Tutorial: Create and Manage Windows VMs with Azure PowerShell

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create resource group

Create a resource group with the New-AzResourceGroup command.

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine. In the following example, a resource group named *myRGroupVM* is created in the *WestUS* region:

```
New-AzResourceGroup `
   -ResourceGroupName "myRGroupVM" `
   -Location "WestUS"
```

Create a VM

When creating a VM, several options are available like operating system image, network configuration, and administrative credentials. This example creates a VM named *myVM*, running the default version of Windows Server 2016 Datacenter.

Set the username and password needed for the administrator account on the VM with Get-Credential:

```
$cred = Get-Credential
```

Create the VM with New-AzVM.

```
New-AzVm `
    -ResourceGroupName "myRGroupVM" `
    -Name "myVM" `
    -Location "WestUS" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -Credential $cred
```

Connect to VM

Run the following commands to return the public IP address of the VM. Take note of this IP Address so you can connect to it with your browser to test web connectivity in a future step.

```
Get-AzPublicIpAddress `
   -ResourceGroupName "myResourceGroupVM"  | Select IpAddress
```

Use the following command, on your local machine, to create a remote desktop session with the VM. Replace the IP address with the *publicIPAddress* of your VM. When prompted, enter the credentials used when creating the VM.

```
mstsc /v:<publicIpAddress>
```

In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username and password you created for the VM and then click **OK**.

Understand marketplace images

Use the Get-AzVMImagePublisher command to return a list of image publishers:

```
Get-AzVMImagePublisher -Location "WestUS"
```

Use the Get-AzVMImageOffer to return a list of image offers. With this command, the returned list is filtered on the specified publisher named `MicrosoftWindowsServer`:

```
Get-AzVMImageOffer `
   -Location "WestUS" `
   -PublisherName "MicrosoftWindowsServer"
```

The results will look something like this example:

```
Offer              PublisherName          Location
-----              -------------          --------
Windows-HUB        MicrosoftWindowsServer EastUS
```

```
WindowsServer     MicrosoftWindowsServer EastUS
WindowsServer-HUB MicrosoftWindowsServer EastUS
```

The Get-AzVMImageSku command will then filter on the publisher and offer name to return a list of image names.

```
Get-AzVMImageSku `
   -Location "WestUS" `
   -PublisherName "MicrosoftWindowsServer" `
   -Offer "WindowsServer"
```

The results will look something like this example:

PowerShellCopy

```
Skus                                      Offer         PublisherName
Location
----                                      -----         -------------          ------
--
2008-R2-SP1                               WindowsServer MicrosoftWindowsServer EastUS
2008-R2-SP1-smalldisk                     WindowsServer MicrosoftWindowsServer EastUS
2012-Datacenter                           WindowsServer MicrosoftWindowsServer EastUS
2012-Datacenter-smalldisk                 WindowsServer MicrosoftWindowsServer EastUS
2012-R2-Datacenter                        WindowsServer MicrosoftWindowsServer EastUS
2012-R2-Datacenter-smalldisk              WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter                           WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-Server-Core               WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-Server-Core-smalldisk     WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-smalldisk                 WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-with-Containers           WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-with-Containers-smalldisk WindowsServer MicrosoftWindowsServer EastUS
2016-Datacenter-with-RDSH                 WindowsServer MicrosoftWindowsServer EastUS
2016-Nano-Server                          WindowsServer MicrosoftWindowsServer EastUS
```

This information can be used to deploy a VM with a specific image. This example deploys a VM using the latest version of a Windows Server 2016 with Containers image.

```
New-AzVm `
    -ResourceGroupName "myRGroupVM" `
    -Name "myVM2" `
    -Location "WestUS" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress2" `
    -ImageName "MicrosoftWindowsServer:WindowsServer:2016-Datacenter-with-Containers:latest" `
    -Credential $cred `
    -AsJob
```

The `-AsJob` parameter creates the VM as a background task, so the PowerShell prompts return to you. You can view details of background jobs with the `Get-Job` cmdlet.

Understand VM sizes

The VM size determines the amount of compute resources like CPU, GPU, and memory that are made available to the VM. Virtual machines should be created using a VM size appropriate for the workload. If a workload increases, an existing virtual machine can also be resized.

VM Sizes

The following table categorizes sizes into use cases.

Type	Common sizes	Description
General purpose	B, Dsv3, Dv3, DSv2, Dv2, Av2, DC	Balanced CPU-to-memory. Ideal for dev / test and small to medium applications and data solutions.
Compute optimized	Fsv2	High CPU-to-memory. Good for medium traffic applications, network appliances, and batch processes.
Memory optimized	Esv3, Ev3, M, DSv2, Dv2	High memory-to-core. Great for relational databases, medium to large caches, and in-memory analytics.
Storage optimized	Lsv2, Ls	High disk throughput and IO. Ideal for Big Data, SQL, and NoSQL databases.
GPU	NV, NVv2, NC, NCv2, NCv3, ND	Specialized VMs targeted for heavy graphic rendering and video editing.
High performance	H	Our most powerful CPU VMs with optional high-throughput network interfaces (RDMA).

To see a list of VM sizes available in a particular region, use the Get-AzVMSize command.

```
Get-AzVMSize -Location "EastUS"
```

Resize a VM

Before resizing a VM, check if the size you want is available on the current VM cluster. The Get-AzVMSize command returns a list of sizes.

```
Get-AzVMSize -ResourceGroupName "myRGroupVM" -VMName "myVM"
```

If the size is available, the VM can be resized from a powered-on state, however it is rebooted during the operation.

```
$vm = Get-AzVM `
   -ResourceGroupName "myRGroupVM"  `
   -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_DS3_v2"
Update-AzVM `
   -VM $vm `
   -ResourceGroupName "myRGroupVM"
```

If the size you want isn't available on the current cluster, the VM needs to be deallocated before the resize operation can occur. Deallocating a VM will remove any data on the temp disk, and the public IP address will change unless a static IP address is being used.

```
Stop-AzVM `
   -ResourceGroupName "myRGroupVM" `
   -Name "myVM" -Force
$vm = Get-AzVM `
   -ResourceGroupName "myRGroupVM"  `
   -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_E2s_v3"
Update-AzVM -VM $vm `
   -ResourceGroupName "myRGroupVM"
Start-AzVM `
   -ResourceGroupName "myRGroupVM"  `
   -Name $vm.name
```

VM power states

An Azure VM can have one of many power states.

Power State	Description
Starting	The virtual machine is being started.
Running	The virtual machine is running.
Stopping	The virtual machine is being stopped.
Stopped	The VM is stopped. Virtual machines in the stopped state still incur compute charges.
Deallocating	The VM is being deallocated.
Deallocated	Indicates that the VM is removed from the hypervisor but is still available in the control plane. Virtual machines in the `Deallocated` state do not incur compute charges.
-	The power state of the VM is unknown.

To get the state of a particular VM, use the Get-AzVM command. Be sure to specify a valid name for a VM and resource group.

```
Get-AzVM `
    -ResourceGroupName "myRGroupVM" `
    -Name "myVM" `
    -Status | Select @{n="Status"; e={$_.Statuses[1].Code}}
```

The output will look something like this example:

```
Status
------
PowerState/running
```

Management tasks

Stop a VM

Stop and deallocate a VM with Stop-AzVM:

```
Stop-AzVM `
   -ResourceGroupName "myRGroupVM" `
   -Name "myVM" -Force
```

If you want to keep the VM in a provisioned state, use the -StayProvisioned parameter.

```
Start-AzVM `
   -ResourceGroupName "myRGroupVM" `
   -Name "myVM"
```

Delete resource group

Everything inside of a resource group is deleted when you delete the resource group.

```
Remove-AzResourceGroup `
   -Name "myRaGroupVM" `
   -Force
```

ALL DONE FOR NOW!