Manage Azure Virtual Networks and Windows Virtual Machines with Azure PowerShell

10/12/2017 7 minutes to read Contributors

* [davidmu1](https://github.com/davidmu1)

* [Cynthia Nottingham [MSFT]](https://github.com/cynthn)

* [Iain Foulds](https://github.com/iainfoulds)

* [Kris Crider](https://github.com/kriscrider)

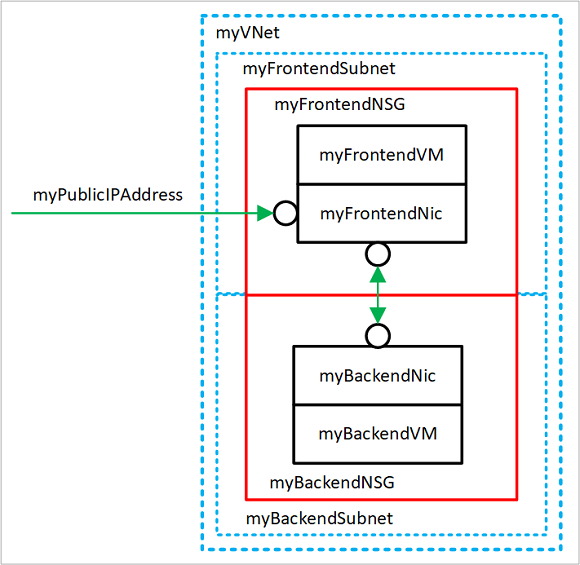
* [](https://github.com/czeumault)

* [all](https://github.com/Microsoft/azure-docs/blob/master/articles/virtual-machines/windows/tutorial-virtual-network.md)

Azure virtual machines use Azure networking for internal and external network communication. This tutorial walks through deploying two virtual machines and configuring Azure networking for these VMs. The examples in this tutorial assume that the VMs are hosting a web application with a database back-end, however an application is not deployed in the tutorial. In this tutorial, you learn how to:

* Create a virtual network and subnet
* Create a public IP address
* Create a front-end VM
* Secure network traffic
* Create back-end VM

While completing this tutorial, you can see these resources created:



* *myVNet* - The virtual network that the VMs use to communicate with each other and the internet.
* *myFrontendSubnet* - The subnet in *myVNet* used by the front-end resources.
* *myPublicIPAddress* - The public IP address used to access *myFrontendVM* from the internet.
* *myFrontentNic* - The network interface used by *myFrontendVM* to communicate with *myBackendVM*.
* *myFrontendVM* - The VM used to communicate between the internet and *myBackendVM*.
* *myBackendNSG* - The network security group that controls communication between the *myFrontendVM* and *myBackendVM*.
* *myBackendSubnet* - The subnet associated with *myBackendNSG* and used by the back-end resources.
* *myBackendNic* - The network interface used by *myBackendVM* to communicate with *myFrontendVM*.
* *myBackendVM* - The VM that uses port 1433 to communicate with *myFrontendVM*.

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

VM networking overview

Azure virtual networks enable secure network connections between virtual machines, the internet, and other Azure services such as Azure SQL database. Virtual networks are broken down into logical segments called subnets. Subnets are used to control network flow, and as a security boundary. When deploying a VM, it generally includes a virtual network interface, which is attached to a subnet.

Create a virtual network and subnet

For this tutorial, a single virtual network is created with two subnets. A front-end subnet for hosting a web application, and a back-end subnet for hosting a database server.

Before you can create a virtual network, create a resource group using [New-AzureRmResourceGroup](https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/new-azurermresourcegroup). The following example creates a resource group named *myRGNetwork*in the *EastUS* location:

Azure PowerShellCopyTry It

New-AzureRmResourceGroup -ResourceGroupName myRGNetwork -Location EastUS

Create subnet configurations

Create a subnet configuration named *myFrontendSubnet* using [New-AzureRmVirtualNetworkSubnetConfig](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermvirtualnetworksubnetconfig):

Azure PowerShellCopyTry It

$frontendSubnet = New-AzureRmVirtualNetworkSubnetConfig `

-Name myFrontendSubnet `

-AddressPrefix 10.0.0.0/24

And, create a subnet configuration named *myBackendSubnet*:

Azure PowerShellCopyTry It

$backendSubnet = New-AzureRmVirtualNetworkSubnetConfig `

-Name myBackendSubnet `

-AddressPrefix 10.0.1.0/24

Create virtual network

Create a VNET named *myVNet* using *myFrontendSubnet* and *myBackendSubnet* using [New-AzureRmVirtualNetwork](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermvirtualnetwork):

Azure PowerShellCopyTry It

$vnet = New-AzureRmVirtualNetwork `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-Name myVNet `

-AddressPrefix 10.0.0.0/16 `

-Subnet $frontendSubnet, $backendSubnet

At this point, a network has been created and segmented into two subnets, one for front-end services, and another for back-end services. In the next section, virtual machines are created and connected to these subnets.

Create a public IP address

A public IP address allows Azure resources to be accessible on the internet. The allocation method of the public IP address can be configured as dynamic or static. By default, a public IP address is dynamically allocated. Dynamic IP addresses are released when a VM is deallocated. This behavior causes the IP address to change during any operation that includes a VM deallocation.

The allocation method can be set to static, which ensures that the IP address remains assigned to a VM, even during a deallocated state. When using a statically allocated IP address, the IP address itself cannot be specified. Instead, it is allocated from a pool of available addresses.

Create a public IP address named *myPublicIPAddress* using [New-AzureRmPublicIpAddress](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermpublicipaddress):

Azure PowerShellCopyTry It

$pip = New-AzureRmPublicIpAddress `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-AllocationMethod Dynamic `

-Name myPublicIPAddress

You could change the -AllocationMethod parameter to Static to assign a static public IP address.

Create a front-end VM

For a VM to communicate in a virtual network, it needs a virtual network interface (NIC). Create a NIC using [New-AzureRmNetworkInterface](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermnetworkinterface):

Azure PowerShellCopyTry It

$frontendNic = New-AzureRmNetworkInterface `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-Name myFrontendNic `

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

-PublicIpAddressId $pip.Id

Set the username and password needed for the administrator account on the VM using [Get-Credential](https://msdn.microsoft.com/powershell/reference/5.1/microsoft.powershell.security/Get-Credential). You use these credentials to connect to the VM in additional steps:

Azure PowerShellCopyTry It

$cred = Get-Credential

Create the VMs using [New-AzureRmVMConfig](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/new-azurermvmconfig), [Set-AzureRmVMOperatingSystem](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/set-azurermvmoperatingsystem), [Set-AzureRmVMSourceImage](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/set-azurermvmsourceimage), [Set-AzureRmVMOSDisk](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/set-azurermvmosdisk), [Add-AzureRmVMNetworkInterface](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/add-azurermvmnetworkinterface), and [New-AzureRmVM](https://docs.microsoft.com/en-us/powershell/module/azurerm.compute/new-azurermvm):

Azure PowerShellCopyTry It

$frontendVM = New-AzureRmVMConfig `

-VMName myFrontendVM `

-VMSize Standard\_D1

$frontendVM = Set-AzureRmVMOperatingSystem `

-VM $frontendVM `

-Windows `

-ComputerName myFrontendVM `

-Credential $cred `

-ProvisionVMAgent `

-EnableAutoUpdate

$frontendVM = Set-AzureRmVMSourceImage `

-VM $frontendVM `

-PublisherName MicrosoftWindowsServer `

-Offer WindowsServer `

-Skus 2016-Datacenter `

-Version latest

$frontendVM = Set-AzureRmVMOSDisk `

-VM $frontendVM `

-Name myFrontendOSDisk `

-DiskSizeInGB 128 `

-CreateOption FromImage `

-Caching ReadWrite

$frontendVM = Add-AzureRmVMNetworkInterface `

-VM $frontendVM `

-Id $frontendNic.Id

New-AzureRmVM `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-VM $frontendVM

Secure network traffic

A network security group (NSG) contains a list of security rules that allow or deny network traffic to resources connected to Azure Virtual Networks (VNet). NSGs can be associated to subnets or individual network interfaces. When an NSG is associated with a network interface, it applies only the associated VM. When an NSG is associated to a subnet, the rules apply to all resources connected to the subnet.

Network security group rules

NSG rules define networking ports over which traffic is allowed or denied. The rules can include source and destination IP address ranges so that traffic is controlled between specific systems or subnets. NSG rules also include a priority (between 1—and 4096). Rules are evaluated in the order of priority. A rule with a priority of 100 is evaluated before a rule with priority 200.

All NSGs contain a set of default rules. The default rules cannot be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

* **Virtual network** - Traffic originating and ending in a virtual network is allowed both in inbound and outbound directions.
* **Internet** - Outbound traffic is allowed, but inbound traffic is blocked.
* **Load balancer** - Allow Azure’s load balancer to probe the health of your VMs and role instances. If you are not using a load balanced set, you can override this rule.

Create network security groups

Create an inbound rule named *myFrontendNSGRule* to allow incoming web traffic on *myFrontendVM* using [New-AzureRmNetworkSecurityRuleConfig](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermnetworksecurityruleconfig):

Azure PowerShellCopyTry It

$nsgFrontendRule = New-AzureRmNetworkSecurityRuleConfig `

-Name myFrontendNSGRule `

-Protocol Tcp `

-Direction Inbound `

-Priority 200 `

-SourceAddressPrefix \* `

-SourcePortRange \* `

-DestinationAddressPrefix \* `

-DestinationPortRange 80 `

-Access Allow

You can limit internal traffic to *myBackendVM* from only *myFrontendVM* by creating an NSG for the back-end subnet. The following example creates an NSG rule named *myBackendNSGRule*:

Azure PowerShellCopyTry It

$nsgBackendRule = New-AzureRmNetworkSecurityRuleConfig `

-Name myBackendNSGRule `

-Protocol Tcp `

-Direction Inbound `

-Priority 100 `

-SourceAddressPrefix 10.0.0.0/24 `

-SourcePortRange \* `

-DestinationAddressPrefix \* `

-DestinationPortRange 1433 `

-Access Allow

Add a network security group named *myFrontendNSG* using [New-AzureRmNetworkSecurityGroup](https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermnetworksecuritygroup):

Azure PowerShellCopyTry It

$nsgFrontend = New-AzureRmNetworkSecurityGroup `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-Name myFrontendNSG `

-SecurityRules $nsgFrontendRule

Now, add a network security group named *myBackendNSG* using New-AzureRmNetworkSecurityGroup:

Azure PowerShellCopyTry It

$nsgBackend = New-AzureRmNetworkSecurityGroup `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-Name myBackendNSG `

-SecurityRules $nsgBackendRule

Add the network security groups to the subnets:

Azure PowerShellCopyTry It

$vnet = Get-AzureRmVirtualNetwork `

-ResourceGroupName myRGNetwork `

-Name myVNet

$frontendSubnet = $vnet.Subnets[0]

$backendSubnet = $vnet.Subnets[1]

$frontendSubnetConfig = Set-AzureRmVirtualNetworkSubnetConfig `

-VirtualNetwork $vnet `

-Name myFrontendSubnet `

-AddressPrefix $frontendSubnet.AddressPrefix `

-NetworkSecurityGroup $nsgFrontend

$backendSubnetConfig = Set-AzureRmVirtualNetworkSubnetConfig `

-VirtualNetwork $vnet `

-Name myBackendSubnet `

-AddressPrefix $backendSubnet.AddressPrefix `

-NetworkSecurityGroup $nsgBackend

Set-AzureRmVirtualNetwork -VirtualNetwork $vnet

Create a back-end VM

The easiest way to create the back-end VM for this tutorial is by using a SQL Server image. This tutorial only creates the VM with the database server, but doesn't provide information about accessing the database.

Create *myBackendNic*:

Azure PowerShellCopyTry It

$backendNic = New-AzureRmNetworkInterface `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-Name myBackendNic `

-SubnetId $vnet.Subnets[1].Id

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

Azure PowerShellCopyTry It

$cred = Get-Credential

Create *myBackendVM*:

Azure PowerShellCopyTry It

$backendVM = New-AzureRmVMConfig `

-VMName myBackendVM `

-VMSize Standard\_D1

$backendVM = Set-AzureRmVMOperatingSystem `

-VM $backendVM `

-Windows `

-ComputerName myBackendVM `

-Credential $cred `

-ProvisionVMAgent `

-EnableAutoUpdate

$backendVM = Set-AzureRmVMSourceImage `

-VM $backendVM `

-PublisherName MicrosoftSQLServer `

-Offer SQL2016SP1-WS2016 `

-Skus Enterprise `

-Version latest

$backendVM = Set-AzureRmVMOSDisk `

-VM $backendVM `

-Name myBackendOSDisk `

-DiskSizeInGB 128 `

-CreateOption FromImage `

-Caching ReadWrite

$backendVM = Add-AzureRmVMNetworkInterface `

-VM $backendVM `

-Id $backendNic.Id

New-AzureRmVM `

-ResourceGroupName myRGNetwork `

-Location EastUS `

-VM $backendVM

The image that is used has SQL Server installed, but is not used in this tutorial. It is included to show you how you can configure a VM to handle web traffic and a VM to handle database management.