**Step 1:**

**Sign in**

If you're using a local install of the Azure PowerShell, you need to sign in before performing any other steps.

Connect-AzureRmAccount

Complete the sign in process by following the steps displayed in your terminal.

**Step 2:**

**Create a resource group**

In Azure, all resources are allocated in a resource management group. Resource groups provide logical groupings of resources that make them easier to work with as a collection. For this tutorial, all of the created resources go into a single group named TutorialResources.

New-AzureRmResourceGroup -Name TutorialResources -Location eastus

OutputCopy

ResourceGroupName : TutorialResources

Location : eastus

ProvisioningState : Succeeded

Tags :

ResourceId : /subscriptions/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/resourceGroups/Tutor

**Step 3: Create admin credentials for the VM**

Before you can create a new virtual machine, you must create a credential object containing the username and password for the administrator account of the Windows VM.

PowerShellCopy

Try It

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

Enter the username and password when prompted. The resulting credential object is passed as a parameter in the next step.

OutputCopy

Windows PowerShell credential request.

Enter a username and password for the virtual machine.

User: tutorAdmin

Password for user tutorAdmin: \*\*\*\*\*\*\*\*\*

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

$vmParams = @{

ResourceGroupName = 'TutorialResources'

Name = 'TutorialVM1'

Location = 'eastus'

ImageName = 'Win2016Datacenter'

PublicIpAddressName = 'tutorialPublicIp'

Credential = $cred

OpenPorts = 3389

}

$newVM1 = New-AzureRmVM @vmParams

Step 4:

**Create a virtual machine**

Virtual machines in Azure have a large number of dependencies. The Azure PowerShell creates these resources for you based on the command-line arguments you specify. For readability, we are using [PowerShell splatting](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_splatting) to pass parameters to the Azure PowerShell cmdlets.

Create a new virtual machine running Windows.

$vmParams = @{

ResourceGroupName = 'TutorialResources'

Name = 'TutorialVM1'

Location = 'eastus'

ImageName = 'Win2016Datacenter'

PublicIpAddressName = 'tutorialPublicIp'

Credential = $cred

OpenPorts = 3389

}

$newVM1 = New-AzureRmVM @vmParams

As the VM is created, you see the parameter values used and Azure resources being created. PowerShell will displace a progress bar as shown below.

OutputCopy

Creating Azure resources

39% \

[ooooooooooooooooooooooooooooooooooo ]

Creating TutorialVM1 virtual machine.

Once the VM is ready, we can view the results in the Azure Portal or by inspecting the $newVM1 variable.

$newVM1

Property values listed inside of braces are nested objects. In the next step we will show you how to view specific values in these nested objects.

## Get VM information with queries

L et's get some more detailed information from the VM we just created. In this example, we verify the Name of the VM and the admin account we created.

$newVM1.OSProfile | Select-Object ComputerName,AdminUserName

We can use other Azure PowerShell commands to get specific information about the network configuration.

$newVM1 | Get-AzureRmNetworkInterface |

Select-Object -ExpandProperty IpConfigurations |

Select-Object Name,PrivateIpAddress

To confirm that the VM is running, we need to connect via Remote Desktop. For that, we need to know the Public IP address.

$publicIp = Get-AzureRmPublicIpAddress -Name tutorialPublicIp -ResourceGroupName TutorialResources

$publicIp | Select-Object Name,IpAddress,@{label='FQDN';expression={$\_.DnsSettings.Fqdn}}

In this example, we use the Get-AzureRmPublicIpAddress and store the results in the $publicIp variable. From this variable we are selecting properties and using an expression to retrieve the nested Fqdn property.

Name IpAddress FQDN

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

tutorialPublicIp <PUBLIC\_IP\_ADDRESS> tutorialvm1-8a0999.eastus.cloudapp.azure.com

From your local machine you can run the following command to connect to the VM over Remote Desktop.

mstsc.exe /v <PUBLIC\_IP\_ADDRESS>

Go to cmd prompt

mstsc.exe /v 13.82.220.248

# Install IIS

$PublicSettings = '{"commandToExecute":"powershell Add-WindowsFeature Web-Server"}'

Set-AzVMExtension -ExtensionName "IIS" -ResourceGroupName $resourceGroup -VMName $vmName `

-Publisher "Microsoft.Compute" -ExtensionType "CustomScriptExtension" -TypeHandlerVersion 1.4 `

-SettingString $PublicSettings -Location $location

