Infrastructure as code

# Create the VM

Launch the Cloud Shell

$cred = Get-Credential

azureadmin

letmein1234!

New-AzVm `

-ResourceGroupName "ResourceGroupAutomate" `

-Name "VM1" `

-Location "UK South" `

-VirtualNetworkName "Vnet1" `

-SubnetName "Subnet1" `

-SecurityGroupName "NetworkSecurityGroup1" `

-PublicIpAddressName "PublicIpAddress1" `

-OpenPorts 80 `

-Credential $cred

# Add a custom script extension.

Use [Set-AzVMExtension](https://learn.microsoft.com/en-us/powershell/module/az.compute/set-azvmextension) to install the Custom Script Extension. The extension runs powershell Add-WindowsFeature Web-Server to install the IIS webserver and then updates the Default.htm page to show the hostname of the VM:

Set-AzVMExtension -ResourceGroupName "ResourceGroupAutomate" `

-ExtensionName "IIS" `

-VMName "VM1" `

-Location "UK South" `

-Publisher Microsoft.Compute `

-ExtensionType CustomScriptExtension `

-TypeHandlerVersion 1.8 `

-SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -Value $($env:computername)"}'

A picture containing chart

Description automatically generated

## Test the Website

Get-AzPublicIPAddress `

-ResourceGroupName "ResourceGroupAutomate" `

-Name "PublicIPAddress1" | select IpAddress

Enter the IP address in a browser and it will display the name of the VM.

# Clean up.

Remove-AzResourceGroup -Name ' ResourceGroupAutomate '

# Custom Images

Get-AzVMImagePublisher -Location "UK South"

Get-AzVMImageOffer `

-Location "UK South" `

-PublisherName "MicrosoftWindowsServer"

Get-AzVMImageSku `

-Location "UK South" `

-PublisherName "MicrosoftWindowsServer" `

-Offer "WindowsServer"

New-AzVm `

-ResourceGroupName "myResourceGroupVM" `

-Name "myVM2" `

-Location "UK South" `

-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](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-job) cmdlet.

# Create another VM specifying size and disk.

$azRegion = "UK South"

$azResourceGroup = "myDemoResourceGroup"

$azVMName = "myDemoVM"

$azDataDiskName = "myDemoDataDisk"

New-AzVm `

-Location $azRegion `

-ResourceGroupName $azResourceGroup `

-Name $azVMName `

-Size "Standard\_D2s\_v3" `

-VirtualNetworkName "myDemoVnet" `

-SubnetName "myDemoSubnet" `

-SecurityGroupName "myDemoNetworkSecurityGroup" `

-PublicIpAddressName "myDemoPublicIpAddress"

## Add a disk.

$diskConfig = New-AzDiskConfig `

-Location $azRegion `

-CreateOption Empty `

-DiskSizeGB 128 `

-SkuName "Standard\_LRS"

## Provision the disk.

$dataDisk = New-AzDisk `

-ResourceGroupName $azResourceGroup `

-DiskName $azDataDiskName `

-Disk $diskConfig

## Check the disk.

Get-AzDisk `

-ResourceGroupName $azResourceGroup `

-DiskName $azDataDiskName

## Attach the disk.

### Get the VM you wish to attach the disk to.

$vm = Get-AzVM `

-ResourceGroupName $azResourceGroup `

-Name $azVMName

### 

### Next, attach the data disk to the VM's configuration with the [Add-AzVMDataDisk](https://learn.microsoft.com/en-us/powershell/module/az.compute/add-azvmdatadisk) cmdlet.

$vm = Add-AzVMDataDisk `

-VM $vm `

-Name $azDataDiskName `

-CreateOption Attach `

-ManagedDiskId $dataDisk.Id `

-Lun 1

### Finally, update the VM's configuration with the [Update-AzVM](https://learn.microsoft.com/en-us/powershell/module/az.compute/add-azvmdatadisk) cmdlet.

Update-AzVM `

-ResourceGroupName $azResourceGroup `

-VM $vm