Microsoft Cloud Workshop

Create a Virtual Network Using PowerShell

Hands-on lab step-by-step

October 2018

Contents

Create a Virtual Network Using PowerShell hands-on lab step-by-step

Abstract and learning objectives

The student will create an Azure virtual network using PowerShell, create 2 VMs, and verify network connectivity between VMs works over a private endpoint.

Networking References

1) Azure Virtual Network Overview
2) Azure Virtual Network FAQ
3) IP Addresses
    a. Public IP Addresses
    b. Internal IP Addresses
4) DNS
    a. Azure DNS
    b. Name Resolution for Azure VNets
5) Connectivity for Azure Virtual Networks
    a. Site-to-Site VPN
    b. VNet-to-VNet VPN
    c. Point-to-Site VPN
    d. Regional VNet Peering
    e. Global VNet Peering
    f. ExpressRoute Overview
6) Load Balancers
    a. Azure Load Balancer
    b. Azure Traffic Manager
    c. Azure Application Gateway
7) Network Security Strategies
    a. DMZ Between Azure and On-Premises
    b. DMZ Between Azure and the Internet
    c. Network Security Groups
    d. User Defined Routes
    e. Virtual Network Service Tunneling
    f. Web Application Firewall
    g. Service Endpoints
    h. Network Virtual Appliances
8) Monitoring
    a. Network Watcher
    b. Network Performance Monitor Overview & Solution
    c. ExpressRoute Monitor
    d. DNS Analytics
    e. Service Endpoint Monitoring

Exercise 1: 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. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

Click **Try It** in the upper right corner of a code block.

Open Cloud Shell in your browser.

Click the **Cloud Shell** button on the menu in the upper right of the Azure portal.

If you choose to install and use PowerShell locally, this tutorial requires the AzureRM PowerShell module version 5.4.1 or later. To find the installed version, run `Get-Module -ListAvailable AzureRM`. If you need to upgrade, see Install Azure PowerShell module. If you are running PowerShell locally, you also need to run `Connect-AzureRmAccount` to create a connection with Azure.

Exercise 2: Create a Virtual Network

Before you can create a virtual network, you must create a resource group to contain the virtual network. Create a resource group with New-AzureRmResourceGroup. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

```
New-AzureRmResourceGroup -Name myResourceGroup -Location EastUS
```

Create a virtual network with New-AzureRmVirtualNetwork. The following example creates a default virtual network named *myVirtualNetwork* in the *EastUS* location:

```
$virtualNetwork = New-AzureRmVirtualNetwork `
  -ResourceGroupName myResourceGroup `
  -Location EastUS `
  -Name myVirtualNetwork `
  -AddressPrefix 10.0.0.0/16
```

Azure resources are deployed to a subnet within a virtual network, so you need to create a subnet. Create a subnet configuration with New-AzureRmVirtualNetworkSubnetConfig.

```
$subnetConfig = Add-AzureRmVirtualNetworkSubnetConfig `
  -Name default `
  -AddressPrefix 10.0.0.0/24 `
  -VirtualNetwork $virtualNetwork
```


Write the subnet configuration to the virtual network with Set-AzureRmVirtualNetwork, which creates the subnet within the virtual network:

```
$virtualNetwork | Set-AzureRmVirtualNetwork
```

Exercise 3: Create Two Virtual Machines in the Virtual Network

Create the first VM

Create a VM with New-AzureRmVM. When running the command that follows, you are prompted for credentials. The values that you enter are configured as the user name and password for the VM. The `-AsJob` option creates the VM in the background, so that you can continue to the next step.

```
New-AzureRmVm `
    -ResourceGroupName "myResourceGroup" `
    -Location "East US" `
    -VirtualNetworkName "myVirtualNetwork" `
    -SubnetName "default" `
    -Name "myVm1" `
    -AsJob
```

Output similar to the following example output is returned, and Azure starts creating the VM in the background.

```
Id      Name            PSJobTypeName   State         HasMoreData     Location
Command
--      ----            -------------   -----         -----------     --------
-------
1       Long Running... AzureLongRun... Running       True            localhost
```

Create the second VM

Enter the following command:

```
New-AzureRmVm `
  -ResourceGroupName "myResourceGroup" `
  -VirtualNetworkName "myVirtualNetwork" `
  -SubnetName "default" `
  -Name "myVm2" `
  -AsJob
```

The VM takes a few minutes to create. Do not continue with the next step until the previous command executes and output is returned to PowerShell.

Exercise 4: Connect to a VM from the Internet

Use Get-AzureRmPublicIpAddress to return the public IP address of a VM. The following example returns the public IP address of the *myVm1* VM:

```
Get-AzureRmPublicIpAddress `
  -Name myVm1 `
  -ResourceGroupName myResourceGroup `
  | Select IpAddress
```

Replace `<publicIpAddress>` in the following command, with the public IP address returned from the previous command, and then enter the following command:
Copy

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

A Remote Desktop Protocol (.rdp) file is created and downloaded to your computer. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM. Select **OK**. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.

Exercise 5: Communicate Between VMs

From PowerShell on the *myVm1* VM, enter `ping myvm2`. Ping fails, because ping uses the Internet Control Message Protocol (ICMP), and ICMP is not allowed through the Windows firewall, by default.

To allow *myVm2* to ping *myVm1* in a later step, enter the following command from PowerShell, which allows ICMP inbound through the Windows firewall:

```
New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
```

Close the remote desktop connection to *myVm1*.

Complete the steps in Connect to a VM from the internet again, but connect to *myVm2*.

From a command prompt on the *myVm2* VM, enter `ping myvm1`.

You receive replies from *myVm1*, because you allowed ICMP through the Windows firewall on the *myVm1* VM in a previous step.

Close the remote desktop connection to *myVm2*.

Exercise 6: Clean Up Resources

When no longer needed, you can use Remove-AzureRmResourceGroup to remove the resource group and all of the resources it contains:

```
Remove-AzureRmResourceGroup -Name myResourceGroup -Force
```