Microsoft

Desktop-as-a-Service (DaaS)
Using Windows Virtual Desktop (WVD)

Line-of-Business (LOB) Applications Integration

Prepared for:
Service Provider Partners
Oct. 2019

Prepared by:
Microsoft – **O**ne **C**ommercial **P**artner (OCP)

MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation. Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, our provision of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers. © 2016 Microsoft Corporation. All rights reserved. Any use or distribution of these materials without express authorization of Microsoft Corp. is strictly prohibited. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

1. Overview

Windows Virtual Desktop provides a complete user experience right out-of-the-box. During the setup process, default values may be applied to a variety of WVD components and objects. In this document, we'll show you how to customize the WVD user experience by modifying some of those default values.

2. Prerequisites

Azure & Windows Active Directory Prerequisites

Before getting started, **all** items listed below **must** be checked/validated to ensure the most basic requirements are in place to proceed with executing the remaining steps in this guide.

- An Azure Active Directory
- A Windows Server Active Directory in sync with Azure Active Directory. This can be enabled through:
    - Azure AD Connect
    - Azure AD Domain Services
- An Azure subscription, containing a virtual network that either contains or is connected to the Windows Server Active Directory
- A functioning Windows Virtuel Desktop environment

General Best Practices

Since everyone's business and technical requirements vary across the board, it is always a good idea to familiarize yourselves with the standard best practices across the different Azure technologies & services.

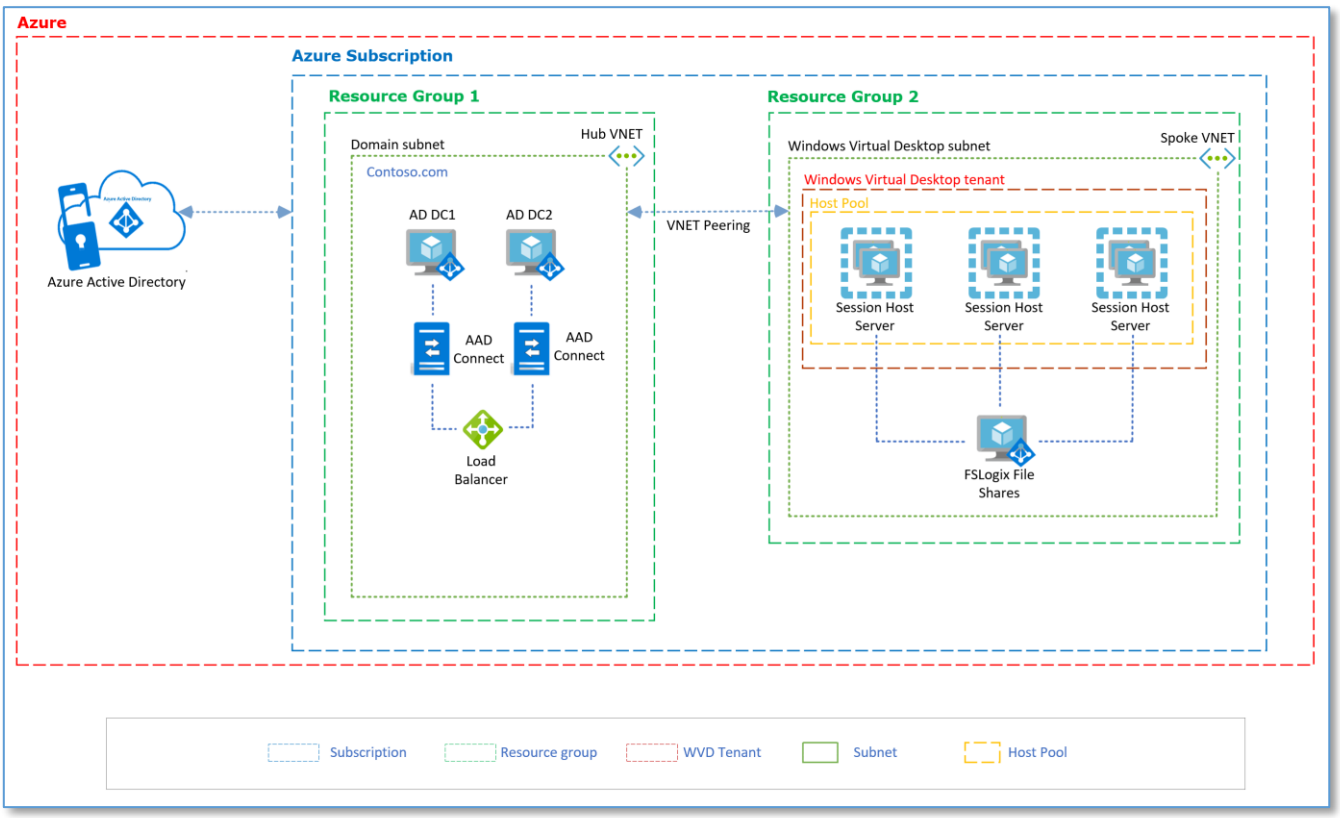
- Please follow the guidance here to maintain a consistent naming convention across your resources, unless you are already using a naming system.
- Azure security best practices and patterns
- Azure Active Directory Hybrid Identity best practices
- Azure identity management and access control security best practices
- Azure Networking & security Best Practices
- Azure Storage security overview
- Best practices for Azure VM security

Azure Networking

The recommendation is to design your Azure Networking using a [Hub-Spoke topology](). Consider the HUB like a DMZ deployed with your Virtual network Gateways and other security/edge appliances like Firewalls Etc. while the Spoke will act as the backend zone where your session hosts servers are deployed to and is peered with the HUB. This is our design for this walk-through, so you'll need this already setup before proceeding.

Azure Architectural Diagram

Below is a diagram of the Azure environment that we'll use. It shows the objects created in Azure and their relationships within the environment. In this example, the company name will be Contoso.



3. Set Friendly Names for Published Desktops

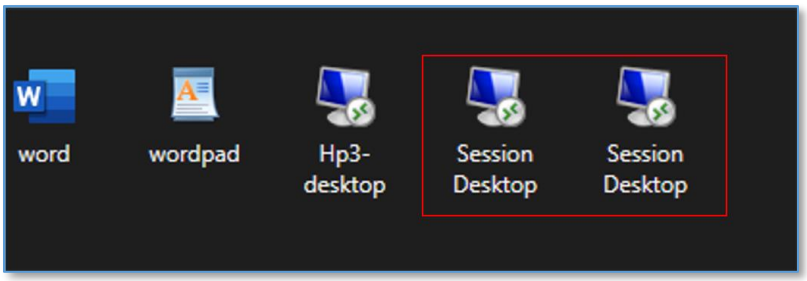
When you deploy a WVD published desktop for the first time, the default name of "Session Desktop" is applied. Each subsequent deployment of a Published Desktop will be automatically given the same name. If a user elects to work with multiple published desktops, then they are unable to determine which is which, since they both have the same name.

1. Download and import the Windows Virtual Desktop PowerShell module to use in a PowerShell session if you haven't already.
2. Using the below command, you can set friendly desktop names to uniquely identify multiple desktops published to a user.

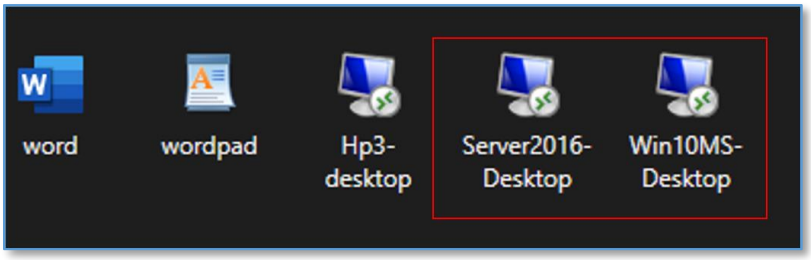
```
#Update the respective variables first and then execute command:
$tenantName = "ContosoCorpWVD"
$AppGroupName = "W10MS"

Set-RdsRemoteDesktop -TenantName $tenantName -HostPoolName $HostPoolName `
-AppGroupName $AppGroupName -FriendlyName "Win10MS-Desktop"
```

Before...



After...



4. Customizing the user experience

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.

Add or edit a single custom RDP property

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

```
# Enable audio/microphone redirection:
$tenantName = "ContosoCorpWVD"
$hostpoolName = "HP0"
$RDPProperty = "audiocapturemode:i:1"

Set-RdsHostPool -TenantName $tenantName -Name $hostpoolName `
-CustomRdpProperty $RDPProperty
```

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

Add or edit multiple custom RDP properties

You may add or edit more than one property at a time by using a semicolon separator.

In the following PowerShell cmdlet, we enable sound to play on the local system, and enable audio redirection to the remote desktop:

```
# Play sound locally & enable audio redirection:
$tenantName = "ContosoCorpWVD"
$hostpoolName = "HP0"
$RDPProperty = "audiomode:i:0;audiocapturemode:i:0"

Set-RdsHostPool -TenantName $tenantName -Name $hostpoolName `
-CustomRdpProperty $RDPProperty
```

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

Reset all custom RDP properties

You can reset all properties back to their defaults as well. Below, we'll do just that to our Contoso HP0 host pool:

```
# Reset all properties:
$tenantName = "ContosoCorpWVD"
$hostpoolName = "HP0"

Set-RdsHostPool -TenantName $tenantName -Name $hostpoolName `
-CustomRdpProperty ""
```

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

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

5. Add Custom & Published Apps to the Desktop

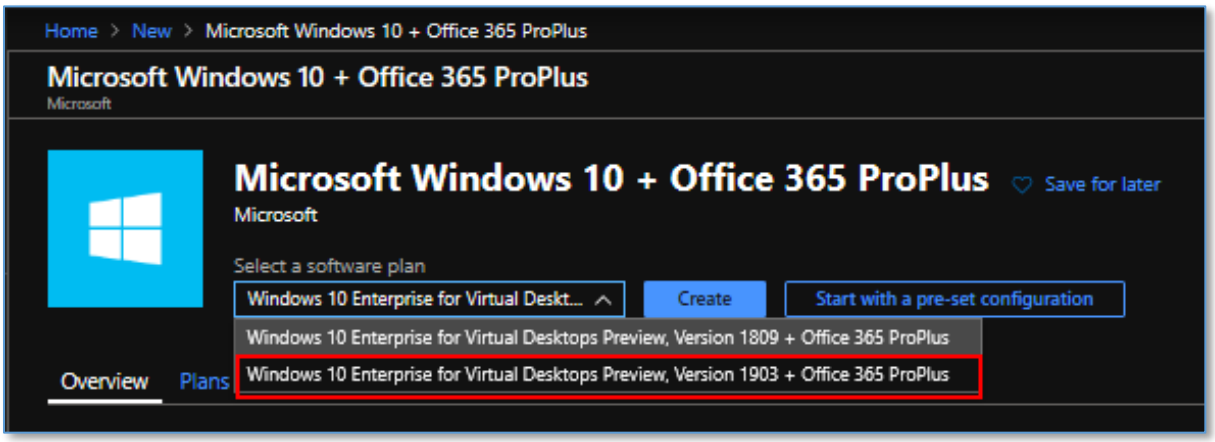
If your organization uses a custom application that's usually installed on the users' desktop, then you'll have to create a template, or master, image of a VM. Most often, this will be based on the new Windows 10 Enterprise for Virtual Desktops (Multi-User) Operating System, which is now available from the Azure Marketplace. After provisioning a VM, you can start installing custom applications on the machine. When you're done, you must capture the machine as an image to use as a base for your Windows Virtual Desktop deployment.

In this section, we will create, make a snapshot of, and deploy a custom image to a host pool. In our example, we'll be using the Windows 10 O365 SKU, and will install an application on to it.

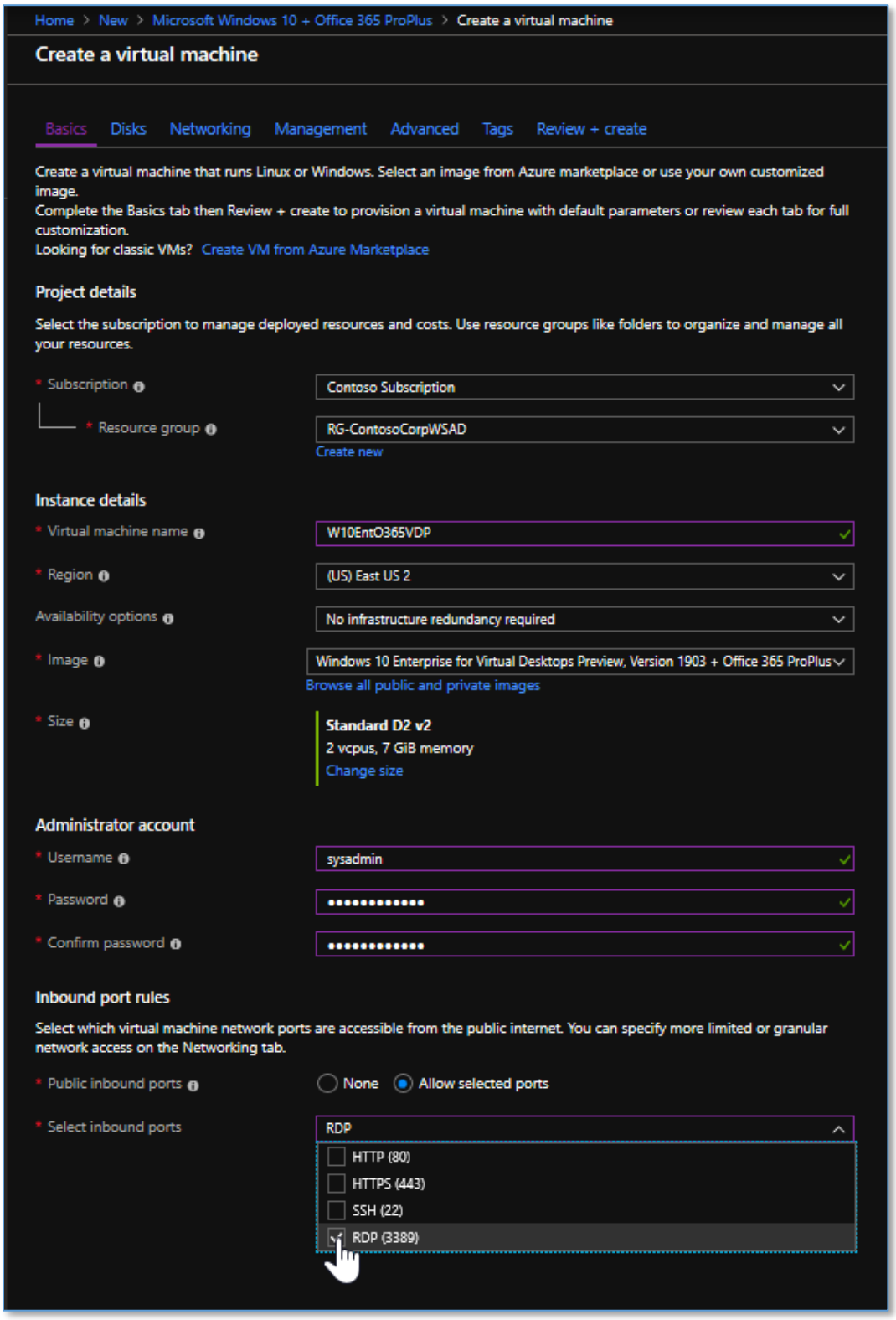
Create a custom, master, Windows 10 multi-session VM

We'll build a custom VM and use it as a master for creating future images. These images are then deployed to WVD host pools.

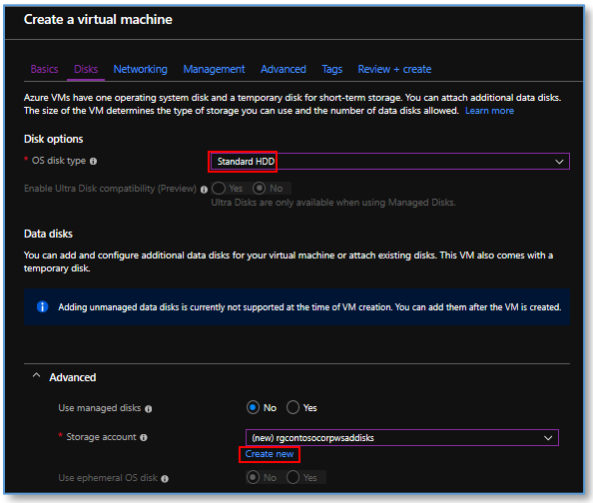
1. In the **Azure Portal**, search for "**windows 10**" and select the latest version of **Windows 10 Enterprise for Virtual Desktops Preview + Office 365 ProPlus**

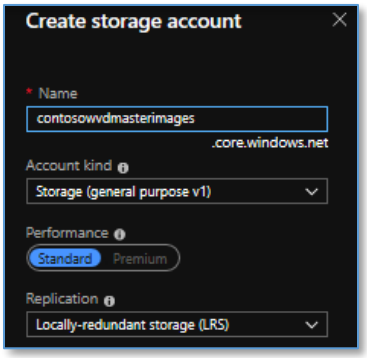


2. Click **Create** and complete the parameters for your environment. Example below shows our walk-through parameters. We're also allowing Remote Desktop port access from the Web:

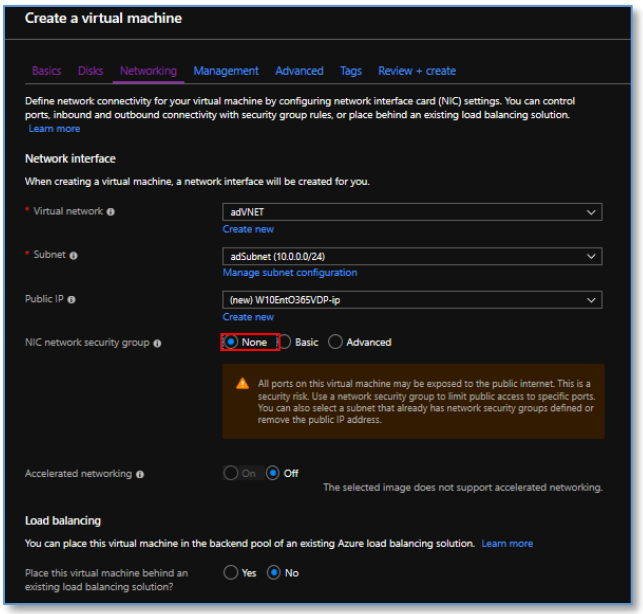


3. Click **Disks,** select **Standard Disks** and create a new storage account:

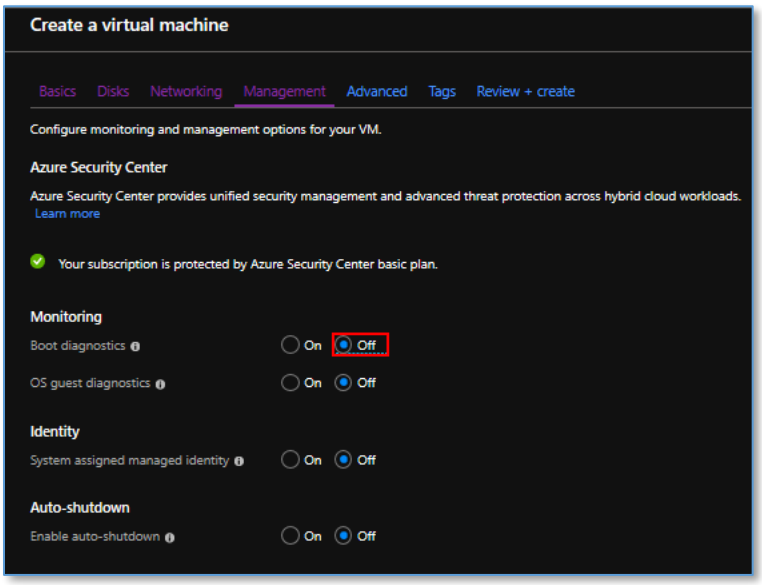




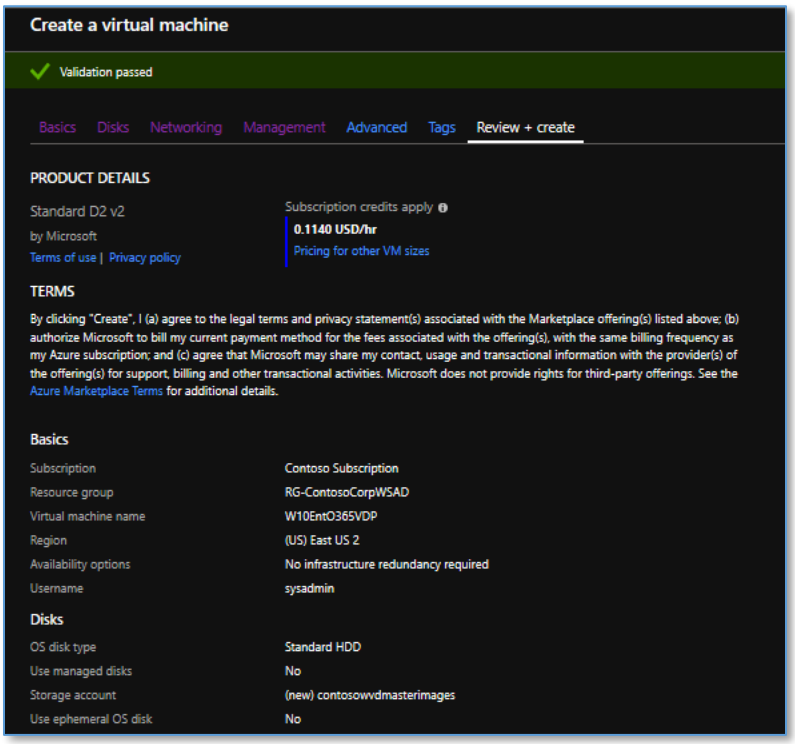
4. Click **Networking** and set **NIC network security group** to **None**:

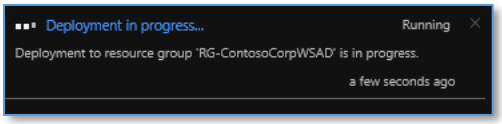


5. Click the **Management** tab and disable **Boot diagnostics**:



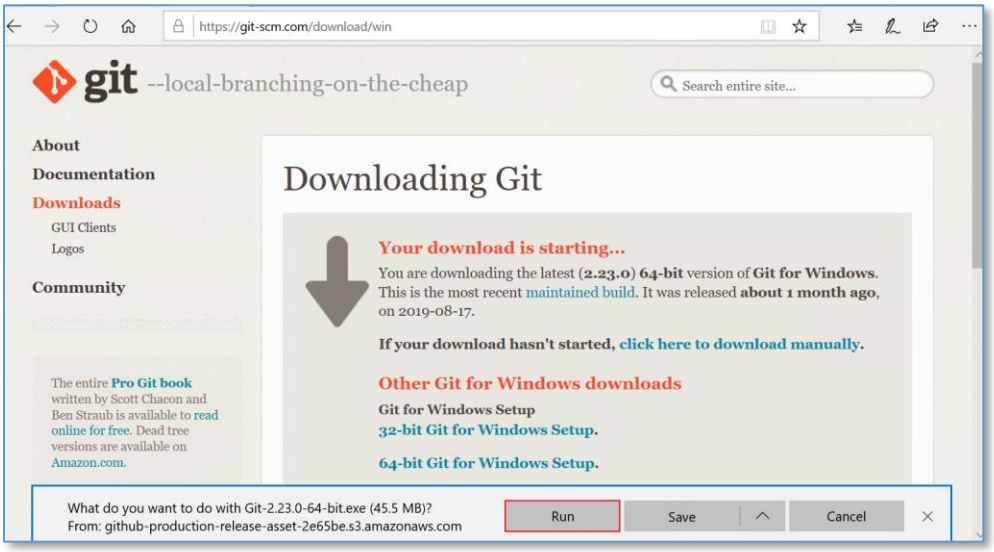
6. Click **Review + create**, then **create**:

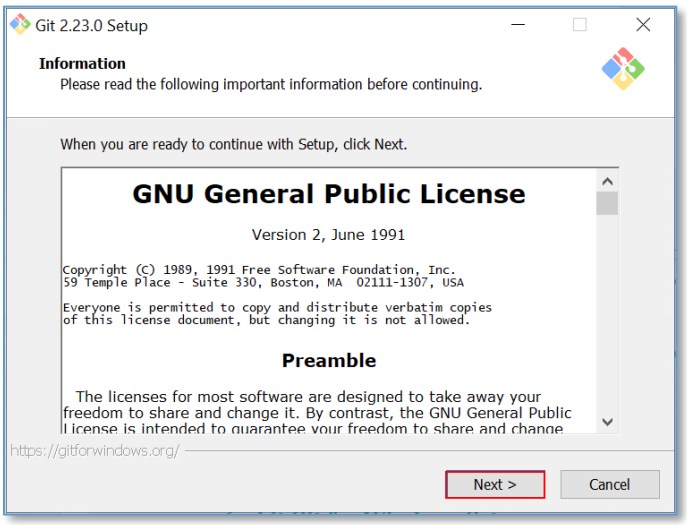


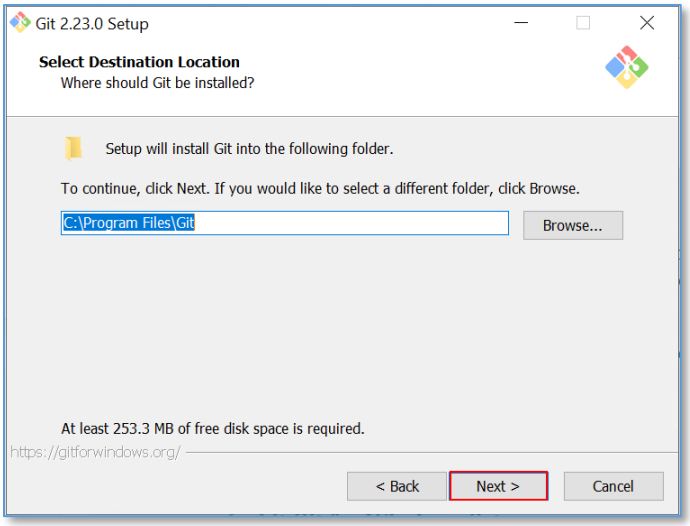


7. Once the deployment is completed, **logon** to the VM and **customize** it for capturing. Detailed instructions may be found here for each topic:
    a. Disable Automatic Updates
    b. Setup FSLogix if in use
    c. Configure Windows Defender
    d. Configure session time-out policies
    e. Time zone redirection
    f. Disable Storage Sense

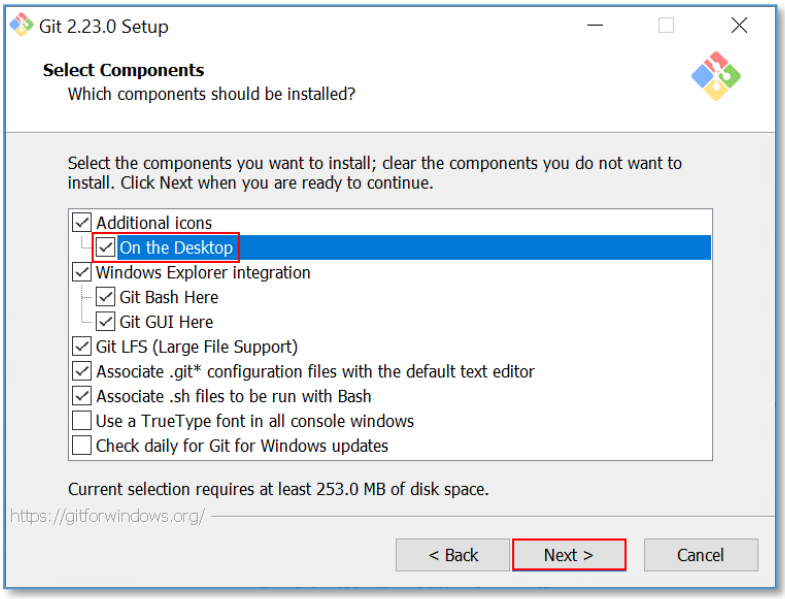
8. **Install** any additional **applications** and **add shortcuts** to any hosted apps. In our example below, we will be installing the Microsoft Windows GitHub client:



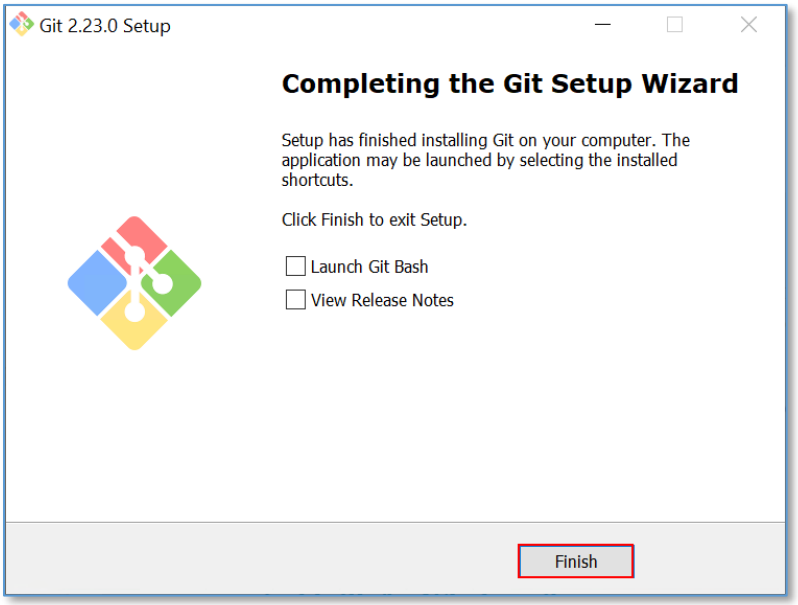




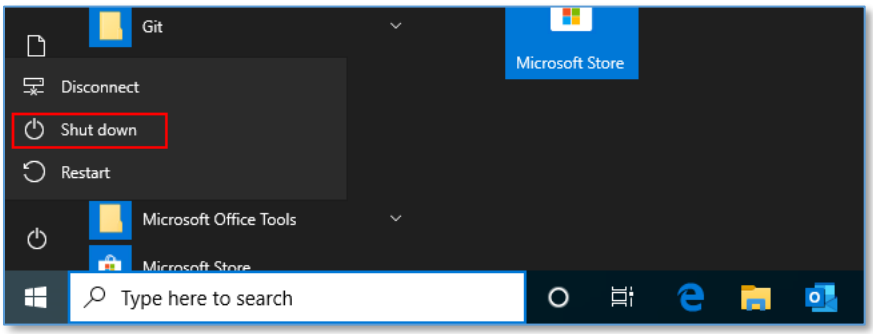
add a desktop icon:



accept all defaults for remaining dialog windows and finish the install:



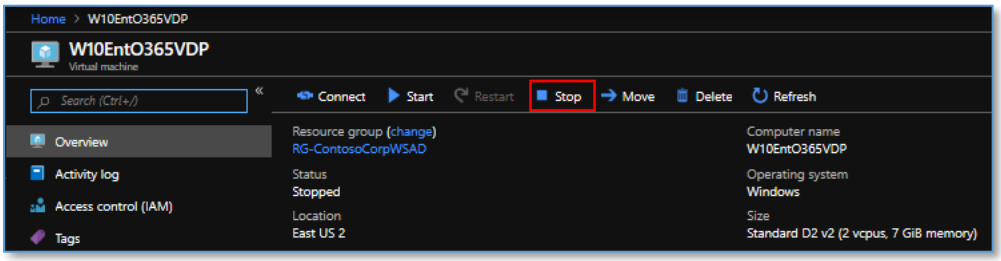
9. After installing all apps and making any additional changes, **shut down** the VM:

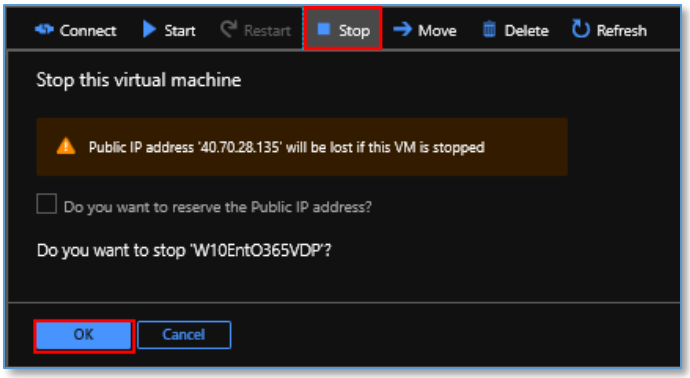


Create a WVD image from a template VM snapshot

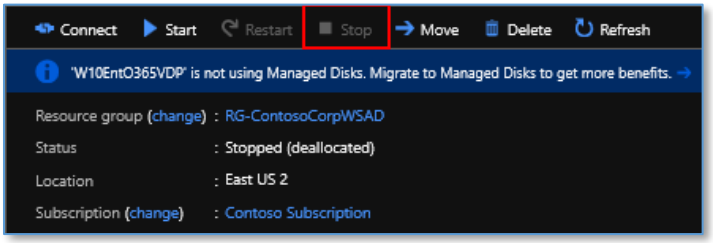
We'll use a snapshot of a VM to make the WVD deployment image. This way, we can update the original VM with changes, but still go back to that, or any other, snapshot at any time.

1. From the Azure Portal, **Stop** the template VM:

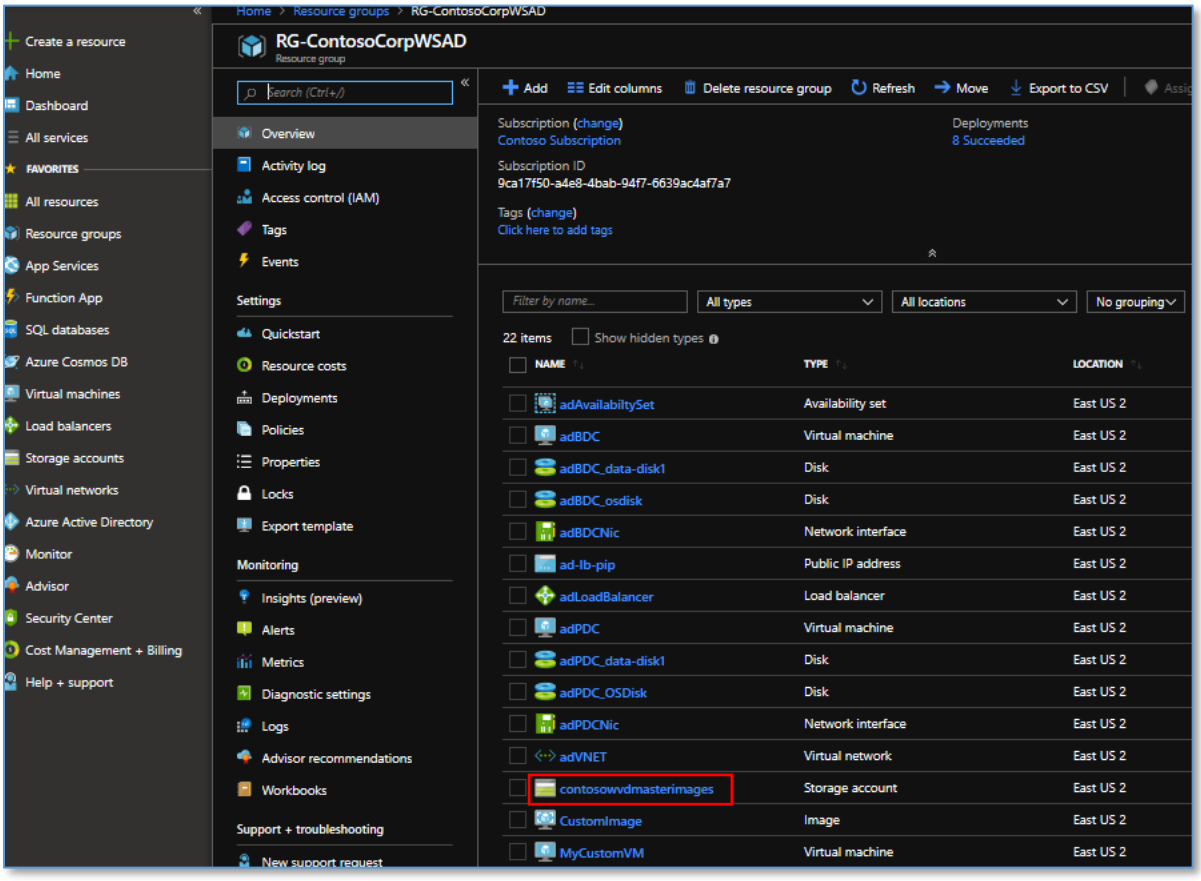




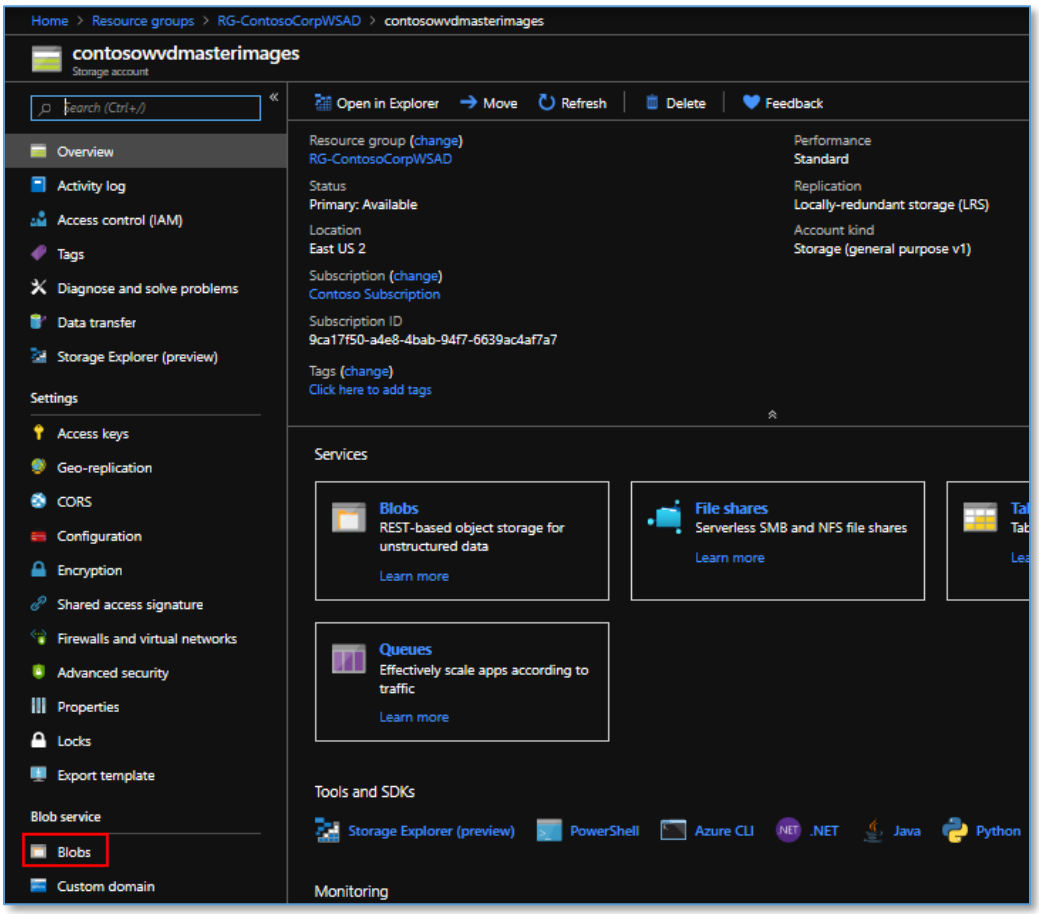
Click **Refresh** until Stop is greyed-out:

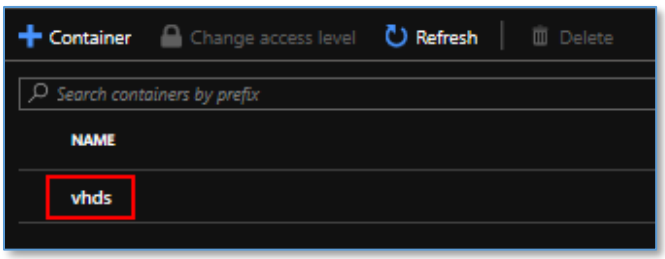


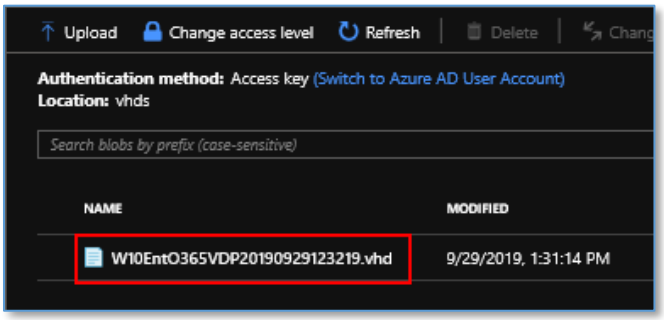
2. Within the VMs resource group, click the storage account name:



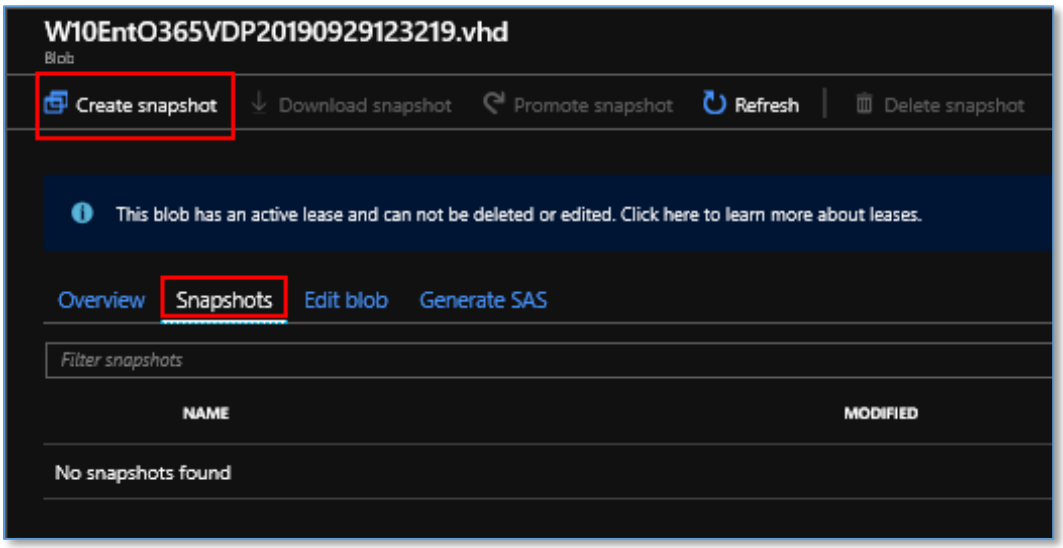
3. Click **Blobs** then **vhds** then the **VHD file name:**:



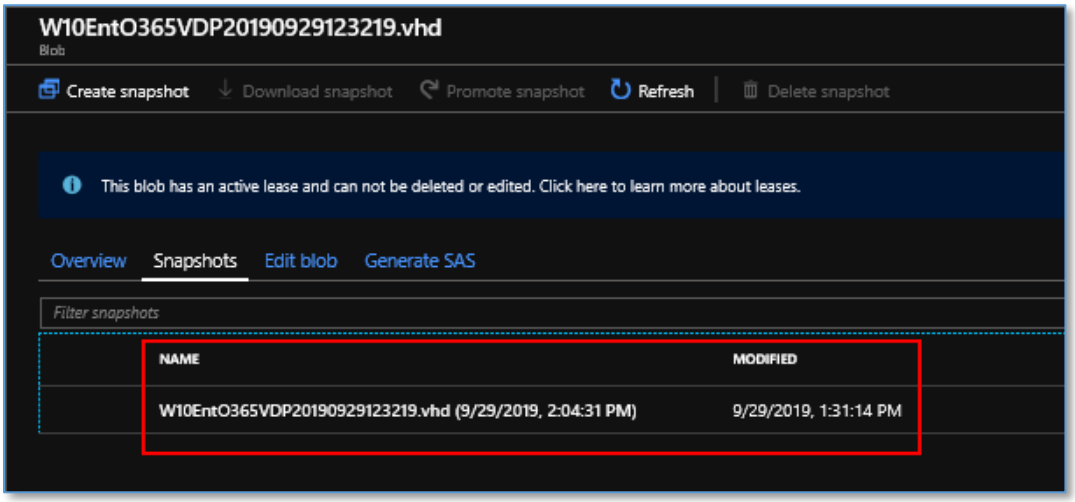




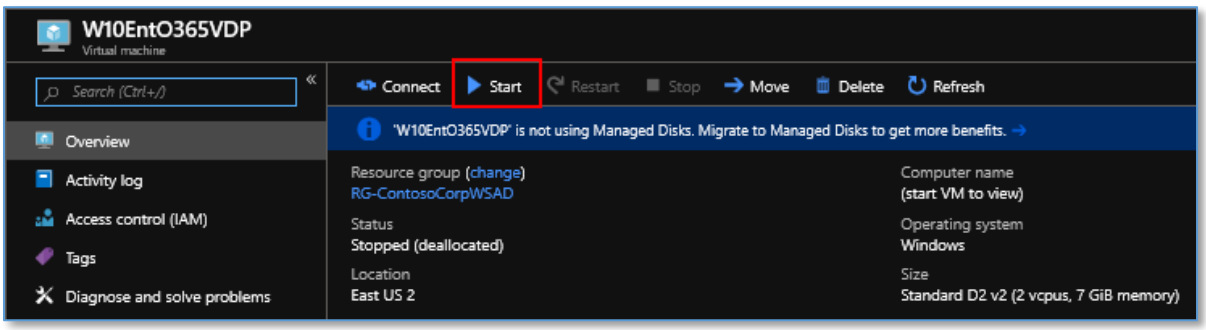
4. Select the **Snapshots** tab and click **Create snapshot:**



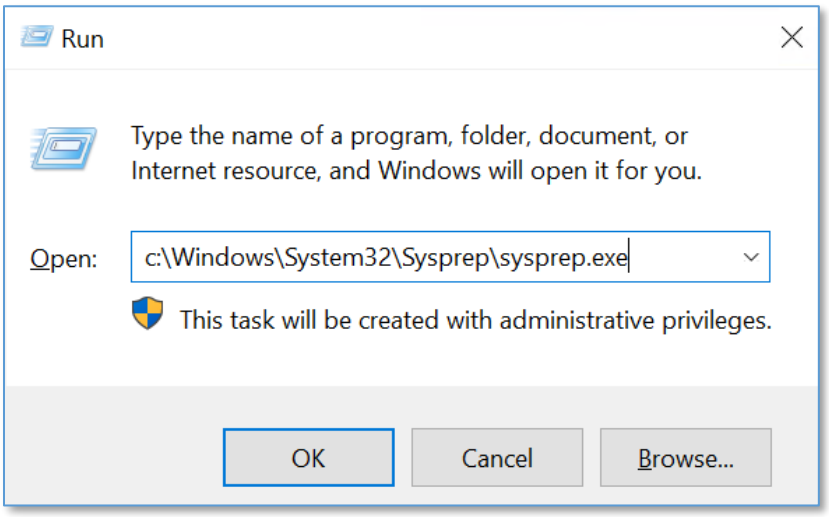
The snapshot is created:



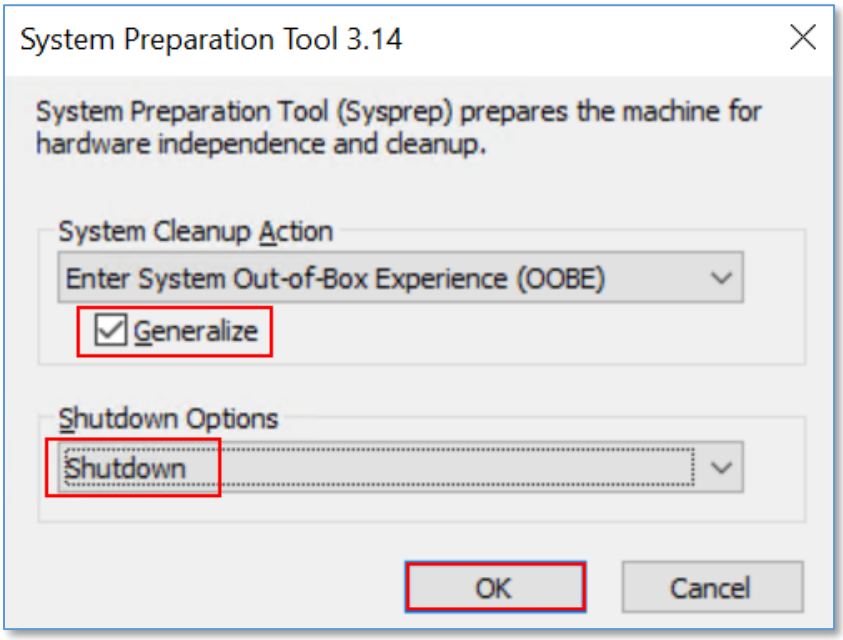
5. **Restart** the VM in Azure:

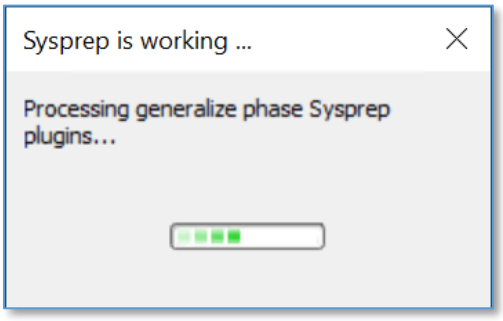


6. Logon to the VM and **run** c:\Windows\System32\Sysprep**sysprep**.exe:

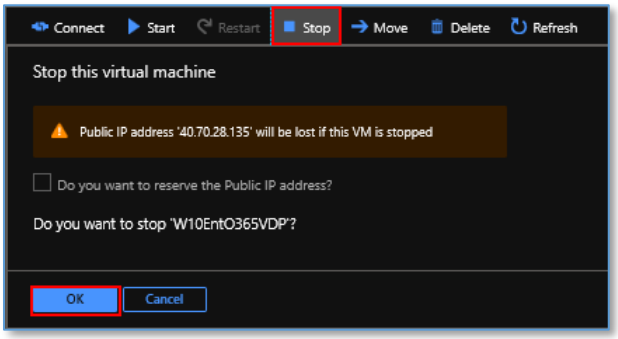


7. Select **Generalize, Shutdown,** and **OK**:

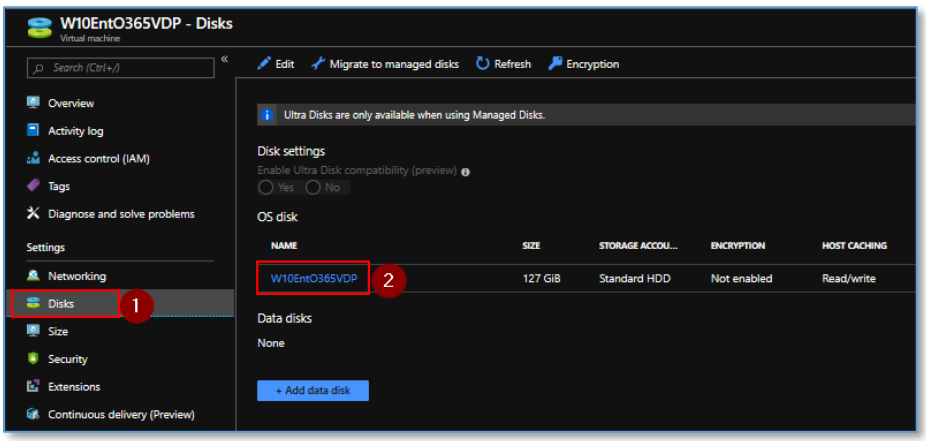




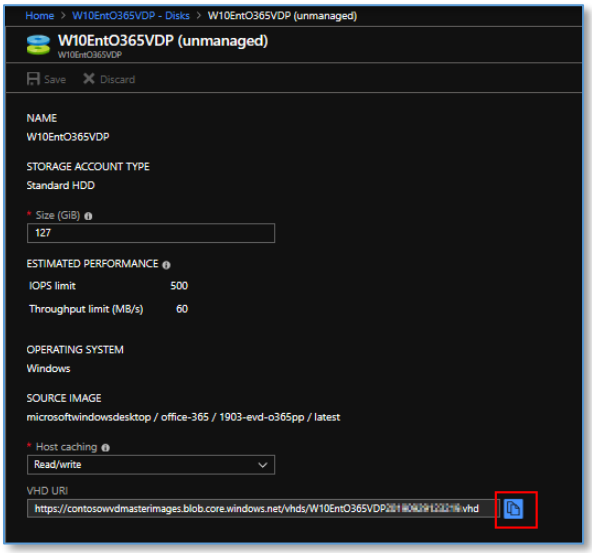
8. Back in Azure Portal, once the VM stops, deallocate it with the **Stop** button:



9. Click the VM name (open it's blade), clisk **Disks**, then click the OS **disk** name:



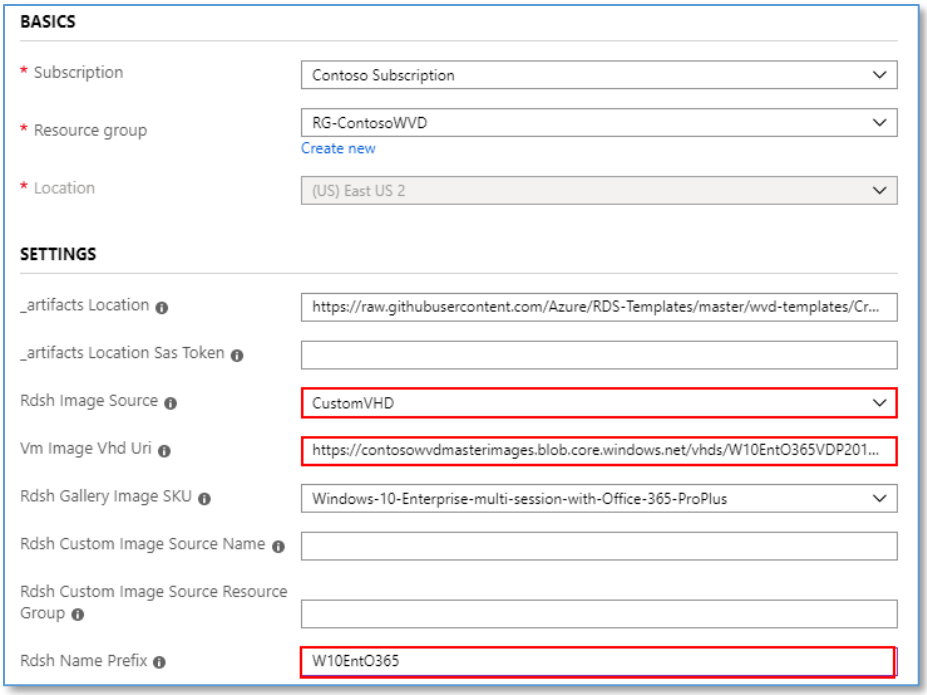
10. Copy the disks **VHD URI** and paste it to **Notepad** for use later:



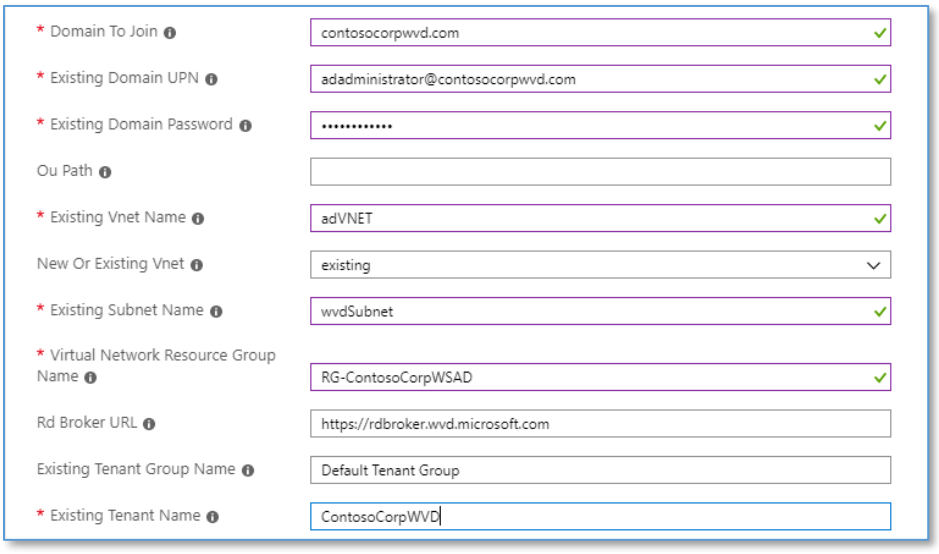
The image is now ready for deployment into a Windows Virtual Desktop host pool.

Deploy a custom image to a WVD host pool using an ARM template

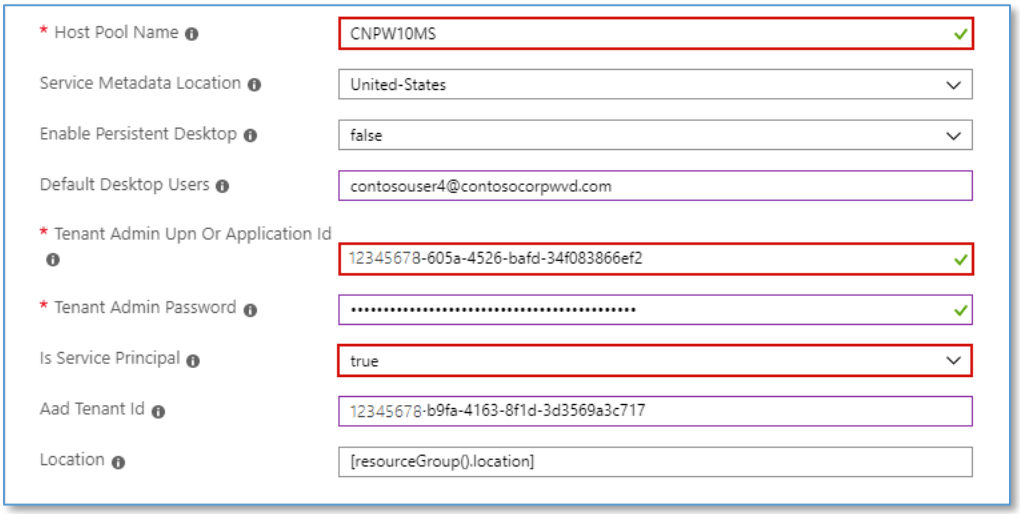
1. Browse to the Create and provision WVD host pool ARM Template, click **Deploy to Azure**, and complete the parameters as shown below, substituting your own values:
    - Rdsh Image Source = Select: CustomVHD
    - Vm Image Vhd Uri = VHD URI copied from Notepad (OS disk URI)
    - Rdsh Name Prefix = As shown below, we chose "W10EntO365"



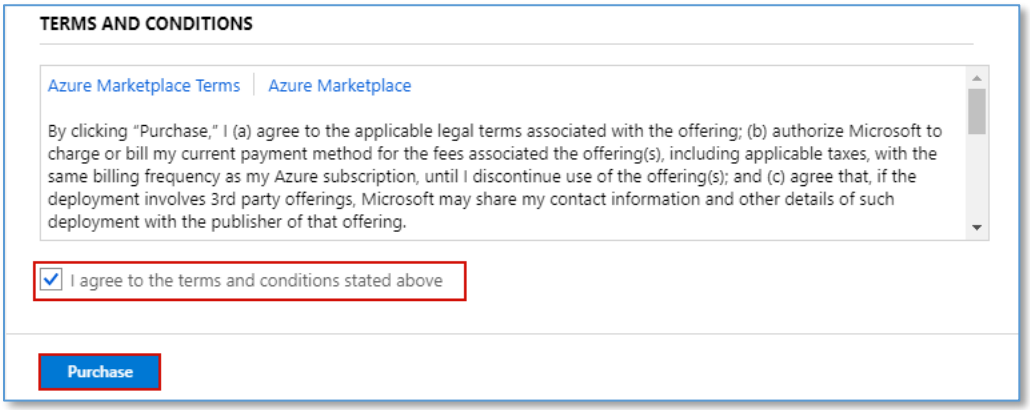
Additional required parameters:

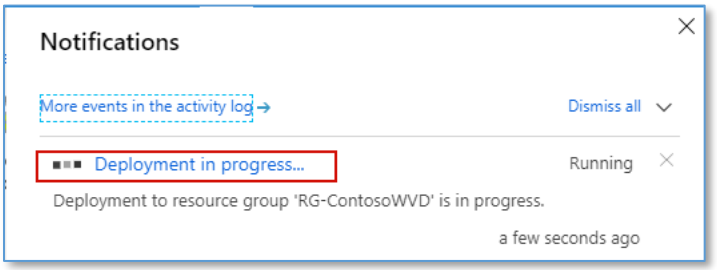


A few more, including the new host pool name and Service Principal credentials:

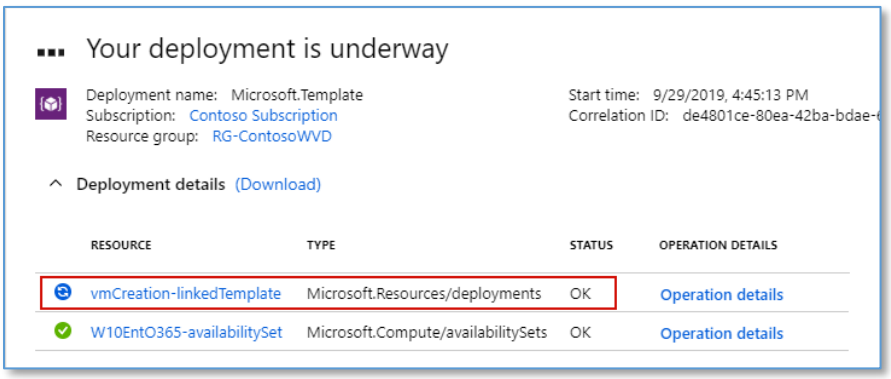


2. Click **Purchase** to deploy the new host pool.

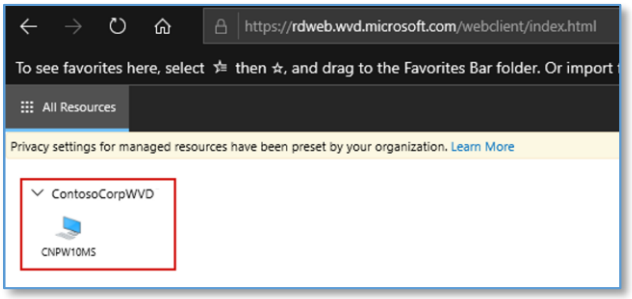


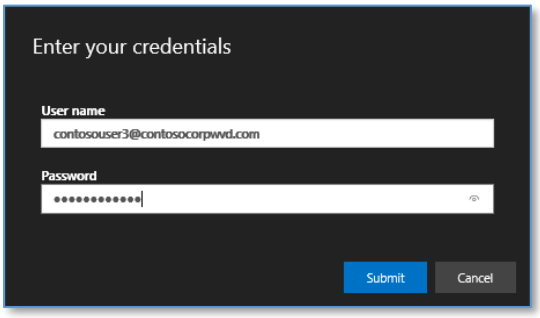


Expanding the notification details indicates the deployment progress:

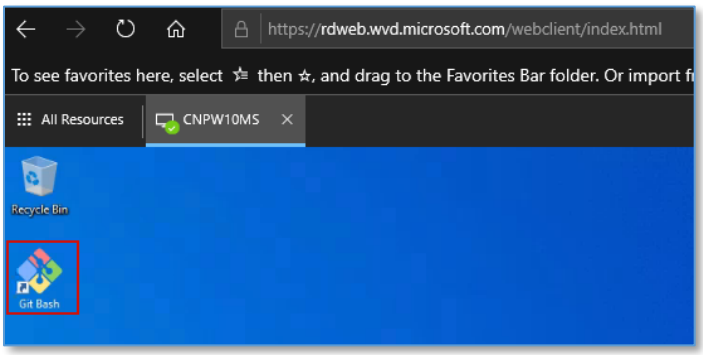


Once the deployment completes, added default desktop users may access the host pool:





The desktop opens and shows GIT, our custom application, installed:



6. Support

Opening tickets

In case of an issue for Windows Virtual Desktop go to the Azure Portal and open a technical ticket based on your existing support plan at https://azure.microsoft.com/en-us/support/create-ticket/

Look for Service under **COMPUTE** and select **Windows Virtual Desktop-Preview**. You will find options to create tickets for the WVD service itself and for Office:

For Office issues you can file tickets during public preview in the Azure Portal when using Office in context of Windows Virtual Desktop.

Information you should provide for failed connection or management interactions when using the service:

- Use the diagnostics service to retrieve the **Activity ID** for failed connections or management interactions.
- Provide the approximate timeframe the issue happened

NOTE: This workflow will change post general availability.

Other resources you can leverage

Windows Virtual Desktop contains a number of knowledge articles as well as trouble shooting guides. Pay attention to the updated diagnostics chapter that provides Error scenarios you can mitigate: https://docs.microsoft.com/azure/virtual-desktop/overview

Exchange on our community forum on issues important to you for Windows Virtual Desktop: https://techcommunity.microsoft.com/t5/Windows-Virtual-Desktop/bd-p/WindowsVirtualDesktop

When setting up your environment you will be using other Azure Services. You can watch the health dashboard here to verify health state on any Azure service you are consuming: https://azure.microsoft.com/en-us/status/