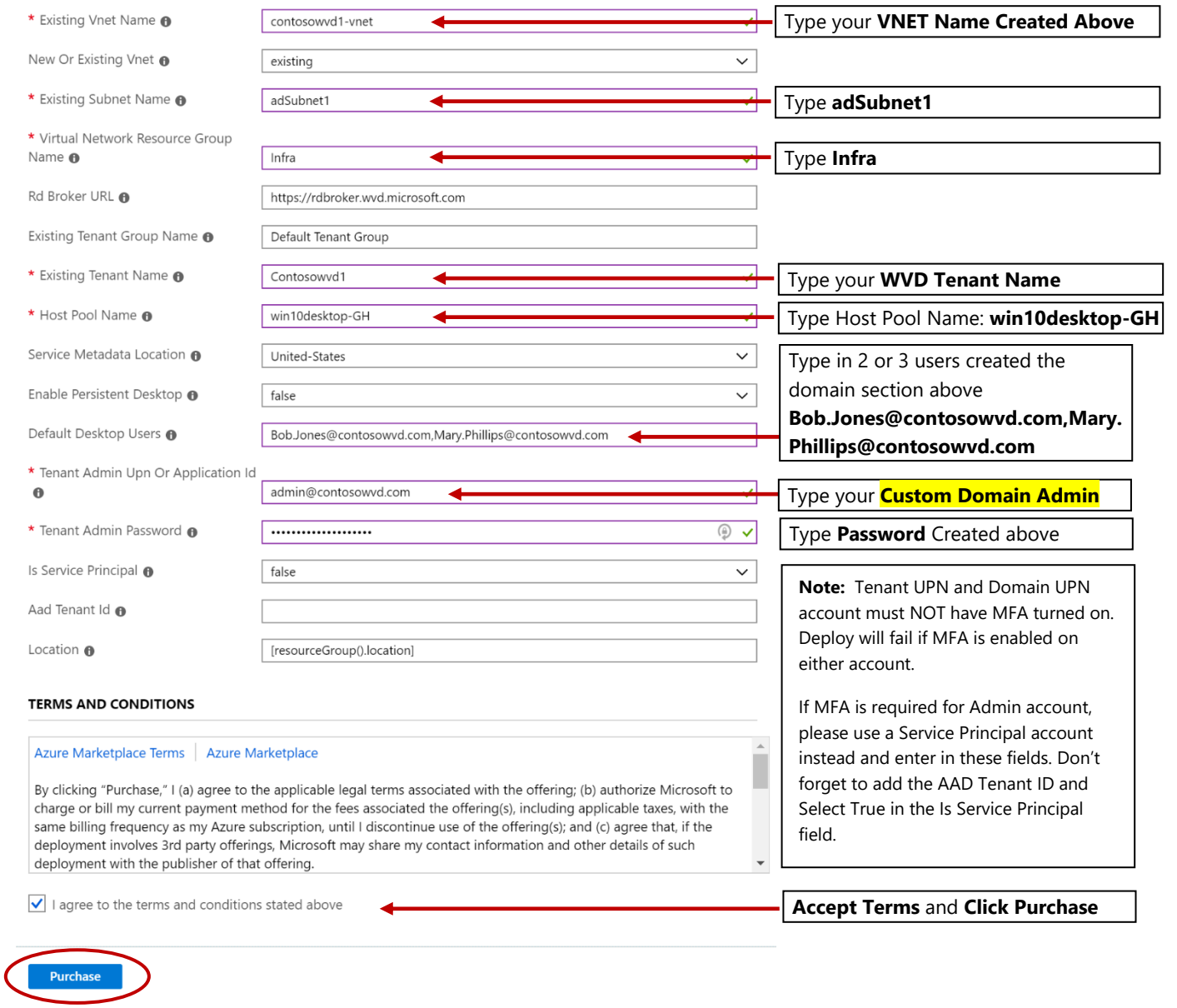
Follow each step for field entries. **Accept Defaults unless otherwise noted**.



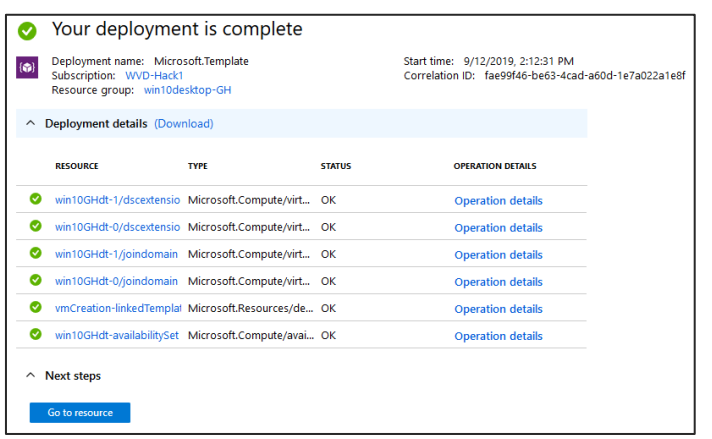
Rdsh VM Disk Type: *For Visual Studio Account HDD is the recommended Disk Type Size. For POC, LAB, Pilots, and Production environments StandardSSD_LRS or PremiumSSD_LRS is recommended*

Rdsh VM Size: *This can vary heavily depending on customer use case, workload type, and number of users desired per Host.*

Continue Filling out all remaining fields with required information



When Complete the Deployment will show green on each step.



Common Errors:

- **Domain Join Fail**: Typically caused by bad domain UPN credentials, or not enough rights in AD to add a new machine
- **Dscextension Fail**: Typically caused by bad AAD Tenant UPN Credentials or incorrect entered WVD Tenant Name

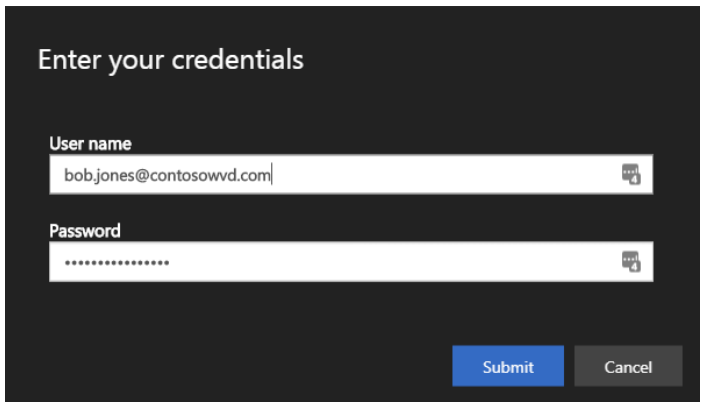
Task 2: Test with the HTML5 client

Launch a browser in privacy or incognito mode and visit http://aka.ms/wvdweb to access the HTML5 client.

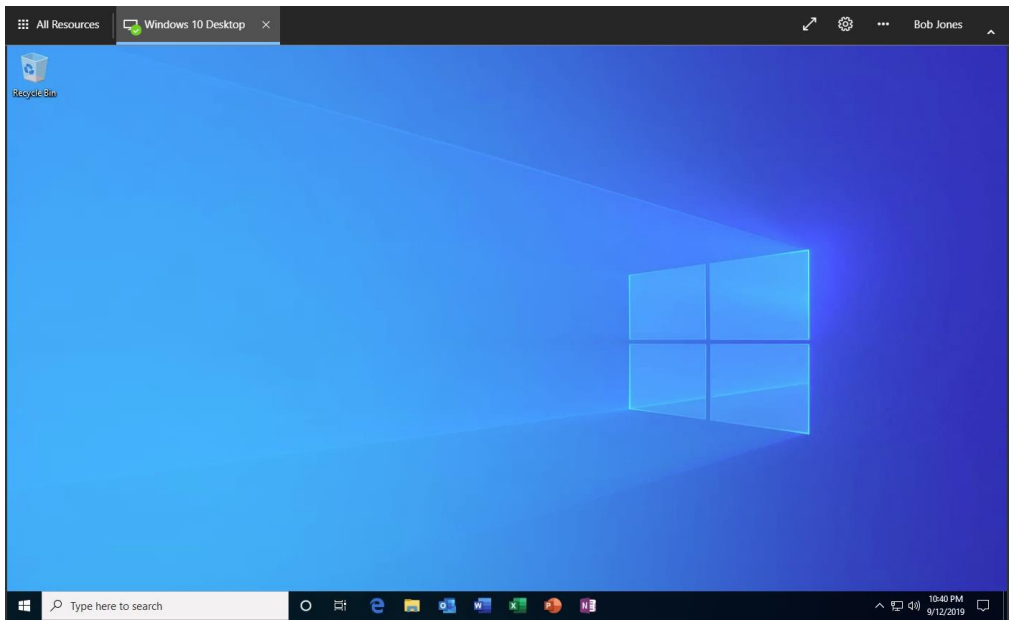
Click the Windows 10 Desktop Icon. Click Allow for Local Resources

Authenticate using the login information

User Name: bob.jones@contosowvd.com
Password: **WVD@zureL@b2019!**



If you are successful, the Remote Desktop Client Connection opened in the previous task should close, and you should be able to view the same desktop via your browser window.



Log off from the WVD session.

NOTE: Host Pool Validation Tag

By default, the WVD agent on each host will auto update as new versions release. You can prevent this behavior by setting the **ValidationEng** flag in a host pool to true (by default it's set to false). With the ValidationEng flag set to true ONLY the host in that host pool will auto update giving administrators control to test new WVD host images, and the newest agents before rolling to production.

A single host pool in a WVD Tenant with the ValidationEng flag set to true will prevent all other hosts in the WVD Tenant from auto updating. You can toggle the ValidationEng flag via PowerShell to false at any time to allow the hosts in that tenant to start auto updating.

For production WVD Tenants it's a good practice to create a single host pool with the ValidationEng flag set to true. When an administrator is comfortable with the newest agent build the ValidationEng flag can be set to false.

To enable this setting follow the PowerShell steps below.

Open a PowerShell window.
Log onto the WVD Service with an WVD Tenant Owner and your Azure AD GA account once prompted

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

Choose a Host Pool for the validation pool

Run

```
Set-RdsHostPool -Name <HostPoolName> -TenantName <yourTenantName> -ValidationEnv $True or $False
```

```
PS C:\Users\dawhitla> Set-RdsHostPool -Name ValidationPool -TenantName contosowvd1 -ValidationEnv 1


TenantName            : contosowvd1
TenantGroupName       : Default Tenant Group
HostPoolName          : ValidationPool
FriendlyName          :
Description           :
Persistent            : False
DiskPath              :
EnableUserProfileDisk :
ExcludeFolderPath     :
ExcludeFilePath       :
IncludeFilePath       :
IncludeFolderPath     :
CustomRdpProperty     :
MaxSessionLimit       : 999999
LoadBalancerType      : BreadthFirst
ValidationEnv         : True
Ring                  :



PS C:\Users\dawhitla>
```

NOTE: Other Common Settings

WVD Tenant Friendly Name: You can give you WVD Tenant a friendly name that will look less technical

Set-RdsTenant -Name <WVD Tenant> -FriendlyName "Contoso WVD"

DesktopFriendlyName: Using the under the Set-RdsRemoteDesktop PowerShell cmdlet the FriendlyName option allows you customize the name users see on the Web Client or Thick Client.

```
Set-RdsRemoteDesktop -TenantName <tenant> -HostPoolName <Host Pool> -AppGroupName "Desktop
Application Group" -FriendlyName "Eng Host"
```

Description: This field can be used by administrators to give additional detail about the HostPool

Persistent: By default, this is set to False. If you want to set a Host Pool for persistence - designates a single Host to a single user - you can set this flag to True. The first time the user logs in they will be assigned a host by the WVD Service. Every subsequent time that user logs in they will be directed to that same host. NOTE: This can ONLY be set at HostPool creation and cannot be switched. Set by default to False. If you do not set the flag to True at HP creation, you must destroy the HostPool and re-create to change.

MaxSessionLimit: Controls the number of users who can log onto HostPool Hosts

LoadBalancerType: Two options

Breadth: As users log in their sessions are spread across all hosts in the Host Pool evenly, giving users the most possible resources.

Depth: As users log in a host is filled to max user session before assigning users to another host. This helps maximize host resource utilization and can help reduce cost.

Exercise 9: Deploy Windows Virtual Desktop Host Pool – Market Place

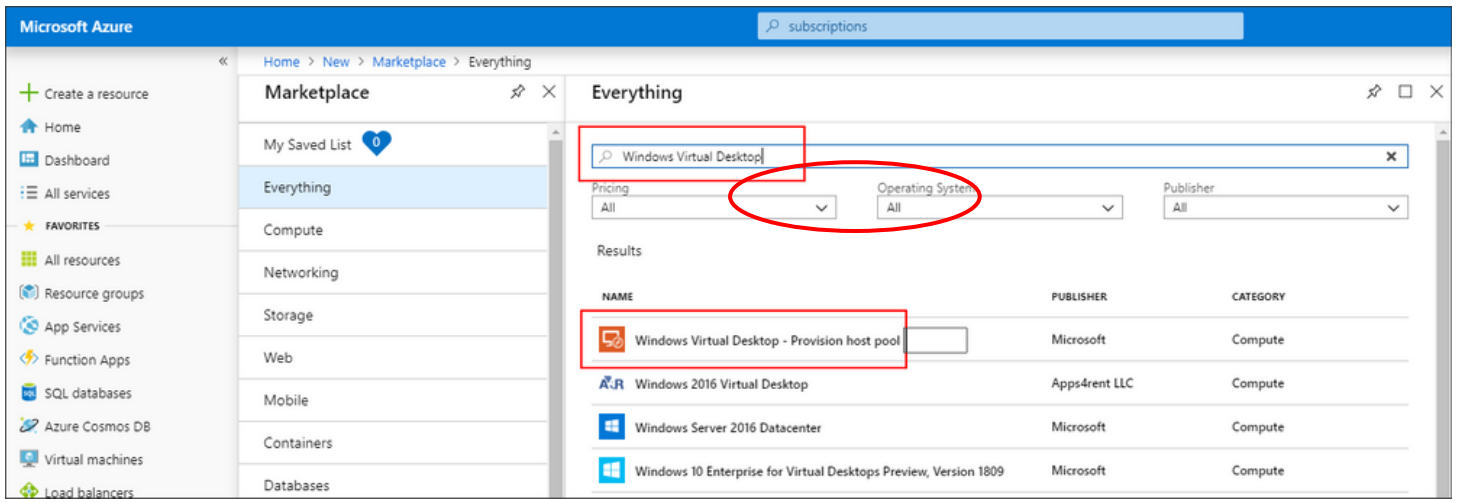
A host pool is a collection of VMs that offer a similar service and have the same underlining hardware. Examples of host pools are serving up a full desktop experience to users or a set of applications. When users connect, they can access a virtual desktop session on any of the hosts in the pool.

Host Pools consist of:

- A collection of session hosts (servers)
- Application groups
- Desktop groups

Task 1: Configure basic settings

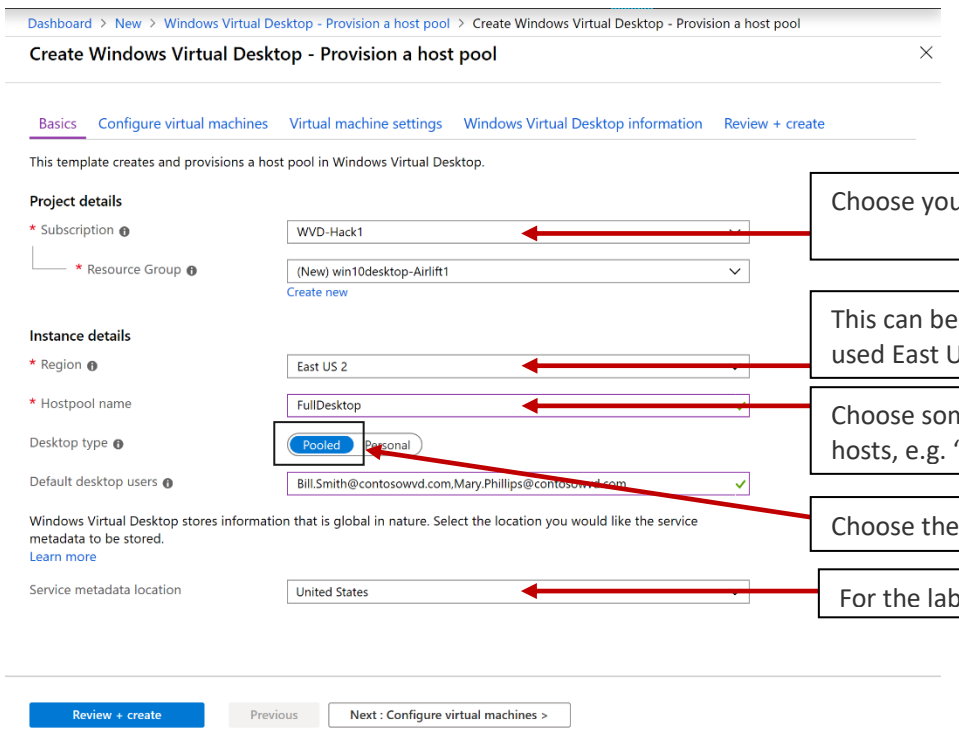
To get started provisioning your Host pool, visit the Microsoft Azure Portal, select **Create a Resource** and search for **Windows Virtual Desktop**. Select **Windows Virtual Desktop – Provision a host pool**.



Select **Windows Virtual Desktop – Provision a host pool** and click **Create**. The details to **Provision a host pool** should be entered as below:

- **Subscription** – Azure subscription
- **Resource group** - Use an empty Resource Group or enter a name to create a new one.
- **Host Pool name** - Choose something descriptive for the pool of hosts, e.g. “Full Desktop”
- **Desktop type: Pooled** or **Personal** - Choose **Pooled** unless you are deploying a virtual desktop infrastructure (VDI) configuration wherein every user has their own dedicated VM.
- **Default desktop users** - Add a comma separated list of users with **no spaces**. (Group support will follow later.) You can also use PowerShell to add users to this host pool at a later point. There was sample user created in your Active directory as part of the template (Bill.smith@yourdomain.com and Patty.Phillips@yourdomain.com are examples of the user’s name)

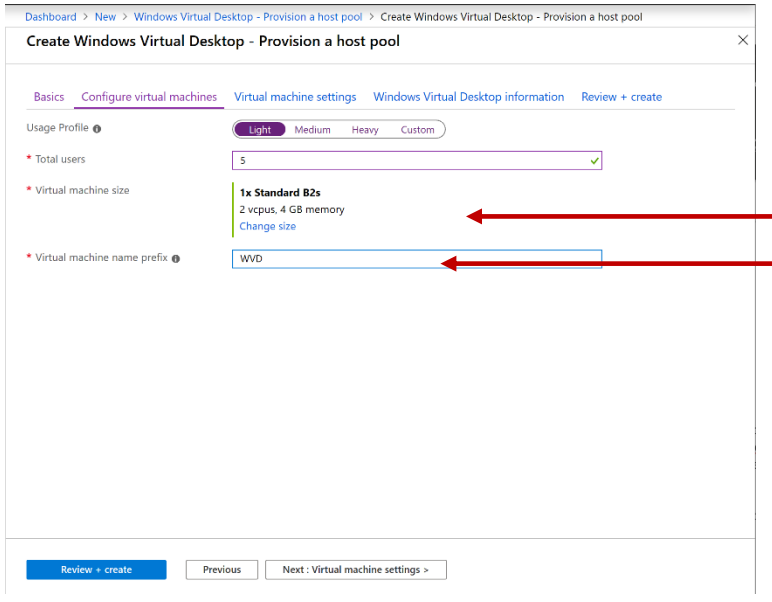
- **Location** - Enter the location where the VM resources will be created. This can be any existing Azure region of your choice. **For the Purposes of this Lab Select East US 2**
- After this blade is filled out, we will continue the journey using the button **"Next: configure virtual machines" at the bottom of the screen**. Do not advance to review and create until all tabs of the form are filled out



Task 2: Configure the virtual machines

Next, you'll enter the VM size details:

- Select a **Usage Profile** that matches your environment: **Light**, **Medium**, **Heavy**, or **Custom**.
- Enter the number of **Total users** that will be using this host pool.
- If desired, change the **Virtual machine size**. For your test environment, which will likely have very few users, you could opt for a smaller size. You can find additional examples and size guidance in the Windows Virtual Desktop pricing guide.
- Add a prefix for the VMs. This will become the VM name.
    - Note:
        - Make sure your prefix is unique in your environment; don't reuse your prefixes.
        - Windows VMs have a 15 character name limit. WVD Template will append minimum 2 characters at the end: "-0, -1, -2", etc
- Click **Next: Virtual Machine Settings** when complete



Task 3: Configure VM settings- Gallery Image

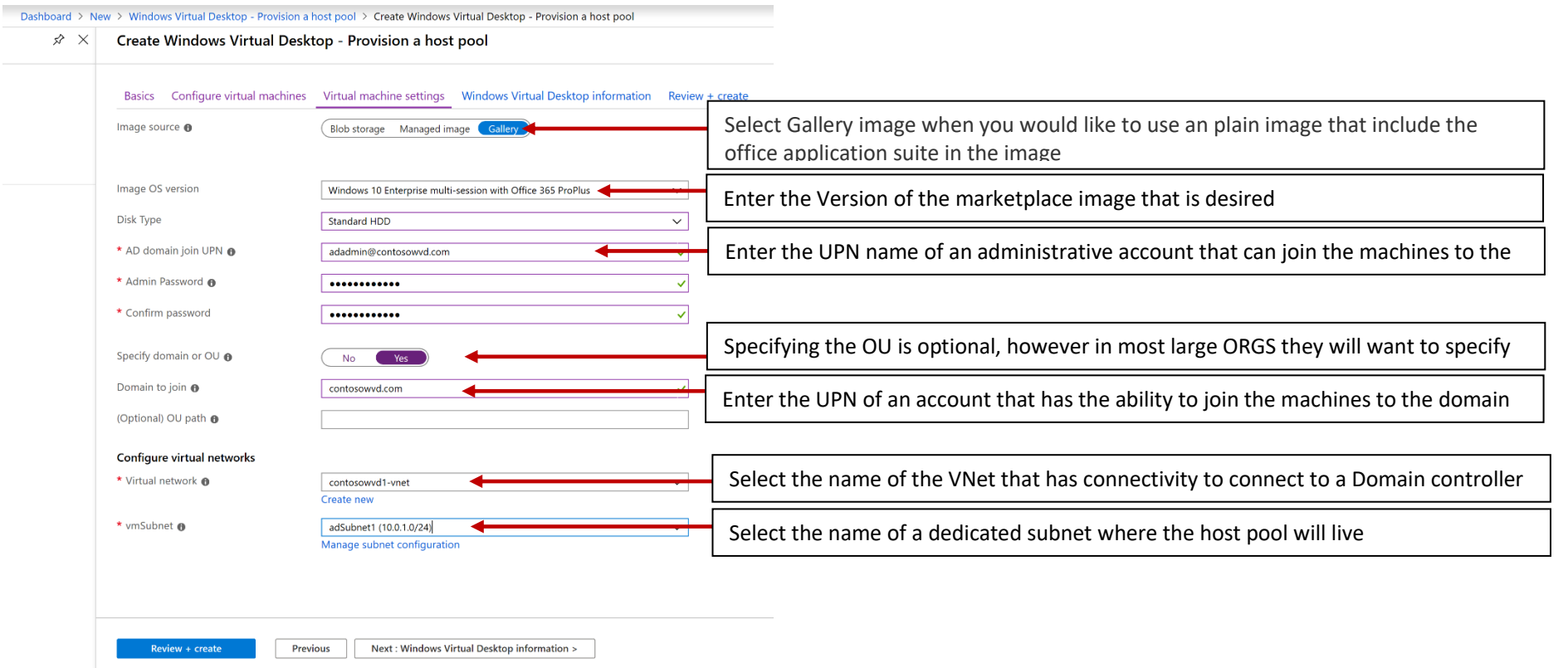
This step has 3 options:

1. Task 3 outlines how to deploy using a blank vanilla marketplace Image
2. Task 3.1 outlines how to deploy using a managed Image
3. Task 3.2 Outlines how to deploy using a custom .vhd file built on premise

This section outlines using a Blank Vanilla Market place image.

To configure the VMs for Azure, you will need to:

- Select a custom image from **Blob storage**, a **Managed image** in Azure, or one from the **Gallery**. For this lab use "Windows 10 Enterprise multi-session with Office 365 Pro Plus" from the Azure Gallery. Office 365 Pro Plus has been preconfigured for the ideal state of Windows 10 multi-session.
- Select the **Disk Type**. For MS internal subscriptions, SSD is a good choice. For MSDN, Trial, and PAYG Subscriptions HDD is recommended. We suggest Premium SSD for Production environments.
- Enter credentials that have permissions to join a VM to the Active Directory you created earlier.
- Click **Yes** to specify Domain Specify the domain and/or OU. Enter the domain you created earlier. Note Domain is mandatory. The OU can be specified by adding the domain path with no quotes, and comma separated.
    - For example: OU=wvd,dc=contosowvd,dc=com
- Select the Virtual network and the subnet created for the Infra deployment of Active directory
- Here an example of what step 3 of the wizard could look like:



Task 3.1 : Configure VM settings- Managed Image

Before selecting this option, you need to publish a managed image in Azure.

Using this option will allow you to create an Image in Azure using one of the supported OS Platforms for WVD, inclusive of custom applications, agents and startup processes, and then use it to: repeatedly deploy new or expand existing host pools in an standardized procedure.

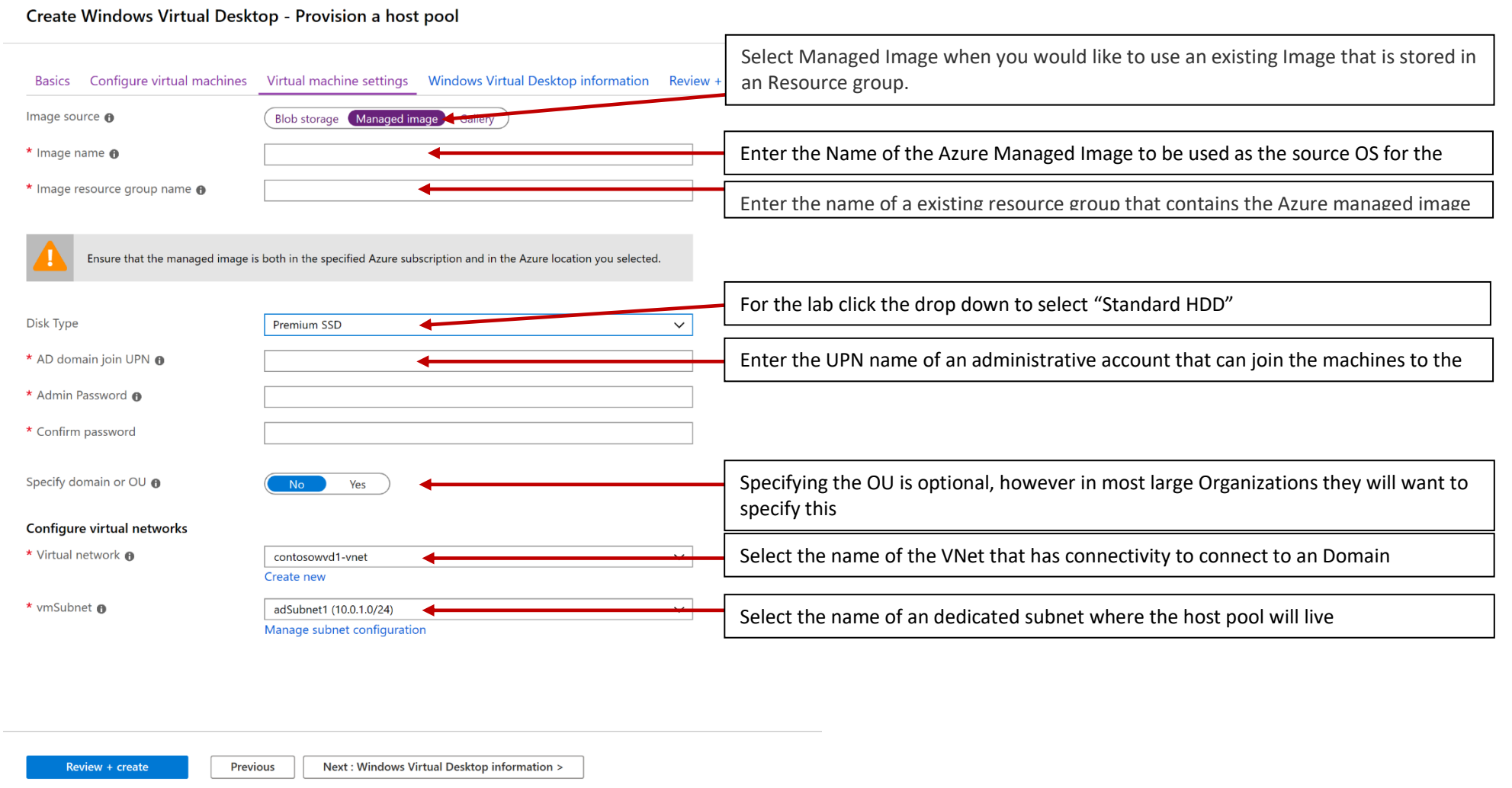
The detailed guidance for creating a managed Image can be found here:

https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-generalized-managed

https://docs.microsoft.com/en-us/azure/virtual-desktop/set-up-customize-master-image#software-preparation-and-installation

The above links are referenced in the WVD image creation guide found here:

https://docs.microsoft.com/en-us/azure/virtual-desktop/set-up-customize-master-image



Task 3.2: Configure VM settings- Blob Storage Image

Before selecting this option, you should have already created an Image using the guidance found here:

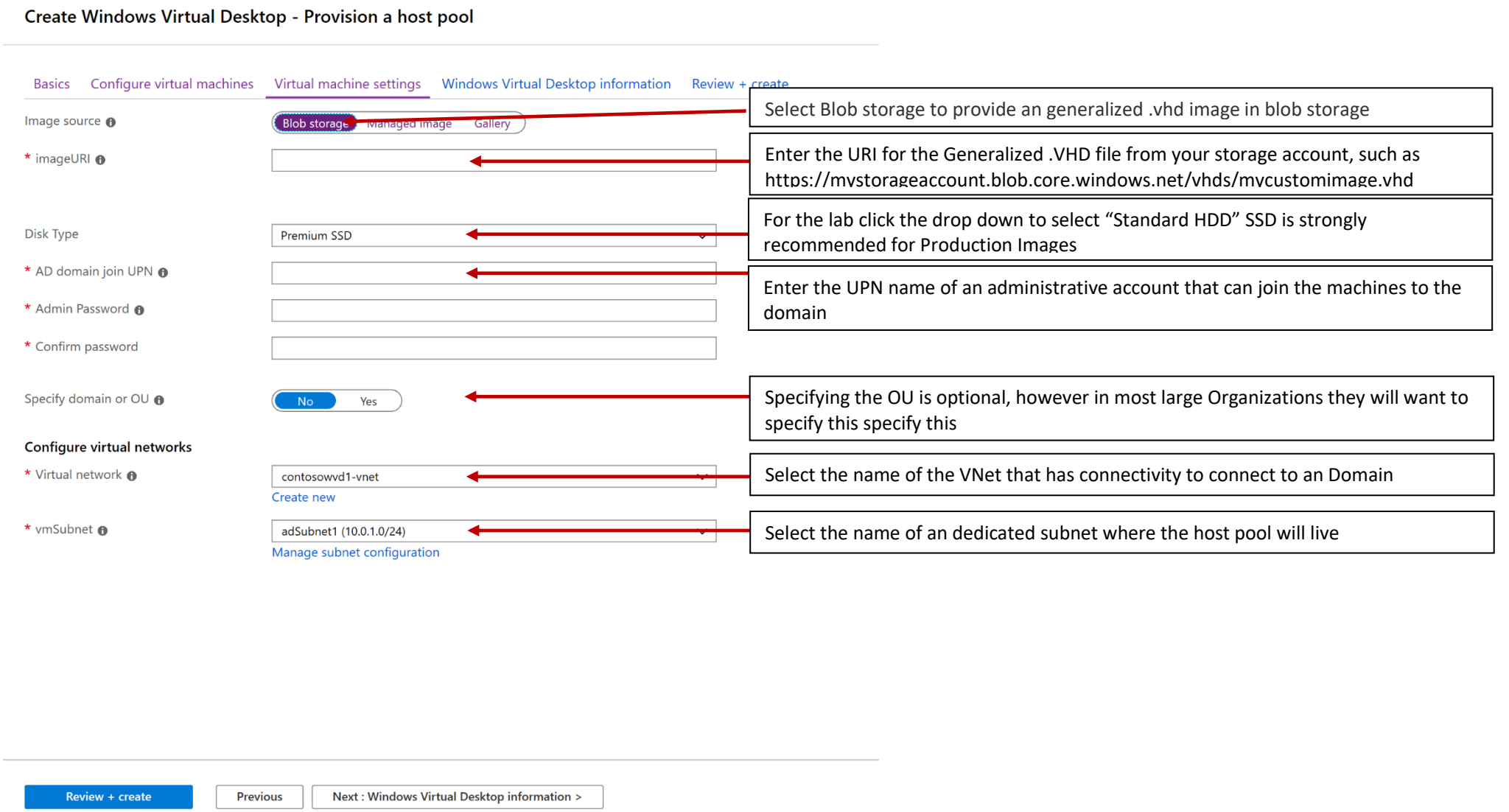
https://docs.microsoft.com/en-us/azure/virtual-desktop/set-up-customize-master-image

For integrating the Office Application suite into your custom Image please review the guidance found here:

https://docs.microsoft.com/en-us/azure/virtual-desktop/install-office-on-wvd-master-image

Using this procedure will allow you to build the image on premise and using one of the supported OS platforms as published by the WVD product group.

This is a sample of how to use an Image that is stored on an Blob storage account:

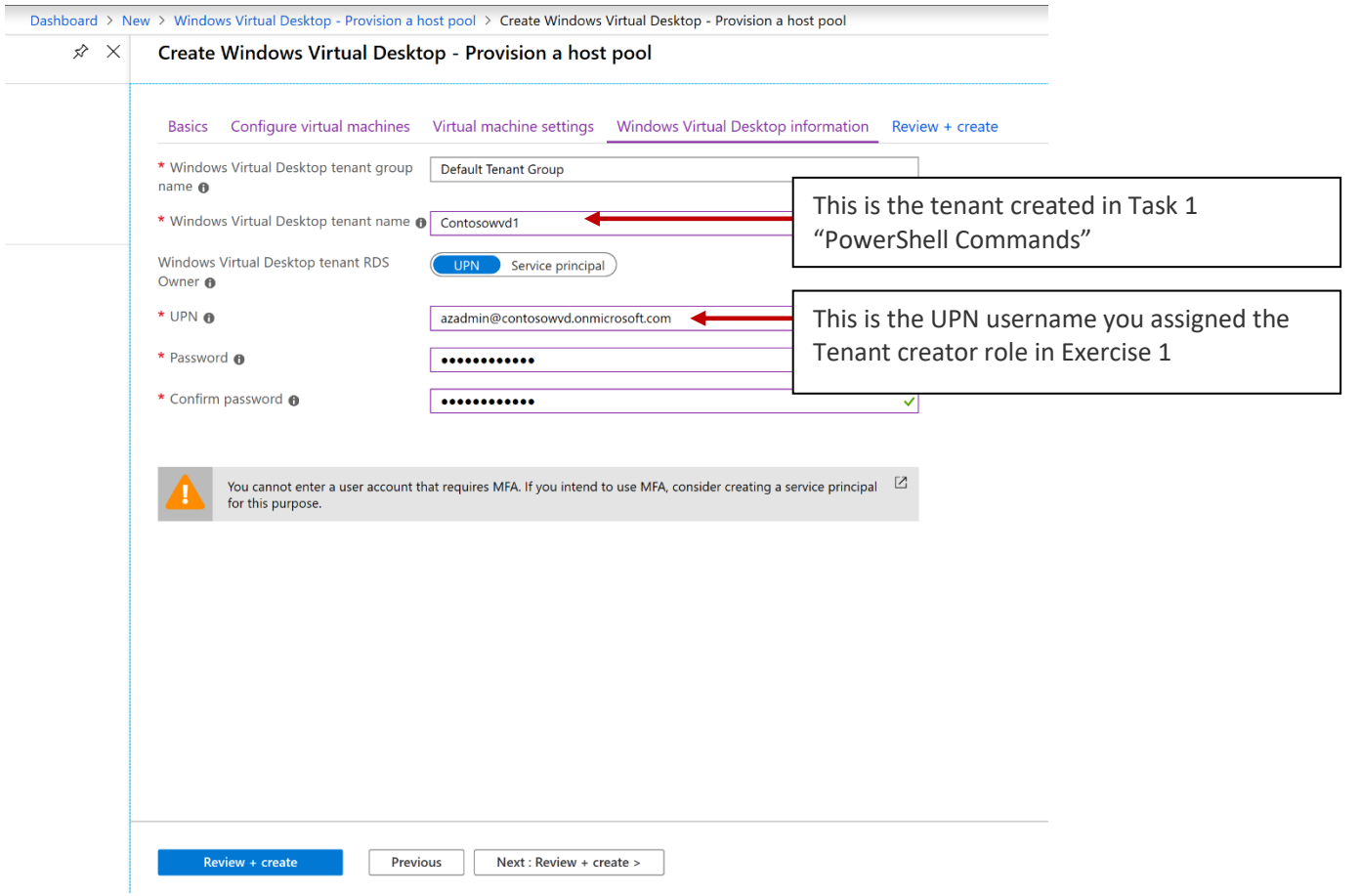


Task 4: Enter Authentication Details

Once you have configured your VM settings, you will need to enter details about your Windows Virtual Desktop tenant and Azure AD tenant. Unless otherwise directed, leave the **Windows Virtual Desktop tenant group name** as “Default Tenant Group.” For the Windows Virtual Desktop tenant name, enter the name of the tenant you created earlier in this process.

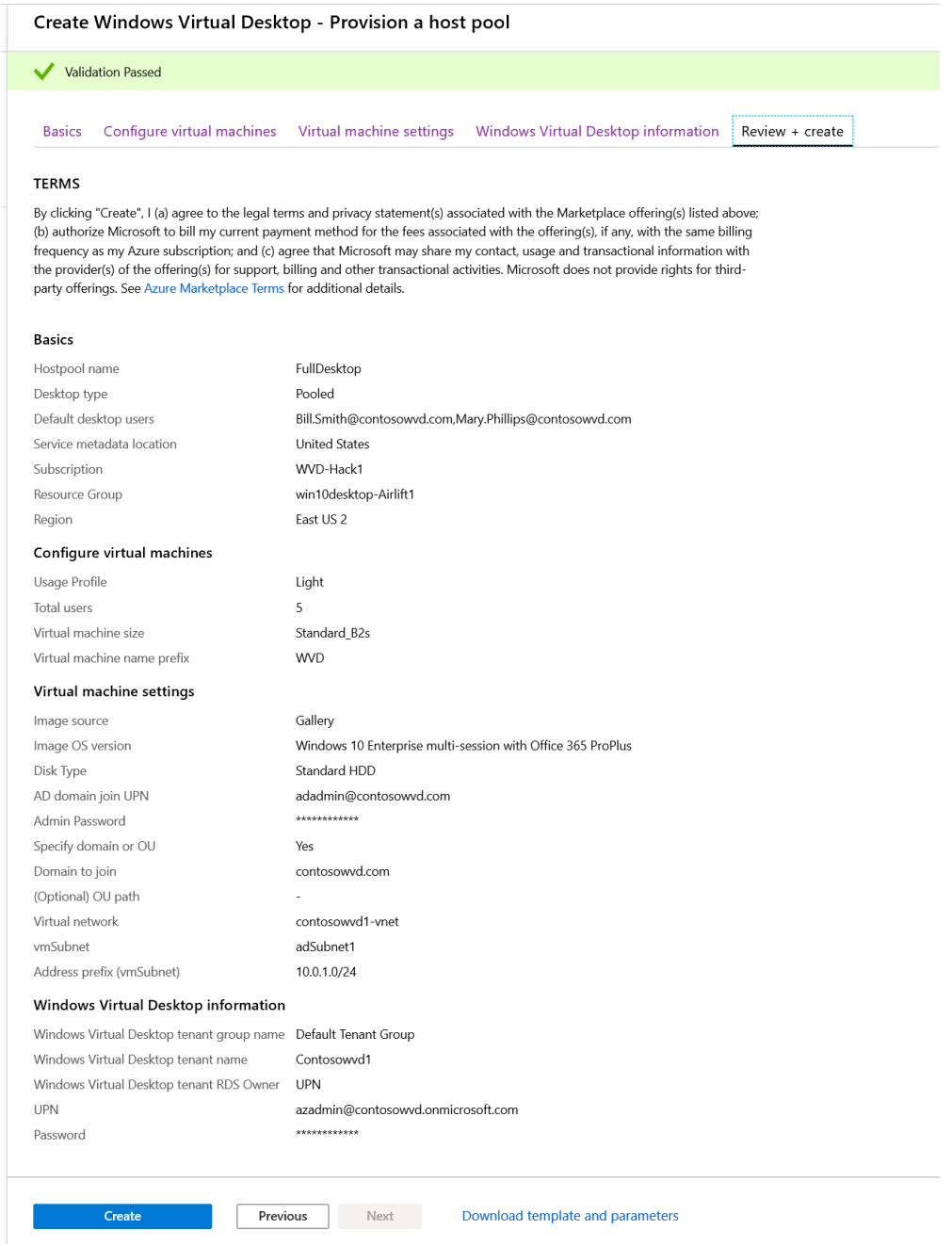
Note: If you are unsure what your Windows Virtual Desktop tenant name is, use the PowerShell command “Get-RdsTenant” to obtain it.

Enter valid credentials for your Azure AD environment (UPN and password).



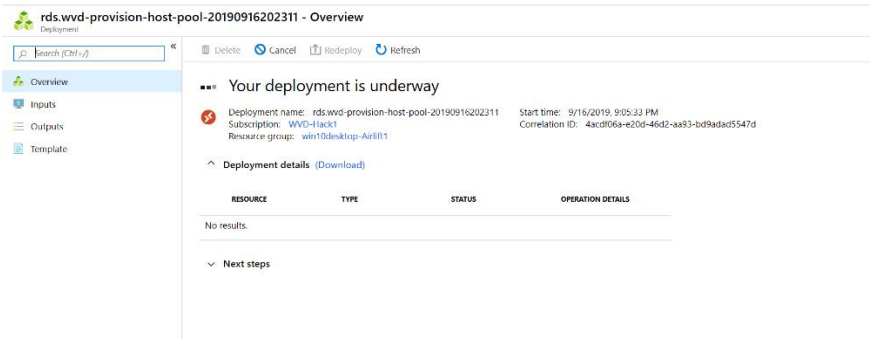
Task 5: Check the summary

Check the summary windows to see your setup passed validation, then click **OK**.

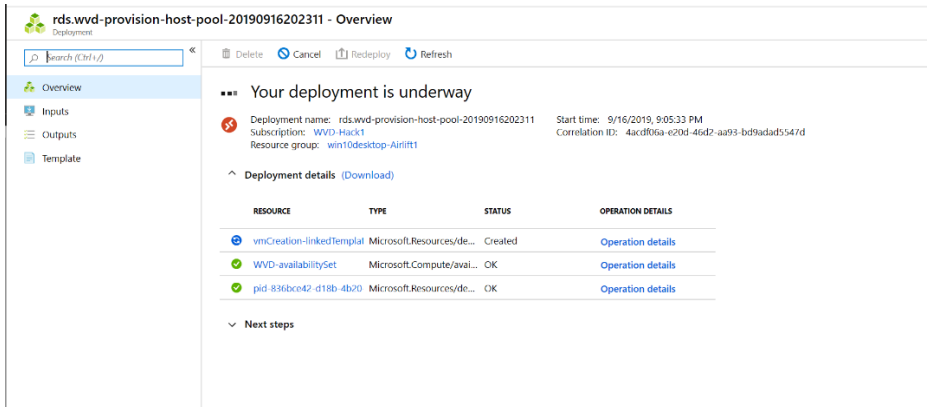


Task 6: Finalize the creation of your host pool

Hit **Create**, sit back, and relax. Wait for the deployment to finish. The process takes roughly 20 minutes.



You can watch the deployment continue to progress in the view



NOTE: Customize RDP Properties

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 PowerShell -CustomRdpProperty parameter in the Set-RdsHostPool cmdlet. For more details check out the link below:

https://docs.microsoft.com/en-us/azure/virtual-desktop/customize-rdp-properties

Supported RDP Properties can be found here:

https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/rdp-files

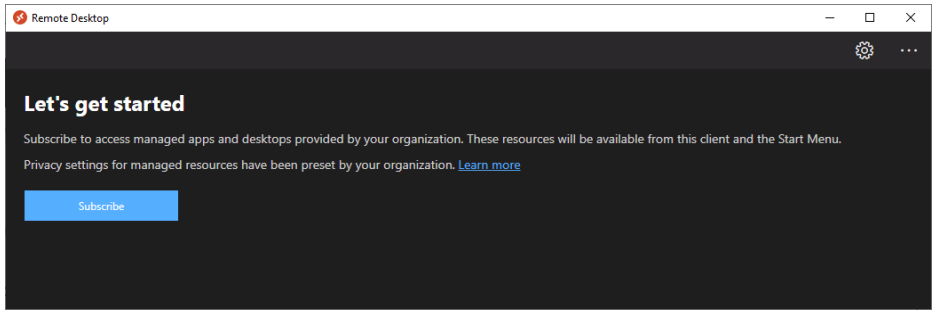
Exercise 10: Test WVD Deployment

Once you have created your Windows Virtual Desktop host pool, you can download the client for Android or Windows, or use the HTML5 client. (The Microsoft Remote Desktop Beta for iOS can be tested using TestFlight.) Here's how to test with Windows or the RDP client.

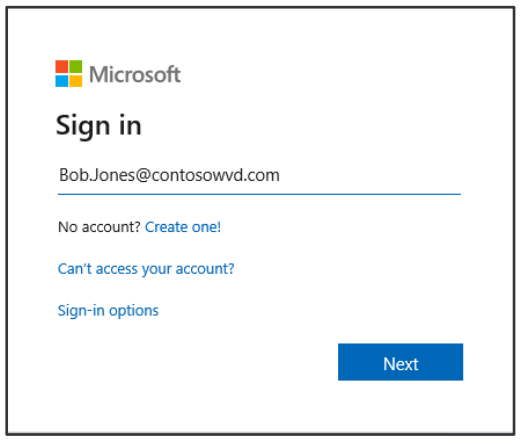
Task 1: Test with the Windows client

Download the latest Windows Remote Desktop client: Once you click subscribe you will be prompted to login. This login should be a VALID WVD user that has been synced from Active Directory to Azure Active Directory and have either a Desktop or Application group assigned to them. From the above steps we added bill smith to the 'Default desktop users' group for example. Once you login you will find the virtualized apps and desktops in the Start menu on a Windows 10 machine.

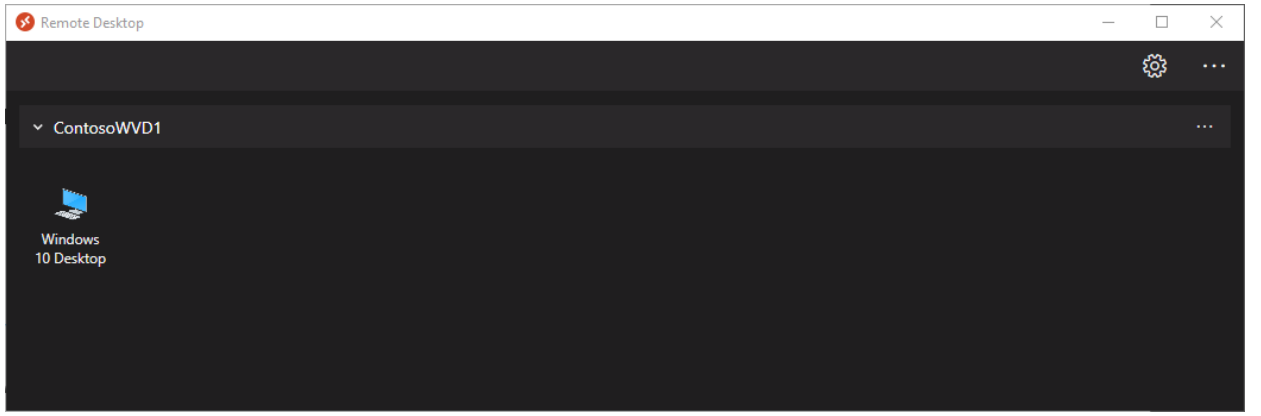
Download the latest Windows Remote Desktop Client. **Install**. Open the app, then click **Subscribe**.



User Name: bob.jones@contosowvd.com or any one of the users created in the AD deploy process
Password: **WVD@zureL@b2019!**

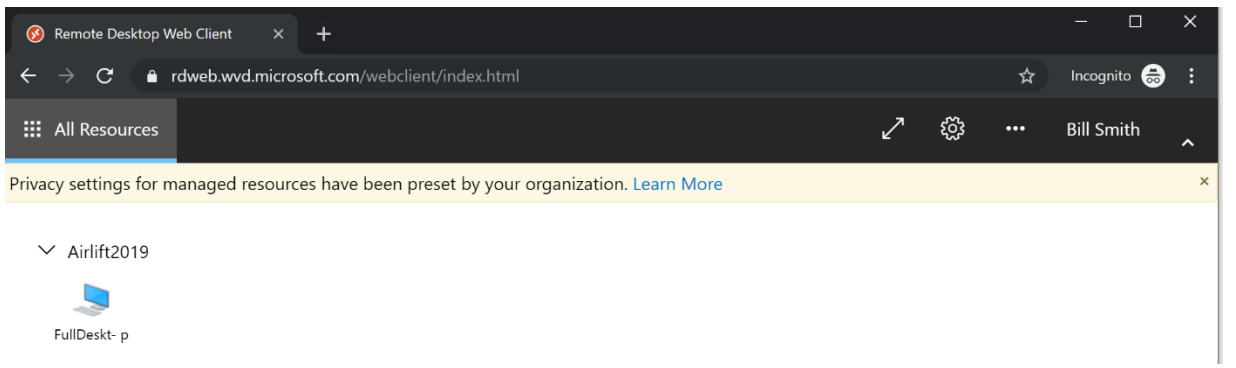


Task 2: Test with the HTML5 client



Launch a browser in privacy or incognito mode and visit http://aka.ms/wvdweb to access the HTML5 client. Authenticate using the login information to which you assigned a full desktop session.

When connecting to the desktop be sure to allow access to the clipboard/Printers if you need to use those devices.



If you are successful, you should be able to view the desktop:

Exercise 11: FSLogix Agent Installation

Task 1: 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. If you still have the previous RDP session to the domain controller please maximize that session. If not, then please refer to Exercise 4 task 1 to find the Public IP of your DC.
2. In Active Directory Users and Computer (ADUC) navigate to the **OrgUsers** OU. Create a **New Global Group** with a Group Type of **Security** called **WVD**.
3. Add the four AD users from the ORGUsers OU into the new group called **WVD**
4. This security group will be used to authenticate the Windows Virtual Desktop users to the file share virtual machine you just created.
5. On the domain controller create a folder on the **C drive** called **WVDFSPROF** that will be used as the profile share.
6. Right-click the new folder, select **Properties**, select **Sharing**, then select **Advanced sharing...**.
7. Select **Share this folder**, select **Permissions...**, then select **Add...**.
8. Search for the **WVD** group to which you added the Windows Virtual Desktop users, then make sure that group has **Full Control**.
9. After adding the security group, right-click the folder, select **Properties**, select **Sharing**, then copy down the **Network Path** to use for later.

Task 2: Installing the FSLogix agent

Installing the FSLogix agent can be done a few different ways. In this exercise we will install it via double hop RDP method where we will leverage the existing RDP session to the domain controller and then from there RDP into the session hosts to install the agent.

1. From within the RDP session of the domain controller open **MSTSC.** Establish an RDP session to the first session host which should be 10.0.1.5, however you should verify the IP is correct by locating it in the Azure portal.
2. Login with the domain administrator credentials **contosowvd\adadmin** and password of **WVD@zureL@b2019!**
3. In the previous labs the end user Bob Jones was able to login to the session host. Once that user logged in a locally cached profile was created. For the FSLogix profile container to take priority over the locally cached profile, you will need to **Delete** the locally cached profile for user **C:\Users\BJones**. *Note: You may need to stop the Windows Search service prior to deleting the profile.*
4. Launch an internet browser and navigate to this link to download the FSLogix agent.
5. **Unzip** the file.
6. Navigate to \\X64\Release in the .zip file and run **FSLogixAppsSetup** to install the FSLogix agent.
7. Navigate to **Program Files** > **FSLogix** > **Apps** to confirm the agent installed.
8. From the start menu, run **RegEdit** as an administrator. Navigate to **Computer\HKEY_LOCAL_MACHINE\software\FSLogix\Profiles**
9. Create\Set the following registry keys

Name	Type	Data/Value
Enabled	DWORD	1
VHDLocations	Multi-String Value	"Network path for file share"
SizeInMBs	DWORD	"integer for size of profile" for example 3000
IsDynamic	DWORD	1
LockedRetryCount	DWORD	1
LockedRetryInterval	DWORD	0

For VHDLocations, use the share folder previously created on the DC.

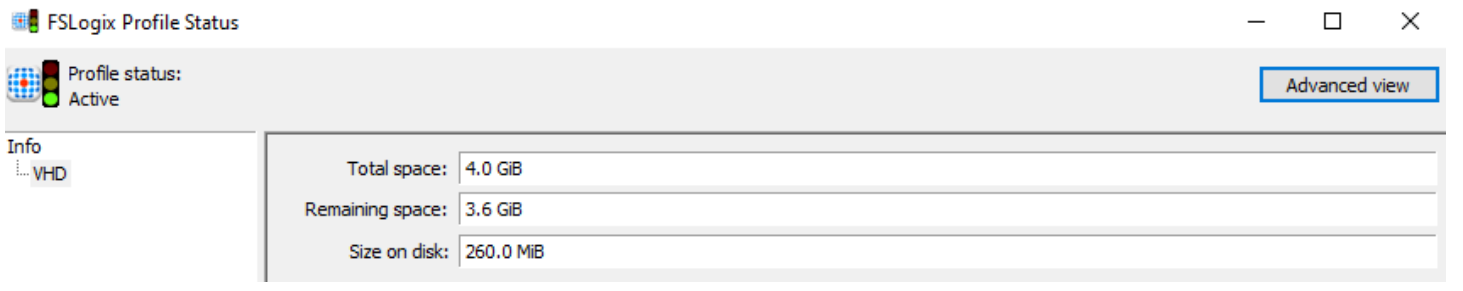
```
#Add FSLogix settings
New-Item -Path HKLM:\Software\FSLogix\ -Name Profiles –Force
Set-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "Enabled" -Type "Dword" -Value "1"
New-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "VHDLocations" -Value \\share\volume -PropertyType MultiString
-Force
Set-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "SizeInMBs" -Type "Dword" -Value "32768"
Set-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "IsDynamic" -Type "Dword" -Value "1"
Set-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "LockedRetryCount" -Type "Dword" -Value "1"
Set-ItemProperty -Path HKLM:\Software\FSLogix\Profiles -Name "LockedRetryInterval" -Type "Dword" -Value "0"
```

If you want to know what the settings are for go check this web page:
https://docs.microsoft.com/en-us/fslogix/configure-profile-container-tutorial

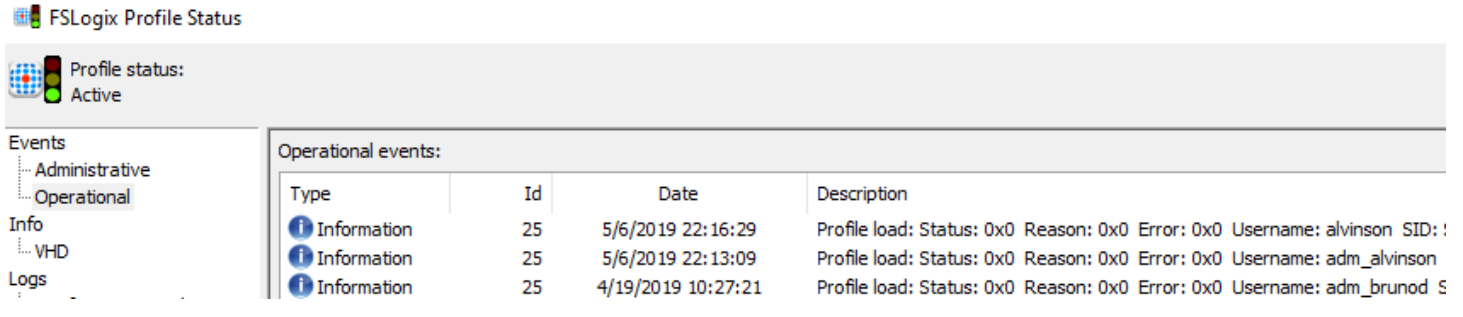
10. Restart the session host at this time and login again to create the user profile. Login as the user:
    1. User Name: bob.jones@contosowvd.com
    2. Password: **WVD@zureL@b2019!**

11. You can get a status if FSLogix agent works by launching FRXTray that in: “C:\Program Files\FSLogix\Apps”
12. Then click on the traffic light that is in the tray. The light should be green.

Note: Sometimes the Profile status may not show green right away.



13. Click on "**Advanced view**"
14. Go to "**Operational**" and look if you have no issue in the logs.

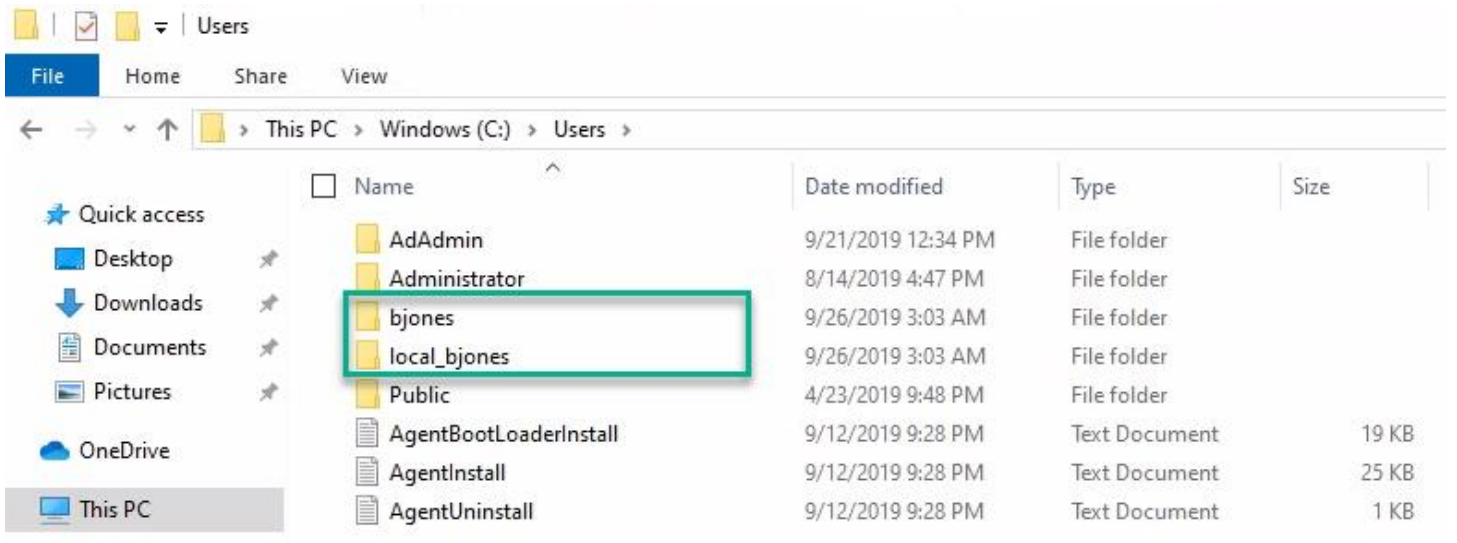


15. The agent needs to be installed on **each WVD session host**. Repeat steps 1

Task 3: Testing user profile containers with the FSLogix agent

Once the FSLogix agent has been installed a session needs to be established to the session host to verify that the profile container is mounted.

1. Launch a browser in privacy or incognito mode and visit http://aka.ms/wvdweb to access the HTML5 client.
2. Authenticate using the login information
    1. User Name: bob.jones@contosowvd.com
    2. Password: **WVD@zureL@b2019!**
3. Click the Windows 10 Desktop Icon. Click Allow for Local Resources
4. Once logged into the WVD session as Bob Jones, Navigate to C:\windows\users
5. If the Fslogix agent is contacting the shared path there will be two directories for Bob Jones.
    1. Local_bjones is the local Windows User Profile
    2. Bjones is the VHD attached profile mounted via the filter driver for FSLogix



3. The VHDX mounted profile container can be seen also by going to the \\servername\Sharename created earlier.
4. If the VHDX is there with a SID after the user logs in, then the process is complete
    A. If the VHD is not there, then open the log file **C:\ProgramData\FSLogix\Logs\Profile** from the session host the user is logged into and look for potential errors, resolve, then test user login again.
6. Once step 4 is complete we can test a simple file redirection to the VHDX file.
7. On the desktop of Bob Jones create a **SuperSecretWorldDomination.docx** file or similar

8. To explore the VHD that is mounted to the user Bob Jones in the **run** window type **diskmgmt.msc** and then click **Run As administrator** then enter the domain admin credentials
   Note: this is should NOT be done in a customer's production environment and solely for the purpose of education.
9. Locate the Disk 2 or similar that is the 29GB or so VHD container in disk management.
10. Add a drive letter **E:** and mount the VHD drive
11. Open **explorer** located the newly mounted E: drive
12. Navigate to E:\Profile\Desktop
13. The SuperSecretWorldDomination.docx file will appear.
14. From the desktop of Bob Jones, **delete** the SuperSecretWorldDomination.docx file. The file will then disappear from E:\Profile\Desktop
15. From Diskmgmt.msc detach the E: drive.

Task 4: Install DICOM viewer

Use Application Masking to manage user access of installed components. Application Masking may be used in both physical and virtual environments. Application Masking is most often applied to manage non-persistent, virtual environments, such as Virtual Desktops. In order to test and apply application masking to a sample application like Notepad ++, the application needs to be installed.

1. In the previous exercise a HTML 5 session as a test user Bob Jones was established.
2. session to the desktop.
3. Download Dicom viewer to the session host from the link here
4. Locate the exe for Dicom exe and **Run as Administrator** to start the install
5. Run the DICOM viewer after the install
6. Click **Yes** to set DICOM viewer as the default.
7. On the desktop the MicroDicom app will appear
8. Repeat the installation of the DICOM application on any additional sessions hosts created by establishing an RDP session from your domain controller to the IP of those sessions hosts.

Task 5: Install FSLogix Rule Editor

In this next task a rule set for denying access to a Bob Jones will be applied for the recently installed application DICOM.

1. Install the **FSLogix Rule Editor**. This exe can be located in the zip file that was downloaded earlier in Task
2. Right click on the **FSLogixAppsRuleEditorSetup.exe** and **run as administrator** to start the install

Task 6: Create and Test Rule Set

1. **Run As Administrator** the FSLogix Rule Editor in the session **host C:\Program Files\FSLogix\Apps\RuleEditor.exe**
2. Click **File** then **New** to create a new Rule Set
3. Enter the name that you would like for your Rule Set. Call this **DICOM**
4. Click **Enter file Name** to create the rule set
5. Select the application that you would like to manage. Select **MicroDicom DICOM viewer 3.0.1**
6. Click **Scan** to have the Rules Editor detect the application settings
7. When scanning is complete, Click **OK**
8. The Rules Editor now shows your first Rule Set

Task 6: Assign users to Rule Set

1. Click **File** then **click manage assignments**
2. **The everyone group is who the rule set is applied to.** Click **remove** and click **Add.** Click **User**
3. Look up **Bob Jones** account in AD**.** Click **Ok** to apply this **rule.**
4. Click **File** and **Apply Rules to System**
5. The Rules that are within your Rule Set will be applied to your system. The icon for DICOM will disappear from Bob Jones desktop.
6. Apply the reverse process to make the application appear back for Bob Jones on his desktop.

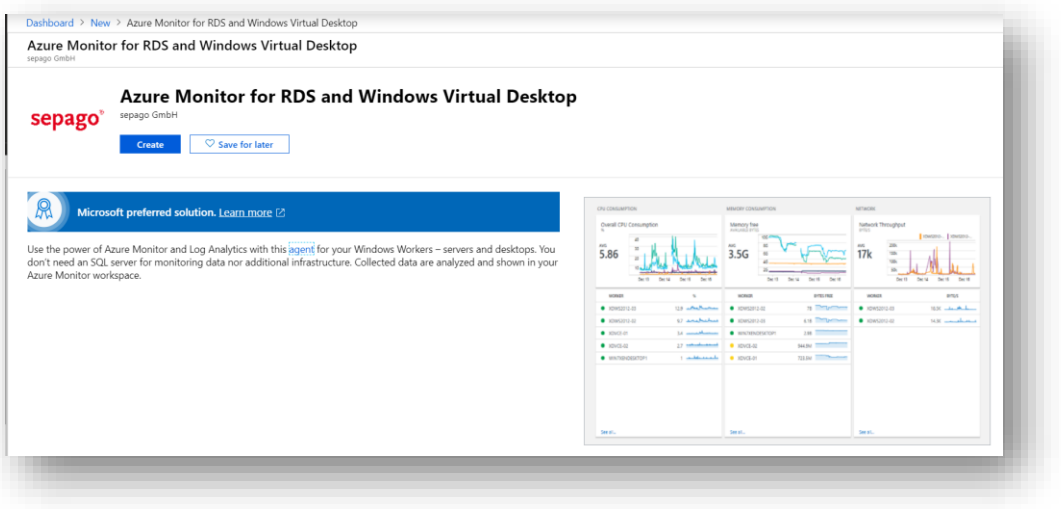
Exercise 12: WVD Monitoring

The following describes the process for you to set up monitoring for your WVD environment. This solution uses a 3rd party partner available in our Azure Gallery. This is referred to as a 'Microsoft Preferred Solution'. A Microsoft preferred solution is a cloud application selected for its quality, performance, and ability to address customer needs in a certain industry vertical or solution area. A team of Microsoft experts validates solutions from partners with specific proven competencies and capabilities. These solutions are featured in our cloud marketplace storefronts, Azure Marketplace, and AppSource, as well as in the Azure portal. Preferred solutions on AppSource can be discovered by industry verticals. Preferred solutions on Azure Marketplace and the Azure portal can be discovered across horizontal solution categories.

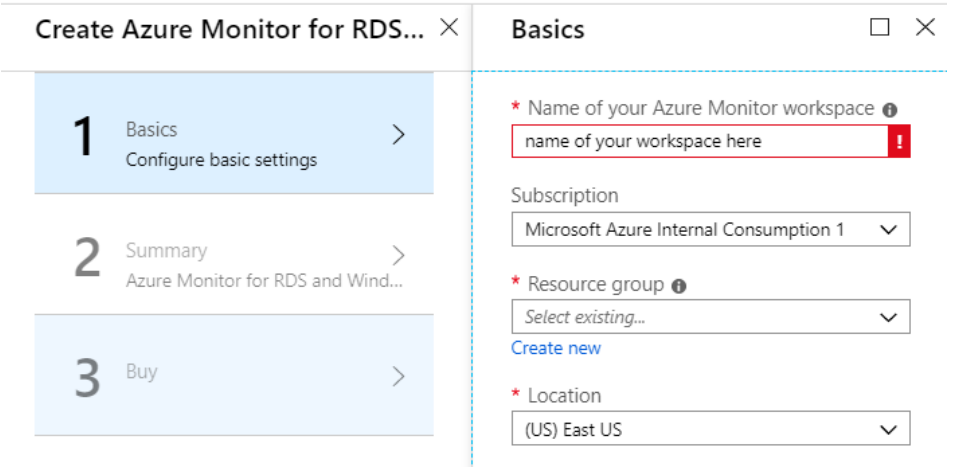
The Sepago Agent monitors each worker in your RDS or Citrix environment. The agent is focused on events, performance consumption, network activities and more regarding each user's IT experiences. Workers in this context are Windows Remote Desktop Server or Windows 10\MS, XenApp Servers and of course Windows Client VDI's (XenDesktop). The agents combine data from different sources and send them to your OMS Log Analytics workspace in Azure.

Task 1: Installation for a new workspace

From the Azure Portal, you select create a resource and search for Azure Monitor for RDS and Windows Virtual Desktop. Make sure you deploy the workspace in **East US as that is where you can leverage the Interdependency agent** to show application and port mappings to your servers.

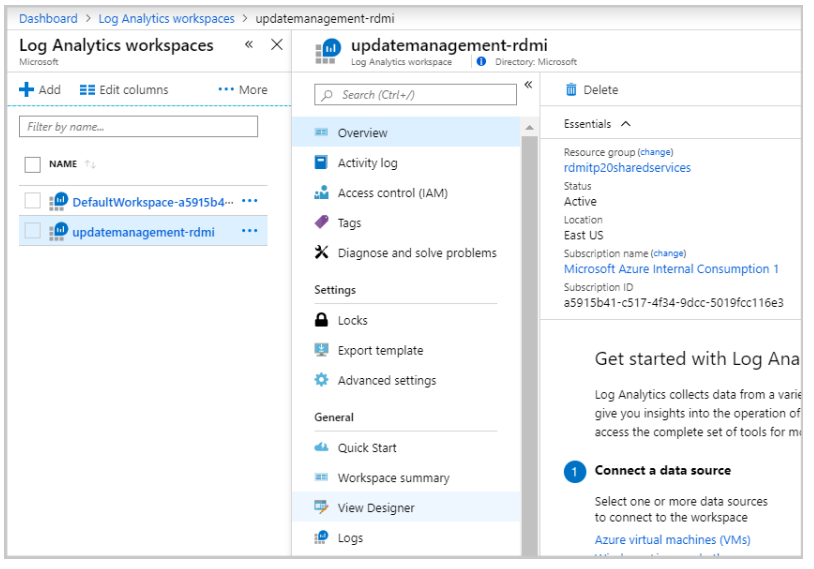


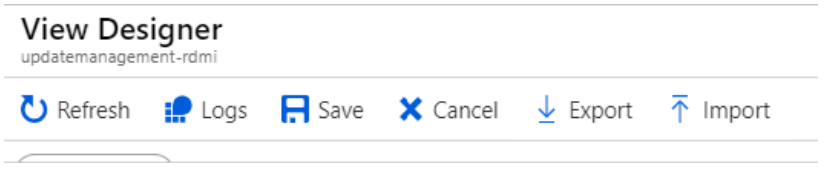
Click Create and run through the steps:



Task 2: Installation for an already existing workspace

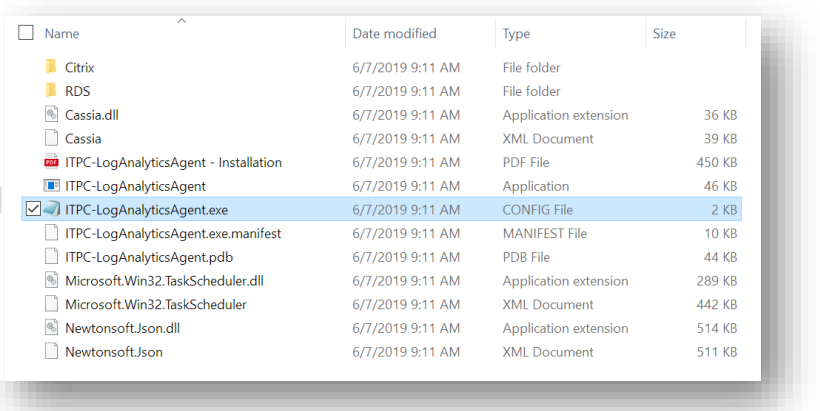
If you want to leverage an already existing Log Analytics workspace in **EAST US**, then all you need to do is import the views which can be downloaded from here, https://github.com/MarcelMeurer/LogAnalytics-for-Citrix-and-RDS . Extract the views to a temp folder for now. Open up Log Analytics workspace and in the middle blade select view designer. Click import and browse to the view files you downloaded.

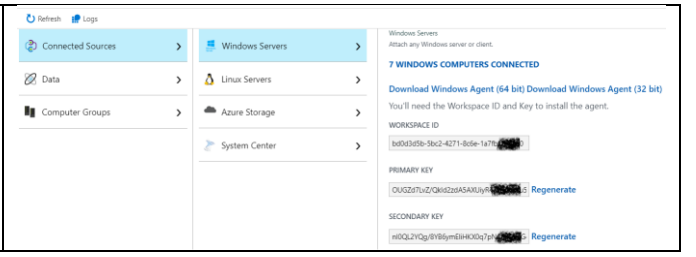




Task 3: Agent Installation

Once you have the workspace created above or you are using a workspace already built you need to deploy the agents to all your session hosts and\or add the agent to a master image. You can grab the agent from here. Once you have it downloaded and extracted you need to configure it for your Log Analytics Workspace. Find the file marked 'config' at the end. Open it in notepad or what editor you like. Change the two key values "Customer ID and Shared Key".



Customer ID: Workspace ID Shared Key: Primary Key	


```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2"/>
  </startup>
  <appSettings>
    <add key="CustomerId" value="bd0d3d5b-5XXXXXXXXXfba84c320"/>
    <add key="SharedKey" value="BT7nP/U1b/g9ICFspp7JPep7hNXXXXXXXXXXXXXXWuU6pU4hXkiVmD6zsYJZBMfjDA=="/>
    <add key="UpdateIntervalInSeconds" value="60"/>
```

Open a cmd window as administrator. Change directory to where the agent has been copied.
ITPC-LogAnalyticsAgent.exe -test

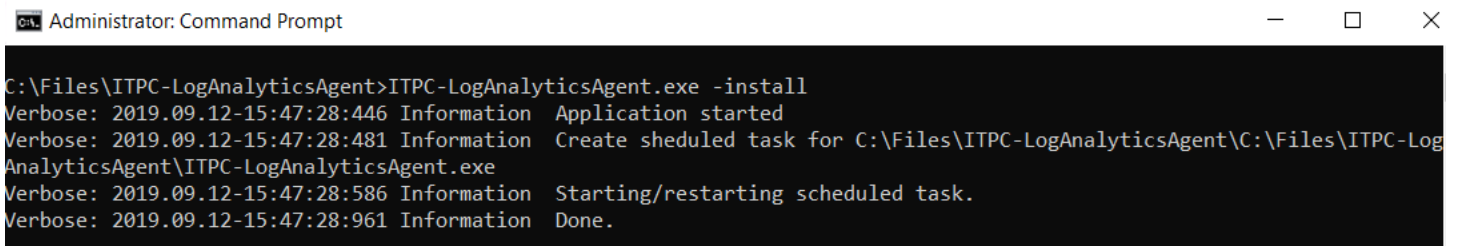
If you see the below output stating "error" there is a potential communication problem , either the workspace ID or Key is incorrect or the session host can't communicate to the LA Workspace due to networking.

```
P-Tcp 1
Verbose: 2019.09.12-15:42:25:027 Information  PerfomanceCounter: Adding counter: RemoteFX Network,Tot
p 1
Verbose: 2019.09.12-15:42:25:034 Information  PerfomanceCounter: Adding counter: RemoteFX Network,Cur
p 1
Verbose: 2019.09.12-15:42:25:037 Information  PerfomanceCounter: Adding counter: RemoteFX Network,Cur
RDP-Tcp 1
Verbose: 2019.09.12-15:42:25:117 Information  Stop init



Entering test mode. Try to send data to LogAnalytics

UseProxy:               False
ProxyUri:               127.0.0.1:8088
ProxyNeedAuthentication: True
ProxyUserName:          user
ProxyPassword:          pwd

Sending test data
Verbose: 2019.09.12-15:42:25:276 Error        LogAnalytics: Cannot send data
                                              Error Message:
                                              The remote server returned an error: (403) Forbidden.
Verbose: 2019.09.12-15:42:25:286 Error        LogAnalytics: Is proxy enabled: False
Done.
```

Make sure all is working as expected, Sending Test data ‘DONE’.

```
p 1
Verbose: 2019.09.12-15:46:30:308 Information  PerfomanceCounter: Adding counter: RemoteFX Network,(
RDP-Tcp 1
Verbose: 2019.09.12-15:46:30:313 Information  PerfomanceCounter: Adding counter: RemoteFX Network,T
P-Tcp 1
Verbose: 2019.09.12-15:46:30:318 Information  PerfomanceCounter: Adding counter: RemoteFX Network,T
p 1
Verbose: 2019.09.12-15:46:30:328 Information  PerfomanceCounter: Adding counter: RemoteFX Network,(
p 1
Verbose: 2019.09.12-15:46:30:333 Information  PerfomanceCounter: Adding counter: RemoteFX Network,(
RDP-Tcp 1
Verbose: 2019.09.12-15:46:30:413 Information  Stop init



Entering test mode. Try to send data to LogAnalytics

UseProxy:               False
ProxyUri:               127.0.0.1:8088
ProxyNeedAuthentication: True
ProxyUserName:          user
ProxyPassword:          pwd

Sending test data
Done.

C:\Files\ITPC-LogAnalyticsAgent>
```

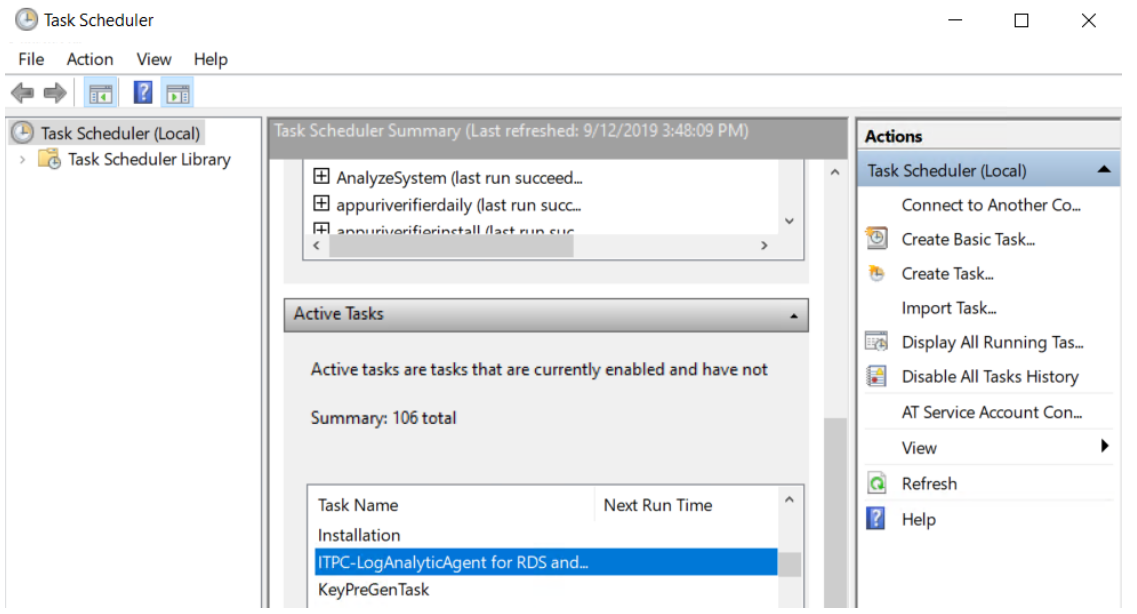
If it is, then execute this command,
ITPC-LogAnalyticsAgent.exe -install



The time it takes to see the data varies, but in my experience its about 24-36 hours.

Please reboot the session host in order for metrics to start to collect.

If you look at the task scheduler you should see the below:



You can then publish a new dashboard for example below:

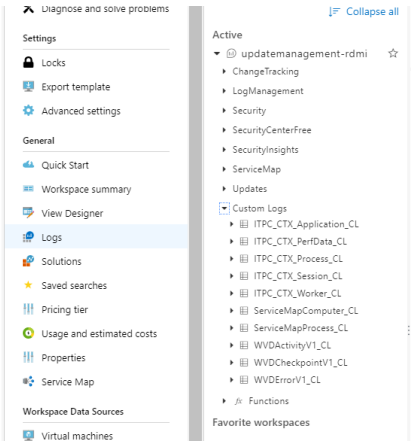
Task 4: Windows Virtual Desktop Service Tenant injection – Log Analytics

Taking your Log Analytics workspace ID and primary key, you can apply this to your Windows Virtual Desktop Tenant(s) by following the below. Open PowerShell with an elevated account if you do not have an already open session.

```
Import-module Microsoft.RDInfra.RDPowershell.dll

Add-RDSAccount -DeploymentUrl https://rdbroker.wvd.microsoft.com

Set-RdsTenant -Name "jojenner" -LogAnalyticsWorkspaceId "96f3255d-547f-4df4-XXXX-fef7e" -LogAnalyticsPrimaryKey "sBN
WCfjb92E19dYHcGsXY4o3Lck0aDPUP3NOogrZC8QoTwV+ZBQXXXXXXXXXo1Mu3eZExEyZXW4fB="
```

```
PS C:\WINDOWS\system32> get-rdstenant "jojenner"


TenantGroupName        : Default Tenant Group
AadTenantId            : XXXXXXXX
Description            : John Jenner WVD Environment
FriendlyName           :
SsoAdfsAuthority       :
SsoClientId            :
SsoClientSecret        :
AzureSubscriptionId    :
LogAnalyticsWorkspaceId : 96f3255d-547f-4df4-9a55-fef7XX9808bc
LogAnalyticsPrimaryKey  : ********
```

Once you have executed the PowerShell command above, login a few times to your session hosts and click some applications. Add a user to an application group. Try and login with an account you know does not have permissions as well. Browse to your log analytics workspace and look for logs in the middle blade. Expand the tree under ‘custom logs’ and you should see something like the below: WVDActivityV1, WVDCheckpointV1 and WVDErrorV1.



The other ITPC ones listed are from the above steps (Sepago). If you have followed this document fully, you should see similar items to what is above in custom logs providing you have errors in WVD to report into this log (WVDErrorV1_CL).

Here are some sample queries that have been requested from our clients:

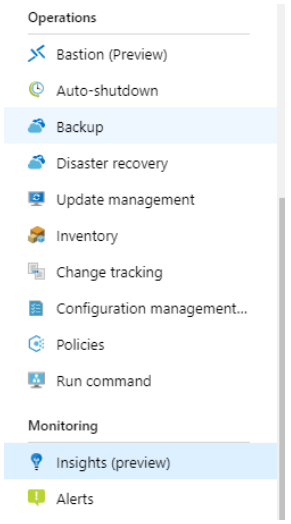
```
ITPC_CTX_Process_CL
| summarize avg(PercentProcessorTime_d) by Name_s, TimeGenerated
| where Name_s != "System Idle Process"
| order by avg_PercentProcessorTime_d desc
| take 20
| render timechart


ITPC_CTX_Worker_CL | summarize count() by VmSize_s | render piechart


ITPC_CTX_Process_CL
| summarize max(PercentProcessorTime_d) by Name_s, TimeGenerated | where Name_s != "System
Idle Process"
| order by max_PercentProcessorTime_d desc | take 20
| render timechart


ITPC_CTX_PerfData_CL
| distinct DesktopGroup_s

WVDActivityV1_CL
| limit 50

WVDCheckpointV1_CL
| limit 50

WVDErrorV1_CL
| limit 50
```

Task 5: Interdependency Agent Metrics

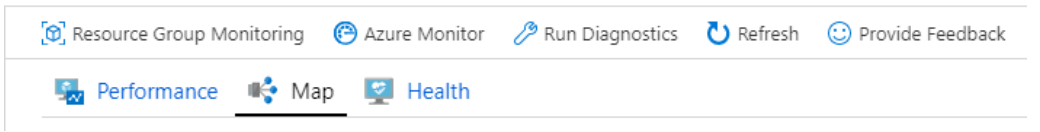
The Map feature in Azure Monitor for VMs gets its data from the Microsoft Dependency agent. The Dependency agent relies on the Log Analytics agent for its connection to Log Analytics. Your system must have the Log Analytics agent installed and configured with the Dependency agent. https://docs.microsoft.com/en-us/azure/azure-monitor/insights/service-map

Whether you enable Azure Monitor for VMs for a single Azure VM or you use the at-scale deployment method, use the Azure VM Dependency agent extension to install the agent as part of the experience.

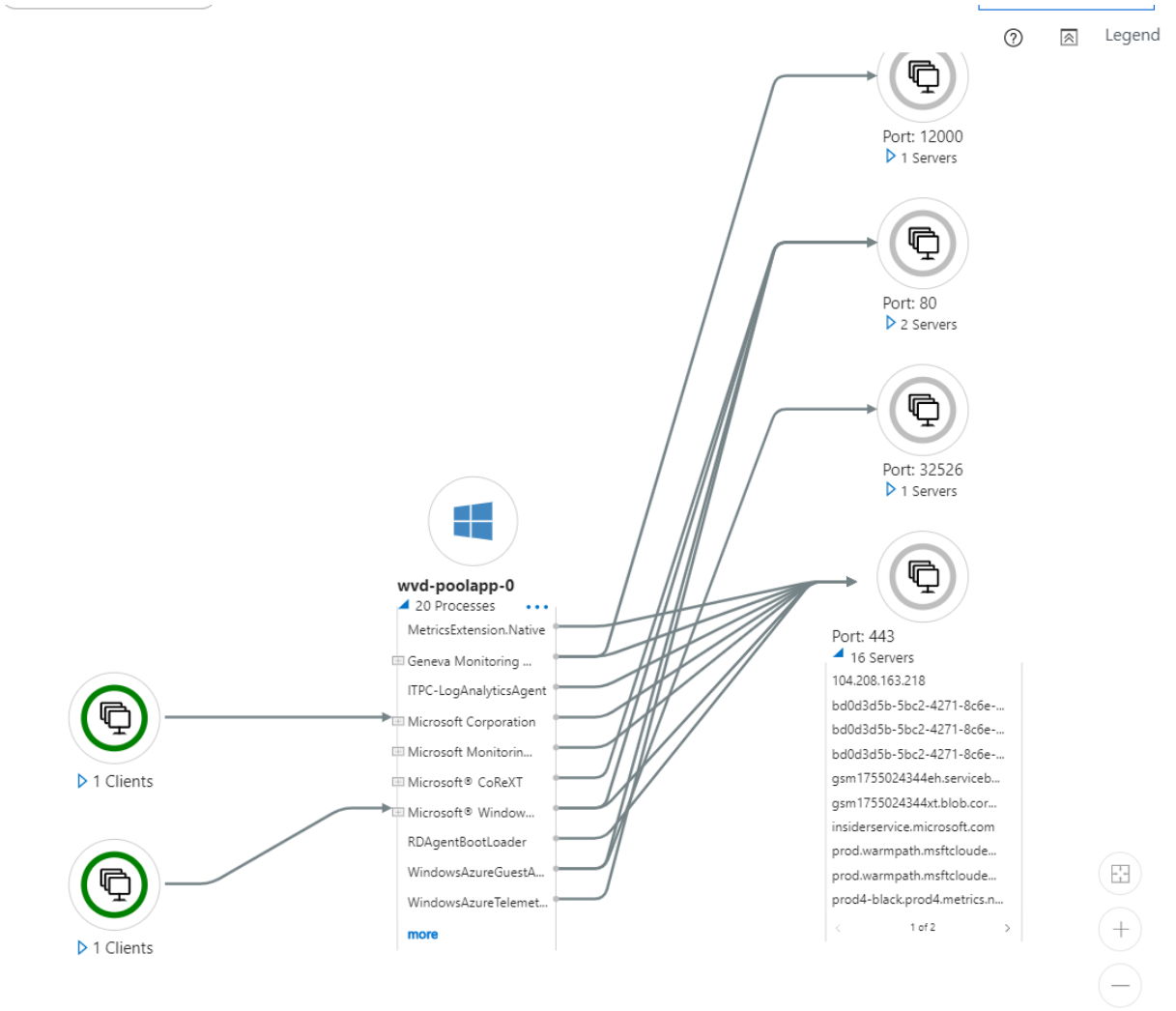
Once you install this agent on your session hosts or golden image, you can show the below data to your clients by going to the virtual machine object, and in the middle blade selecting ‘insights’.



Once you click ‘insights’ a menu on the right will show,



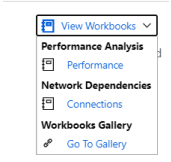
It is the MAP we are interested in for this exercise. You can also use the other 2 options to show value add (Performance and Health). You might be prompted to enable the solution, go ahead and enable it and then wait a few minutes for the data to start flowing through.

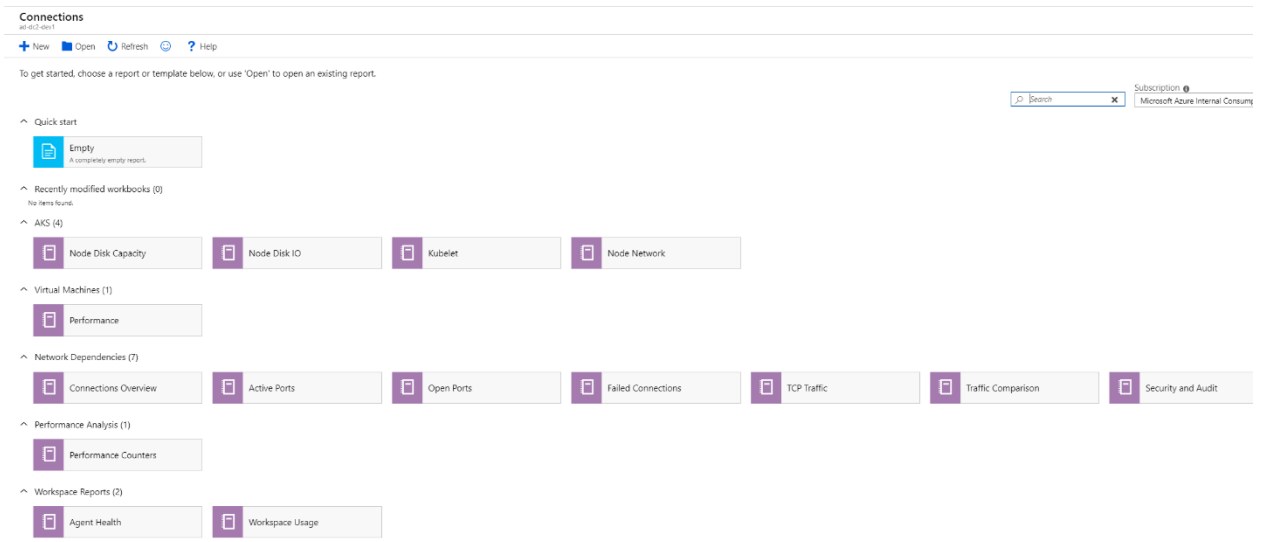


In the middle we can see the processes running on our session host and where they are communicating to.

Port 443, these are all outbound calls from your session host (wvd-poolapp-0)

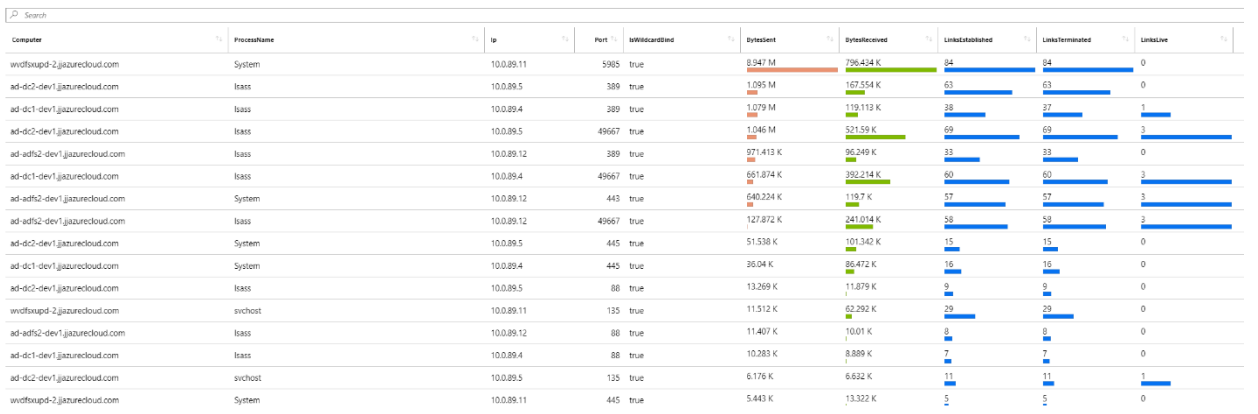
If you click the top right box,
For example, network dependencies will show you these options





Click the active ports for example and you will see the below data,

Port activity by Computer, Process, IP and Port



Search

Computer	ProcessName	Ip	Port	IsWildcardBind	BytesSent	BytesReceived	LinksEstablished	LinksTerminated	LinksLive
wvdfsxupd-2.jjazurecloud.com	System	10.0.89.11	5985	true	8.947 M	796.434 K	84	84	0
ad-dc2-dev1.jjazurecloud.com	lsass	10.0.89.5	389	true	1.095 M	167.554 K	63	63	0
ad-dc1-dev1.jjazurecloud.com	lsass	10.0.89.4	49667	true	1.079 M	119.113 K	38	37	1
ad-dc2-dev1.jjazurecloud.com	lsass	10.0.89.5	49667	true	1.046 M	521.59 K	69	69	3
ad-adfs2-dev1.jjazurecloud.com	lsass	10.0.89.12	389	true	971.413 K	96.249 K	33	33	0
ad-dc1-dev1.jjazurecloud.com	lsass	10.0.89.4	49667	true	661.874 K	392.214 K	60	60	3
ad-adfs2-dev1.jjazurecloud.com	System	10.0.89.12	443	true	640.224 K	119.7 K	57	57	3
ad-adfs2-dev1.jjazurecloud.com	lsass	10.0.89.12	49667	true	127.872 K	241.014 K	58	58	3
ad-dc2-dev1.jjazurecloud.com	System	10.0.89.5	445	true	51.538 K	101.342 K	15	15	0
ad-dc1-dev1.jjazurecloud.com	System	10.0.89.4	445	true	36.04 K	86.472 K	16	16	0
ad-dc2-dev1.jjazurecloud.com	lsass	10.0.89.5	88	true	13.269 K	11.879 K	9	9	0
wvdfsxupd-2.jjazurecloud.com	svchost	10.0.89.11	135	true	11.512 K	62.292 K	29	29	0
ad-adfs2-dev1.jjazurecloud.com	lsass	10.0.89.12	88	true	11.407 K	10.01 K	8	8	0
ad-dc1-dev1.jjazurecloud.com	lsass	10.0.89.4	88	true	10.283 K	8.889 K	7	7	0
ad-dc2-dev1.jjazurecloud.com	svchost	10.0.89.5	135	true	6.176 K	6.632 K	11	11	1
wvdfsxupd-2.jjazurecloud.com	System	10.0.89.11	445	true	5.443 K	13.322 K	5	5	0

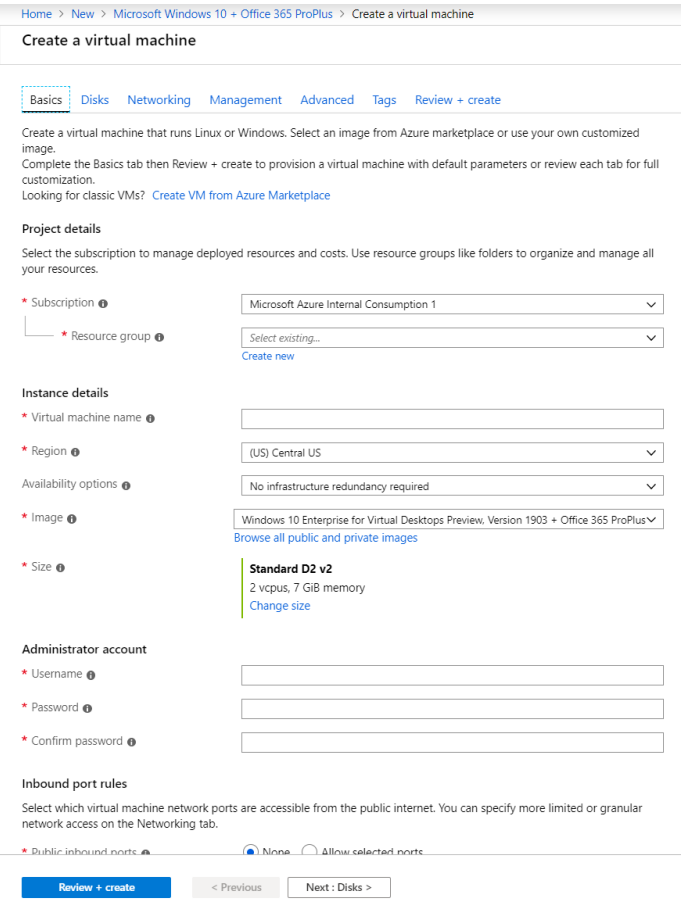
Exercise 13: Creating a WVD Master Image

There are many ways to create images and for WVD we are going to cover off on one method that is quite simple to use.

Task 1: Create a new server in Azure

In your Azure Portal, go to create a resource, and in the search field type “Microsoft Windows 10 + Office 365 ProPlus”

Once you pick which version you want, go ahead and build the server.



We are not going to cover off on building an azure server in detail, however, please make sure you open the public inbound ports for RDP when you create it.

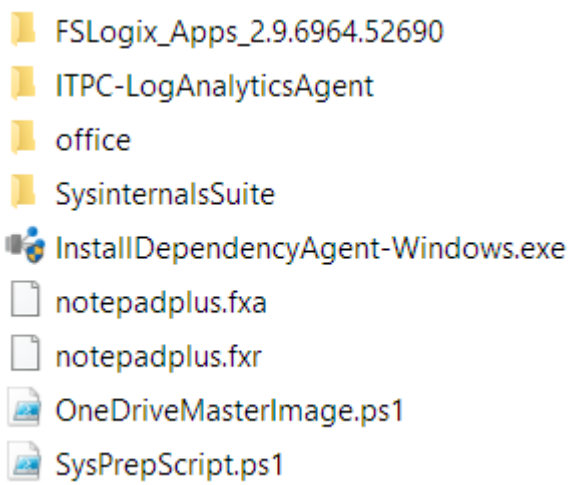
Once the server has been provisioned go ahead and login to the server with the credentials you supplied.

Task 2: Copy agent files over to your master image

The files discussed in this section can be found in the Airlift folder located here,

For WVD demos, the team has been copying over to the master image the following to C:\Files:

1) Interdependency agent
2) Sepago agent
3) Notepad ++
4) Office setup and custom config.xml \ OneDrive setup
5) FSLogix software (this will be part of our gallery image soon if not already) and sample notepad ++ rules
6) SysInternalsSuite
7) Powershell scripts



Task 3: Install agents and applications

To install Office and OneDrive on a master image for WVD please consult this link below.
https://docs.microsoft.com/en-us/azure/virtual-desktop/install-office-on-wvd-master-image

In the office folder above, onedrive.exe and officesetup.exe are already there with the appropriate .xml config. Use the OneDriveMasterImage script for the OneDrive install and the setup.exe for Office. There are notepad.txt instruction files and instructions in the master image prep script. You might notice when trying to remove OneDrive from CMD or PS that it does not actually remove. You will need to go into Add Remove programs and uninstall it from there.

The notepadplus.fxa\fxr need to be copied to the fslogix program files directory “C:\Program Files\FSLogix\Apps\Rules”

Task 4: Run Sysprep Script

Once you have completed installing all the agents needed, go ahead and open the SysPrepScript.ps1 in an editor

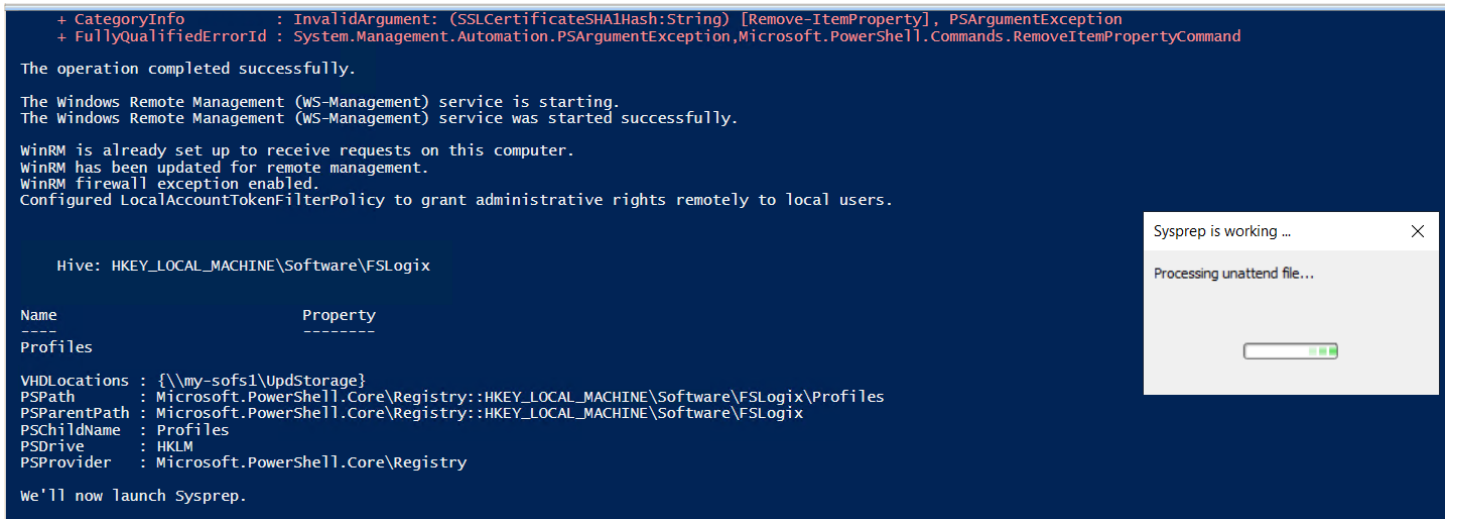
The sysprep PowerShell file consists of the registry settings posted on our public site here,
https://docs.microsoft.com/en-us/azure/virtual-desktop/set-up-customize-master-image

Once you have it opened, make sure to change the FSLogix location based on your configuration,

```
# This section is for OneDrive Configuration.
# Master Script for WVD Image Prep
# Script authors: Chris Nylen <Chris.Nylen@microsoft.com>; John Jenner <John.Jenner@microsoft.com>
#
#
#<Please run the first section for OneDrive if needed, then proceed to run the rest of the script>#

Set-ExecutionPolicy -ExecutionPolicy Unrestricted

# Uninstall OneDrive
Run c:\Files\office\OneDriveSetup.exe /uninstall

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

#Install OneDrive
Run c:\Files\office\OneDriveSetup.exe /allusers

REG ADD "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v OneDrive /t REG_SZ /d "C:\Program Files (x86)\Microsc

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

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



# ------------------------------Once the above is run continue to run the rest of the script-------------------------



# The following steps are from: https://docs.microsoft.com/en-us/azure/virtual-desktop/set-up-customize-master-image
# Set-ExecutionPolicy -ExecutionPolicy Unrestricted

# Set this variable to your FSLogix profile directory
$FSLUNC = "\\my-sofs1\UpdStorage"

Write-Host "This script will prepare your image for capture and eventual upload to Azure."
```

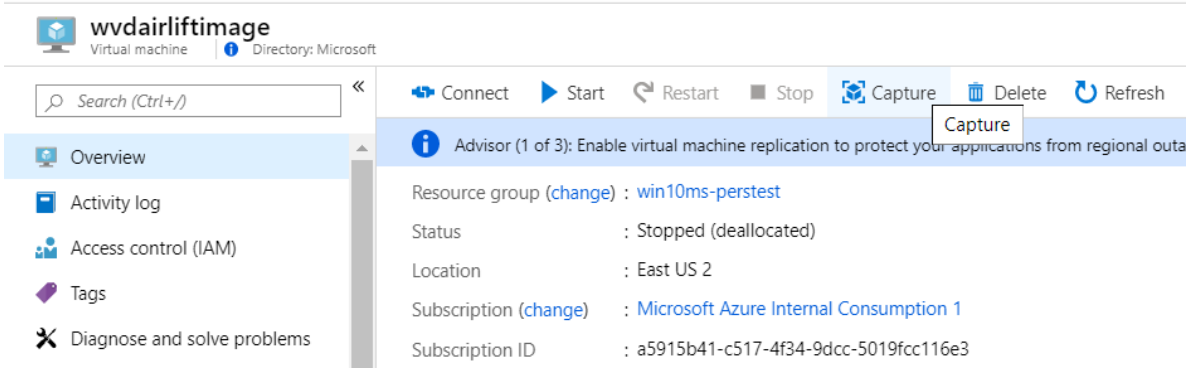
Once you have changed this setting, execute “Set-ExecutionPolicy -ExecutionPolicy Unrestricted” saying ‘yes to all’
Go ahead and run this script.

You will see some errors in red with access is denied, do not worry about those they are fine and expected.

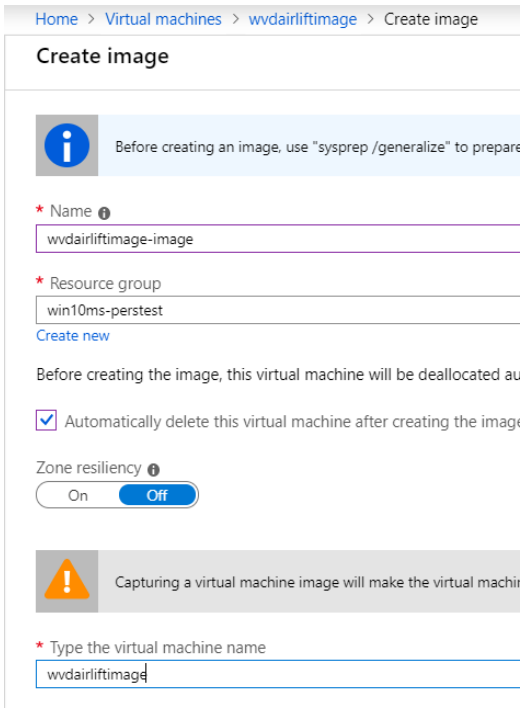


Task 5: Capturing the image

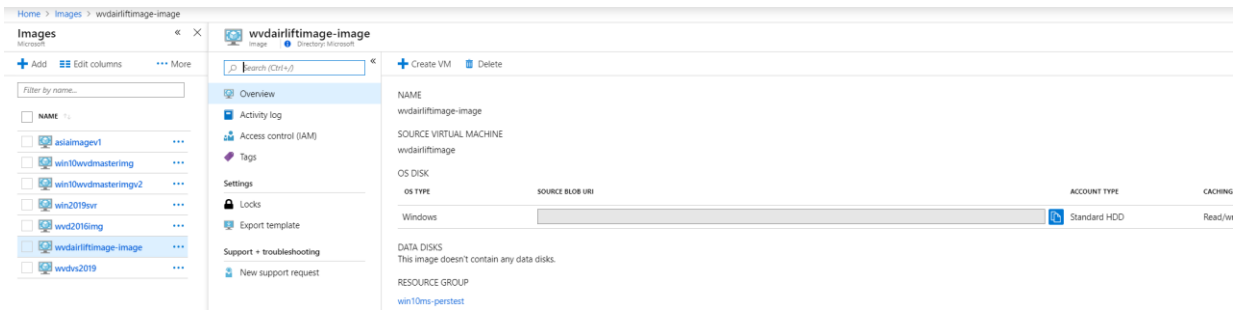
Go into the Azure Portal and find this server that you ran sysprep on, it should be in a stopped state. Please deallocate the server currently from a stopped state to a stopped state (deallocated). Once this is done, on the top of the same screen, click capture.



Once you click capture, fill in all the associated fields.

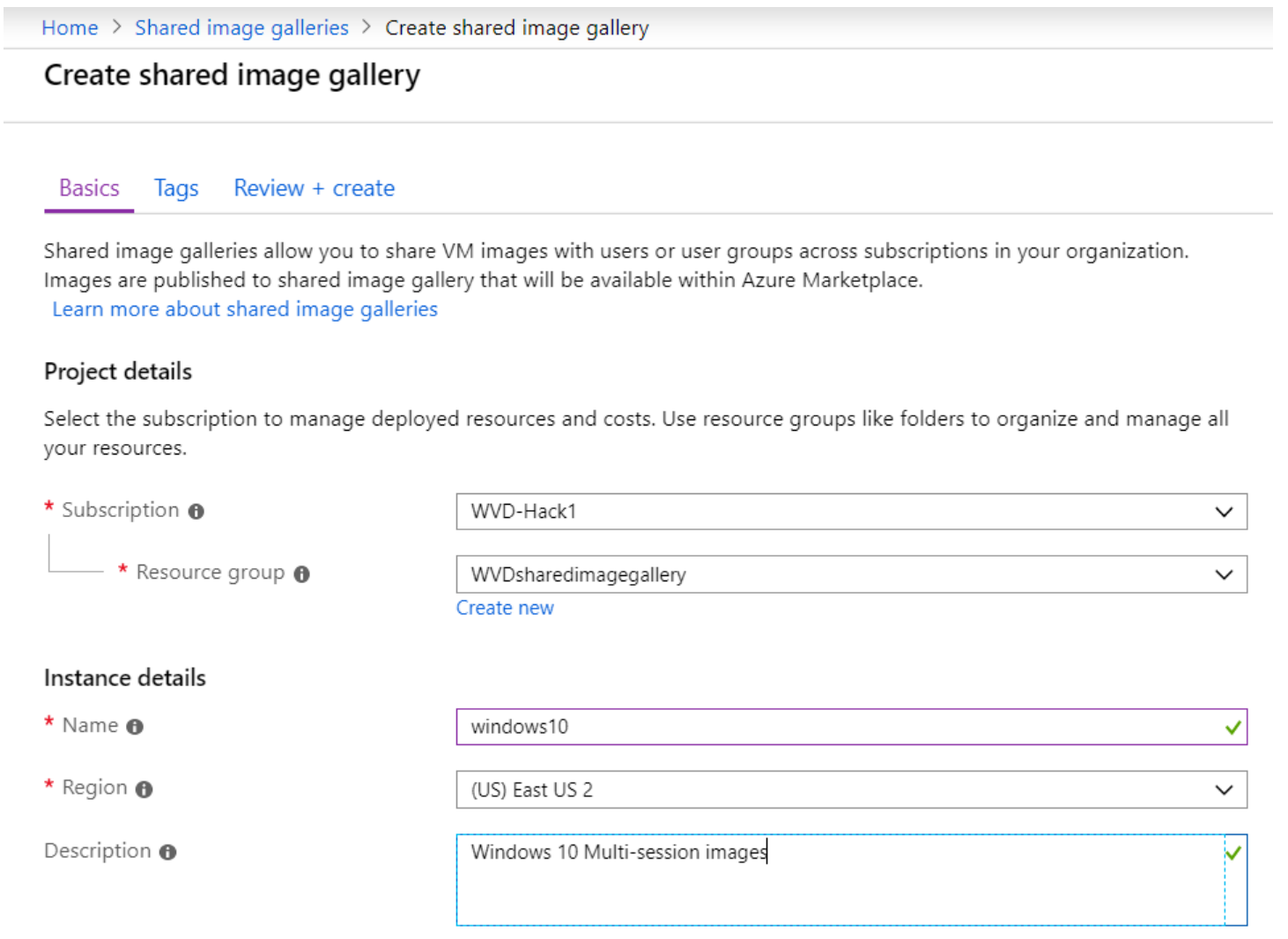


Once completed you should now see the image,



Task 6: Azure Shared Image Galleries

Launch Azure Shared Image Galleries from the Azure Portal. We are following this below link to our public documentation.
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/shared-image-galleries



Next step is to create an image definition,

Add new image definition to shared image gallery

Basics | Version | Publishing options | Tags | Review + create

Images are defined within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements.
Learn more about image definitions

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: WVD-Hack1
* Resource group: WVDsharedimagegallery

Instance details

* Region: (US) East US 2

Image definition details

* Target shared image gallery: windows10
* Image definition name: windows10master
* Operating system: Windows / Linux
* Publisher: Microsoft
* Offer: InformationTechnology
* SKU: Standard

Add new image definition to shared image gallery

Basics | Version | Publishing options | Tags | Review + create

Image version

An image version is what you use to create a VM when using a gallery. You can create a VM in any region where the image version is replicated. You can have multiple versions of an image as needed for your environment. Image versions can be used multiple times. Learn more about image versions and version replication.

Version name: 0.0.1

Location: (US) East US 2

Source image: baseimage-image-20190924182527

Exclude from latest: Yes / No

Image version end of life date: MM/DD/YYYY

Replication

An image version can be replicated to different regions depending on what makes sense for your organization. One example is to always replicate the latest image in multiple regions while all older versions are only available in 1 region. This can help save on storage costs for image versions.

* Default replica count: 1

TARGET REGIONS	TARGET REGION REPLICA COUNT	STORAGE ACCOUNT TYPE
(US) East US 2	1	Standard HDD
(US) West Central US	1	Standard HDD

Add new image definition to shared image gallery

Basics Version Publishing options Tags Review + create

Provide additional metadata about the image, including recommended VM specifications, and links to release notes and privacy policies.

Publishing meta data

EULA link

Description

Release notes URI

Privacy URI

Purchase plan name

Purchase plan publisher name

Purchase plan product name

VM deployment

Provide recommendations for VM specifications for this image. These recommendations are informational only, and do not constrain VM specification.

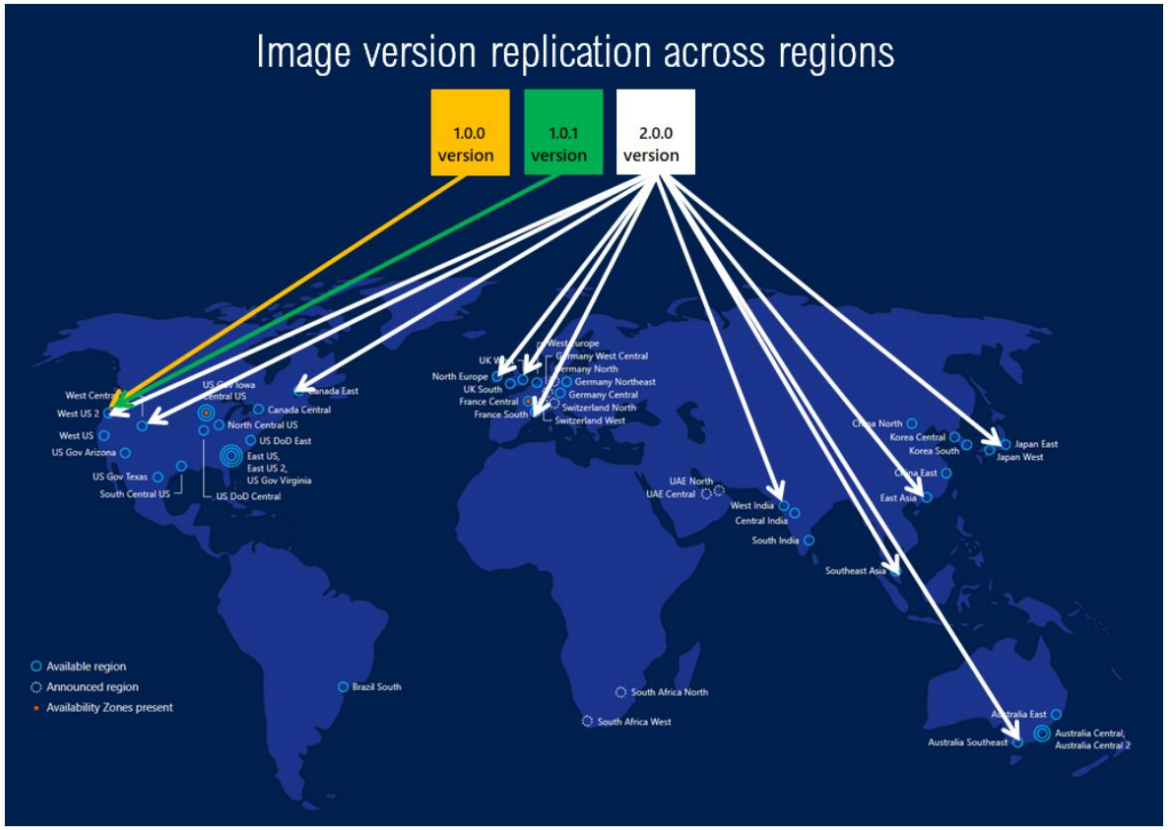
Recommended VM vCPUs 16 64

Recommended VM memory 500 1024

Excluded disk types 0 selected

Image definition end of life date MM/DD/YYYY

Now that you have an image in the gallery with versions you can create vms. In order to leverage this with WVD, you will need to incorporate the json fields for using an image from a shared image gallery with the WVD deployment json script.



Demo building a session host with the shared image gallery features via GitHub.

I have modified our formal Repo to add this shared image gallery function, my repo is located here.
https://github.com/hardeights/RDS-Templates/tree/master/wvd-templates/Create%20and%20provision%20WVD%20host%20pool

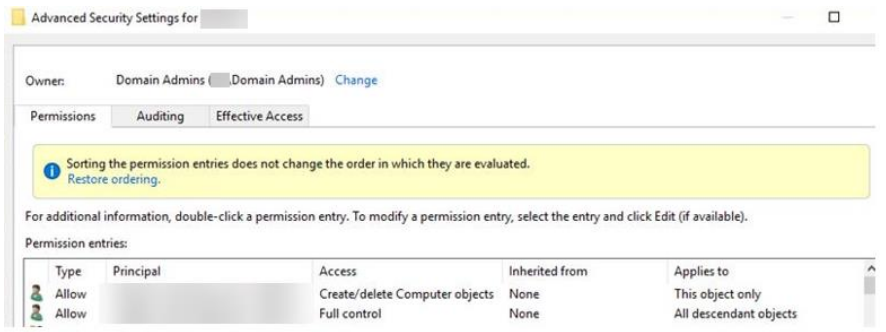
This repo is only to use the shared Image gallery function.

Exercise 14: WVD Troubleshooting

Often during the deployment of WVD there may be challenges with either having machines join the domain or installing the agent via automation.

Domain troubleshooting

1) Verify that the AD account has the correct permissions for domain joining machines based on the blog domain join blog.

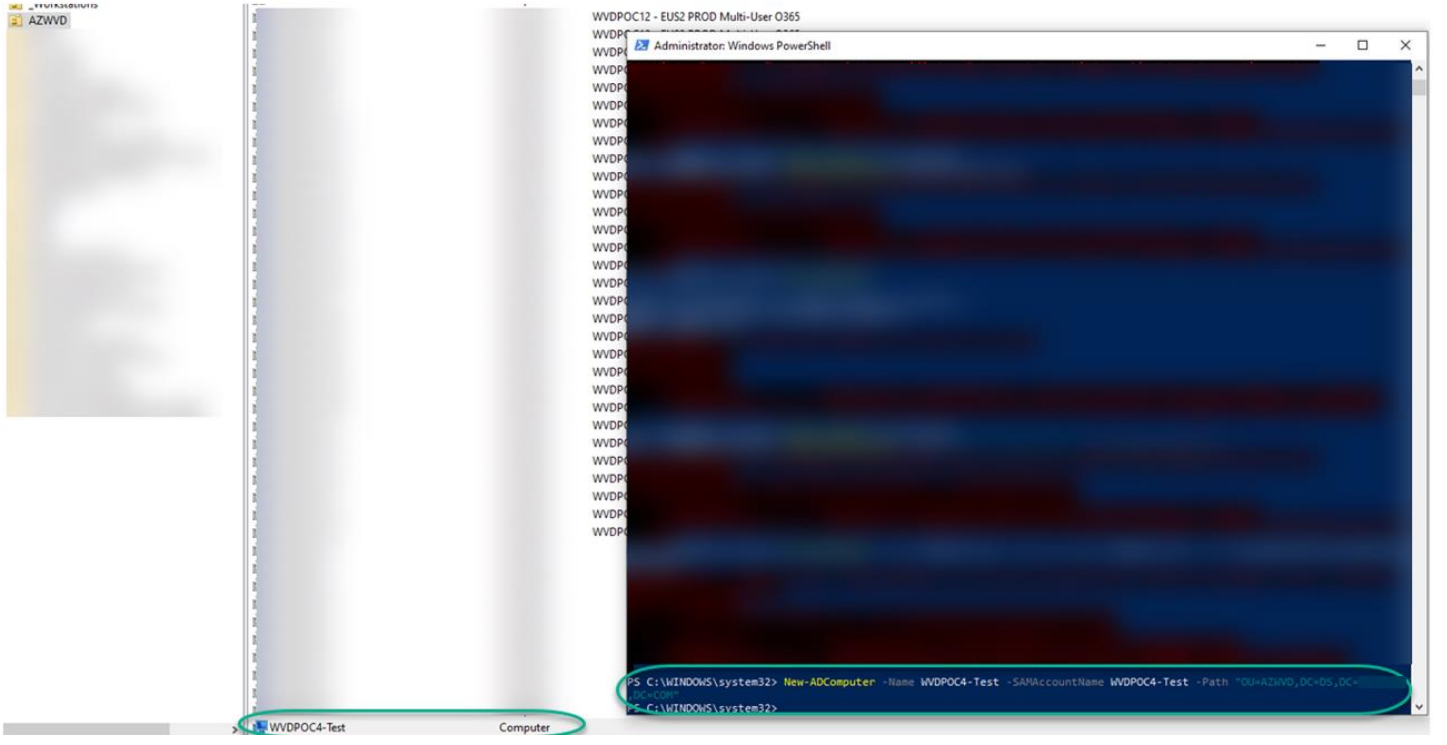


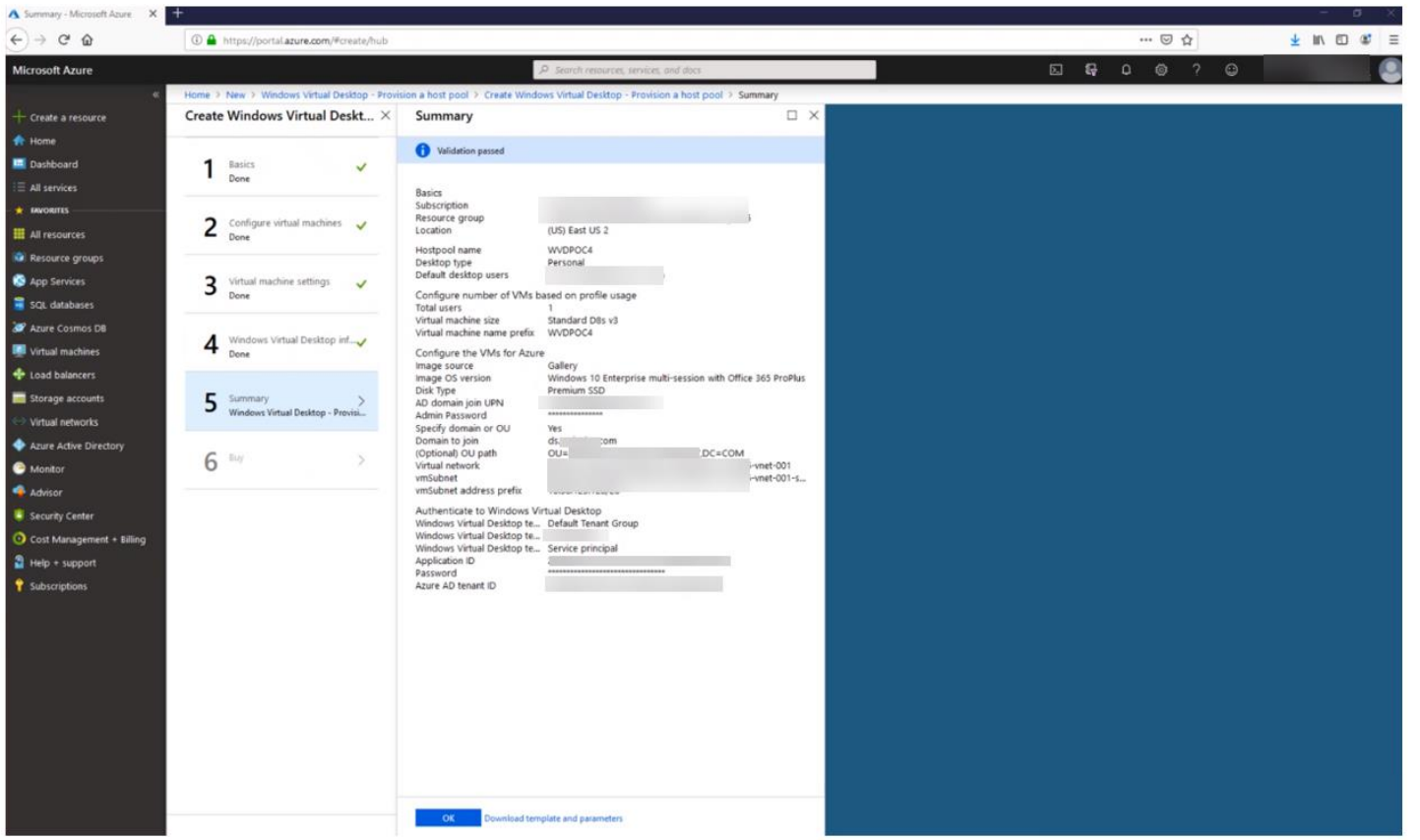
When deploying WVD machines to the OU path below. This is the OU where the AD account AdAdmin has create and delete rights for machine objects. I've validated that he can manually create and delete AD objects in this OU.

MachineAccountOU: OU=AZWVD,DC=DS,DC=CUSTOMERDOMAIN,DC=COM

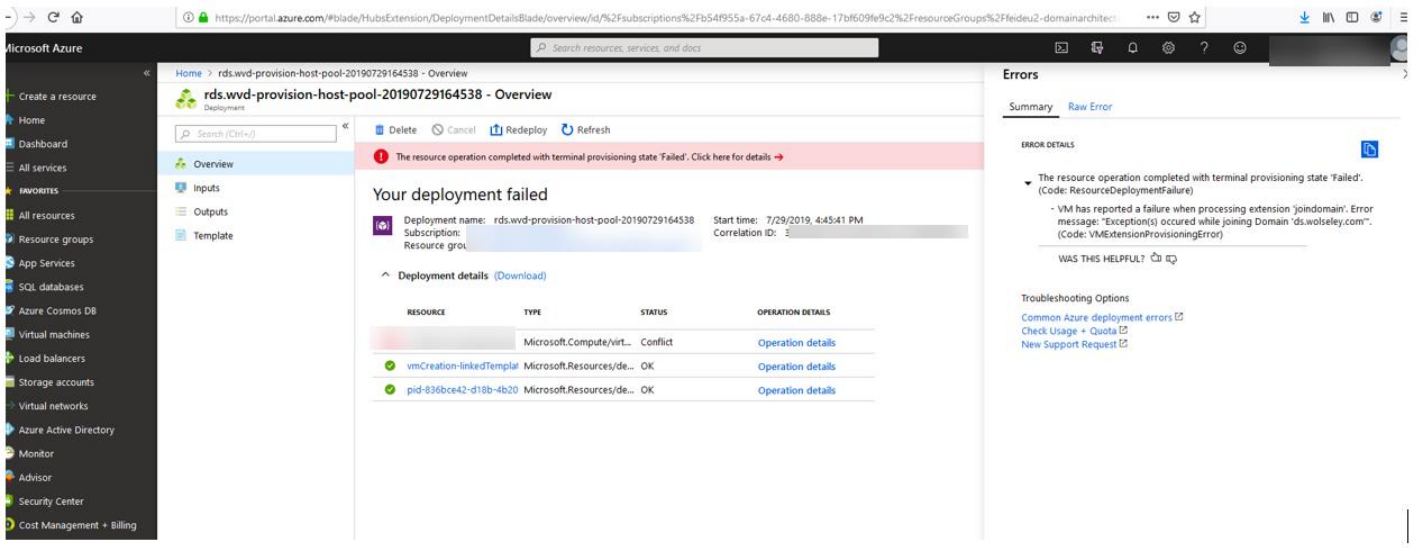
Verify using PowerShell that the AD account can create AD object in the OU

C:\PS>New-ADComputer -Name "'WVDPOC4-0'" -SamAccountName "'WVDPOC4-0'" -Path "
OU=AZWVD,DC=DS,DC=CUSTOMERDOMAIN,DC=COM "





Deployment failure for domain joining

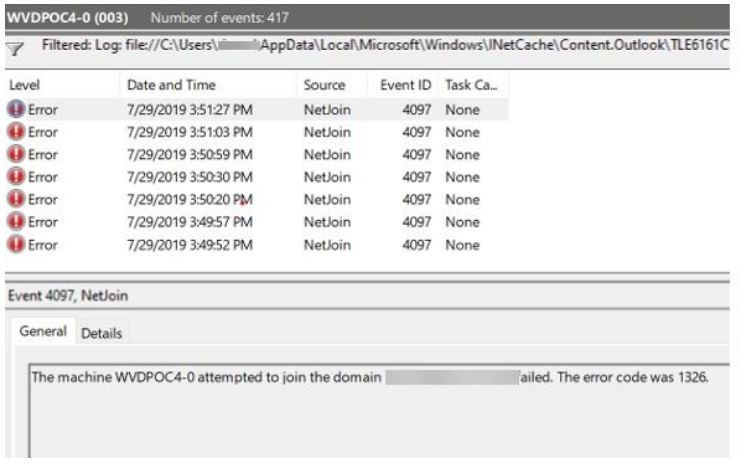


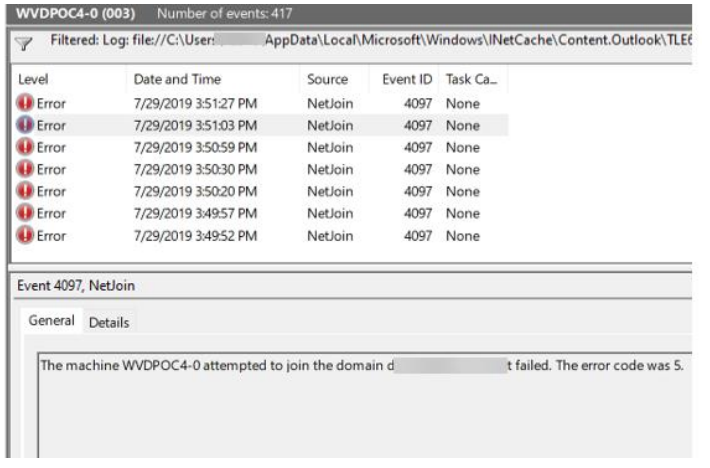
{"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list deployment operations for details. Please see https://aka.ms/arm-debug for usage details.","details":[{"code":"Conflict","message":"{\r\n \"status\": \"Failed\",\r\n \"error\": {\r\n \"code\": \"ResourceDeploymentFailure\",\r\n \"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) occured while joining Domain 'ds.CUSTOMERDOMAIN.com'\\\".\"\r\n }\r\n]\r\n }\r\n}"}]}

Troubleshooting process

In the session host we see these errors in event viewer.





After digging into the "NetSetup.log" file in the C:\Windows\Debug directory we see this.

07/29/2019 20:51:18:281 ----------

07/29/2019 20:51:18:281 NetpDoDomainJoin

07/29/2019 20:51:18:281 NetpDoDomainJoin: using current computer names

07/29/2019 20:51:18:281 NetpDoDomainJoin: NetpGetComputerNameEx(NetBios) returned 0x0

07/29/2019 20:51:18:281 NetpDoDomainJoin: NetpGetComputerNameEx(DnsHostName) returned 0x0

07/29/2019 20:51:18:281 NetpMachineValidToJoin: 'WVDPOC4-0'

07/29/2019 20:51:18:281 NetpMachineValidToJoin: status: 0x0

07/29/2019 20:51:18:281 NetpJoinDomain

07/29/2019 20:51:18:281 HostName: WVDPOC4-0

07/29/2019 20:51:18:281 NetbiosName: WVDPOC4-0

07/29/2019 20:51:18:281 Domain: ds.CUSTOMERDOMAIN.com

07/29/2019 20:51:18:281 MachineAccountOU: OU=AZWVD,DC=DS,DC=CUSTOMERDOMAIN,DC=COM

07/29/2019 20:51:18:281 Account: admin@customerdomain.com

07/29/2019 20:51:18:281 Options: 0x3

07/29/2019 20:51:18:301 NetpValidateName: checking to see if 'ds.CUSTOMERDOMAIN.com' is valid as type 3 name

07/29/2019 20:51:18:621 NetpCheckDomainNameIsValid [Exists] for 'ds.CUSTOMERDOMAIN.com' returned 0x0

07/29/2019 20:51:18:621 NetpValidateName: name 'ds.CUSTOMERDOMAIN.com' is valid for type 3

07/29/2019 20:51:18:621 NetpDsGetDcName: trying to find DC in domain 'ds.CUSTOMERDOMAIN.com', flags: 0x40001010

07/29/2019 20:51:22:556 NetpDsGetDcName: status of verifying DNS A record name resolution for 'a07308.DS.CUSTOMERDOMAIN.COM': 0x0

07/29/2019 20:51:22:556 NetpDsGetDcName: found DC '\\a07308.DS.CUSTOMERDOMAIN.COM' in the specified domain

07/29/2019 20:51:22:556 NetpJoinDomainOnDs: NetpDsGetDcName returned: 0x0

07/29/2019 20:51:22:556 NetpDisableIDNEncoding: using FQDN DS.CUSTOMERDOMAIN.COM from dcinfo

07/29/2019 20:51:22:556 NetpDisableIDNEncoding: DnsDisableIdnEncoding(UNTILREBOOT) on 'DS.CUSTOMERDOMAIN.COM' succeeded

07/29/2019 20:51:22:556 NetpJoinDomainOnDs: NetpDisableIDNEncoding returned: 0x0

07/29/2019 20:51:27:687 NetUseAdd to \\a07308.DS.CUSTOMERDOMAIN.COM\IPC$ returned 1326

07/29/2019 20:51:27:687 NetpJoinDomainOnDs: status of connecting to dc '\\a07308.DS.CUSTOMERDOMAIN.COM': 0x52e

07/29/2019 20:51:27:687 NetpJoinDomainOnDs: Function exits with status of: 0x52e

07/29/2019 20:51:27:687 NetpResetIDNEncoding: DnsDisableIdnEncoding(RESETALL) on 'DS.CUSTOMERDOMAIN.COM' returned 0x0

07/29/2019 20:51:27:687 NetpJoinDomainOnDs: NetpResetIDNEncoding on 'DS.CUSTOMERDOMAIN.COM': 0x0

07/29/2019 20:51:27:687 NetpDoDomainJoin: status: 0x52e

07/29/2019 20:51:27:703 ----------

07/29/2019 20:51:27:703 NetpDoDomainJoin

07/29/2019 20:51:27:703 NetpDoDomainJoin: using current computer names

07/29/2019 20:51:27:703 NetpDoDomainJoin: NetpGetComputerNameEx(NetBios) returned 0x0

07/29/2019 20:51:27:703 NetpDoDomainJoin: NetpGetComputerNameEx(DnsHostName) returned 0x0

07/29/2019 20:51:27:703 NetpMachineValidToJoin: 'WVDPOC4-0'

07/29/2019 20:51:27:703 NetpMachineValidToJoin: status: 0x0

07/29/2019 20:51:27:703 NetpJoinDomain

07/29/2019 20:51:27:703 HostName: WVDPOC4-0

07/29/2019 20:51:27:703 NetbiosName: WVDPOC4-0

07/29/2019 20:51:27:703 Domain: ds.CUSTOMERDOMAIN.com

07/29/2019 20:51:27:703 MachineAccountOU: OU=AZWVD,DC=DS,DC=CUSTOMERDOMAIN,DC=COM

07/29/2019 20:51:27:703 Account: admin@customerdomain.com

07/29/2019 20:51:27:703 Options: 0x1

07/29/2019 20:51:27:722 NetpValidateName: checking to see if 'ds.CUSTOMERDOMAIN.com' is valid as type 3 name

07/29/2019 20:51:28:028 NetpCheckDomainNameIsValid [Exists] for 'ds.CUSTOMERDOMAIN.com' returned 0x0

07/29/2019 20:51:28:028 NetpValidateName: name 'ds.CUSTOMERDOMAIN.com' is valid for type 3

07/29/2019 20:51:28:028 NetpDsGetDcName: trying to find DC in domain 'ds.CUSTOMERDOMAIN.com', flags: 0x40001010

07/29/2019 20:51:31:261 NetpDsGetDcName: status of verifying DNS A record name resolution for 'a07300.DS.CUSTOMERDOMAIN.COM': 0x0

07/29/2019 20:51:31:261 NetpDsGetDcName: found DC '\\a07300.DS.CUSTOMERDOMAIN.COM' in the specified domain

07/29/2019 20:51:31:261 NetpJoinDomainOnDs: NetpDsGetDcName returned: 0x0

07/29/2019 20:51:31:261 NetpDisableIDNEncoding: using FQDN DS.CUSTOMERDOMAIN.COM from dcinfo

07/29/2019 20:51:31:261 NetpDisableIDNEncoding: DnsDisableIdnEncoding(UNTILREBOOT) on 'DS.CUSTOMERDOMAIN.COM' succeeded

07/29/2019 20:51:31:261 NetpJoinDomainOnDs: NetpDisableIDNEncoding returned: 0x0

07/29/2019 20:51:32:008 NetUseAdd to \\a07300.DS.CUSTOMERDOMAIN.COM\IPC$ returned 1326

07/29/2019 20:51:32:008 Trying add to \\a07300.DS.CUSTOMERDOMAIN.COM\IPC$ using NULL Session

07/29/2019 20:51:32:133 NullSession NetUseAdd to \\a07300.DS.CUSTOMERDOMAIN.COM\IPC$ returned 5

07/29/2019 20:51:32:133 NetpJoinDomainOnDs: status of connecting to dc '\\a07300.DS.CUSTOMERDOMAIN.COM': 0x5

07/29/2019 20:51:32:133 NetpJoinDomainOnDs: Function exits with status of: 0x5

07/29/2019 20:51:32:133 NetpResetIDNEncoding: DnsDisableIdnEncoding(RESETALL) on 'DS.CUSTOMERDOMAIN.COM' returned 0x0

07/29/2019 20:51:32:133 NetpJoinDomainOnDs: NetpResetIDNEncoding on 'DS.CUSTOMERDOMAIN.COM': 0x0

07/29/2019 20:51:32:133 NetpDoDomainJoin: status: 0x5

07/29/2019 20:51:47:212 ----------

Resolution: Try the following steps to resolve domain join issues.

When using an ARM template you man encounter domain join errors. Use the following tips to verify what information is being displayed.

Step 1) Verify the AD permissions for the domain account. In some enterprises the account being used to join the domain is not a domain admin and only has limited rights to a specific OU for create and delete of objects.

Step 2) If you receive a domain join failure ensure that credentials are correct.
Step 3) Attempt to provision a new windows machine with just the Azure portal and manually domain join the machine in the exact OU that the admin states he/she has permissions to.

i. Also if there may be enforced permissions to the OU you can try other Ous.

Step4) Attempt to use another elevated account from an IT person that has rights with other Ous.

Step5) If there is a failure for the domain join, RDP into the WVD machine that is created with the local account created that mirrors the domain account entered in the field. Open event viewer and in
Generally error 5 means lack of permission. Which is what is implied from your comment that he has access only to a specific OU.

Please review the following guide - https://support.microsoft.com/en-us/help/4341920/troubleshoot-errors-that-occur-when-you-join-windows-based-computers-t

Step 6) Grab additional logs from the path
Windows clients log the details of domain join operations in the %windir%\debug\Netsetup.log file.

C:\Windows\TEMP\scriptlogs.log

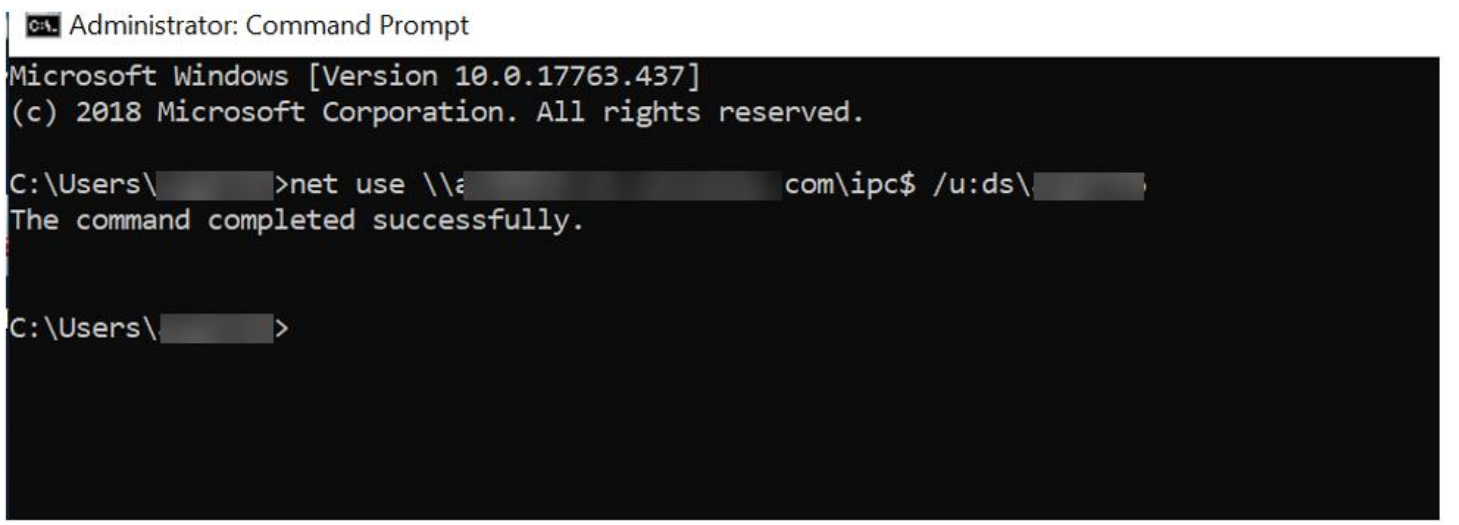
C:WindowsdebugNetSetup.log

Step 7) also can you try this and provide the output?

To troubleshoot the issue, run a similar command from the command prompt to confirm the above analysis.

```
net use \\dcname\ipc$ /u:< domain\user > < password >
```

try the command above with a output that should say something like below:



Step 8) Verify that the password used for the account does not have a non-standard character that could cause issues. Characters like "/" and others can cause automation to fail.

https://social.technet.microsoft.com/Forums/ie/en-US/49b4e082-c189-4b82-9c0d-e3f0b925d562/domain-join-account-still-fails-with-error-1326?forum=configmgrgeneral

ERROR 2 – DSC Failure to install agents, but domain join is successful.

{"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 '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}"}]}

WVD Agent troubleshooting

Troubleshooting WVD Agent issues

Agent Installation and Update process:

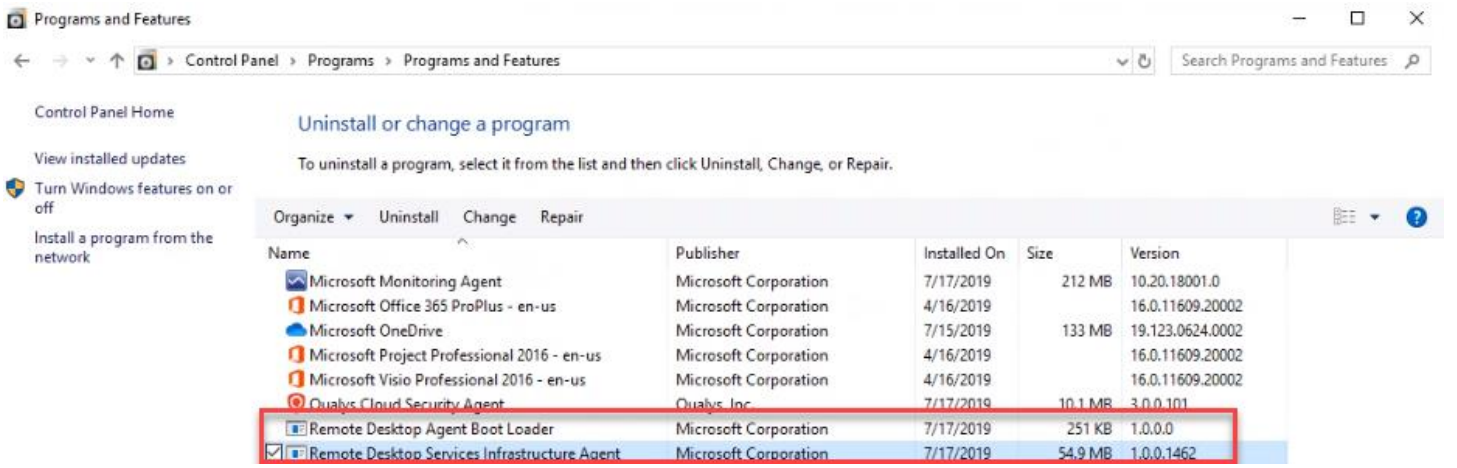
- Initial Version of the Windows Virtual Desktop Agent is downloaded and installed from an externally accessible download location (either manually or via Azure Marketplace)
- Once the Initial version of Windows Virtual Desktop Agent is installed, it queries the WVD service to determine the desired version of the WVD agent, Remote Desktop Services Infrastructure Geneva Agent and SXS stack
- Desired version of the Agent, Remote Desktop Services Infrastructure Geneva Agent and SXS stack (in the same order) are then pulled from an Azure blob location and installed on the Virtual Machine.
- This upgrade operation normally lasts about few minutes

- There may be few scenarios where upgrade of the WVD agent may fail. When this scenario occurs output of the Get-RDSSessionnHost -TenantName <tenantname> -HostPoolName <hostpoolname> will be like below:

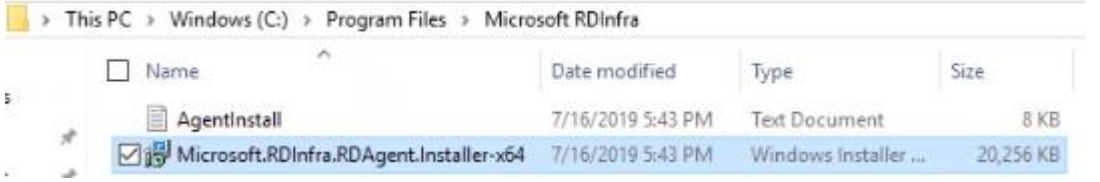
```
PS C:\> Get-RdsSessionHost -TenantName wvdtraining -HostPoolName hp1-win10evd


SessionHostName     : HP1-WIN10EVD-0.azureguides.com
TenantName          : wvdtraining
TenantGroupName     : Default Tenant Group
HostPoolName        : hp1-win10evd
AllowNewSession     : True
Sessions            : 0
LastHeartBeat       :
AgentVersion        :
AssignedUser        :
OsVersion           : 10.0.17763
SxSStackVersion     : rdp-sxs190429002
Status              : Upgrading
UpdateState         :
LastUpdateTime      :
UpdateErrorMessage  :
```

When this issue occurs, follow below steps to troubleshoot:

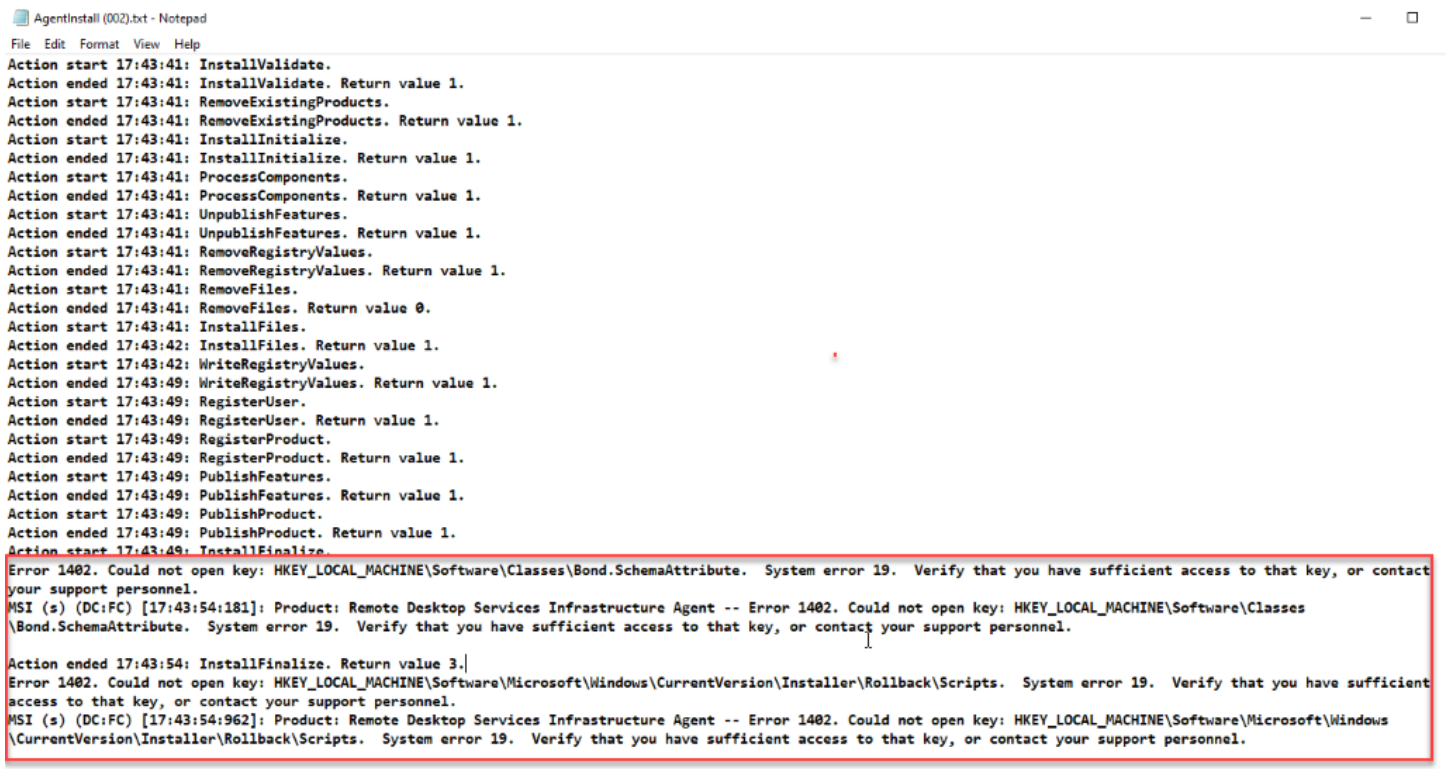
1. Review WVD Agent installation C:\Program Files\Microsoft RDInfra\AgentInstall.txt to see if the installation failed
2. Review version of WVD Agent already installed on your VM.
    - Go to Control Panel\Programs\Programs and Features. Look for latest version of “Remote Desktop Services Infrastructure Agent” installed on the Virtual Machine and note the same.
    -



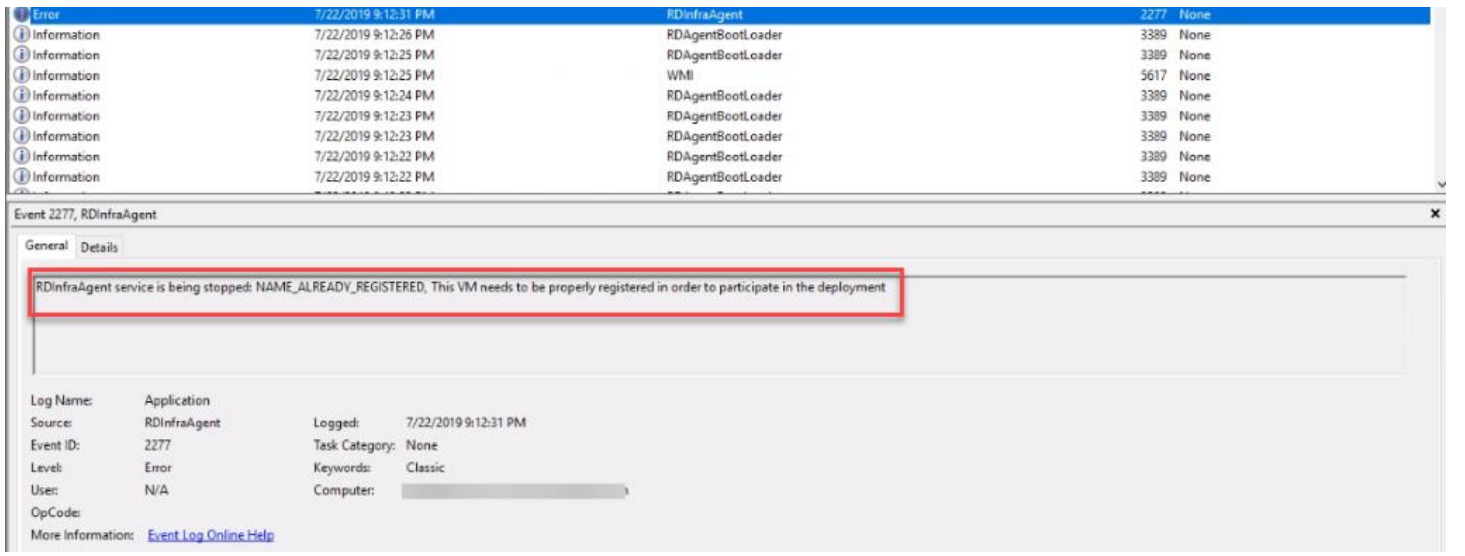
3. Review if there is a later version of this agent downloaded onto the Virtual machine. Go to C:\program files\Microsoft RdInfra and look for versions later than one installed.



a. The Agentinstall.txt log files has data regarding the agent installation. It may be helpful to open that log and verify that there are no errors that may related permissions used to install the agent.



- In case there is a newer version:
    - Attempt to manually install the MSI. Note this requires you to acquire and supply a new registration token.
    - Run the below 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 > <PathToRegFile>
- In case there is no newer version:
    - Uninstall all the versions of the agents from Control Panel\Programs\Programs and Features
    - Remove the session host from the host pool also before reinstalling the agent. Leaving the session host in the host pool can result in errors like the event ID 2277 NAME_ALREADY_REGISTERED image below. Remove-RdsSessionHost [-TenantName] <String> [-HostPoolName] <String> [-Name] <string> [-Force]



Manually install Windows Virtual Desktop agent and register it using the steps described here.

4. Once the agent is manually installed, you should see the Get-RDSSessionHost -TenantName <tenantname> -HostPoolName <hostpoolname> output as below:

```
PS C:\> Get-RdsSessionHost -TenantName wvdtraining -HostPoolName hp1-win10evd

SessionHostName : HP1-WIN10EVD-0.azureguides.com
TenantName      : wvdtraining
TenantGroupName : Default Tenant Group
HostPoolName    : hp1-win10evd
AllowNewSession : True
Sessions        : 2
LastHeartBeat   : 7/22/2019 8:09:49 PM
AgentVersion    : 1.0.833.5
AssignedUser    :
Status          : Available
StatusTimestamp :
```

5. In case the manual installation fails, review
   - Contents of “UpdateErrorMessage” in Get-RDSSessionHost -TenantName <tenantname> -HostPoolName <hostpoolname>
   - Application Event logs in Event Viewer:
     - Event Viewer - > Windows Logs -> Application. Look for events with the source
       - MsiInstaller
       - RDAgentBootLoader
       - WVD-Agent-Updater

Error Code	Description
112	There is not enough space on the disk
1603	Fatal error during installation

1605	This action is only valid for products that are currently installed
1618	Another installation is already in progress. Complete that installation before proceeding with this install
1638	Another version of this product is already installed. Installation of this version cannot continue. To configure or remove the existing version of this product, use Add/Remove Programs on the Control Panel
2277	RD

- Enable Verbose MSI logging , reattempt install and review the output logs

Internal Logging:

In case Geneva Monitoring Agent is installed on the VM , we should be able to see data about agent upgrade failures in our Kusto database.
Below is an example Kusto query

Kusto

```
RDInfraTrace
| where PreciseTimeStamp >= datetime(2019-07-01 00:00:37)
//| where ActivityId == "db23a6e3-e3a1-4159-bcad-496f215aa1b2"
| where HostInstance contains "VDI-0.customerdomain.local"
//| where Msg contains "c27f9442-81ef-4274-a6a0-fde78807b9ca"
| order by PreciseTimeStamp asc
| project PreciseTimeStamp, Msg, ActivityId, Category, Machine, HostInstance, HostPool,
Role,Env, Level, Ring, Cluster, Ver, Pid
| take 10000
```

Agent Registration issues

External Logging:

Event Viewer: Windows Logs\Applications

```
WVDEventLogger (3019, "WVD-Agent-Transport");
```

Error	Description	
	WVD-Agent service is being stopped: ENDPOINT_NOT_FOUND , This VM needs to be properly	

	registered in order to participate in the deployment	
	WVD-Agent service is being stopped: INVALID_REGISTRATION_TOKEN, This VM needs to be properly registered in order to participate in the deployment	
	WVD-Agent service is being stopped: NAME_ALREADY_REGISTERED, This VM needs to be properly registered in order to participate in the deployment	

Internal logging

No additional logging

Demo of Management UX

The installation of the Management UX can be found here.

Discuss Automation Options

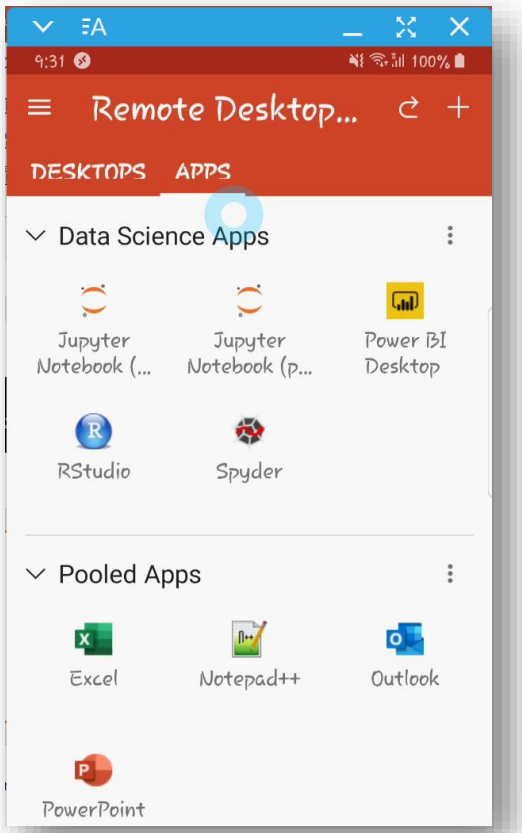
Sepago has created a WVD scaling automation script located here, https://github.com/MarcelMeurer/Project-MySmartScale We also have a automation script you can leverage by following the following document, https://microsoft.sharepoint.com/:w:/t/WVDAirlift2019Trainers/Ec_x2G_xOJZFt2l9tE0Oj0ABWRX5t8W_0Swe676QwSVEnQ?e=8ev0II

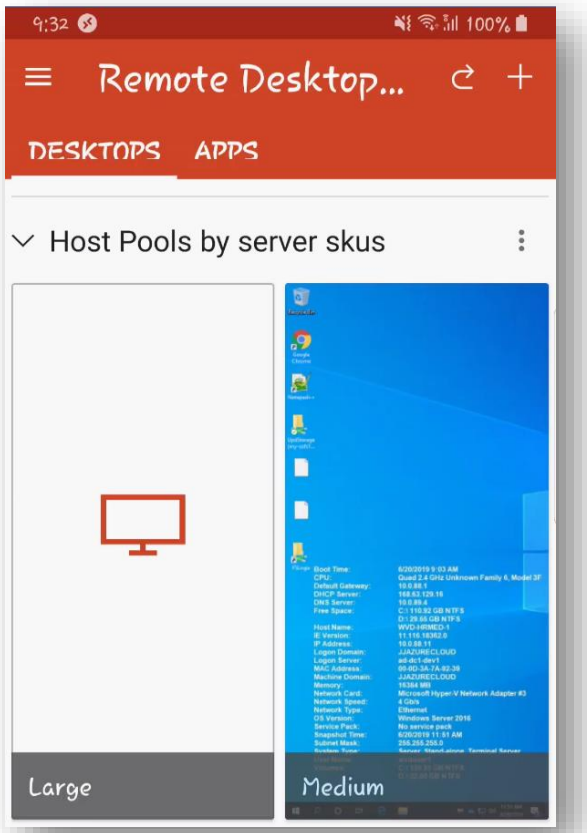
Discuss Mobile Options – Android and IOS

WVD is compatible with Android and IOS.

Android users can browse here to get the latest client and install it on Android supported devices.

IOS users can browse here to get the latest client and install it on IOS supported devices.





Authors

The following authors contributed to the creation of this deliverable.

Tony Sanchez	Tony.Sanchez@Microsoft.com
John Jenner	John.Jenner@microsoft.com
James Prendergast	James.Prendergast@microsoft.com
Adam Whitlatch	Adam.Whitlatch@microsoft.com

Revision History

Revision	Change Description	Updated By	Date
1.02	Minor Updates	Tony Sanchez	September 2019