How to setup and upload custom images

For Azure

Irizarry, Irvin

Irvin.irizarry@bauschhealth.com

Contents

[How to Copy a server disk to VHD/VHDX File 2](#_Toc24382718)

[How to convert VHDX disk to VHD 3](#_Toc24382719)

[Adding the VHD File to a new Hyper-V VM: 4](#_Toc24382720)

[Set Windows configurations for Azure 7](#_Toc24382721)

[**Check the Windows services** 8](#_Toc24382722)

[**Update remote-desktop registry settings** 9](#_Toc24382723)

[**Configure Windows Firewall rules** 12](#_Toc24382724)

[**Verify the VM** 13](#_Toc24382725)

[**Install Windows updates** 16](#_Toc24382726)

[**Generalize a VHD** 16](#_Toc24382727)

[**Upload a generalized VHD to Azure to create a new VM** 18](#_Toc24382728)

[**Upload the VHD** 18](#_Toc24382729)

[**Log in to Azure** 18](#_Toc24382730)

[**Get the storage account** 18](#_Toc24382731)

[**Start the upload** 19](#_Toc24382732)

[**----------------------REVIEW------------------------------------** 20](#_Toc24382733)

[**Create a new VM** 20](#_Toc24382734)

[**Set the URI of the VHD** 20](#_Toc24382735)

[**Create a virtual network** 20](#_Toc24382736)

[**Create a public IP address and network interface** 21](#_Toc24382737)

[**Create the network security group and an RDP rule** 21](#_Toc24382738)

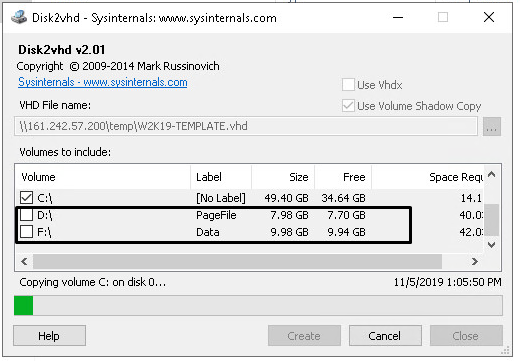
[**Create a variable for the virtual network** 22](#_Toc24382739)

[**Create the VM** 22](#_Toc24382740)

[**Verify that the VM was created** 23](#_Toc24382741)

# How to Copy a server disk to VHD/VHDX File

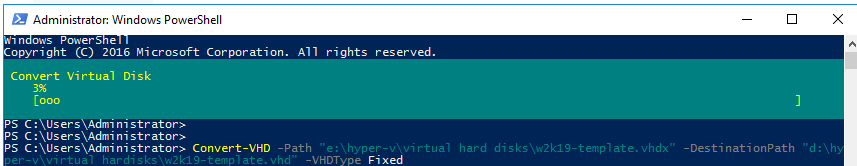
1. Hyper-V Server – Virtual server running on swvedscorvcrp04
2. IP: 161.242.57.200 – Server is running out of the domain
3. Download: Disk2vhd Software – Microsoft/Sysinternals
4. URL: <https://download.sysinternals.com/files/Disk2vhd.zip>
5. Run: Disk2vhd.exe
6. Deselect “Use Vhdx” for Biios or Select “Use Vhdx” for EFI
7. Deselect D:\ and E:\ drives and keep only the C:\ drive and any not labeled volume
8. Change the VHD/VHDX File Name path to: [\\161.242.57.200\temp\FileName.vhd](file:///\\161.242.57.200\temp\FileName.vhd) or .vhdx
9. Click on “Create”



# How to convert VHDX disk to VHD

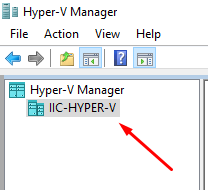
1. Open PowerShell
2. Run the following command:

*Convert-VHD –Path c:\test\MY-VM.vhdx –DestinationPath c:\test\MY-NEW-VM.vhd -VHDType Fixed*

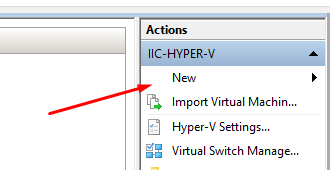


# Adding the VHD File to a new Hyper-V VM:

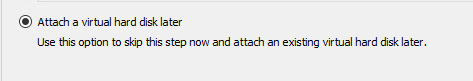
1. RDP to server 161.242.57.200 – if the server is power off, connect to swvedscorvcrp04 and power on **”IIC-Hyper-V”** Server
2. Open the Hyper-V Manager
3. Select the Hyper-V server:



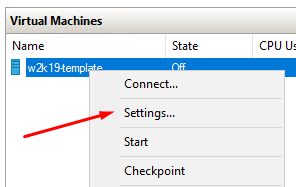
1. On the right-side menu, select New, then “Virtual Machine…”



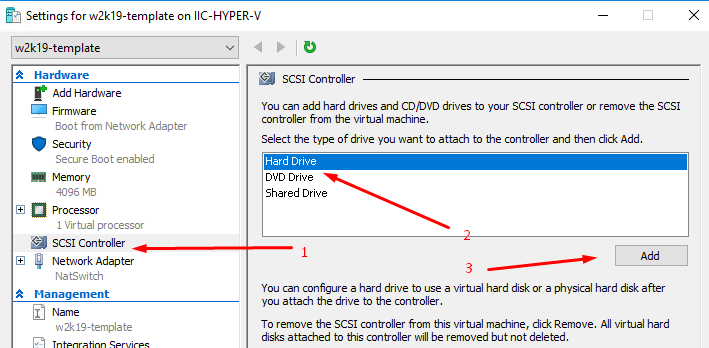
1. Click on Next
2. Enter the server name and click next
3. Select Generation 1 for BIOS or Generation 2 for EFI, then click next
4. Assign the memory, click next
5. Click on Connection and Select “NatSwitch”, click next
6. Select the option “Attach a virtual hard disk later” and Finish



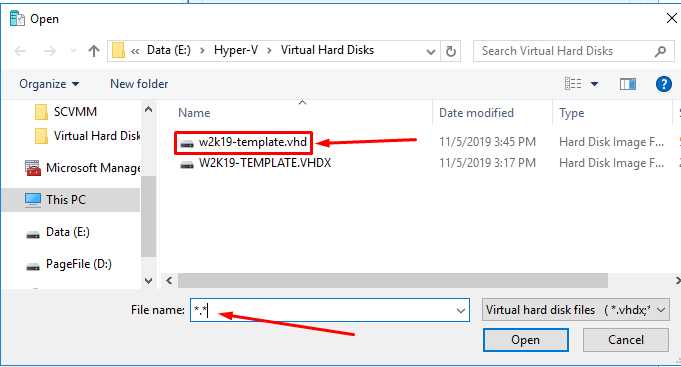
1. Right-Click on the new server and choose “Settings…”



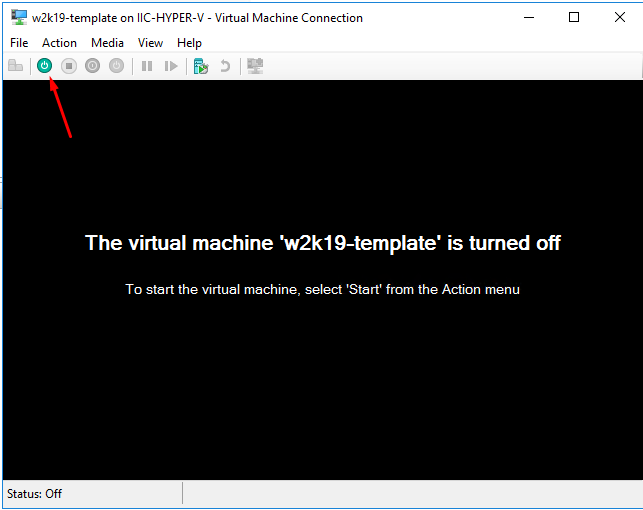
1. Select the “SCSI Controller”, then “Hard Disk”, then Click “Add”



1. Click on Browse
2. On the File Name: type \*.\* and press Enter, select the VHD file



1. Click “Apply” and “OK”
2. Power on VM



# Set Windows configurations for Azure

On the VM that you plan to upload to Azure, run the following commands from an elevated command prompt window:

1. Remove any static persistent route on the routing table:
   * To view the route table, run route print at the command prompt.
   * Check the Persistence Routes sections. If there's a persistent route, use the route delete command to remove it.
2. Remove the WinHTTP proxy:

netsh winhttp reset proxy

1. Set the disk SAN policy to [Onlineall](https://technet.microsoft.com/library/gg252636.aspx):

diskpart

In the open command prompt window, type the following commands:

san policy=onlineall

exit

1. Set Coordinated Universal Time (UTC) time for Windows. Also set the startup type of the Windows time service (w32time) to Automatic:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\TimeZoneInformation' -Name "RealTimeIsUniversal" -Value 1 -Type DWord -Force

Set-Service -Name w32time -StartupType Automatic

1. Set the power profile to high performance:

powercfg /setactive SCHEME\_MIN

1. Make sure the environmental variables TEMP and TMP are set to their default values:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name "TEMP" -Value "%SystemRoot%\TEMP" -Type ExpandString -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name "TMP" -Value "%SystemRoot%\TEMP" -Type ExpandString -Force

**Check the Windows services**

Make sure that each of the following Windows services is set to the Windows default values. These services are the minimum that must be set up to ensure VM connectivity. To reset the startup settings, run the following commands:

Get-Service -Name bfe | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name dhcp | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name dnscache | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name IKEEXT | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name iphlpsvc | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name netlogon | Where-Object { $\_.StartType -ne 'Manual' } | Set-Service -StartupType 'Manual'

Get-Service -Name netman | Where-Object { $\_.StartType -ne 'Manual' } | Set-Service -StartupType 'Manual'

Get-Service -Name nsi | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name TermService | Where-Object { $\_.StartType -ne 'Manual' } | Set-Service -StartupType 'Manual'

Get-Service -Name MpsSvc | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

Get-Service -Name RemoteRegistry | Where-Object { $\_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'

**Update remote-desktop registry settings**

Make sure the following settings are configured correctly for remote access:

**Note**

You might receive an error message when you run Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services -Name <object name> -Value <value>. You can safely ignore this message. It means only that the domain isn't pushing that configuration through a Group Policy Object.

1. Remote Desktop Protocol (RDP) is enabled:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server' -Name "fDenyTSConnections" -Value 0 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name "fDenyTSConnections" -Value 0 -Type DWord -Force

1. The RDP port is set up correctly. The default port is 3389:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "PortNumber" -Value 3389 -Type DWord -Force

When you deploy a VM, the default rules are created against port 3389. If you want to change the port number, do that after the VM is deployed in Azure.

1. The listener is listening in every network interface:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "LanAdapter" -Value 0 -Type DWord -Force

1. Configure the network-level authentication (NLA) mode for the RDP connections:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -Name "UserAuthentication" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -Name "SecurityLayer" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -Name "fAllowSecProtocolNegotiation" -Value 1 -Type DWord -Force

1. Set the keep-alive value:

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name "KeepAliveEnable" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name "KeepAliveInterval" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "KeepAliveTimeout" -Value 1 -Type DWord -Force

1. Reconnect:

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name "fDisableAutoReconnect" -Value 0 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "fInheritReconnectSame" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "fReconnectSame" -Value 0 -Type DWord -Force

1. Limit the number of concurrent connections:

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name "MaxInstanceCount" -Value 4294967295 -Type DWord -Force

1. Remove any self-signed certificates tied to the RDP listener:

if ((Get-Item -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp').Property -contains "SSLCertificateSHA1Hash")

{

Remove-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -Name "SSLCertificateSHA1Hash" -Force

}

This code ensures that you can connect at the beginning when you deploy the VM. If you need to review this later, you can do so after the VM is deployed in Azure.

1. If the VM will be part of a domain, check the following policies to make sure the former settings aren't reverted.

| **Goal** | **Policy** | **Value** |
| --- | --- | --- |
| RDP is enabled | Computer Configuration\Policies\Windows Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Allow users to connect remotely by using Remote Desktop |
| NLA group policy | Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Security | Require user authentication for remote access by using NLA |
| Keep-alive settings | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Configure keep-alive connection interval |
| Reconnect settings | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Reconnect automatically |
| Limited number of connection settings | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Limit number of connections |

**Configure Windows Firewall rules**

1. Turn on Windows Firewall on the three profiles (domain, standard, and public):

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True

1. Run the following command in PowerShell to allow WinRM through the three firewall profiles (domain, private, and public), and enable the PowerShell remote service:

Enable-PSRemoting -Force

Set-NetFirewallRule -DisplayName "Windows Remote Management (HTTP-In)" -Enabled True

1. Enable the following firewall rules to allow the RDP traffic:

Set-NetFirewallRule -DisplayGroup "Remote Desktop" -Enabled True

1. Enable the rule for file and printer sharing so the VM can respond to a ping command inside the virtual network:

Set-NetFirewallRule -DisplayName "File and Printer Sharing (Echo Request - ICMPv4-In)" -Enabled True

1. If the VM will be part of a domain, check the following Azure AD policies to make sure the former settings aren't reverted.

| **Goal** | **Policy** | **Value** |
| --- | --- | --- |
| Enable the Windows Firewall profiles | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall | Protect all network connections |
| Enable RDP | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall | Allow inbound Remote Desktop exceptions |
|  | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall | Allow inbound Remote Desktop exceptions |
| Enable ICMP-V4 | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall | Allow ICMP exceptions |
|  | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall | Allow ICMP exceptions |

**Verify the VM**

Make sure the VM is healthy, secure, and RDP accessible:

1. To make sure the disk is healthy and consistent, check the disk at the next VM restart:

Chkdsk /f

Make sure the report shows a clean and healthy disk.

1. Set the Boot Configuration Data (BCD) settings.

**Note**

Use an elevated PowerShell window to run these commands.

bcdedit /set "{bootmgr}" integrityservices enable

bcdedit /set "{default}" device partition=C:

bcdedit /set "{default}" integrityservices enable

bcdedit /set "{default}" recoveryenabled Off

bcdedit /set "{default}" osdevice partition=C:

bcdedit /set "{default}" bootstatuspolicy IgnoreAllFailures

#Enable Serial Console Feature

bcdedit /set "{bootmgr}" displaybootmenu yes

bcdedit /set "{bootmgr}" timeout 5

bcdedit /set "{bootmgr}" bootems yes

bcdedit /ems "{current}" ON

bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200

1. The dump log can be helpful in troubleshooting Windows crash issues. Enable the dump log collection:

# Set up the guest OS to collect a kernel dump on an OS crash event

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name CrashDumpEnabled -Type DWord -Force -Value 2

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name DumpFile -Type ExpandString -Force -Value "%SystemRoot%\MEMORY.DMP"

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name NMICrashDump -Type DWord -Force -Value 1

# Set up the guest OS to collect user mode dumps on a service crash event

$key = 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps'

if ((Test-Path -Path $key) -eq $false) {(New-Item -Path 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error Reporting' -Name LocalDumps)}

New-ItemProperty -Path $key -Name DumpFolder -Type ExpandString -Force -Value "c:\CrashDumps"

New-ItemProperty -Path $key -Name CrashCount -Type DWord -Force -Value 10

New-ItemProperty -Path $key -Name DumpType -Type DWord -Force -Value 2

Set-Service -Name WerSvc -StartupType Manual

1. Verify that the Windows Management Instrumentation (WMI) repository is consistent:

winmgmt /verifyrepository

1. Make sure no other application is using port 3389. This port is used for the RDP service in Azure. To see which ports are used on the VM, run netstat -anob:

netstat -anob

1. Make sure you know the built-in administrator account and password. You might want to reset the current local administrator password and make sure you can use this account to sign in to Windows through the RDP connection. This access permission is controlled by the "Allow log on through Remote Desktop Services" Group Policy Object. View this object in the Local Group Policy Editor here:

Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment

1. Check the following Azure AD policies to make sure you're not blocking your RDP access through RDP or from the network:
   * Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny access to this computer from the network
   * Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny log on through Remote Desktop Services
2. Check the following Azure AD policy to make sure you're not removing any of the required access accounts:
   * Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Access this computer from the network

The policy should list the following groups:

* + Administrators
  + Backup Operators
  + Everyone
  + Users

1. Restart the VM to make sure that Windows is still healthy and can be reached through the RDP connection. At this point, you might want to create a VM in your local Hyper-V to make sure the VM starts completely. Then test to make sure you can reach the VM through RDP.
2. Remove any extra Transport Driver Interface (TDI) filters. For example, remove software that analyzes TCP packets or extra firewalls. If you need to review this later, you can do so after the VM is deployed in Azure.
3. Uninstall any other third-party software or driver that's related to physical components or any other virtualization technology.

**Install Windows updates**

Ideally, you should keep the machine updated at the *patch level*. If this isn't possible, make sure the following updates are installed:

**Note**

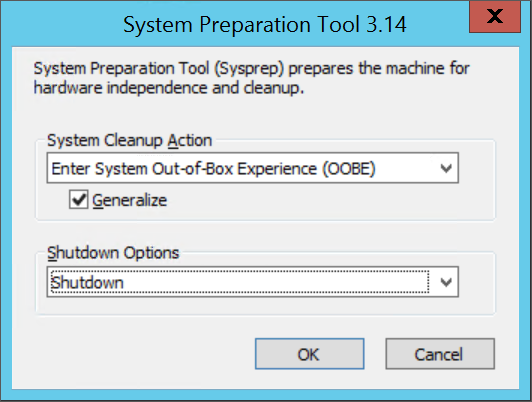
To avoid an accidental reboot during VM provisioning, we recommend ensuring that all Windows Update installations are finished and that no updates are pending. One way to do this is to install all possible Windows updates and reboot once before you run the Sysprep command.

**Generalize a VHD**

**Note**

After you run sysprep.exe in the following steps, turn off the VM. Don't turn it back on until you create an image from it in Azure.

1. Sign in to the Windows VM.
2. Run **Command Prompt** as an administrator.
3. Change the directory to %windir%\system32\sysprep. Then run sysprep.exe.
4. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that the **Generalize** check box is selected.



1. In **Shutdown Options**, select **Shutdown**.
2. Select **OK**.
3. When Sysprep finishes, shut down the VM. Don't use **Restart** to shut down the VM. Do not Power-on the VM again.

**Upload a generalized VHD to Azure to create a new VM**

**Upload the VHD**

Upload the VHD to an Azure storage account.

**Log in to Azure**

1. Open Azure PowerShell and sign in to your Azure account. A pop-up window opens for you to enter your Azure account credentials.

Connect-AzAccount

1. Get the subscription IDs for your available subscriptions.

Get-AzSubscription

1. Set the correct subscription using the subscription ID. Replace <subscriptionID> with the ID of the correct subscription.

Select-AzSubscription -SubscriptionId "<subscriptionID>"

**Get the storage account**

For windows 2019 use:

**Resource Group:** *NPS-INF-AMER-WIN2K19STD* on EAST US

**Storage Account:** *stow2k19aannpsslh01* in EAST US

You need a storage account in Azure to store the uploaded VM image. You can either use an existing storage account or create a new one.

To show the available storage accounts, type:

Get-AzStorageAccount

If you want to use an existing storage account, proceed to the Upload the VM image section.

If you need to create a storage account, follow these steps:

1. You need the name of the resource group where the storage account should be created. To find out all the resource groups that are in your subscription, type:

Get-AzResourceGroup

To create a resource group named **myResourceGroup** in the **East US** region, type:

New-AzResourceGroup -Name myResourceGroup -Location "East US"

1. Create a storage account named **mystorageaccount** in this resource group by using the New-AzStorageAccount cmdlet:

New-AzStorageAccount -ResourceGroupName myResourceGroup -Name mystorageaccount -Location "East US" -SkuName "Standard\_LRS" -Kind "Storage"

**Start the upload**

Use the Add-AzVhd cmdlet to upload the image to a container in your storage account. This example uploads the file **myVHD.vhd** from "C:\Users\Public\Documents\Virtual hard disks\" to a storage account named **mystorageaccount** in the **myResourceGroup** resource group. The file will be placed into the container named **mycontainer** and the new file name will be **myUploadedVHD.vhd**.

$rgName = "myResourceGroup"

$urlOfUploadedImageVhd = "https://mystorageaccount.blob.core.windows.net/mycontainer/myUploadedVHD.vhd"

Add-AzVhd -ResourceGroupName $rgName -Destination $urlOfUploadedImageVhd `

-LocalFilePath "C:\Users\Public\Documents\Virtual hard disks\myVHD.vhd"

*$rgName = "NPS-INF-AMER-WIN2K19STD"*

*$urlOfUploadedImageVhd = "https://stow2k19aannpsslh01.blob.core.windows.net/stow2k19aannpsslh01/w2k19-Template.vhd"*

*Add-AzVhd -ResourceGroupName $rgName -Destination $urlOfUploadedImageVhd -LocalFilePath "E:\Hyper-V\Virtual Hard Disks\w2k19-template.vhd"*

If successful, you get a response that looks similar to this:

MD5 hash is being calculated for the file C:\Users\Public\Documents\Virtual hard disks\myVHD.vhd.

MD5 hash calculation is completed.

Elapsed time for the operation: 00:03:35

Creating new page blob of size 53687091712...

Elapsed time for upload: 01:12:49

LocalFilePath DestinationUri

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

C:\Users\Public\Doc... https://mystorageaccount.blob.core.windows.net/mycontainer/myUploadedVHD.vhd

Depending on your network connection and the size of your VHD file, this command may take a while to complete.

**----------------------REVIEW------------------------------------**

**Create a new VM**

You can now use the uploaded VHD to create a new VM.

**Set the URI of the VHD**

The URI for the VHD to use is in the format: https://**mystorageaccount**.blob.core.windows.net/**mycontainer**/**MyVhdName**.vhd. In this example the VHD named **myVHD** is in the storage account **mystorageaccount** in the container **mycontainer**.

$imageURI = "https://mystorageaccount.blob.core.windows.net/mycontainer/myVhd.vhd"

**Create a virtual network**

Create the vNet and subnet of the [virtual network](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview).

1. Create the subnet. The following sample creates a subnet named **mySubnet** in the resource group **myResourceGroup** with the address prefix of **10.0.0.0/24**.

$rgName = "myResourceGroup"

$subnetName = "mySubnet"

$singleSubnet = New-AzVirtualNetworkSubnetConfig -Name $subnetName -AddressPrefix 10.0.0.0/24

1. Create the virtual network. The following sample creates a virtual network named **myVnet** in the **West US** location with the address prefix of **10.0.0.0/16**.

$location = "WestUS"

$vnetName = "myVnet"

$vnet = New-AzVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $location `

-AddressPrefix 10.0.0.0/16 -Subnet $singleSubnet

**Create a public IP address and network interface**

To enable communication with the virtual machine in the virtual network, you need a [public IP address](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-ip-addresses-overview-arm) and a network interface.

1. Create a public IP address. This example creates a public IP address named **myPip**.

$ipName = "myPip"

$pip = New-AzPublicIpAddress -Name $ipName -ResourceGroupName $rgName -Location $location `

-AllocationMethod Dynamic

1. Create the NIC. This example creates a NIC named **myNic**.

$nicName = "myNic"

$nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $location `

-SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id

**Create the network security group and an RDP rule**

To be able to log in to your VM using RDP, you need to have a security rule that allows RDP access on port 3389.

This example creates an NSG named **myNsg** that contains a rule called **myRdpRule** that allows RDP traffic over port 3389. For more information about NSGs, see [Opening ports to a VM in Azure using PowerShell](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/nsg-quickstart-powershell?toc=%2fazure%2fvirtual-machines%2fwindows%2ftoc.json).

$nsgName = "myNsg"

$rdpRule = New-AzNetworkSecurityRuleConfig -Name myRdpRule -Description "Allow RDP" `

-Access Allow -Protocol Tcp -Direction Inbound -Priority 110 `

-SourceAddressPrefix Internet -SourcePortRange \* `

-DestinationAddressPrefix \* -DestinationPortRange 3389

$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $rgName -Location $location `

-Name $nsgName -SecurityRules $rdpRule

**Create a variable for the virtual network**

Create a variable for the completed virtual network.

$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name $vnetName

**Create the VM**

The following PowerShell script shows how to set up the virtual machine configurations and use the uploaded VM image as the source for the new installation.

# Enter a new user name and password to use as the local administrator account

# for remotely accessing the VM.

$cred = Get-Credential

# Name of the storage account where the VHD is located. This example sets the

# storage account name as "myStorageAccount"

$storageAccName = "myStorageAccount"

# Name of the virtual machine. This example sets the VM name as "myVM".

$vmName = "myVM"

# Size of the virtual machine. This example creates "Standard\_D2\_v2" sized VM.

# See the VM sizes documentation for more information:

# https://azure.microsoft.com/documentation/articles/virtual-machines-windows-sizes/

$vmSize = "Standard\_D2\_v2"

# Computer name for the VM. This examples sets the computer name as "myComputer".

$computerName = "myComputer"

# Name of the disk that holds the OS. This example sets the

# OS disk name as "myOsDisk"

$osDiskName = "myOsDisk"

# Assign a SKU name. This example sets the SKU name as "Standard\_LRS"

# Valid values for -SkuName are: Standard\_LRS - locally redundant storage, Standard\_ZRS - zone redundant

# storage, Standard\_GRS - geo redundant storage, Standard\_RAGRS - read access geo redundant storage,

# Premium\_LRS - premium locally redundant storage.

$skuName = "Standard\_LRS"

# Get the storage account where the uploaded image is stored

$storageAcc = Get-AzStorageAccount -ResourceGroupName $rgName -AccountName $storageAccName

# Set the VM name and size

$vmConfig = New-AzVMConfig -VMName $vmName -VMSize $vmSize

#Set the Windows operating system configuration and add the NIC

$vm = Set-AzVMOperatingSystem -VM $vmConfig -Windows -ComputerName $computerName `

-Credential $cred -ProvisionVMAgent -EnableAutoUpdate

$vm = Add-AzVMNetworkInterface -VM $vm -Id $nic.Id

# Create the OS disk URI

$osDiskUri = '{0}vhds/{1}-{2}.vhd' `

-f $storageAcc.PrimaryEndpoints.Blob.ToString(), $vmName.ToLower(), $osDiskName

# Configure the OS disk to be created from the existing VHD image (-CreateOption fromImage).

$vm = Set-AzVMOSDisk -VM $vm -Name $osDiskName -VhdUri $osDiskUri `

-CreateOption fromImage -SourceImageUri $imageURI -Windows

# Create the new VM

New-AzVM -ResourceGroupName $rgName -Location $location -VM $vm

**Verify that the VM was created**

When complete, you should see the newly created VM in the [Azure portal](https://portal.azure.com/) under **Browse** > **Virtual machines**, or by using the following PowerShell commands:

$vmList = Get-AzVM -ResourceGroupName $rgName

$vmList.Name