Create a custom image of an Azure VM using PowerShell

Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap configurations such as preloading applications, application configurations, and other OS configurations. In this tutorial, you create your own custom image of an Azure virtual machine. You learn how to:

* Sysprep and generalize VMs
* Create a custom image
* Create a VM from a custom image
* List all the images in your subscription
* Delete an image

This tutorial requires the Azure PowerShell module version 3.6 or later. Run Get-Module -ListAvailable AzureRM to find the version. If you need to upgrade, see [Install Azure PowerShell module](https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps).

Before you begin

The steps below detail how to take an existing VM and turn it into a re-usable custom image that you can use to create new VM instances.

To complete the example in this tutorial, you must have an existing virtual machine. If needed, this [script sample](https://docs.microsoft.com/en-us/azure/virtual-machines/scripts/virtual-machines-windows-powershell-sample-create-vm) can create one for you. When working through the tutorial, replace the resource group and VM names where needed.

Prepare VM

To create an image of a virtual machine, you need to prepare the VM by generalizing the VM, deallocating, and then marking the source VM as generalized in Azure.

Generalize the Windows VM using Sysprep

Sysprep removes all your personal account information, among other things, and prepares the machine to be used as an image. For details about Sysprep, see [How to Use Sysprep: An Introduction](http://technet.microsoft.com/library/bb457073.aspx).

1. Connect to the virtual machine.
2. Open the Command Prompt window as an administrator. Change the directory to *%windir%\system32\sysprep*, and then run *sysprep.exe*.
3. 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.
4. In **Shutdown Options**, select *Shutdown* and then click **OK**.
5. When Sysprep completes, it shuts down the virtual machine. **Do not restart the VM**.

Deallocate and mark the VM as generalized

To create an image, the VM needs to be deallocated and marked as generalized in Azure.

Deallocated the VM using [Stop-AzureRmVM](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/stop-azurermvm).

PowerShellCopy

Stop-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Force

Set the status of the virtual machine to -Generalized using [Set-AzureRmVm](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/set-azurermvm).

PowerShellCopy

Set-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Generalized

Create the image

Now you can create an image of the VM by using [New-AzureRmImageConfig](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/new-azurermimageconfig) and [New-AzureRmImage](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/new-azurermimage). The following example creates an image named *myImage* from a VM named *myVM*.

Get the virtual machine.

PowerShellCopy

$vm = Get-AzureRmVM -Name myVM -ResourceGroupName myResourceGroup

Create the image configuration.

PowerShellCopy

$image = New-AzureRmImageConfig -Location EastUS -SourceVirtualMachineId $vm.ID

Create the image.

PowerShellCopy

New-AzureRmImage -Image $image -ImageName myImage -ResourceGroupName myResourceGroup

Create VMs from the image

Now that you have an image, you can create one or more new VMs from the image. Creating a VM from a custom image is very similar to creating a VM using a Marketplace image. When you use a Marketplace image, you have to provide the information about the image, image provider, offer, SKU and version. With a custom image, you just need to provide the ID of the custom image resource.

In the following script, we create a variable *$image* to store information about the custom image using [Get-AzureRmImage](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/get-azurermimage) and then we use [Set-AzureRmVMSourceImage](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/set-azurermvmsourceimage) and specify the ID using the *$image* variable we just created.

The script creates a VM named *myVMfromImage* from our custom image in a new resource group named *myResourceGroupFromImage* in the *West US* location.

PowerShellCopy

$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

New-AzureRmResourceGroup -Name myResourceGroupFromImage -Location EastUS

$subnetConfig = New-AzureRmVirtualNetworkSubnetConfig `

-Name mySubnet `

-AddressPrefix 192.168.1.0/24

$vnet = New-AzureRmVirtualNetwork `

-ResourceGroupName myResourceGroupFromImage `

-Location EastUS `

-Name MYvNET `

-AddressPrefix 192.168.0.0/16 `

-Subnet $subnetConfig

$pip = New-AzureRmPublicIpAddress `

-ResourceGroupName myResourceGroupFromImage `

-Location EastUS `

-Name "mypublicdns$(Get-Random)" `

-AllocationMethod Static `

-IdleTimeoutInMinutes 4

$nsgRuleRDP = New-AzureRmNetworkSecurityRuleConfig `

-Name myNetworkSecurityGroupRuleRDP `

-Protocol Tcp `

-Direction Inbound `

-Priority 1000 `

-SourceAddressPrefix \* `

-SourcePortRange \* `

-DestinationAddressPrefix \* `

-DestinationPortRange 3389 `

-Access Allow

$nsg = New-AzureRmNetworkSecurityGroup `

-ResourceGroupName myResourceGroupFromImage `

-Location EastUS `

-Name myNetworkSecurityGroup `

-SecurityRules $nsgRuleRDP

$nic = New-AzureRmNetworkInterface `

-Name myNic `

-ResourceGroupName myResourceGroupFromImage `

-Location EastUS `

-SubnetId $vnet.Subnets[0].Id `

-PublicIpAddressId $pip.Id `

-NetworkSecurityGroupId $nsg.Id

$vmConfig = New-AzureRmVMConfig `

-VMName myVMfromImage `

-VMSize Standard\_D1 | Set-AzureRmVMOperatingSystem -Windows `

-ComputerName myComputer `

-Credential $cred

# Here is where we create a variable to store information about the image

$image = Get-AzureRmImage `

-ImageName myImage `

-ResourceGroupName myResourceGroup

# Here is where we specify that we want to create the VM from and image and provide the image ID

$vmConfig = Set-AzureRmVMSourceImage -VM $vmConfig -Id $image.Id

$vmConfig = Add-AzureRmVMNetworkInterface -VM $vmConfig -Id $nic.Id

New-AzureRmVM `

-ResourceGroupName myResourceGroupFromImage `

-Location EastUS `

-VM $vmConfig

Image management

Here are some examples of common management image tasks and how to complete them using PowerShell.

List all images by name.

PowerShellCopy

$images = Find-AzureRMResource -ResourceType Microsoft.Compute/images

$images.name

Delete an image. This example deletes the image named *myOldImage* from the *myResourceGroup*.

PowerShellCopy

Remove-AzureRmImage `

-ImageName myOldImage `

-ResourceGroupName myResourceGroup