

Contents

[Virtual Network documentation](#)

[Overview](#)

[About Virtual Network](#)

[Quickstarts](#)

[Create virtual network - Portal](#)

[Create virtual network - PowerShell](#)

[Create virtual network - Azure CLI](#)

[Create virtual network - ARM template](#)

[Tutorials](#)

[Filter network traffic](#)

[Route network traffic](#)

[Restrict network access to resources](#)

[Connect virtual networks](#)

[Create NAT gateway](#)

[Create and validate NAT gateway](#)

[Samples](#)

[Azure CLI](#)

[Azure PowerShell](#)

[Resource Manager templates](#)

[Concepts](#)

[Concepts and best practices](#)

[Business continuity](#)

[Connectivity](#)

[NAT](#)

[Overview](#)

[Resource](#)

[Metrics and alerts](#)

[Troubleshoot](#)

[Routing](#)

Backend Connectivity Interoperability

- Preface and Test Setup
- Test Setup Configuration
- Control Plane Analysis
- Data Plane Analysis
- Container networking
- Cross-network connectivity
- Peering
- Integrate Azure services
 - Overview
 - Deploy a dedicated service
 - Private Link
 - Service endpoints
 - Service tags
- IP services
 - Public IPs
 - Private IPs
 - Public IP prefix
 - IPv6 for Azure VNet
 - Public IPv6 address prefix
 - Routing preference
 - Public IP security baseline
- Security
 - Virtual network security baseline
 - Virtual network NAT security baseline
 - Network isolation of Azure services
 - Network security groups
 - Overview
 - How it works
 - Application security groups
 - Service tags
 - Service endpoint policies

- [Kubernetes network policies](#)
- [DDoS protection](#)
- [Virtual network TAP](#)
- [Security controls by Azure Policy](#)
- [Subnets](#)
 - [Subnet extension](#)
 - [Subnet delegation](#)
- [Classic deployment](#)
- [How-to guides](#)
 - [Plan and configure](#)
 - [Plan virtual networks](#)
 - [Name resolution for VMs and cloud services](#)
 - [Use dynamic DNS with your own DNS server](#)
 - [Optimize network throughput](#)
 - [View and modify hostnames](#)
 - [Logs](#)
- [Connectivity](#)
 - [Route network traffic](#)
 - [Azure portal](#)
 - [Azure PowerShell](#)
 - [Azure CLI](#)
 - [Ansible](#)
 - [Manage route tables](#)
 - [Manage NICs](#)
 - [Manage virtual network](#)
 - [Manage subnets](#)
 - [Manage subnet delegation](#)
 - [Create VNet peering](#)
 - [Same deployment model - same subscription](#)
 - [Azure PowerShell](#)
 - [Azure CLI](#)
 - [Same deployment model - different subscriptions](#)

- Different deployment models - same subscription
- Different deployment models - different subscriptions
- Manage VNet peering
- Connectivity scenarios
 - Virtual network to Virtual network
 - VNet (Resource Manager) to a VNet (Classic)
 - VNet to on-premises network (VPN)
 - VNet to on-premises network (ExpressRoute)
 - Highly available hybrid network architecture
- Configure Virtual network TAP
- Deploy Container networking
- IP services
 - IPv6 for VNet
 - Add IPv6 to IPv4 application
 - Azure PowerShell
 - Azure CLI
 - Basic Public Load Balancer
 - Azure PowerShell
 - Azure CLI
 - Resource Manager template
 - Standard Public Load Balancer
 - Azure PowerShell
 - Azure CLI
 - Resource Manager template
 - Standard Internal Load Balancer
 - Virtual machine scale set with IPv6
 - Public IPv6 address prefix
 - IP addressing
 - Create public IP addresses - portal
 - Create public IP addresses - PowerShell
 - Create public IP addresses - Azure CLI
 - Manage public IP addresses

- Manage public IP prefix
 - Upgrade public IP addresses
 - Add or remove IP address from NICs
 - Routing preference
 - Create public IP address
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Configure VM with internet routing preference
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Security
 - Filter network traffic
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Network isolation
 - Using private endpoints
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Using service endpoints
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Using service endpoint policies
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Manage DDoS protection
 - Onboard partners to DDoS Protection Standard

Manage network security groups

Security scenarios

Secure networks with virtual appliances

DMZ between Azure and the Internet

VM networking

Create VM - static public IP

Azure portal

Azure PowerShell

Azure CLI

Add public IP address to existing VM

Dissociate public IP address from a VM

Create VM - static private IP

Azure portal

Azure PowerShell

Azure CLI

Create VM - static private IP (classic)

Azure portal

Azure PowerShell

Create VM - multiple IPs

Azure portal

Azure PowerShell

Azure CLI

Add or remove network interfaces

Create VM - multiple NICs

Azure PowerShell

Azure CLI

Create VM - accelerated networking

Azure PowerShell

Azure CLI

Setup DPDK

TCP/IP Performance Tuning for Azure VMs

Virtual machine network throughput

[Move across regions](#)

[Network security groups \(NSG\)](#)

[Virtual networks](#)

[Network interfaces \(NICs\)](#)

[Public IPs](#)

[Manual move](#)

[Network security groups - Portal](#)

[Network security groups - PowerShell](#)

[Virtual networks - Portal](#)

[Virtual networks - PowerShell](#)

[Public IPs - Portal](#)

[Public IPs - PowerShell](#)

[NAT](#)

[Create NAT gateway - PowerShell](#)

[Create NAT gateway - CLI](#)

[Create NAT gateway - ARM template](#)

[Create and validate NAT gateway - PowerShell](#)

[Create and validate NAT gateway - CLI](#)

[Troubleshoot](#)

[Troubleshoot virtual network peering issues](#)

[Configure and validate VNet or VPN connections](#)

[Network security groups](#)

[Routes](#)

[Throughput testing](#)

[Test VM network latency](#)

[Cannot delete virtual networks](#)

[VM to VM connectivity problems](#)

[Configure PTR for SMTP Banner Check](#)

[Troubleshooting checklist for virtual appliances](#)

[Troubleshooting outbound SMTP connectivity](#)

[About IP address 168.63.129.16](#)

[Troubleshoot Azure VM connectivity problems](#)

Classic deployment

Reference

[Azure CLI](#)

[Azure PowerShell](#)

[.NET](#)

[Java](#)

[Node.js](#)

[Python](#)

[REST](#)

[Resource Manager template](#)

[Code samples](#)

[Azure Policy built-ins](#)

Resources

[Build your skills with Microsoft Learn](#)

[Azure roadmap](#)

[Networking blog](#)

[Microsoft Q&A question page for Networking](#)

[Networking feedback](#)

[Pricing](#)

[Pricing calculator](#)

[Stack Overflow](#)

[FAQ](#)

What is Azure Virtual Network?

2/18/2021 • 5 minutes to read • [Edit Online](#)

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

Why use an Azure Virtual network?

Azure virtual network enables Azure resources to securely communicate with each other, the internet, and on-premises networks. Key scenarios that you can accomplish with a virtual network include - communication of Azure resources with the internet, communication between Azure resources, communication with on-premises resources, filtering network traffic, routing network traffic, and integration with Azure services.

Communicate with the internet

All resources in a VNet can communicate outbound to the internet, by default. You can communicate inbound to a resource by assigning a public IP address or a public Load Balancer. You can also use public IP or public Load Balancer to manage your outbound connections. To learn more about outbound connections in Azure, see [Outbound connections](#), [Public IP addresses](#), and [Load Balancer](#).

NOTE

When using only an internal [Standard Load Balancer](#), outbound connectivity is not available until you define how you want [outbound connections](#) to work with an instance-level public IP or a public Load Balancer.

Communicate between Azure resources

Azure resources communicate securely with each other in one of the following ways:

- **Through a virtual network:** You can deploy VMs, and several other types of Azure resources to a virtual network, such as Azure App Service Environments, the Azure Kubernetes Service (AKS), and Azure Virtual Machine Scale Sets. To view a complete list of Azure resources that you can deploy into a virtual network, see [Virtual network service integration](#).
- **Through a virtual network service endpoint:** Extend your virtual network private address space and the identity of your virtual network to Azure service resources, such as Azure Storage accounts and Azure SQL Database, over a direct connection. Service endpoints allow you to secure your critical Azure service resources to only a virtual network. To learn more, see [Virtual network service endpoints overview](#).
- **Through VNet Peering:** You can connect virtual networks to each other, enabling resources in either virtual network to communicate with each other, using virtual network peering. The virtual networks you connect can be in the same, or different, Azure regions. To learn more, see [Virtual network peering](#).

Communicate with on-premises resources

You can connect your on-premises computers and networks to a virtual network using any combination of the following options:

- **Point-to-site virtual private network (VPN):** Established between a virtual network and a single computer in your network. Each computer that wants to establish connectivity with a virtual network must configure its connection. This connection type is great if you're just getting started with Azure, or for developers, because it requires little or no changes to your existing network. The communication between your computer and a virtual network is sent through an encrypted tunnel over the internet. To learn more,

see [Point-to-site VPN](#).

- **Site-to-site VPN:** Established between your on-premises VPN device and an Azure VPN Gateway that is deployed in a virtual network. This connection type enables any on-premises resource that you authorize to access a virtual network. The communication between your on-premises VPN device and an Azure VPN gateway is sent through an encrypted tunnel over the internet. To learn more, see [Site-to-site VPN](#).
- **Azure ExpressRoute:** Established between your network and Azure, through an ExpressRoute partner. This connection is private. Traffic does not go over the internet. To learn more, see [ExpressRoute](#).

Filter network traffic

You can filter network traffic between subnets using either or both of the following options:

- **Network security groups:** Network security groups and application security groups can contain multiple inbound and outbound security rules that enable you to filter traffic to and from resources by source and destination IP address, port, and protocol. To learn more, see [Network security groups](#) or [Application security groups](#).
- **Network virtual appliances:** A network virtual appliance is a VM that performs a network function, such as a firewall, WAN optimization, or other network function. To view a list of available network virtual appliances that you can deploy in a virtual network, see [Azure Marketplace](#).

Route network traffic

Azure routes traffic between subnets, connected virtual networks, on-premises networks, and the Internet, by default. You can implement either or both of the following options to override the default routes Azure creates:

- **Route tables:** You can create custom route tables with routes that control where traffic is routed to for each subnet. Learn more about [route tables](#).
- **Border gateway protocol (BGP) routes:** If you connect your virtual network to your on-premises network using an Azure VPN Gateway or ExpressRoute connection, you can propagate your on-premises BGP routes to your virtual networks. Learn more about using BGP with [Azure VPN Gateway](#) and [ExpressRoute](#).

Virtual network integration for Azure services

Integrating Azure services to an Azure virtual network enables private access to the service from virtual machines or compute resources in the virtual network. You can integrate Azure services in your virtual network with the following options:

- Deploying [dedicated instances of the service](#) into a virtual network. The services can then be privately accessed within the virtual network and from on-premises networks.
- Using [Private Link](#) to access privately a specific instance of the service from your virtual network and from on-premises networks.
- You can also access the service using public endpoints by extending a virtual network to the service, through [service endpoints](#). Service endpoints allow service resources to be secured to the virtual network.

Azure VNet limits

There are certain limits around the number of Azure resources you can deploy. Most Azure networking limits are at the maximum values. However, you can [increase certain networking limits](#) as specified on the [VNet limits page](#).

Pricing

There is no charge for using Azure VNet, it is free of cost. Standard charges are applicable for resources, such as Virtual Machines (VMs) and other products. To learn more, see [VNet pricing](#) and the Azure [pricing calculator](#).

Next steps

- Learn about [Azure Virtual Network concepts and best practices](#) .
- To get started using a virtual network, create one, deploy a few VMs to it, and communicate between the VMs. To learn how, see the [Create a virtual network](#) quickstart.

Quickstart: Create a virtual network using the Azure portal

2/18/2021 • 5 minutes to read • [Edit Online](#)

In this quickstart, you learn how to create a virtual network using the Azure portal. You deploy two virtual machines (VMs). Next, you securely communicate between VMs and connect to VMs from the internet. A virtual network is the fundamental building block for your private network in Azure. It enables Azure resources, like VMs, to securely communicate with each other and with the internet.

Prerequisites

- An Azure account with an active subscription. [Create one for free](#).

Sign in to Azure

Sign in to the [Azure portal](#).

Create a virtual network

1. From the Azure portal menu, select **Create a resource**. From the Azure Marketplace, select **Networking > Virtual network**.
2. In **Create virtual network**, enter or select this information:

SETTING	VALUE
Subscription	Select your subscription.
Resource group	Select Create new , enter <i>myResourceGroup</i> , then select OK .
Name	Enter <i>myVirtualNetwork</i> .
Location	Select East US .

3. Select **Next: IP Addresses**, and for **IPv4 address space**, enter *10.1.0.0/16*.
4. Select **Add subnet**, then enter *myVirtualSubnet* for **Subnet name** and *10.1.0.0/24* for **Subnet address range**.
5. Select **Add**, then select **Review + create**. Leave the rest as default and select **Create**.
6. In **Create virtual network**, select **Create**.

Create virtual machines

Create two VMs in the virtual network:

Create the first VM

1. From the Azure portal menu, select **Create a resource**.

2. From the Azure Marketplace, select **Compute > Windows Server 2019 Datacenter**. Select **Create**.

3. In **Create a virtual machine - Basics**, enter or select this information:

SETTING	VALUE
Project details	
Subscription	Select your subscription.
Resource group	Select myResourceGroup . You created this resource group in the previous section.
Instance details	
Virtual machine name	Enter <i>myVm1</i> .
Region	Select East US .
Availability options	Default to No infrastructure redundancy required .
Image	Default to Windows Server 2019 Datacenter .
Size	Default to Standard DS1 v2 .
Administrator account	
Username	Enter a username of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements .
Confirm Password	Re-enter password.
Inbound port rules	
Public inbound ports	Select Allow selected ports .
Select inbound ports	Enter <i>HTTP (80)</i> and <i>RDP (3389)</i> .
Save money	
Already have a Windows license?	Default to No .

4. Select **Next: Disks**.

5. In **Create a virtual machine - Disks**, keep the defaults and select **Next: Networking**.

6. In **Create a virtual machine - Networking**, select this information:

SETTING	VALUE
Virtual network	Default to myVirtualNetwork .

SETTING	VALUE
Subnet	Default to myVirtualSubnet (10.1.0.0/24) .
Public IP	Default to (new) myVm-ip .
NIC network security group	Default to Basic .
Public inbound ports	Default to Allow selected ports .
Select inbound ports	Default to HTTP and RDP .

7. Select **Next: Management**.

8. In **Create a virtual machine - Management**, for **Diagnostics storage account**, select **Create New**.

9. In **Create storage account**, enter or select this information:

SETTING	VALUE
Name	Enter <i>myvmstorageaccount</i> . If this name is taken, create a unique name.
Account kind	Default to Storage (general purpose v1) .
Performance	Default to Standard .
Replication	Default to Locally-redundant storage (LRS) .

10. Select **OK**, then select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.

11. When you see the **Validation passed** message, select **Create**.

Create the second VM

Repeat the procedure in the previous section to create another virtual machine.

IMPORTANT

For the **Virtual machine name**, enter *myVm2*.

For **Diagnosis storage account**, make sure you select **myvmstorageaccount**, instead of creating one.

Connect to a VM from the internet

After you've created *myVm1*, connect to the internet.

- In the Azure portal, search for and select *myVm1*.
- Select **Connect**, then **RDP**.

The screenshot shows the Azure portal interface for a virtual machine named 'myVm1'. At the top, there's a search bar labeled 'Search (Ctrl+ /)' and several action buttons: 'Connect' (highlighted with a red box), 'Start', and 'Restart'. Below these are sections for 'Resource group (change)', 'myResourceGroup', 'Status' (Running), and 'Location' (East US). On the left, a sidebar lists navigation options: 'Overview' (selected and highlighted in blue), 'Activity log', 'Access control (IAM)', and 'Tags'.

The **Connect** page opens.

3. Select **Download RDP File**. Azure creates a Remote Desktop Protocol (.rdp) file and downloads it to your computer.
4. Open the RDP file. If prompted, select **Connect**.
5. Enter the username and password you specified when creating the VM.

NOTE

You may need to select **More choices > Use a different account**, to specify the credentials you entered when you created the VM.

6. Select **OK**.
7. You may receive a certificate warning when you sign in. If you receive a certificate warning, select **Yes** or **Continue**.
8. Once the VM desktop appears, minimize it to go back to your local desktop.

Communicate between VMs

1. In the Remote Desktop of *myVm1*, open PowerShell.
2. Enter `ping myVm2`.

You'll receive a message similar to this output:

```
Pinging myVm2.0v0zze1s0uiedpvtxz5z0r0cxg.bx.internal.clouda
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.0.5:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The `ping` fails, because `ping` uses the Internet Control Message Protocol (ICMP). By default, ICMP isn't allowed through the Windows firewall.

3. To allow *myVm2* to ping *myVm1* in a later step, enter this command:

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

This command allows ICMP inbound through the Windows firewall:

4. Close the remote desktop connection to *myVm1*.
5. Complete the steps in [Connect to a VM from the internet](#) again, but connect to *myVm2*.
6. From a command prompt, enter `ping myvm1`.

You'll get back something like this message:

```
Pinging myVm1.0v0zze1s0uiedpvtxz5z0r0cxg.bx.internal.cloudapp.net [10.1.0.4] with 32 bytes of data:  
Reply from 10.1.0.4: bytes=32 time=1ms TTL=128  
Reply from 10.1.0.4: bytes=32 time<1ms TTL=128  
Reply from 10.1.0.4: bytes=32 time<1ms TTL=128  
Reply from 10.1.0.4: bytes=32 time<1ms TTL=128  
  
Ping statistics for 10.1.0.4:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

You receive replies from *myVm1*, because you allowed ICMP through the Windows firewall on the *myVm1* VM in step 3.

7. Close the remote desktop connection to *myVm2*.

Clean up resources

In this quickstart, you created a default virtual network and two VMs. You connected to one VM from the internet and securely communicated between the two VMs.

When you're done using the virtual network and the VMs, delete the resource group and all of the resources it contains:

1. Search for and select *myResourceGroup*.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME** and select **Delete**.

Next steps

To learn more about virtual network settings, see [Create, change, or delete a virtual network](#).

By default, Azure allows secure communication between VMs. Azure only allows inbound remote desktop connections to Windows VMs from the internet. To learn more about types of VM network communications, see [Filter network traffic](#).

NOTE

Azure services cost money. Azure Cost Management helps you set budgets and configure alerts to keep spending under control. Analyze, manage, and optimize your Azure costs with Cost Management. To learn more, see the [quickstart on analyzing your costs](#).

Quickstart: Create a virtual network using PowerShell

11/2/2020 • 5 minutes to read • [Edit Online](#)

A virtual network lets Azure resources, like virtual machines (VMs), communicate privately with each other, and with the internet. In this quickstart, you learn how to create a virtual network. After creating a virtual network, you deploy two VMs into the virtual network. You then connect to the VMs from the internet, and communicate privately over the virtual network.

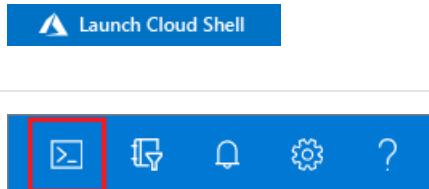
Prerequisites

If you don't have an Azure subscription, create a [free account](#) now.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you decide to install and use PowerShell locally instead, this quickstart requires you to use Azure PowerShell module version 1.0.0 or later. To find the installed version, run `Get-Module -ListAvailable Az`. See [Install Azure PowerShell module](#) for install and upgrade info.

Finally, if you're running PowerShell locally, you'll also need to run `Connect-AzAccount`. That command creates a connection with Azure.

Create a resource group and a virtual network

There are a handful of steps you have to walk through to get your resource group and virtual network configured.

Create the resource group

Before you can create a virtual network, you have to create a resource group to host the virtual network. Create a resource group with [New-AzResourceGroup](#). This example creates a resource group named *myResourceGroup* in the *eastus* location:

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS
```

Create the virtual network

Create a virtual network with [New-AzVirtualNetwork](#). This example creates a default virtual network named *myVirtualNetwork* in the *EastUS* location:

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

Add a subnet

Azure deploys resources to a subnet within a virtual network, so you need to create a subnet. Create a subnet configuration named *default* with [Add-AzVirtualNetworkSubnetConfig](#):

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

Associate the subnet to the virtual network

You can write the subnet configuration to the virtual network with [Set-AzVirtualNetwork](#). This command creates the subnet:

```
$virtualNetwork | Set-AzVirtualNetwork
```

Create virtual machines

Create two VMs in the virtual network.

Create the first VM

Create the first VM with [New-AzVM](#). When you run the next command, you're prompted for credentials. Enter a user name and password for the VM:

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

The `-AsJob` option creates the VM in the background. You can continue to the next step.

When Azure starts creating the VM in the background, you'll get something like this back:

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

Create the second VM

Create the second VM with this command:

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

You'll have to create another user and password. Azure takes a few minutes to create the VM.

IMPORTANT

Don't continue with the next step until Azure's finished. You'll know it's done when it returns output to PowerShell.

Connect to a VM from the internet

Use `Get-AzPublicIpAddress` to return the public IP address of a VM. This example returns the public IP address of the *myVm1* VM:

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

Open a command prompt on your local computer. Run the `mstsc` command. Replace `<publicIpAddress>` with the public IP address returned from the last step:

NOTE

If you've been running these commands from a PowerShell prompt on your local computer, and you're using the Az PowerShell module version 1.0 or later, you can continue in that interface.

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

1. If prompted, select **Connect**.
2. Enter the user name and password you specified when creating the VM.

NOTE

You may need to select **More choices > Use a different account**, to specify the credentials you entered when you created the VM.

3. Select OK.
4. You may receive a certificate warning. If you do, select Yes or Continue.

Communicate between VMs

1. In the Remote Desktop of *myVm1*, open PowerShell.
2. Enter `ping myVm2`.

You'll get something like this back:

```
PS C:\Users\myVm1> ping myVm2

Pinging myVm2.ovvzzdcazhb5iczfvonhg2zrb.bx.internal.cloudapp.net
Request timed out.

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The ping fails, because it uses the Internet Control Message Protocol (ICMP). By default, ICMP isn't allowed through your Windows firewall.

3. To allow *myVm2* to ping *myVm1* in a later step, enter this command:

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

That command lets ICMP inbound through the Windows firewall.

4. Close the remote desktop connection to *myVm1*.
5. Repeat the steps in [Connect to a VM from the internet](#). This time, connect to *myVm2*.
6. From a command prompt on the *myVm2* VM, enter `ping myVm1`.

You'll get something like this back:

```
C:\windows\system32>ping myVm1

Pinging myVm1.e5p2dibbrqtejhq04lqrusvd4g.bx.internal.cloudapp.net [10.0.0.4] with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=2ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

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

7. Close the remote desktop connection to *myVm2*.

Clean up resources

When you're done with the virtual network and the VMs, use [Remove-AzResourceGroup](#) to remove the resource group and all the resources it has:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

Next steps

In this quickstart, you created a default virtual network and two VMs. You connected to one VM from the internet and communicated privately between the two VMs. Azure lets unrestricted private communication between VMs. By default, Azure only lets inbound remote desktop connections to Windows VMs from the internet. Advance to the next article to learn more about configuring different types of VM network communications:

[Filter network traffic](#)

Quickstart: Create a virtual network using the Azure CLI

11/2/2020 • 3 minutes to read • [Edit Online](#)

A virtual network enables Azure resources, like virtual machines (VMs), to communicate privately with each other, and with the internet. In this quickstart, you learn how to create a virtual network. After creating a virtual network, you deploy two VMs into the virtual network. You then connect to the VMs from the internet, and communicate privately over the new virtual network.

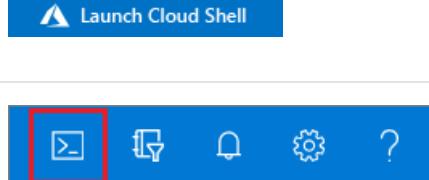
Prerequisites

If you don't have an Azure subscription, create a [free account](#) now.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you decide to install and use Azure CLI locally instead, this quickstart requires you to use Azure CLI version 2.0.28 or later. To find your installed version, run `az --version`. See [Install Azure CLI](#) for install or upgrade info.

Create a resource group and a virtual network

Before you can create a virtual network, you have to create a resource group to host the virtual network. Create

a resource group with [az group create](#). This example creates a resource group named *myResourceGroup* in the *eastus* location:

```
az group create --name myResourceGroup --location eastus
```

Create a virtual network with [az network vnet create](#). This example creates a default virtual network named *myVirtualNetwork* with one subnet named *default*:

```
az network vnet create \
--name myVirtualNetwork \
--resource-group myResourceGroup \
--subnet-name default
```

Create virtual machines

Create two VMs in the virtual network.

Create the first VM

Create a VM with [az vm create](#). If SSH keys don't already exist in a default key location, the command creates them. To use a specific set of keys, use the `--ssh-key-value` option. The `--no-wait` option creates the VM in the background, so that you can continue to the next step. This example creates a VM named *myVm1*:

```
az vm create \
--resource-group myResourceGroup \
--name myVm1 \
--image UbuntuLTS \
--generate-ssh-keys \
--no-wait
```

Create the second VM

Since you used the `--no-wait` option in the previous step, you can go ahead and create the second VM named *myVm2*.

```
az vm create \
--resource-group myResourceGroup \
--name myVm2 \
--image UbuntuLTS \
--generate-ssh-keys
```

Azure CLI output message

The VMs take a few minutes to create. After Azure creates the VMs, the Azure CLI returns output like this:

```
{
  "fqdns": "",
  "id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVm2",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.5",
  "publicIpAddress": "40.68.254.142",
  "resourceGroup": "myResourceGroup"
  "zones": ""}
```

Take note of the `publicIpAddress`. You will use this address to connect to the VM from the internet in the next step.

Connect to a VM from the internet

In this command, replace `<publicIpAddress>` with the public IP address of your `myVm2` VM:

```
ssh <publicIpAddress>
```

Communicate between VMs

To confirm private communication between the `myVm2` and `myVm1` VMs, enter this command:

```
ping myVm1 -c 4
```

You'll receive four replies from `10.0.0.4`.

Exit the SSH session with the `myVm2` VM.

Clean up resources

When no longer needed, you can use [az group delete](#) to remove the resource group and all the resources it has:

```
az group delete --name myResourceGroup --yes
```

Next steps

In this quickstart, you created a default virtual network and two VMs. You connected to one VM from the internet and communicated privately between the two VMs. Azure lets unrestricted private communication between VMs. By default, Azure only lets inbound remote desktop connections to Windows VMs from the internet. Advance to the next article to learn more about configuring different types of VM network communications:

[Filter network traffic](#)

Quickstart: Create a virtual network - Resource Manager template

2/18/2021 • 2 minutes to read • [Edit Online](#)

In this quickstart, you learn how to create a virtual network with two subnets using the Azure Resource Manager template. A virtual network is the fundamental building block for your private network in Azure. It enables Azure resources, like VMs, to securely communicate with each other and with the internet.

An [ARM template](#) is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. In declarative syntax, you describe your intended deployment without writing the sequence of programming commands to create the deployment.

You can also complete this quickstart using the [Azure portal](#), [Azure PowerShell](#), or [Azure CLI](#).

Prerequisites

If you don't have an Azure subscription, create a [free account](#) before you begin.

Review the template

The template used in this quickstart is from [Azure Quickstart templates](#)

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "vnetName": {  
      "type": "string",  
      "defaultValue": "VNet1",  
      "metadata": {  
        "description": "VNet name"  
      }  
    },  
    "vnetAddressPrefix": {  
      "type": "string",  
      "defaultValue": "10.0.0.0/16",  
      "metadata": {  
        "description": "Address prefix"  
      }  
    },  
    "subnet1Prefix": {  
      "type": "string",  
      "defaultValue": "10.0.0.0/24",  
      "metadata": {  
        "description": "Subnet 1 Prefix"  
      }  
    },  
    "subnet1Name": {  
      "type": "string",  
      "defaultValue": "Subnet1",  
      "metadata": {  
        "description": "Subnet 1 Name"  
      }  
    },  
    "subnet2Prefix": {  
      "type": "string",  
      "defaultValue": "10.0.1.0/24",  
      "metadata": {  
        "description": "Subnet 2 Prefix"  
      }  
    }  
  }  
}
```

```

    "metadata": {
      "description": "Subnet 2 Prefix"
    },
    "subnet2Name": {
      "type": "string",
      "defaultValue": "Subnet2",
      "metadata": {
        "description": "Subnet 2 Name"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    }
  },
  "variables": {},
  "resources": [
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2020-05-01",
      "name": "[parameters('vnetName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[parameters('vnetAddressPrefix')]"
          ]
        }
      }
    },
    "resources": [
      {
        "type": "subnets",
        "apiVersion": "2020-05-01",
        "location": "[parameters('location')]",
        "name": "[parameters('subnet1Name')]",
        "dependsOn": [
          "[parameters('vnetName')]"
        ],
        "properties": {
          "addressPrefix": "[parameters('subnet1Prefix')]"
        }
      },
      {
        "type": "subnets",
        "apiVersion": "2020-05-01",
        "location": "[parameters('location')]",
        "name": "[parameters('subnet2Name')]",
        "dependsOn": [
          "[parameters('vnetName')]",
          "[parameters('subnet1Name')]"
        ],
        "properties": {
          "addressPrefix": "[parameters('subnet2Prefix')]"
        }
      }
    ]
  }
}

```

The following Azure resources have been defined in the template:

- **Microsoft.Network/virtualNetworks**: create an Azure virtual network.
- **Microsoft.Network/virtualNetworks/subnets** - create a subnet.

Deploy the template

Deploy Resource Manager template to Azure:

1. Select **Deploy to Azure** to sign in to Azure and open the template. The template creates a virtual network with two subnets.



2. In the portal, on the **Create a Virtual Network with two Subnets** page, type or select the following values:

- **Resource group:** Select **Create new**, type a name for the resource group, and select **OK**.
- **Virtual Network Name:** Type a name for the new virtual network.

3. Select **Review + create**, and then select **Create**.

Review deployed resources

Explore the resources that were created with the virtual network.

To learn about the JSON syntax and properties for a virtual network in a template, see [Microsoft.Network/virtualNetworks](#).

Clean up resources

When you no longer need the resources that you created with the virtual network, delete the resource group. This removes the virtual network and all the related resources.

To delete the resource group, call the `Remove-AzResourceGroup` cmdlet:

```
Remove-AzResourceGroup -Name <your resource group name>
```

Next steps

In this quickstart, you deployed an Azure virtual network with two subnets. To learn more about Azure virtual networks, continue to the tutorial for virtual networks.

[Filter network traffic](#)

Tutorial: Filter network traffic with a network security group using the Azure portal

2/18/2021 • 7 minutes to read • [Edit Online](#)

You can filter network traffic inbound to and outbound from a virtual network subnet with a network security group. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet. In this tutorial, you learn how to:

- Create a network security group and security rules
- Create a virtual network and associate a network security group to a subnet
- Deploy virtual machines (VM) into a subnet
- Test traffic filters

If you prefer, you can complete this tutorial using the [Azure CLI](#) or [PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

Create a virtual network

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Select **Networking**, and then select **Virtual network**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	myVirtualNetwork
Address space	10.0.0.0/16
Subscription	Select your subscription.
Resource group	Select Create new and enter <i>myResourceGroup</i> .
Location	Select East US .
Subnet- Name	mySubnet
Subnet - Address range	10.0.0.0/24

Create application security groups

An application security group enables you to group together servers with similar functions, such as web servers.

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. In the **Search the Marketplace** box, enter *Application security group*. When **Application security group** appears in the search results, select it, select **Application security group** again under **Everything**, and then select **Create**.
3. Enter, or select, the following information, and then select **Create**:

SETTING	VALUE
Name	myAsgWebServers
Subscription	Select your subscription.
Resource group	Select Use existing and then select myResourceGroup .
Location	East US

4. Complete step 3 again, specifying the following values:

SETTING	VALUE
Name	myAsgMgmtServers
Subscription	Select your subscription.
Resource group	Select Use existing and then select myResourceGroup .
Location	East US

Create a network security group

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Select **Networking**, and then select **Network security group**.
3. Enter, or select, the following information, and then select **Create**:

SETTING	VALUE
Name	myNsg
Subscription	Select your subscription.
Resource group	Select Use existing and then select myResourceGroup .
Location	East US

Associate network security group to subnet

1. In the **Search resources, services, and docs** box at the top of the portal, begin typing *myNsg*. When **myNsg** appears in the search results, select it.

2. Under SETTINGS, select Subnets and then select + Associate, as shown in the following picture:

The screenshot shows the 'myNsg - Subnets' page in the Azure portal. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, SETTINGS (Inbound security rules, Outbound security rules, Network interfaces), and Subnets. The 'Subnets' link is highlighted with a red box. On the right, there's a search bar labeled 'Search subnets' and a table with columns 'NAME' and 'ADDRESS RANGE'. A red box highlights the '+ Associate' button at the top of this section.

3. Under Associate subnet, select Virtual network and then select myVirtualNetwork. Select Subnet, select mySubnet, and then select OK.

Create security rules

1. Under SETTINGS, select Inbound security rules and then select + Add, as shown in the following picture:

The screenshot shows the 'myNsg - Inbound security rules' page. The left sidebar has links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Inbound security rules (which is highlighted with a red box). The main area shows a table of existing security rules with columns Priority, Name, Port, Protocol, Source, Destination, and Action. A red box highlights the '+ Add' button at the top of the table.

2. Create a security rule that allows ports 80 and 443 to the myAsgWebServers application security group. Under Add inbound security rule, enter, or select the following values, accept the remaining defaults, and then select Add:

SETTING	VALUE
Destination	Select Application security group, and then select myAsgWebServers for Application security group.

SETTING	VALUE
Destination port ranges	Enter 80,443
Protocol	Select TCP
Name	Allow-Web-All

3. Complete step 2 again, using the following values:

SETTING	VALUE
Destination	Select Application security group, and then select myAsgMgmtServers for Application security group.
Destination port ranges	Enter 3389
Protocol	Select TCP
Priority	Enter 110
Name	Allow-RDP-All

In this tutorial, RDP (port 3389) is exposed to the internet for the VM that is assigned to the *myAsgMgmtServers* application security group. For production environments, instead of exposing port 3389 to the internet, it's recommended that you connect to Azure resources that you want to manage using a VPN or private network connection.

Once you've completed steps 1-3, review the rules you created. Your list should look like the list in the following picture:

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
100	Allow-Web-All	80,443	TCP	Any	myAsgWebServers	Allow
110	Allow-RDP-All	3389	TCP	Any	myAsgMgmtServ...	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalanc...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Create virtual machines

Create two VMs in the virtual network.

Create the first VM

- On the Azure portal menu or from the Home page, select **Create a resource**.
- Select **Compute**, and then select **Windows Server 2016 Datacenter**.
- Enter, or select, the following information, and accept the defaults for the remaining settings:

SETTING	VALUE
Subscription	Select your subscription.
Resource group	Select Use existing and select myResourceGroup .
Name	myVmWeb
Location	Select East US .
User name	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements .

4. Select a size for the VM and then select **Select**.

5. Under **Networking**, select the following values, and accept the remaining defaults:

SETTING	VALUE
Virtual network	Select myVirtualNetwork .
NIC network security group	Select None .

6. Select **Review + Create** at the bottom, left corner, select **Create** to start VM deployment.

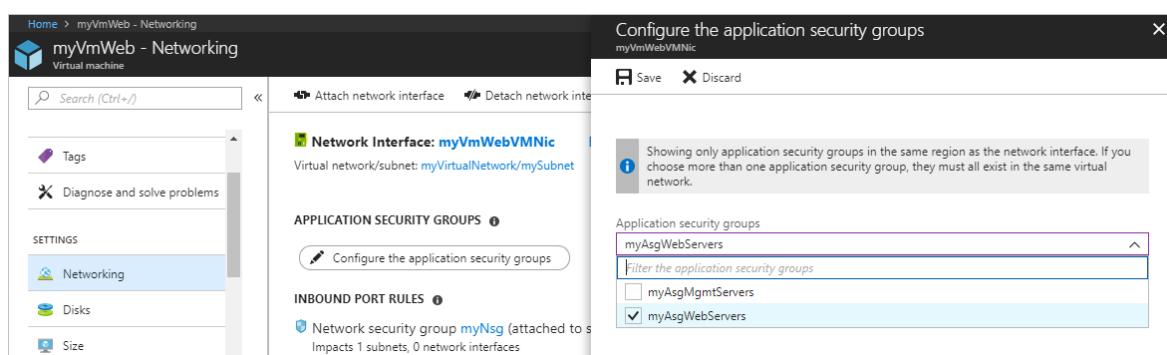
Create the second VM

Complete steps 1-6 again, but in step 3, name the VM *myVmMgmt*. The VM takes a few minutes to deploy. Do not continue to the next step until the VM is deployed.

Associate network interfaces to an ASG

When the portal created the VMs, it created a network interface for each VM, and attached the network interface to the VM. Add the network interface for each VM to one of the application security groups you created previously:

- In the *Search resources, services, and docs* box at the top of the portal, begin typing *myVmWeb*. When the **myVmWeb** VM appears in the search results, select it.
- Under **SETTINGS**, select **Networking**. Select **Configure the application security groups**, select **myAsgWebServers** for **Application security groups**, and then select **Save**, as shown in the following picture:



3. Complete steps 1 and 2 again, searching for the **myVmMgmt** VM and selecting the **myAsgMgmtServers** ASG.

Test traffic filters

1. Connect to the *myVmMgmt* VM. Enter *myVmMgmt* in the search box at the top of the portal. When **myVmMgmt** appears in the search results, select it. Select the **Connect** button.
2. Select **Download RDP file**.
3. Open the downloaded rdp file and 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.
4. Select **OK**.
5. 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.

The connection succeeds, because port 3389 is allowed inbound from the internet to the *myAsgMgmtServers* application security group that the network interface attached to the *myVmMgmt* VM is in.

6. Connect to the *myVmWeb* VM from the *myVmMgmt* VM by entering the following command in a PowerShell session:

```
mstsc /v:myVmWeb
```

You are able to connect to the *myVmWeb* VM from the *myVmMgmt* VM because VMs in the same virtual network can communicate with each other over any port, by default. You can't however, create a remote desktop connection to the *myVmWeb* VM from the internet, because the security rule for the *myAsgWebServers* doesn't allow port 3389 inbound from the internet and inbound traffic from the Internet is denied to all resources, by default.

7. To install Microsoft IIS on the *myVmWeb* VM, enter the following command from a PowerShell session on the *myVmWeb* VM:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

8. After the IIS installation is complete, disconnect from the *myVmWeb* VM, which leaves you in the *myVmMgmt* VM remote desktop connection.
9. Disconnect from the *myVmMgmt* VM.
10. In the *Search resources, services, and docs* box at the top of the Azure portal, begin typing *myVmWeb* from your computer. When **myVmWeb** appears in the search results, select it. Note the **Public IP address** for your VM. The address shown in the following picture is 137.135.84.74, but your address is different:

The screenshot shows the Azure portal interface for a virtual machine named 'myVmWeb'. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The main pane displays the VM's configuration, including its resource group ('myResourceGroup'), status ('Running'), location ('East US'), subscription information, and network settings. A red box highlights the 'Public IP address' field, which contains the value '137.135.84.74'. Other visible details include the computer name ('myVmWeb'), operating system ('Windows'), size ('Standard DS1 v2 (1 vcpus, 3.5 GB memory)'), and network information ('Virtual network/subnet: myVirtualNetwork/mySubnet').

11. To confirm that you can access the *myVmWeb* web server from the internet, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`. You see the IIS welcome screen, because port 80 is allowed inbound from the internet to the *myAsgWebServers* application security group that the network interface attached to the *myVmWeb* VM is in.

Clean up resources

When no longer needed, delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this tutorial, you created a network security group and associated it to a virtual network subnet. To learn more about network security groups, see [Network security group overview](#) and [Manage a network security group](#).

Azure routes traffic between subnets by default. You may instead, choose to route traffic between subnets through a VM, serving as a firewall, for example. To learn how to create a route table, advance to the next tutorial.

[Create a route table](#)

Tutorial: Route network traffic with a route table using the Azure portal

2/18/2021 • 9 minutes to read • [Edit Online](#)

Azure routes traffic between all subnets within a virtual network, by default. You can create your own routes to override Azure's default routing. Custom routes are helpful when, for example, you want to route traffic between subnets through a network virtual appliance (NVA). In this tutorial, you learn how to:

- Create an NVA that routes traffic
- Create a route table
- Create a route
- Associate a route table to a subnet
- Deploy virtual machines (VM) into different subnets
- Route traffic from one subnet to another through an NVA

This tutorial uses the [Azure portal](#). You can also use [Azure CLI](#) or [Azure PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Create an NVA

Network virtual appliances (NVAs) are virtual machines that help with network functions, such as routing and firewall optimization. This tutorial assumes you're using **Windows Server 2016 Datacenter**. You can select a different operating system if you want.

1. On the [Azure portal](#) menu or from the **Home** page, select **Create a resource**.
2. Choose **Security > Windows Server 2016 Datacenter**.

The screenshot shows the Microsoft Azure Marketplace interface. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+)" and a user profile icon. Below the search bar, the word "New" is displayed. The main content area is titled "Popular" and lists various Azure services. One service, "Windows Server 2016 Datacenter", is highlighted with a red box around its thumbnail and title. Other services listed include Ubuntu Server 18.04 LTS, Web App, SQL Database, Function App, Azure Cosmos DB, Kubernetes Service, DevOps Project, Storage account - blob, file, table, queue, and more. On the left side, there's a sidebar with categories like Get started, Recently created, AI + Machine Learning, Analytics, Blockchain, Compute, Containers, Databases, Developer Tools, DevOps, Identity, Integration, Internet of Things, Media, Mixed Reality, IT & Management Tools, Networking, Software as a Service (SaaS), Security (which is highlighted with a red box), Storage, and Web.

3. In the **Create a virtual machine** page, under **Basics**, enter or select this information:

SECTION	SETTING	ACTION
Project details	Subscription	Choose your subscription.
	Resource group	Select Create new , enter <i>myResourceGroup</i> , and select OK .
Instance details	Virtual machine name	Enter <i>myVmName</i> .
	Region	Choose (US) East US .
	Availability options	Choose No infrastructure redundancy required .
	Image	Choose Windows Server 2016 Datacenter .
	Size	Keep the default, Standard DS1 v2 .
Administrator account	Username	Enter a user name of your choosing.

SECTION	SETTING	ACTION
	Password	Enter a password of your choosing, which must be at least 12 characters long and meet the defined complexity requirements .
	Confirm Password	Enter the password again.
Inbound port rules	Public inbound ports	Pick None .
Save money	Already have a Windows Server license?	Pick No .

The screenshot shows the 'Create a virtual machine' wizard on the 'Basics' tab. A red box highlights the 'Project details' section, which includes fields for Subscription (Contoso), Resource group (myResourceGroup), Virtual machine name (myVmNva), Region (US East), Availability options (No infrastructure redundancy required), and Image (Windows Server 2016 Datacenter). Another red box highlights the 'Inbound port rules' section, where 'None' is selected for Public inbound ports. At the bottom, the 'Review + create' button is visible.

Microsoft Azure ! Search resources, services, and docs (G+)

Home > New > Create a virtual machine

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * (1) Contoso

Resource group * (1) myResourceGroup Create new

Virtual machine name * (1) myVmNva

Region * (1) (US) East US

Availability options (1) No infrastructure redundancy required

Image * (1) Windows Server 2016 Datacenter Browse all public and private images

Azure Spot instance (1) Yes No

Size * (1) Standard DS1 v2
1 vcpu, 3.5 GiB memory (\$??./month) Change size

Administrator account

Username * (1) adminuser

Password * (1) Confirm password * (1)

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * (1) None Allow selected ports

Select inbound ports Select one or more ports

! All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Review + create < Previous Next : Disks >

Then select **Next : Disks >**.

4. Under Disks, select the settings that are right for your needs, and then select **Next : Networking >**.

5. Under **Networking**:

a. For **Virtual network**, select **Create new**.

b. In the **Create virtual network** dialog box, under **Name**, enter *myVirtualNetwork*.

c. In **Address space**, replace the existing address range with *10.0.0.0/16*.

d. In **Subnets**, select the **Delete** icon to delete the existing subnet, and then enter the following combinations of **Subnet name** and **Address range**. Once a valid name and range is entered, a new empty row appears below it.

SUBNET NAME	ADDRESS RANGE
<i>Public</i>	<i>10.0.0.0/24</i>
<i>Private</i>	<i>10.0.1.0/24</i>
<i>DMZ</i>	<i>10.0.2.0/24</i>

e. Select **OK** to exit the dialog box.

f. In **Subnet**, choose **DMZ (10.0.2.0/24)**.

g. In **Public IP**, choose **None**, since this VM won't connect over the internet.

h. Select **Next : Management >**.

6. Under **Management**:

a. In **Diagnostics storage account**, select **Create New**.

b. In the **Create storage account** dialog box, enter or select this information:

SETTING	VALUE
Name	<i>mynvastorageaccount</i>
Account kind	Storage (general purpose v1)
Performance	Standard
Replication	Locally-redundant storage (LRS)

c. Select **OK** to exit the dialog box.

d. Select **Review + create**. You're taken to the **Review + create** page, and Azure validates your configuration.

7. When you see the **Validation passed** message, select **Create**.

The VM takes a few minutes to create. Wait until Azure finishes creating the VM. The **Your deployment is underway** page shows you deployment details.

8. When your VM is ready, select **Go to resource**.

Create a route table

1. On the [Azure portal](#) menu or from the Home page, select **Create a resource**.
2. In the search box, enter *Route table*. When **Route table** appears in the search results, select it.
3. In the **Route table** page, select **Create**.
4. In **Create route table**, enter or select this information:

SETTING	VALUE
Name	<i>myRouteTablePublic</i>
Subscription	Your subscription
Resource group	myResourceGroup
Location	(US) East US
Virtual network gateway route propagation	Enabled

The screenshot shows the 'Create route table' dialog box. At the top, there's a breadcrumb navigation: Home > New > Route table > Create route table. The main title is 'Create route table' with a close button. Below the title, a note says 'You can add routes to this table after it's created.' The form fields are as follows:

- Name ***: myRouteTablePublic
- Subscription ***: Contoso
- Resource group ***: myResourceGroup (with a 'Create new' link)
- Location ***: (US) East US
- Virtual network gateway route propagation**: Enabled (button)

At the bottom of the dialog are two buttons: 'Create' (in blue) and 'Automation options'.

5. Select **Create**.

Create a route

1. Go to the [Azure portal](#) to manage your route table. Search for and select **Route tables**.
2. Pick the name of your route table (**myRouteTablePublic**).
3. Choose **Routes > Add**.

myRouteTablePublic - Routes

Route table

Search (Ctrl+ /) <> Add

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Configuration

Routes **Add**

Subnets

Properties

Locks

Export template

Support + troubleshooting

Effective routes

New support request

4. In **Add route**, enter or select this information:

SETTING	VALUE
Route name	<i>ToPrivateSubnet</i>
Address prefix	<i>10.0.1.0/24</i> (the address range of the <i>Private</i> subnet created earlier)
Next hop type	Virtual appliance
Next hop address	<i>10.0.2.4</i> (an address within the address range of the <i>DMZ</i> subnet)

5. Select **OK**.

Associate a route table to a subnet

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Virtual networks**.
2. Pick the name of your virtual network (**myVirtualNetwork**).

3. In the virtual network's menu bar, choose **Subnets**.
4. In the virtual network's subnet list, choose **Public**.
5. In **Route table**, choose the route table you created (**myRouteTablePublic**), and then select **Save** to associate your route table to the *Public* subnet.

Home > Virtual networks > myVirtualNetwork | Subnets > Public

Public
myVirtualNetwork

Save **Discard** **Delete** **Refresh**

Address range (CIDR block) ①
10.0.0.0/24
10.0.0.0 - 10.0.0.255 (256 addresses)

Available addresses ①
250

NAT gateway ①
None

Add IPv6 address space

Network security group
None

Route table
myRouteTablePublic

Users >
Manage users

Service endpoints

Services ①
0 selected

Subnet delegation

Delegate subnet to a service ①
None

Turn on IP forwarding

Next, turn on IP forwarding for your new NVA virtual machine, *myVmNva*. When Azure sends network traffic to *myVmNva*, if the traffic is destined for a different IP address, IP forwarding sends the traffic to the correct location.

1. Go to the [Azure portal](#) to manage your VM. Search for and select **Virtual machines**.
2. Pick the name of your VM (**myVmNva**).
3. In your NVA virtual machine's menu bar, select **Networking**.
4. Select **myvmnva123**. That's the network interface Azure created for your VM. Azure adds numbers to ensure a unique name.

myVmNva - Networking

Virtual machine

Search (Ctrl+ /) Attach network interface Detach network interface

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Networking

Connect Disks Size Security Extensions Continuous delivery (Preview) Availability + scaling Configuration Identity Properties Locks Export template

Operations

Bastion

Network Interface: myvmnva123 Effective security rule

Virtual network/subnet: myVirtualNetwork/DMZ NIC Public IP: Accelerated networking: Disabled

Inbound port rules Outbound port rules Application security rules

Network security group myVmNva-nsg (attached to network interface) Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port
65000	AllowVnetInBound	Any
65001	AllowAzureLoadBalanc...	Any
65500	DenyAllInBound	Any

5. In the network interface menu bar, select **IP configurations**.
6. In the **IP configurations** page, set **IP forwarding** to **Enabled**, and select **Save**.

The screenshot shows the 'IP configurations' section of the Azure portal for a specific network interface. The 'Save' button is highlighted with a red box. The 'IP forwarding' setting is set to 'Enabled', also highlighted with a red box. The subnet is set to 'DMZ (10.0.2.0/24)'. The left sidebar shows various settings like Overview, Activity log, Access control (IAM), Tags, and IP configurations, which is currently selected.

Create public and private virtual machines

Create a public VM and a private VM in the virtual network. Later, you'll use them to see that Azure routes the *Public* subnet traffic to the *Private* subnet through the NVA.

To create the public VM and the private VM, follow the steps of [Create an NVA](#) earlier. You don't need to wait for deployment to finish or go to the VM resource. You'll use most of the same settings, except as described below.

Before you select **Create** to create the public or private VM, go to the following two subsections ([Public VM](#) and [Private VM](#)), which show the values that have to be different. You may continue to the next section ([Route traffic through an NVA](#)) after Azure finishes deploying both VMs.

Public VM

TAB	SETTING	VALUE
Basics	Resource group	myResourceGroup
	Virtual machine name	myVmPublic
	Public inbound ports	Allow selected ports
	Select inbound ports	RDP
Networking	Virtual network	myVirtualNetwork
	Subnet	Public (10.0.0.0/24)
	Public IP address	The default
Management	Diagnostics storage account	mynvastorageaccount

Private VM

TAB	SETTING	VALUE
Basics	Resource group	myResourceGroup
	Virtual machine name	myVmPrivate
	Public inbound ports	Allow selected ports
	Select inbound ports	RDP
Networking	Virtual network	myVirtualNetwork
	Subnet	Private (10.0.1.0/24)
	Public IP address	The default
Management	Diagnostics storage account	mynvastorageaccount

Route traffic through an NVA

Sign in to myVmPrivate over remote desktop

1. Go to the [Azure portal](#) to manage your private VM. Search for and select **Virtual machines**.
2. Pick the name of your private VM (**myVmPrivate**).
3. In the VM menu bar, select **Connect** to create a remote desktop connection to the private VM.
4. In the **Connect with RDP** page, select **Download RDP File**. Azure creates a Remote Desktop Protocol (.rdp) file and downloads it to your computer.
5. Open the downloaded .rdp file. If prompted, select **Connect**. Select **More choices > Use a different account**, and then enter the user name and password you specified when creating the private VM.
6. Select **OK**.
7. If you receive a certificate warning during the sign-in process, select **Yes** to connect to the VM.

Enable ICMP through the Windows firewall

In a later step, you'll use the trace route tool to test routing. Trace route uses the Internet Control Message Protocol (ICMP), which the Windows Firewall denies by default. Enable ICMP through the Windows firewall.

1. In the Remote Desktop of *myVmPrivate*, open PowerShell.
2. Enter this command:

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

You'll be using trace route to test routing in this tutorial. For production environments, we don't recommend allowing ICMP through the Windows Firewall.

Turn on IP forwarding within myVmNva

You [turned on IP forwarding](#) for the VM's network interface using Azure. The VM's operating system also has to forward network traffic. Turn on IP forwarding for *myVmNva* VM's operating system with these commands.

- From a command prompt on the *myVmPrivate* VM, open a remote desktop to the *myVmNva* VM:

```
mstsc /v:myvmnva
```

- From PowerShell on the *myVmNva* VM, enter this command to turn on IP forwarding:

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRouter -Value 1
```

- Restart the *myVmNva* VM: From the taskbar, select **Start > Power, Other (Planned) > Continue**.

This also disconnects the remote desktop session.

- After the *myVmNva* VM restarts, create a remote desktop session to the *myVmPublic* VM. While still connected to the *myVmPrivate* VM, open a command prompt and run this command:

```
mstsc /v:myVmPublic
```

- In the remote desktop of *myVmPublic*, open PowerShell.

- Enable ICMP through the Windows firewall by entering this command:

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

Test the routing of network traffic

First, let's test routing of network traffic from the *myVmPublic* VM to the *myVmPrivate* VM.

- From PowerShell on the *myVmPublic* VM, enter this command:

```
tracert myVmPrivate
```

The response is similar to this example:

```
Tracing route to myVmPrivate.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.1.4]
over a maximum of 30 hops:

 1  <1 ms      *      1 ms  10.0.2.4
 2  1 ms      1 ms  1 ms  10.0.1.4

Trace complete.
```

You can see the first hop is to 10.0.2.4, which is NVA's private IP address. The second hop is to the private IP address of the *myVmPrivate* VM: 10.0.1.4. Earlier, you added the route to the *myRouteTablePublic* route table and associated it to the *Public* subnet. As a result, Azure sent the traffic through the NVA and not directly to the *Private* subnet.

- Close the remote desktop session to the *myVmPublic* VM, which leaves you still connected to the *myVmPrivate* VM.
- From a command prompt on the *myVmPrivate* VM, enter this command:

```
tracert myVmPublic
```

This command tests the routing of network traffic from the *myVmPrivate* VM to the *myVmPublic* VM. The response is similar to this example:

```
Tracing route to myVmPublic.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.0.4]
over a maximum of 30 hops:

1      1 ms      1 ms      1 ms  10.0.0.4

Trace complete.
```

You can see that Azure routes traffic directly from the *myVmPrivate* VM to the *myVmPublic* VM. By default, Azure routes traffic directly between subnets.

4. Close the remote desktop session to the *myVmPrivate* VM.

Clean up resources

When the resource group is no longer needed, delete *myResourceGroup* and all resources it has:

1. Go to the [Azure portal](#) to manage your resource group. Search for and select **Resource groups**.
2. Pick the name of your resource group (**myResourceGroup**).
3. Select **Delete resource group**.
4. In the confirmation dialog box, enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME**, and then select **Delete**. Azure deletes the *myResourceGroup* and all resources tied to that resource group, including your route tables, storage accounts, virtual networks, VMs, network interfaces, and public IP addresses.

Next steps

In this tutorial, you created a route table and associated it to a subnet. You created a simple NVA that routed traffic from a public subnet to a private subnet. Now you can deploy different preconfigured NVAs from the [Azure Marketplace](#), which provide many useful network functions. To learn more about routing, see [Routing overview](#) and [Manage a route table](#).

While you can deploy many Azure resources within a virtual network, Azure can't deploy resources for some PaaS services into a virtual network. It's possible to restrict access to the resources of some Azure PaaS services, though the restriction must only be traffic from a virtual network subnet. To learn how to restrict network access to Azure PaaS resources, see the next tutorial.

[Restrict network access to PaaS resources](#)

NOTE

Azure services cost money. Azure Cost Management helps you set budgets and configure alerts to keep spending under control. Analyze, manage, and optimize your Azure costs with Cost Management. To learn more, see the [quickstart on analyzing your costs](#).

Tutorial: Restrict network access to PaaS resources with virtual network service endpoints using the Azure portal

2/18/2021 • 10 minutes to read • [Edit Online](#)

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network. In this tutorial, you learn how to:

- Create a virtual network with one subnet
- Add a subnet and enable a service endpoint
- Create an Azure resource and allow network access to it from only a subnet
- Deploy a virtual machine (VM) to each subnet
- Confirm access to a resource from a subnet
- Confirm access is denied to a resource from a subnet and the internet

If you prefer, you can complete this tutorial using the [Azure CLI](#) or [Azure PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Log in to Azure

Log in to the Azure portal at <https://portal.azure.com>.

Create a virtual network

1. Select + **Create a resource** on the upper, left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual networks**.
3. Click + **Add** and enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select Create new and enter <i>myResourceGroup</i> .
Name	Enter <i>myVirtualNetwork</i>
Region	Select (US) East US

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

Project details

Subscription *

Resource group *
[Create new](#)

Instance details

Name *

Region *

Buttons: [Review + create](#) (highlighted in blue), [< Previous](#), [Next : IP Addresses >](#), [Download a template for automation](#)

4. Click **Next: IP addresses >**

SETTING	VALUE
IPv4Address space	Leave as default
Subnet name	Click default and change the name from "default" to "Public"
Subnet Address Range	Leave as default

5. Click **Next: Security >**

SETTING	VALUE
BastionHost	Disable
DDoS protection	Disable
Firewall	Disable

6. When complete, click **Review and create**.

7. If the validation checks pass, click **Create**.

8. Wait for the deployment to finish, then click **Go to resource** or move on to the next section.

Enable a service endpoint

Service endpoints are enabled per service, per subnet. To create a subnet and enable a service endpoint for the subnet:

- If you are not already on the virtual network resource page, you can search for the newly created network in the **Search resources, services, and docs** box at the top of the portal, enter *myVirtualNetwork*, and select it from the list.

2. In the **Settings** menu (left), select **Subnets**, and then select **+ Subnet**, as shown:

3. Under **Add subnet**, select or enter the following information, and then select **OK**:

SETTING	VALUE
Name	Private
Address range	Leave as default
Service endpoints	Select Microsoft.Storage
Service endpoint policies	Leave default as 0

Caution

Before enabling a service endpoint for an existing subnet that has resources in it, see [Change subnet settings](#).

4. Click **Save**, then close the Subnet window on the right. The newly created subnet should appear the list.

Restrict network access for a subnet

By default, all virtual machine instances in a subnet can communicate with all resources. You can limit communication to and from all resources in a subnet by creating a network security group, and associating it to the subnet:

1. Select **All services** in the upper left corner of the Azure portal.
2. Select **Networking**, and then select (or search for) **Network security groups**.
3. From the **Network security groups** page, click **+ Add**.
4. Enter the following information

SETTING	VALUE
Subscription	Select your subscription

SETTING	VALUE
Resource group	Select <i>myResourceGroup</i> from the list
Name	Enter myNsgPrivate
Location	Select East US

5. Click **Review + create**, and when the validation check is passed, click **Create**.
6. After the network security group is created, click **Go to resource** or search for *myNsgPrivate*.
7. Under **Settings** on the left, select **Outbound security rules**.
8. Select **+ Add**.
9. Create a rule that allows outbound communication to the Azure Storage service. Enter, or select, the following information, and then select **Add**:

SETTING	VALUE
Source	Select VirtualNetwork
Source port ranges	*
Destination	Select Service Tag
Destination service tag	Select Storage
Destination port ranges	Leave default as <i>8080</i>
Protocol	Any
Action	Allow
Priority	100
Name	Rename to Allow-Storage-All

10. Create another outbound security rule that denies communication to the internet. This rule overrides a default rule in all network security groups that allows outbound internet communication. Complete steps 6-9 from above using the following values:

SETTING	VALUE
Source	Select VirtualNetwork
Source port ranges	*
Destination	Select Service Tag
Destination service tag	Select Internet
Destination port ranges	*

SETTING	VALUE
Protocol	Any
Action	Change default to <i>Deny</i>
Priority	110
Name	Change to <i>Deny-Internet-All</i>

11. Create an *inbound security rule* that allows Remote Desktop Protocol (RDP) traffic to the subnet from anywhere. The rule overrides a default security rule that denies all inbound traffic from the internet. Remote desktop connections are allowed to the subnet so that connectivity can be tested in a later step.
12. Under **Settings**, select **Inbound security rules**.
13. Select **+ Add** and use the following values:

SETTING	VALUE
Source	Any
Source port ranges	*
Destination	Select VirtualNetwork
Destination port ranges	Change to <i>3389</i>
Protocol	Any
Action	Allow
Priority	120
Name	Change to <i>Allow-RDP-All</i>

WARNING

RDP port 3389 is exposed to the Internet. This is only recommended for testing. For *Production environments*, we recommend using a VPN or private connection.

1. Under **Settings**, select **Subnets**.
2. Click **+ Associate**.
3. Under **Virtual network**, select **myVirtualNetwork**.
4. Under **Subnet**, select **Private**, and then select **Ok**.

Restrict network access to a resource

The steps required to restrict network access to resources created through Azure services, which are enabled for service endpoints will vary across services. See the documentation for individual services for specific steps for each service. The remainder of this tutorial includes steps to restrict network access for an Azure Storage account, as an example.

Create a storage account

1. Select + **Create a resource** on the upper, left corner of the Azure portal.
2. Enter "Storage account" in the search bar, and select it from the drop-down menu.
3. Click + **Add**.
4. Enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select <i>myResourceGroup</i>
Storage Account Name	Enter a name that is unique across all Azure locations, between 3-24 characters in length, using only numbers and lower-case letters.
Location	Select (US) East US
Performance	Standard
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)

5. Select **Create + review**, and when validation checks have passed, click **Create**.

NOTE

The deployment may take a couple of minutes to complete.

6. After the storage account is created, click **Go to resource**

Create a file share in the storage account

1. Go to your storage account overview page.
2. Select the **File Shares** app icon, then click + **File share**.

SETTING	VALUE
Name	my-file-share
Quota	'Set to maximum'

The screenshot shows the Microsoft Azure Storage Account interface for 'rhodiumstoracct2'. On the left, there's a sidebar with various settings like Overview, Activity log, Tags, and Networking. The main area shows 'File shares' with a single entry: 'You don't have any file shares yet. Click '+ File share' to get started.' A red box highlights the '+ File share' button. To the right, a modal window titled 'New file share' is open. It has a 'Name' field containing 'my-file-share' with a green checkmark. Below it is a 'Quota' field with a 'Set to maximum' option. Under 'Tiers', 'Premium' is selected. At the bottom of the modal are 'Create' and 'Discard' buttons.

3. Click **Create**.

4. The file share should be shown in the Azure window, if not click **Refresh**

Restrict network access to a subnet

By default, storage accounts accept network connections from clients in any network, including the internet. You can restrict network access from the internet, and all other subnets in all virtual networks (except the *Private* subnet in the *myVirtualNetwork* virtual network.) To restrict network access to a subnet:

1. Under **Settings** for your (uniquely named) storage account, select **Networking**.
2. Select **Selected networks**.
3. Select **+ Add existing virtual network**.
4. Under **Add networks**, select the following values, and then select **Add**:

SETTING	VALUE
Subscription	Select your subscription
Virtual networks	myVirtualNetwork
Subnets	Private

The screenshot shows the Microsoft Azure Storage Account Networking settings page for 'rhodiumstorageaccount'. The 'Networking' tab is selected. In the 'Virtual networks' section, there is a red box around the 'Add existing virtual network' button. To the right, a modal window titled 'Add networks' is open, showing a list of virtual networks. Another red box highlights the 'Virtual networks' dropdown which shows 'myVirtualNetwork' selected. Under 'Subnets', 'Private' is selected.

5. Click **Add** and, then immediately click the **Save** icon to save the changes.

6. Under **Settings** for the storage account, select **Access keys**, as shown in the following image:

The screenshot shows the Microsoft Azure Storage account access keys page for 'rhodiumstoracct2'. The 'Access key' tab is selected. In the left sidebar, 'Access key' is also highlighted with a red box. The page displays two access keys: 'key1' and 'key2', each with its corresponding key value and connection string.

7. Click **Show keys** and note the **Key** values, as you'll have to manually enter key1 in a later step when mapping the file share to a drive letter in a VM.

Create virtual machines

To test network access to a storage account, deploy a VM to each subnet.

Create the first virtual machine

1. From the "Search resources ..." bar, search for **Virtual machines**.
2. Select **+ Add > Virtual machine**.
3. Enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select **myResourceGroup, which was created earlier.
Virtual machine name	Enter <i>myVmPublic</i>
Region	(US) East US
Availability options	Availability zone
Availability zone	1
Image	Windows Server 2019 Datacenter - Gen1
Size	Select the VM Instance size you want to use
Username	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements .
Public inbound ports	Allow selected ports
Select inbound ports	Leave default set to <i>RDP (3389)</i>

Microsoft Azure

Search resources, services, and docs (G+ /)

Home > Virtual machines >

Create a virtual machine

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Resource group * Create new

Instance details

Virtual machine name * Region * Availability options Availability zone * Image * See all images

Azure Spot instance

Size * See all sizes

Administrator account

Username * Password * Confirm password *

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * None Allow selected ports

Select inbound ports *

⚠️ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Review + create < Previous Next : Disks >

4. Select the **Networking** tab and then select **myVirtualNetwork**.
5. Select the **Public** subnet.
6. Under **NIC Network Security Group**, select **Advanced**. The portal automatically creates a network security group for you that allows port 3389, which you'll need open to connect to the virtual machine in a later step.

[Home](#) > [Virtual machines](#) >

Create a virtual machine

[Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *  [Create new](#)Subnet *  [Manage subnet configuration](#)Public IP  [Create new](#)NIC network security group  None Basic AdvancedConfigure network security group *  [Create new](#)Accelerated networking  On Off

The selected VM size does not support accelerated networking.

Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution?

 Yes No[Review + create](#)[< Previous](#)[Next : Management >](#)

7. Select **Review and create**, then **Create** and wait for the deployment to finish.

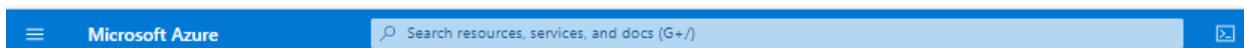
8. Click **Go to resource**, or open the **Home > Virtual machines** page, and select the VM you just created *myVmPublic*, which should be started.

Create the second virtual machine

1. Complete steps 1-8 again, but in step 3, name the virtual machine *myVmPrivate* and set **Public inbound port** to "None".
2. In step 4-5, select the **Private** subnet.

NOTE

The **NIC network security group** and **Public inbound ports** settings should mirror the image shown below, including the blue confirmation window stating: "all public internet traffic will be blocked by default".



Home > Virtual machines >

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
[Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network* [Create new](#)

Subnet* [Manage subnet configuration](#)

Public IP [Create new](#)

NIC network security group Basic Advanced

Public inbound ports* None Allow selected ports

Select inbound ports

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Accelerated networking On Off

The selected VM size does not support accelerated networking.

Load balancing

[Review + create](#)

[< Previous](#)

[Next : Management >](#)

3. Select **Review and create**, then **Create** and wait for the deployment to finish.

WARNING

Please do not continue to the next step until the deployment is complete.

4. Wait for the confirmation window shown below and click **Go to resource**.

Microsoft Azure

Home > CreateVm-MicrosoftWindowsServer.WindowsServer-201-20201209152720 | Overview

Deployment

Overview We'd love your feedback! →

Your deployment is complete

Deployment name: CreateVm-MicrosoftWindowsServer.WindowsSe... Start time: 12/9/2020, 3:42:28 PM
Subscription: C&L App Platforms Correlation ID: 102a0772-1851-4967-950e-7575b912E
Resource group: myResourceGroup

Deployment details [Download](#)

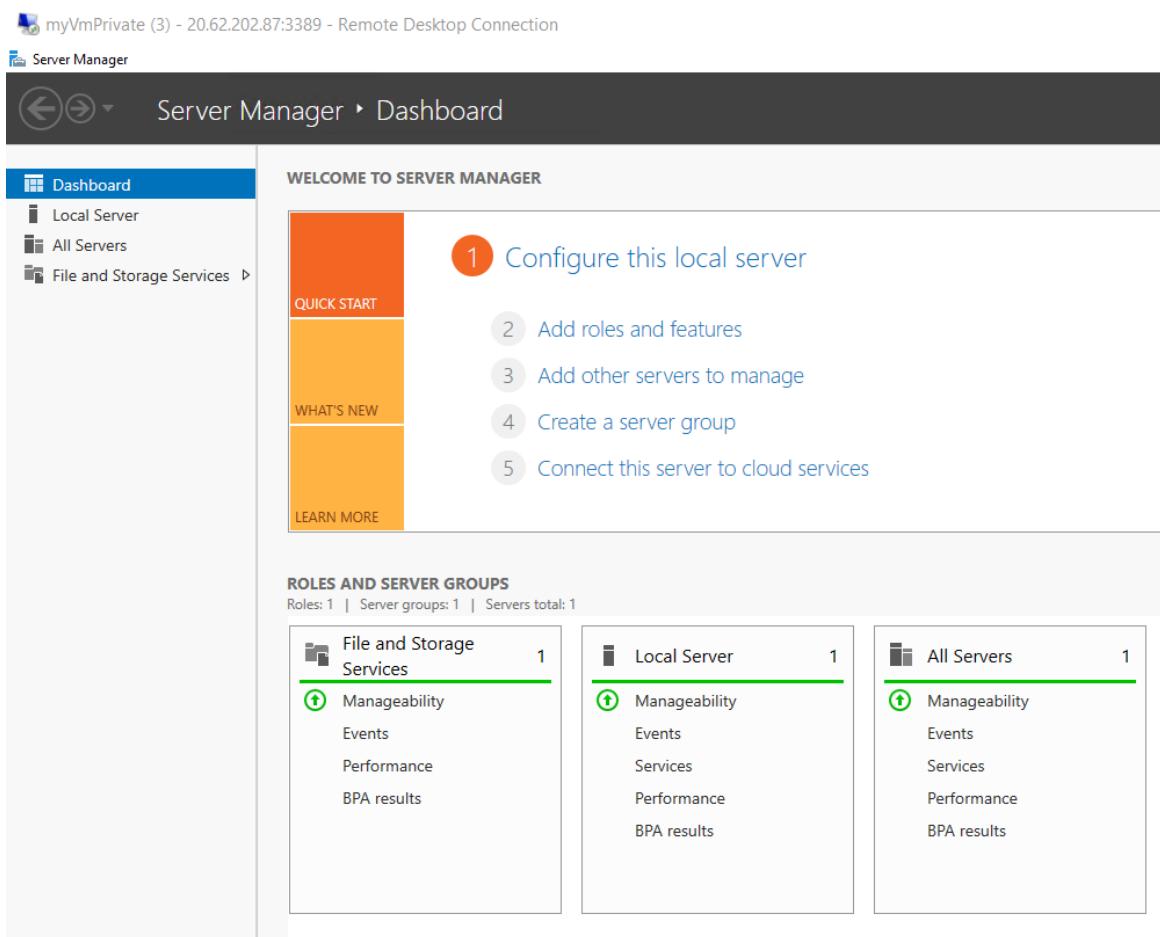
Next steps

[Enable Automanage virtual machine best practices \(preview\)](#) Recommended
[Setup auto-shutdown](#) Recommended
[Monitor VM health, performance and network dependencies](#) Recommended
[Run a script inside the virtual machine](#) Recommended

[Go to resource](#) [Create another VM](#)

Confirm access to storage account

1. Once the *myVmPrivateVM* has been created, click **Go to resource**.
2. Connect to the VM by selecting the **Connect > RDP**.
3. After selecting the **Connect** button, a Remote Desktop Protocol (.rdp) file is created. Click **Download RDP File** to download to your computer.
4. 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. For the email field, enter the "Administrator account: username" credentials you specified earlier.
5. Select **OK**.
6. 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. You should see the VM start as shown:



7. In the VM window, open a PowerShell CLI instance.
8. Using the script below, map the Azure file share to drive Z using PowerShell. Before running the commands that follow, replace `<storage-account-key>` and both `<storage-account-name>` fields with values you supplied and "d earlier in [Create a storage account](#).

```
$acctKey = ConvertTo-SecureString -String "<storage-account-key>" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\<storage-account-name>", $acctKey
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\\<storage-account-name>.file.core.windows.net\my-file-share" -Credential $credential
```

PowerShell returns output similar to the following example output:

Name	Used (GB)	Free (GB)	Provider	Root
Z			FileSystem	\vnt.file.core.windows.net\my-f...

The Azure file share successfully mapped to the Z drive.

9. Close the remote desktop session to the *myVmPrivate* VM.

Confirm access is denied to storage account

1. Enter *myVmPublic* in the **Search resources, services, and docs** box at the top of the portal.
2. When **myVmPublic** appears in the search results, select it.
3. Complete steps 1-8 above in [Confirm access to storage account for the myVmPublic VM](#).

After a short wait, you receive a `New-PSDrive : Access is denied` error. Access is denied because the *myVmPublic* VM is deployed in the *Public* subnet. The *Public* subnet does not have a service endpoint enabled for Azure Storage. The storage account only allows network access from the *Private* subnet, not the *Public* subnet.

4. Close the remote desktop session to the *myVmPublic* VM.
5. Back in the Azure portal, go to the uniquely named storage account you created earlier.
6. Under File Service, select **File shares**, the *my-file-share*, created earlier.

7. You should receive the following error message:

The screenshot shows the Microsoft Azure portal interface. In the center, there's a large error message: "This machine doesn't seem to have access." Below it, a note says: "This storage account is located in a VNET. Recent changes to 'Firewalls and virtual networks' settings may not be in effect yet. If you expect this machine to be able to connect to the content of this file share, check that this machine is a part of the VNET or try waiting a few minutes for changes in settings to take effect, and then refresh this page." At the bottom, there's a summary table with details like Session ID, Resource ID, Extension, Content, and Error code.

Summary	Session ID dd087e494e9e4d0899af8b25b129dd9e	Resource ID /subscriptions/b96d505c-4e34-426c-9e93-7e5e8ff744b...
	Extension Microsoft_Azure_FileStorage	Content FilesGridBlade2
	Error code 403	

NOTE

Access is denied, because your computer is not in the *Private* subnet of the *MyVirtualNetwork* virtual network.

Clean up resources

When no longer needed, delete the resource group and all resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this tutorial, you enabled a service endpoint for a virtual network subnet. You learned that you can enable service endpoints for resources deployed from multiple Azure services. You created an Azure Storage account and restricted network access to the storage account to only resources within a virtual network subnet. To learn more about service endpoints, see [Service endpoints overview](#) and [Manage subnets](#).

If you have multiple virtual networks in your account, you may want to connect two virtual networks together so the resources within each virtual network can communicate with each other. To learn how to connect virtual networks, advance to the next tutorial.

[Connect virtual networks](#)

Tutorial: Connect virtual networks with virtual network peering using the Azure portal

2/13/2020 • 5 minutes to read • [Edit Online](#)

You can connect virtual networks to each other with virtual network peering. These virtual networks can be in the same region or different regions (also known as Global VNet peering). Once virtual networks are peered, resources in both virtual networks are able to communicate with each other, with the same latency and bandwidth as if the resources were in the same virtual network. In this tutorial, you learn how to:

- Create two virtual networks
- Connect two virtual networks with a virtual network peering
- Deploy a virtual machine (VM) into each virtual network
- Communicate between VMs

If you prefer, you can complete this tutorial using the [Azure CLI](#) or [Azure PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Log in to Azure

Log in to the Azure portal at <https://portal.azure.com>.

Create virtual networks

1. On the Azure portal, select **Create a resource**.
2. Select **Networking**, and then select **Virtual network**.
3. On the **Basics** tab, enter or select the following information and accept the defaults for the remaining settings:

SETTING	VALUE
Subscription	Select your subscription.
Resource group	Select Create new and enter <i>myResourceGroup</i> .
Region	Select East US .
Name	myVirtualNetwork1

4. On the **IP Addresses** tab, enter 10.0.0.0/16 for the **Address Space** field. Click the **Add subnet** button below and enter *Subnet1* for **Subnet Name** and 10.0.0.0/24 for **Subnet Address range**.
5. Select **Review + Create** and then select **Create**.
6. Complete steps 1-5 again, with the following changes:

SETTING	VALUE
Name	myVirtualNetwork2

SETTING	VALUE
Address space	10.1.0.0/16
Resource group	Select Use existing and then select myResourceGroup .
Subnet name	Subnet2
Subnet Address range	10.1.0.0/24

Peer virtual networks

1. In the Search box at the top of the Azure portal, begin typing *MyVirtualNetwork1*. When **myVirtualNetwork1** appears in the search results, select it.
2. Select **Peerings**, under **Settings**, and then select **Add**, as shown in the following picture:

The screenshot shows the Azure portal interface for managing a virtual network. The URL in the address bar is "Home > Virtual networks > myVirtualNetwork1 - Peerings". The left sidebar lists various settings: Address space, Connected devices, Subnets, DDoS protection, Firewall, Security, DNS servers, **Peerings** (which is highlighted with a red box), and Service endpoints. The main pane displays a table titled "myVirtualNetwork1 - Peerings" with columns: Name, Peering status, Peer, and Gateway transit. A search bar at the top of the main pane says "Search peerings". A red box also highlights the "+ Add" button in the top right corner of the main pane.

3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**.

SETTING	VALUE
Name of the peering from myVirtualNetwork1 to remote virtual network	myVirtualNetwork1-myVirtualNetwork2 - When the page first loads, you'll see the phrase "remote virtual network" here. After you choose the remote virtual network, the phrase "remote virtual network" will be replaced with the name of the remote virtual network.
Subscription	Select your subscription.
Virtual network	myVirtualNetwork2 - To select the <i>myVirtualNetwork2</i> virtual network, select Virtual network , then select myVirtualNetwork2 (myResourceGroup) . You can select a virtual network in the same region or in a different region.
Name of the peering from myVirtualNetwork2 to myVirtualNetwork1	myVirtualNetwork2-myVirtualNetwork1

Home > myVirtualNetwork1 - Peering > Add peering

Add peering

myVirtualNetwork1

For peering to work, a peering link must be created from myVirtualNetwork1 to myVirtualNetwork2 as well as from myVirtualNetwork2 to myVirtualNetwork1.

* Name of the peering from myVirtualNetwork1 to myVirtualNetwork2
myVirtualNetwork1-myVirtualNetwork2 ✓

Peer details

Virtual network deployment model ⓘ
 Resource manager Classic

I know my resource ID ⓘ

* Subscription ⓘ

* Virtual network
myVirtualNetwork2 (myResourceGroup)

* Name of the peering from myVirtualNetwork2 to myVirtualNetwork1
myVirtualNetwork2-myVirtualNetwork1 ✓

OK

The PEERING STATUS is *Connected*, as shown in the following picture:

Home > myVirtualNetwork1 - Peering

myVirtualNetwork1 - Peering

Virtual network

Search (Ctrl+ /)
Add Refresh

Overview Activity log Access control (IAM)

NAME	PEERING STATUS	PEER	GATEWAY TRANSIT
myVirtualNetwork1-myVi...	Connected	myVirtualNetwork2	Disabled

If you don't see the status, refresh your browser.

Create virtual machines

Create a VM in each virtual network so that you can communicate between them in a later step.

Create the first VM

1. On the Azure portal, select **Create a resource**.
2. Select **Compute**, and then select **Windows Server 2016 Datacenter**. You can select a different operating system, but the remaining steps assume you selected **Windows Server 2016 Datacenter**.
3. Enter, or select, the following information for **Basics**, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Resource group	Select Use existing and then select myResourceGroup .
Name	myVm1
Location	Select East US .
User name	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements .

- Select a VM size for the **Size** option.
- Select the following values for under **Networking**:

SETTING	VALUE
Virtual network	myVirtualNetwork1 - If it's not already selected, select Virtual network and then select myVirtualNetwork1 .
Subnet	Subnet1 - If it's not already selected, select Subnet and then select Subnet1 .

- Select **Networking**. Choose **Allow selected ports** for the **Public inbound ports** option. Choose **RDP** for the **Select inbound ports** option below this.
- Select the **Review + Create** button in the lower, left-hand corner to start the VM deployment.

Create the second VM

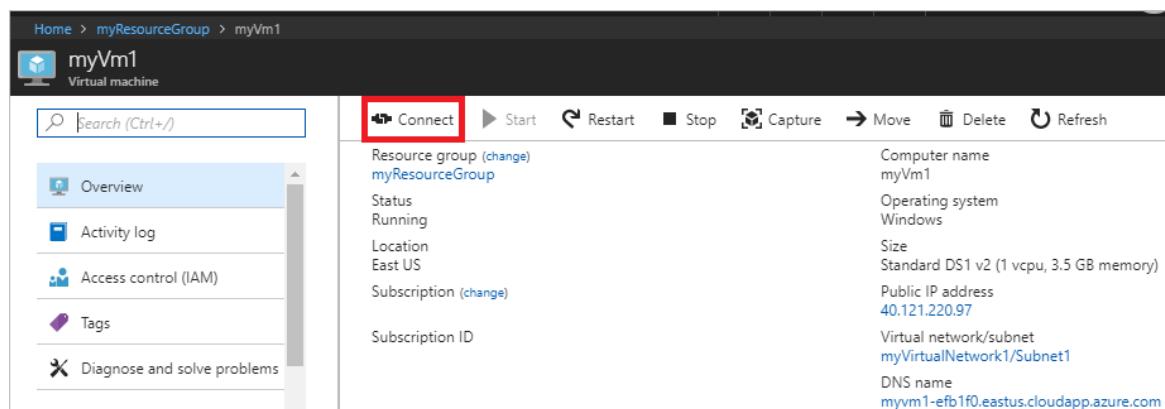
Complete steps 1-6 again, with the following changes:

SETTING	VALUE
Name	myVm2
Virtual network	myVirtualNetwork2

The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

Communicate between VMs

- In the **Search** box at the top of the portal, begin typing *myVm1*. When **myVm1** appears in the search results, select it.
- Create a remote desktop connection to the *myVm1* VM by selecting **Connect**, as shown in the following picture:



- To connect to the VM, 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), then select **OK**.
- You may receive a certificate warning during the sign-in process. Select **Yes** to proceed with the connection.

6. In a later step, ping is used to communicate with the *myVm2* VM from the *myVm1* VM. Ping uses the Internet Control Message Protocol (ICMP), which is denied through the Windows Firewall, by default. On the *myVm1* VM, enable ICMP through the Windows firewall, so that you can ping this VM from *myVm2* in a later step, using PowerShell:

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

Though ping is used to communicate between VMs in this tutorial, allowing ICMP through the Windows Firewall for production deployments is not recommended.

7. To connect to the *myVm2* VM, enter the following command from a command prompt on the *myVm1* VM:

```
mstsc /v:10.1.0.4
```

8. Since you enabled ping on *myVm1*, you can now ping it by IP address:

```
ping 10.0.0.4
```

9. Disconnect your RDP sessions to both *myVm1* and *myVm2*.

Clean up resources

When no longer needed, delete the resource group and all resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this tutorial, you learned how to connect two networks in the same Azure region, with virtual network peering. You can also peer virtual networks in different [supported regions](#) and in [different Azure subscriptions](#), as well as create [hub and spoke network designs](#) with peering. To learn more about virtual network peering, see [Virtual network peering overview](#) and [Manage virtual network peerings](#).

To connect your own computer to a virtual network through a VPN, and interact with resources in a virtual network, or in peered virtual networks, see [Connect your computer to a virtual network](#).

Tutorial: Create a NAT gateway using the Azure portal

11/2/2020 • 5 minutes to read • [Edit Online](#)

This tutorial shows you how to use Azure Virtual Network NAT service. You'll create a NAT gateway to provide outbound connectivity for a virtual machine in Azure.

If you prefer, you can do these steps using the [Azure CLI](#), [Azure PowerShell](#), or deploy a [ARM Template](#) instead of the portal.

Sign in to Azure

Sign in to the [Azure portal](#).

Virtual network and parameters

Before you deploy a VM and can use your NAT gateway, we need to create the resource group and virtual network.

In this section you'll need to replace the following parameters in the steps with the information below:

PARAMETER	VALUE
<resource-group-name>	myResourceGroupNAT
<virtual-network-name>	myVNet
<region-name>	East US 2
<IPv4-address-space>	192.168.0.0\16
<subnet-name>	mySubnet
<subnet-address-range>	192.168.0.0\24

Create the virtual network and subnet

In this section, you'll create a virtual network and subnet.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network** or search for **Virtual network** in the search box.
2. In **Create virtual network**, enter or select this information in the **Basics** tab:

SETTING	VALUE
Project Details	
Subscription	Select your Azure subscription

SETTING	VALUE
Resource Group	Select Create new, enter <resource-group-name>, then select OK, or select an existing <resource-group-name> based on parameters.
Instance details	
Name	Enter <virtual-network-name>
Region	Select <region-name>

3. Select the **IP Addresses** tab or select the **Next: IP Addresses** button at the bottom of the page.

4. In the **IP Addresses** tab, enter this information:

SETTING	VALUE
IPv4 address space	Enter <IPv4-address-space>

5. Under **Subnet name**, select the word **default**.

6. In **Edit subnet**, enter this information:

SETTING	VALUE
Subnet name	Enter <subnet-name>
Subnet address range	Enter <subnet-address-range>

7. Select **Save**.

8. Select the **Review + create** tab or select the **Review + create** button.

9. Select **Create**.

Create a VM to use the NAT gateway

We'll now create a VM to use the NAT service. This VM has a public IP to use as an instance-level Public IP to allow you to access the VM. NAT service is flow direction aware and will replace the default Internet destination in your subnet. The VM's public IP address won't be used for outbound connections.

1. On the upper-left side of the portal, select **Create a resource** > **Compute** > **Ubuntu Server 18.04 LTS**, or search for **Ubuntu Server 18.04 LTS** in the Marketplace search.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

- **Subscription > Resource Group:** Select **myResourceGroupNAT**.
- **Instance Details > Virtual machine name:** Type **myVM**.
- **Instance Details > Region:** select **East US 2**.
- **Administrator account > Authentication type:** Select **Password**.
- **Administrator account > Enter the Username, Password, and Confirm password** information.
- **Inbound port rules > Public inbound ports:** Select **Allow selected ports**.
- **Inbound port rules > Select inbound ports:** Select **SSH (22)**
- Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

3. In the **Networking** tab make sure the following are selected:

- **Virtual network:** myVnet
- **Subnet:** mySubnet
- **Public IP** > Select **Create new**. In the **Create public IP address** window, type **myPublicIPVM** in the **Name** field, and choose **Standard** for the **SKU**. Click **OK**.
- **NIC network security group:** Select **Basic**.
- **Public inbound ports:** Select **Allow selected ports**.
- **Select inbound ports:** Confirm **SSH** is selected.

4. In the **Management** tab, under **Monitoring**, set **Boot diagnostics** to **Off**.

5. Select **Review + create**.

6. Review the settings and click **Create**.

Create the NAT gateway

You can use one or more public IP address resources, public IP prefixes, or both. We'll add a public IP resource, public IP prefix, and a NAT gateway resource.

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a public IP address

1. On the upper-left side of the portal, select **Create a resource** > **Networking** > **Public IP address**, or search for **Public IP address** in the Marketplace search.

2. In **Create public IP address**, enter or select this information:

SETTING	VALUE
IP Version	Select IPv4 .
SKU	Select Standard .
Name	Enter myPublicIP .
Subscription	Select your subscription.
Resource group	Select myResourceGroupNAT .
Location	Select East US 2 .

3. Leave the rest of the defaults and select **Create**.

Create a public IP prefix

1. On the upper-left side of the portal, select **Create a resource** > **Networking** > **Public IP prefix**, or search for **Public IP prefix** in the Marketplace search.

2. In **Create a public IP prefix**, type or select the following values in the **Basics** tab:

- **Subscription > Resource Group:** Select **myResourceGroupNAT**>
- **Instance details > Name:** Type **myPublicIPprefix**.

- **Instance details > Region:** Select East US 2.
 - **Instance details > Prefix size:** Select /31 (2 addresses)
3. Leave the rest the defaults and select **Review + create**.
4. Review the settings, and then select **Create**.

Create a NAT gateway resource

1. On the upper-left side of the portal, select **Create a resource > Networking > NAT gateway**, or search for **NAT gateway** in the Marketplace search.
2. In **Create network address translation (NAT) gateway**, type or select the following values in the **Basics** tab:
 - **Subscription > Resource Group:** Select **myResourceGroupNAT**.
 - **Instance details > NAT gateway name:** Type **myNATgateway**.
 - **Instance details > Region:** Select **East US 2**.
 - **Instance details > Idle timeout (minutes):** Type **10**.
 - Select the **Public IP** tab, or select **Next: Public IP**.
3. In the **Public IP** tab, type or select the following values:
 - **Public IP addresses:** Select **myPublicIP**.
 - **Public IP Prefixes:** Select **myPublicIPprefix**.
 - Select the **Subnet** tab, or select **Next: Subnet**.
4. In the **Subnet** tab, type or select the following values:
 - **Virtual Network:** Select **myResourceGroupNAT > myVnet**.
 - **Subnet name:** Select the box next to **mySubnet**.
5. Select **Review + create**.
6. Review the settings, and then select **Create**.

Discover the IP address of the VM

1. On the left side of the portal, select **Resource groups**.
2. Select **myResourceGroupNAT**.
3. Select **myVM**.
4. In **Overview**, copy the **Public IP address** value, and paste into notepad so you can use it to access the VM.

IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it to access the VM.

Sign in to VM

Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <username>@<ip-address-destination>
```

You're now ready to use the NAT service.

Clean up resources

When no longer needed, delete the resource group, NAT gateway, and all related resources. Select the resource group **myResourceGroupNAT** that contains the NAT gateway, and then select **Delete**.

Next steps

In this tutorial, you created a NAT gateway and a VM to use it.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is addressed by adding additional public IP address resources or public IP prefix resources or both.

- Learn about [Azure Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).
- Quickstart for deploying [NAT gateway resource using Azure CLI](#).
- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

Tutorial: Create a NAT Gateway using the Azure portal and test the NAT service

11/2/2020 • 11 minutes to read • [Edit Online](#)

In this tutorial, you'll create a NAT gateway to provide outbound connectivity for virtual machines in Azure. To test the NAT gateway, you deploy a source and destination virtual machine. You'll test the NAT gateway by making outbound connections to a public IP address from the source to the destination virtual machine. This tutorial deploys source and destination in two different virtual networks in the same resource group for simplicity only.

If you prefer, you can do these steps using the [Azure CLI](#) or [Azure PowerShell](#) instead of the portal.

Sign in to Azure

Sign in to the [Azure portal](#).

Prepare the source for outbound traffic

We'll guide you through configuration of a full test environment and the execution of the tests itself in the next steps. We'll start with the source, which will use the NAT gateway resource we create in later steps.

Virtual network and parameters

Before you deploy a VM and can use your NAT gateway, we need to create the resource group and virtual network.

In this section you'll need to replace the following parameters in the steps with the information below:

PARAMETER	VALUE
<resource-group-name>	myResourceGroupNAT
<virtual-network-name>	myVNetsource
<region-name>	East US 2
<IPv4-address-space>	192.168.0.0/16
<subnet-name>	mySubnetsource
<subnet-address-range>	192.168.0.0/24

Create the virtual network and subnet

In this section, you'll create a virtual network and subnet.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network** or search for **Virtual network** in the search box.
2. In **Create virtual network**, enter or select this information in the **Basics** tab:

SETTING	VALUE
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select Create new , enter <resource-group-name>, then select OK, or select an existing <resource-group-name> based on parameters.
Instance details	
Name	Enter <virtual-network-name>
Region	Select <region-name>

3. Select the **IP Addresses** tab or select the **Next: IP Addresses** button at the bottom of the page.

4. In the **IP Addresses** tab, enter this information:

SETTING	VALUE
IPv4 address space	Enter <IPv4-address-space>

5. Under **Subnet name**, select the word **default**.

6. In **Edit subnet**, enter this information:

SETTING	VALUE
Subnet name	Enter <subnet-name>
Subnet address range	Enter <subnet-address-range>

7. Select **Save**.

8. Select the **Review + create** tab or select the **Review + create** button.

9. Select **Create**.

Create source virtual machine

We'll now create a VM to use the NAT service. This VM has a public IP to use as an instance-level Public IP to allow you to access the VM. NAT service is flow direction aware and will replace the default Internet destination in your subnet. The VM's public IP address won't be used for outbound connections.

To test the NAT gateway, we'll assign a public IP address resource as an instance-level Public IP to access this VM from the outside. This address is only used to access it for the test. We'll demonstrate how the NAT service takes precedence over other outbound options.

You could also create this VM without a public IP and create another VM to use as a jumpbox without a public IP as an exercise.

1. On the upper-left side of the portal, select **Create a resource** > **Compute** > **Ubuntu Server 18.04 LTS**, or search for **Ubuntu Server 18.04 LTS** in the Marketplace search.

2. In **Create a virtual machine**, enter or select the following values in the **Basics** tab:

- Subscription > Resource Group: Select myResourceGroupNAT.
 - Instance Details > Virtual machine name: enter myVMsource.
 - Instance Details > Region > select East US 2.
 - Administrator account > Authentication enter: Select Password.
 - Administrator account > Enter the Username, Password, and Confirm password information.
 - Inbound port rules > Public inbound ports: Select Allow selected ports.
 - Inbound port rules > Select inbound ports: Select SSH (22)
 - Select the Networking tab, or select Next: Disks, then Next: Networking.
3. In the Networking tab make sure the following are selected:
- Virtual network: myVnetsource
 - Subnet: mySubnetsource
 - Public IP > Select Create new. In the Create public IP address window, enter myPublicIPsourceVM in the Name field. Select Standard for the SKU. Leave the rest at the defaults and click OK.
 - NIC network security group: Select Basic.
 - Public inbound ports: Select Allow selected ports.
 - Select inbound ports: Confirm SSH is selected.
4. In the Management tab, under Monitoring, set Boot diagnostics to Off.
5. Select Review + create.
6. Review the settings and click Create.

Create the NAT Gateway

You can use one or more public IP address resources, public IP prefixes, or both with NAT gateway. We'll add a public IP resource, public IP prefix, and a NAT gateway resource.

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a public IP address

- On the upper-left side of the portal, select **Create a resource > Networking > Public IP address**, or search for **Public IP address** in the Marketplace search.
- In **Create public IP address**, enter or select this information:

SETTING	VALUE
IP Version	Select IPv4.
SKU	Select Standard.
Name	Enter myPublicIPsource.
Subscription	Select your subscription.
Resource group	Select myResourceGroupNAT.

SETTING	VALUE
Location	Select East US 2.

3. Leave the rest of the defaults and select **Create**.

Create a public IP prefix

1. On the upper-left side of the portal, select **Create a resource > Networking > Public IP prefix**, or search for **Public IP prefix** in the Marketplace search.
2. In **Create a public IP prefix**, enter or select the following values in the **Basics** tab:
 - **Subscription > Resource Group:** Select **myResourceGroupNAT**.
 - **Instance details > Name:** enter **myPublicIPprefixsource**.
 - **Instance details > Region:** Select **East US 2**.
 - **Instance details > Prefix size:** Select **/31 (2 addresses)**
3. Leave the rest the defaults and select **Review + create**.
4. Review the settings, and then select **Create**.

Create a NAT gateway resource

1. On the upper-left side of the portal, select **Create a resource > Networking > NAT gateway**, or search for **NAT gateway** in the Marketplace search.
2. In **Create network address translation (NAT) gateway**, enter or select the following values in the **Basics** tab:
 - **Subscription > Resource Group:** Select **myResourceGroupNAT**.
 - **Instance details > NAT gateway name:** enter **myNATgateway**.
 - **Instance details > Region:** Select **East US 2**.
 - **Instance details > Idle timeout (minutes):** enter **10**.
 - Select the **Public IP** tab, or select **Next: Public IP**.
3. In the **Public IP** tab, enter or select the following values:
 - **Public IP addresses:** Select **myPublicIPsource**.
 - **Public IP Prefixes:** Select **myPublicIPprefixsource**.
 - Select the **Subnet** tab, or select **Next: Subnet**.
4. In the **Subnet** tab, enter or select the following values:
 - **Virtual Network:** Select **myResourceGroupNAT > myVnetsource**.
 - **Subnet name:** Select the box next to **mySubnetsource**.
5. Select **Review + create**.
6. Review the settings, and then select **Create**.

All outbound traffic to Internet destinations is now using the NAT service. It isn't necessary to configure a UDR.

Prepare destination for outbound traffic

We'll now create a destination for the outbound traffic translated by the NAT service to allow you to test it.

Virtual network and parameters for destination

Before you deploy a VM for the destination, we need to create a virtual network where the destination virtual machine can reside. The following are the same steps as for the source VM with some small changes to expose

the destination endpoint.

In this section you'll need to replace the following parameters in the steps with the information below:

PARAMETER	VALUE
<resource-group-name>	myResourceGroupNAT
<virtual-network-name>	myVNetdestination
<region-name>	East US 2
<IPv4-address-space>	10.1.0.0/16
<subnet-name>	mySubnetdestination
<subnet-address-range>	10.1.0.0/24

Create the virtual network and subnet

In this section, you'll create a virtual network and subnet.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network** or search for **Virtual network** in the search box.
2. In **Create virtual network**, enter or select this information in the **Basics** tab:

SETTING	VALUE
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select Create new , enter <resource-group-name>, then select OK, or select an existing <resource-group-name> based on parameters.
Instance details	
Name	Enter <virtual-network-name>
Region	Select <region-name>

3. Select the **IP Addresses** tab or select the **Next: IP Addresses** button at the bottom of the page.

4. In the **IP Addresses** tab, enter this information:

SETTING	VALUE
IPv4 address space	Enter <IPv4-address-space>

5. Under **Subnet name**, select the word **default**.

6. In **Edit subnet**, enter this information:

SETTING	VALUE
Subnet name	Enter <subnet-name>
Subnet address range	Enter <subnet-address-range>

7. Select **Save**.
8. Select the **Review + create** tab or select the **Review + create** button.
9. Select **Create**.

Create destination virtual machine

1. On the upper-left side of the portal, select **Create a resource > Compute > Ubuntu Server 18.04 LTS**, or search for **Ubuntu Server 18.04 LTS** in the Marketplace search.
2. In **Create a virtual machine**, enter or select the following values in the **Basics** tab:
 - **Subscription > Resource Group:** Select **myResourceGroupNAT**.
 - **Instance Details > Virtual machine name:** enter **myVMdestination**.
 - **Instance Details > Region:** select **East US 2**.
 - **Administrator account > Authentication:** Select **Password**.
 - **Administrator account > Enter the Username, Password, and Confirm password information**.
 - **Inbound port rules > Public inbound ports:** Select **Allow selected ports**.
 - **Inbound port rules > Select inbound ports:** Select **SSH (22) and HTTP (80)**.
 - **Select the Networking tab**, or select **Next: Disks**, then **Next: Networking**.
3. In the **Networking** tab make sure the following are selected:
 - **Virtual network:** **myVnetdestination**
 - **Subnet:** **mySubnetdestination**
 - **Public IP > Select Create new.** In the **Create public IP address** window, enter **myPublicIPdestinationVM** in the **Name** field. Select **Standard** for **SKU**. Leave the rest at the defaults and click **OK**.
 - **NIC network security group:** Select **Basic**.
 - **Public inbound ports:** Select **Allow selected ports**.
 - **Select inbound ports:** Confirm **SSH** and **HTTP** is selected.
4. In the **Management** tab, under **Monitoring**, set **Boot diagnostics** to **Off**.
5. Select **Review + create**.
6. Review the settings, and then select **Create**.

Prepare a web server and test payload on destination VM

First we need to discover the IP address of the destination VM.

1. On the left side of the portal, select **Resource groups**.
2. Select **myResourceGroupNAT**.
3. Select **myVMdestination**.
4. In **Overview**, copy the **Public IP address** value, and paste into notepad so you can use it to access the VM.

IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the destination virtual machine.

Sign in to destination VM

Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <username>@<ip-address-destination>
```

Copy and paste the following commands once you've logged in.

```
sudo apt-get -y update && \
sudo apt-get -y upgrade && \
sudo apt-get -y dist-upgrade && \
sudo apt-get -y autoremove && \
sudo apt-get -y autoclean && \
sudo apt-get -y install nginx && \
sudo ln -sf /dev/null /var/log/nginx/access.log && \
sudo touch /var/www/html/index.html && \
sudo rm /var/www/html/index.nginx-debian.html && \
sudo dd if=/dev/zero of=/var/www/html/100k bs=1024 count=100
```

These commands will update your virtual machine, install nginx, and create a 100-KBytes file. This file will be retrieved from the source VM using the NAT service.

Close the SSH session with the destination VM.

Prepare test on source VM

First we need to discover the IP address of the source VM.

1. On the left side of the portal, select **Resource groups**.
2. Select **myResourceGroupNAT**.
3. Select **myVMsource**.
4. In **Overview**, copy the **Public IP address** value, and paste into notepad so you can use it to access the VM.

IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the source virtual machine.

Log into source VM

Open a new tab for [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <username>@<ip-address-source>
```

Copy and paste the following commands to prepare for testing the NAT service.

```
sudo apt-get -y update && \
sudo apt-get -y upgrade && \
sudo apt-get -y dist-upgrade && \
sudo apt-get -y autoremove && \
sudo apt-get autoclean && \
sudo apt-get install -y nload golang && \
echo 'export GOPATH=${HOME}/go' >> .bashrc && \
echo 'export PATH=${PATH}: ${GOPATH}/bin' >> .bashrc && \
. ~/.bashrc &&
go get -u github.com/rakyll/hey
```

This command will update your virtual machine, install go, install [hey](#) from GitHub, and update your shell environment.

You're now ready to test the NAT service.

Validate NAT service

While logged into the source VM, you can use `curl` and `hey` to generate requests to the destination IP address.

Use `curl` to retrieve the 100-KBytes file. Replace `<ip-address-destination>` in the example below with the destination IP address you have previously copied. The `--output` parameter indicates that the retrieved file will be discarded.

```
curl http://<ip-address-destination>/100k --output /dev/null
```

You can also generate a series of requests using `hey`. Again, replace `<ip-address-destination>` with the destination IP address you have previously copied.

```
hey -n 100 -c 10 -t 30 --disable-keepalive http://<ip-address-destination>/100k
```

This command will generate 100 requests, 10 concurrently, with a timeout of 30 seconds, and without reusing the TCP connection. Each request will retrieve 100 Kbytes. At the end of the run, `hey` will report some statistics about how well the NAT service did.

Clean up resources

When no longer needed, delete the resource group, NAT gateway, and all related resources. Select the resource group `myResourceGroupNAT` that contains the NAT gateway, and then select **Delete**.

Next steps

In this tutorial, you created a NAT gateway, created a source and destination VM, and then tested the NAT gateway.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is easily addressed by adding additional public IP address resources or public IP prefix resources or both.

- Learn about [Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).
- Quickstart for deploying [NAT gateway resource using Azure CLI](#).
- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

Azure CLI samples for virtual network

11/2/2020 • 2 minutes to read • [Edit Online](#)

The following table includes links to bash scripts with Azure CLI commands:

SCRIPT	DESCRIPTION
Create a virtual network for multi-tier applications	Creates a virtual network with front-end and back-end subnets. Traffic to the front-end subnet is limited to HTTP and SSH, while traffic to the back-end subnet is limited to MySQL, port 3306.
Peer two virtual networks	Creates and connects two virtual networks in the same region.
Route traffic through a network virtual appliance	Creates a virtual network with front-end and back-end subnets and a VM that is able to route traffic between the two subnets.
Filter inbound and outbound VM network traffic	Creates a virtual network with front-end and back-end subnets. Inbound network traffic to the front-end subnet is limited to HTTP, HTTPS, and SSH. Outbound traffic to the internet from the back-end subnet is not permitted.
Configure IPv4 + IPv6 dual stack virtual network with Basic Load Balancer	Deploys dual-stack (IPv4+IPv6) virtual network with two VMs and an Azure Basic Load Balancer with IPv4 and IPv6 public IP addresses.
Configure IPv4 + IPv6 dual stack virtual network with Standard Load Balancer	Deploys dual-stack (IPv4+IPv6) virtual network with two VMs and an Azure Standard Load Balancer with IPv4 and IPv6 public IP addresses.
Tutorial: Create and test a NAT gateway - Azure CLI	Create and validate a NAT gateway using a source and destination virtual machine.

Azure PowerShell samples for virtual network

11/2/2020 • 2 minutes to read • [Edit Online](#)

The following table includes links to Azure Powershell scripts:

SCRIPT	DESCRIPTION
Create a virtual network for multi-tier applications	Creates a virtual network with front-end and back-end subnets. Traffic to the front-end subnet is limited to HTTP, while traffic to the back-end subnet is limited to SQL, port 1433.
Peer two virtual networks	Creates and connects two virtual networks in the same region.
Route traffic through a network virtual appliance	Creates a virtual network with front-end and back-end subnets and a VM that is able to route traffic between the two subnets.
Filter inbound and outbound VM network traffic	Creates a virtual network with front-end and back-end subnets. Inbound network traffic to the front-end subnet is limited to HTTP and HTTPS. Outbound traffic to the internet from the back-end subnet is not permitted.
Configure IPv4 + IPv6 dual stack virtual network with Basic Load Balancer	Deploys dual-stack (IPv4+IPv6) virtual network with two VMs and an Azure Basic Load Balancer with IPv4 and IPv6 public IP addresses.
Configure IPv4 + IPv6 dual stack virtual network with Standard Load Balancer	Deploys dual-stack (IPv4+IPv6) virtual network with two VMs and an Azure Standard Load Balancer with IPv4 and IPv6 public IP addresses.

Azure Resource Manager template samples for virtual network

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following table includes links to Azure Resource Manager template samples. You can deploy templates using the Azure [portal](#), Azure [CLI](#), or Azure [PowerShell](#). To learn how to author your own templates, see [Create your first template](#) and [Understand the structure and syntax of Azure Resource Manager templates](#).

For the JSON syntax and properties to use in templates, see [Microsoft.Network resource types](#).

TASK	DESCRIPTION
Create a virtual network with two subnets	Creates a virtual network with two subnets.
Route traffic through a network virtual appliance	Creates a virtual network with three subnets. Deploys a virtual machine into each of the subnets. Creates a route table containing routes to direct traffic from one subnet to another through the virtual machine in the third subnet. Associates the route table to one of the subnets.
Create a virtual network service endpoint for Azure Storage	Creates a new virtual network with two subnets, and a network interface in each subnet. Enables a service endpoint to Azure Storage for one of the subnets and secures a new storage account to that subnet.
Connect two virtual networks	Creates two virtual networks and a virtual network peering between them.
Create a virtual machine with multiple IP addresses	Creates a Windows or Linux VM with multiple IP addresses.
Configure IPv4 + IPv6 dual stack virtual network	Deploys dual-stack (IPv4+IPv6) virtual network with two VMs and an Azure Basic Load Balancer with IPv4 and IPv6 public IP addresses.

Azure Virtual Network concepts and best practices

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes key concepts and best practices for Azure Virtual Network (VNet) .

VNet concepts

- **Address space:** When creating a VNet, you must specify a custom private IP address space using public and private (RFC 1918) addresses. Azure assigns resources in a virtual network a private IP address from the address space that you assign. For example, if you deploy a VM in a VNet with address space, 10.0.0.0/16, the VM will be assigned a private IP like 10.0.0.4.
- **Subnets:** Subnets enable you to segment the virtual network into one or more sub-networks and allocate a portion of the virtual network's address space to each subnet. You can then deploy Azure resources in a specific subnet. Just like in a traditional network, subnets allow you to segment your VNet address space into segments that are appropriate for the organization's internal network. This also improves address allocation efficiency. You can secure resources within subnets using Network Security Groups. For more information, see [Network security groups](#).
- **Regions:** VNet is scoped to a single region/location; however, multiple virtual networks from different regions can be connected together using Virtual Network Peering.
- **Subscription:** VNet is scoped to a subscription. You can implement multiple virtual networks within each Azure [subscription](#) and Azure [region](#).

Best practices

As you build your network in Azure, it is important to keep in mind the following universal design principles:

- Ensure non-overlapping address spaces. Make sure your VNet address space (CIDR block) does not overlap with your organization's other network ranges.
- Your subnets should not cover the entire address space of the VNet. Plan ahead and reserve some address space for the future.
- It is recommended you have fewer large VNets rather than multiple small VNets. This will prevent management overhead.
- Secure your VNet's by assigning Network Security Groups (NSGs) to the subnets beneath them.

Next steps

To get started using a virtual network, create one, deploy a few VMs to it, and communicate between the VMs. To learn how, see the [Create a virtual network quickstart](#).

Virtual Network – Business Continuity

7/14/2019 • 2 minutes to read • [Edit Online](#)

Overview

A Virtual Network (VNet) is a logical representation of your network in the cloud. It allows you to define your own private IP address space and segment the network into subnets. VNets serve as a trust boundary to host your compute resources such as Azure Virtual Machines and Cloud Services (web/worker roles). A VNet allows direct private IP communication between the resources hosted in it. You can link a virtual network to an on-premises network through a VPN Gateway, or ExpressRoute.

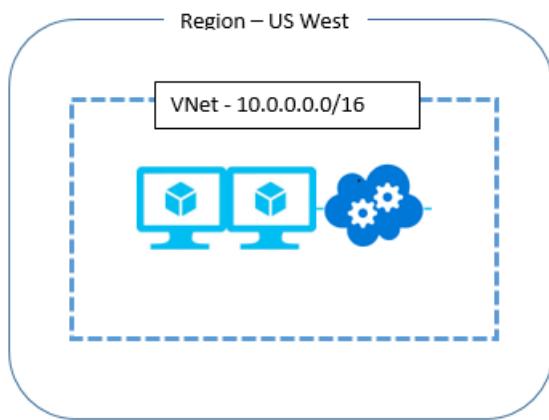
A VNet is created within the scope of a region. You can *create* VNets with same address space in two different regions (For example, US East and US West), but because they have the same address space, you can't connect them together.

Business Continuity

There could be several different ways that your application could be disrupted. A region could be completely cut off due to a natural disaster, or a partial disaster, due to a failure of multiple devices or services. The impact on the VNet service is different in each of these situations.

Q: If an outage occurs for an entire region, what do I do? For example, if a region is completely cut off due to a natural disaster? What happens to the virtual networks hosted in the region?

A: The virtual network and the resources in the affected region remains inaccessible during the time of the service disruption.



Q: What can I do re-create the same virtual network in a different region?

A: Virtual networks are fairly lightweight resources. You can invoke Azure APIs to create a VNet with the same address space in a different region. To recreate the same environment that was present in the affected region, you make API calls to redeploy the Cloud Services web and worker roles, and the virtual machines that you had. If you have on-premises connectivity, such as in a hybrid deployment, you have to deploy a new VPN Gateway, and connect to your on-premises network.

To create a virtual network, see [Create a virtual network](#).

Q: Can a replica of a VNet in a given region be re-created in another region ahead of time?

A: Yes, you can create two VNets using the same private IP address space and resources in two different regions ahead of time. If you are hosting internet-facing services in the VNet, you could have set up Traffic Manager to

geo-route traffic to the region that is active. However, you cannot connect two VNets with the same address space to your on-premises network, as it would cause routing issues. At the time of a disaster and loss of a VNet in one region, you can connect the other VNet in the available region, with the matching address space to your on-premises network.

To create a virtual network, see [Create a virtual network](#).

What is Virtual Network NAT?

11/2/2020 • 4 minutes to read • [Edit Online](#)

Virtual Network NAT (network address translation) simplifies outbound-only Internet connectivity for virtual networks. When configured on a subnet, all outbound connectivity uses your specified static public IP addresses. Outbound connectivity is possible without load balancer or public IP addresses directly attached to virtual machines. NAT is fully managed and highly resilient.

Figure: Virtual Network NAT

Static IP addresses for outbound-only

Outbound connectivity can be defined for each subnet with NAT. Multiple subnets within the same virtual network can have different NATs. A subnet is configured by specifying which NAT gateway resource to use. All UDP and TCP outbound flows from any virtual machine instance will use NAT.

NAT is compatible with standard SKU public IP address resources or public IP prefix resources or a combination of both. You can use a public IP prefix directly or distribute the public IP addresses of the prefix across multiple NAT gateway resources. NAT will groom all traffic to the range of IP addresses of the prefix. Any IP filtering of your deployments is now easy.

All outbound traffic for the subnet is processed by NAT automatically without any customer configuration. User-defined routes aren't necessary. NAT takes precedence over other outbound scenarios and replaces the default Internet destination of a subnet.

On-demand SNAT with multiple IP addresses for scale

NAT uses "port network address translation" (PNAT or PAT) and is recommended for most workloads. Dynamic or divergent workloads can be easily accommodated with on-demand outbound flow allocation. Extensive pre-planning, pre-allocation, and ultimately overprovisioning of outbound resources is avoided. SNAT port resources are shared and available across all subnets using a specific NAT gateway resource and are provided when needed.

A public IP address attached to NAT provides up to 64,000 concurrent flows for UDP and TCP respectively. You can start with a single IP address and scale up to 16 IP addresses using public IP addresses or public IP prefixes or both. A NAT gateway resource will use all IP addresses associated with the resource for outbound connections from all subnets configured with the same NAT gateway resource.

NAT allows flows to be created from the virtual network to the Internet. Return traffic from the Internet is only allowed in response to an active flow.

Unlike load balancer outbound SNAT, NAT has no restrictions on which private IP of a virtual machine instance can make outbound connections. Primary and secondary IP configurations can create outbound Internet connection with NAT.

Coexistence of inbound and outbound

NAT is compatible with the following standard SKU resources:

- Load balancer

- Public IP address
- Public IP prefix

When used together with NAT, these resources provide inbound Internet connectivity to your subnet(s). NAT provides all outbound Internet connectivity from your subnet(s).

NAT and compatible Standard SKU features are aware of the direction the flow was started. Inbound and outbound scenarios can coexist. These scenarios will receive the correct network address translations because these features are aware of the flow direction.

Figure: Virtual Network NAT flow direction

Fully managed, highly resilient

NAT is fully scaled out from the start. There's no ramp up or scale-out operation required. Azure manages the operation of NAT for you. NAT always has multiple fault domains and can sustain multiple failures without service outage.

TCP Reset for unrecognized flows

The private side of NAT sends TCP Reset packets for attempts to communicate on a TCP connection that doesn't exist. One example is connections that have reached idle timeout. The next packet received will return a TCP Reset to the private IP address to signal and force connection closure.

The public side of NAT doesn't generate TCP Reset packets or any other traffic. Only traffic produced by the customer's virtual network is emitted.

Configurable TCP idle timeout

A default TCP idle timeout of 4 minutes is used and can be increased to up to 120 minutes. Any activity on a flow can also reset the idle timer, including TCP keepalives.

Regional or zone isolation with availability zones

NAT is regional by default. When creating [availability zones](#) scenarios, NAT can be isolated in a specific zone (zonal deployment).

Figure: Virtual Network NAT with availability zones

Multi-dimensional metrics for observability

You can monitor the operation of your NAT through multi-dimensional metrics exposed in Azure Monitor. These metrics can be used to observe the usage and for troubleshooting. NAT gateway resources expose the following metrics:

- Bytes
- Packets
- Dropped Packets
- Total SNAT connections
- SNAT connection state transitions per interval.

SLA

At general availability, NAT data path is at least 99.9% available.

Pricing

For pricing details, see [Virtual Network pricing](#).

Availability

Virtual Network NAT and the NAT gateway resource are available in all regions of all Azure clouds [regions](#).

Suggestions

We want to know how we can improve the service. Propose and vote on what we should build next at [UserVoice for NAT](#).

Limitations

- NAT is compatible with standard SKU public IP, public IP prefix, and load balancer resources. Basic resources, such as basic load balancer, and any products derived from them aren't compatible with NAT. Basic resources must be placed on a subnet not configured with NAT.
- IPv4 address family is supported. NAT doesn't interact with IPv6 address family. NAT can't be deployed on a subnet with an IPv6 prefix.
- NAT can't span multiple virtual networks.

Next steps

- Learn about [NAT gateway resource](#).
- [Tell us what to build next for Virtual Network NAT in UserVoice](#).

Designing virtual networks with NAT gateway resources

2/18/2021 • 16 minutes to read • [Edit Online](#)

NAT gateway resources are part of [Virtual Network NAT](#) and provide outbound Internet connectivity for one or more subnets of a virtual network. The subnet of the virtual network states which NAT gateway will be used. NAT provides source network address translation (SNAT) for a subnet. NAT gateway resources specify which static IP addresses virtual machines use when creating outbound flows. Static IP addresses come from public IP address resources (PIP), public IP prefix resources, or both. If a public IP prefix resource is used, all IP addresses of the entire public IP prefix resource are consumed by a NAT gateway resource. A NAT gateway resource can use a total of up to 16 static IP addresses from either.

Figure: Virtual Network NAT for outbound to Internet

How to deploy NAT

Configuring and using NAT gateway is intentionally made simple:

NAT gateway resource:

- Create regional or zonal (zone-isolated) NAT gateway resource,
- Assign IP addresses,
- If necessary, modify TCP idle timeout (optional). Review [timers](#) [before](#) you change the default.

Virtual network:

- Configure virtual network subnet to use a NAT gateway.

User-defined routes aren't necessary.

Resource

The resource is designed to be simple as you can see from the following Azure Resource Manager example in a template-like format. This template-like format is shown here to illustrate the concepts and structure. Modify the example for your needs. This document isn't intended as a tutorial.

The following diagram shows the writeable references between the different Azure Resource Manager resources. The arrow indicates the direction of the reference, originating from where it's writeable. Review

Figure: Virtual Network NAT object model

NAT is recommended for most workloads unless you have a specific dependency on [pool-based Load Balancer outbound connectivity](#).

You can migrate from standard load balancer scenarios, including [outbound rules](#), to NAT gateway. To migrate, move the public ip and public ip prefix resources from load balancer frontends to NAT gateway. New IP addresses for NAT gateway aren't required. Standard public IP address resources and public IP prefix resource can be reused as long as the total doesn't exceed 16 IP addresses. Plan for migration with service interruption in mind during the transition. You can minimize the interruption by automating the process. Test the migration in a

staging environment first. During the transition, inbound originated flows aren't affected.

The following example is a snippet from an Azure Resource Manager template. This template deploys several resources, including a NAT gateway. The template has the following parameters in this example:

- **natgatewayname** - Name of the NAT gateway.
- **location** - Azure region where resource is located.
- **publicipname** - Name of the outbound public IP associated with the NAT gateway.
- **vnetname** - Name of the virtual network.
- **subnetname** - Name of the subnet associated with the NAT gateway.

The total number of IP addresses provided by all IP address and prefix resources can't exceed 16 IP addresses total. Any number of IP addresses between 1 and 16 is allowed.

```
{  
  "type": "Microsoft.Network/natGateways",  
  "apiVersion": "2019-11-01",  
  "name": "[parameters('natgatewayname')]",  
  "location": "[parameters('location')]",  
  "dependsOn": [  
    "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicipname'))]"  
,  
  "sku": {  
    "name": "Standard"  
  },  
  "properties": {  
    "idleTimeoutInMinutes": 4,  
    "publicIpAddresses": "[if(not(empty(parameters('publicipdns'))), variables('publicIpAddresses'),  
  json('null'))]"  
  }  
},
```

When the NAT gateway resource has been created, it can be used on one or more subnets of a virtual network. Specify which subnets use this NAT gateway resource. A NAT gateway isn't able to span more than one virtual network. It isn't required to assign the same NAT gateway to all subnets of a virtual network. Individual subnets can be configured with different NAT gateway resources.

Scenarios that don't use availability zones will be regional (no zone specified). If you're using availability zones, you can specify a zone to isolate NAT to a specific zone. Zone-redundancy isn't supported. Review [NAT availability zones](#).

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "vnetname": {  
      "defaultValue": "myVnet",  
      "type": "String",  
      "metadata": {  
        "description": "Name of the virtual network"  
      }  
    },  
    "subnetname": {  
      "defaultValue": "mySubnet",  
      "type": "String",  
      "metadata": {  
        "description": "Name of the subnet for virtual network"  
      }  
    },  
    "vnetaddresssspace": {  
      "defaultValue": "192.168.0.0/16",  
      "type": "String",  
      "metadata": {  
        "description": "Address space for the virtual network"  
      }  
    }  
  },  
  "variables": {},  
  "resources": [  
    {  
      "type": "Microsoft.Network/natGateways",  
      "apiVersion": "2019-11-01",  
      "name": "[parameters('natgatewayname')]",  
      "location": "[parameters('location')]",  
      "dependsOn": [  
        "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicipname'))]"  
,  
        "[resourceId('Microsoft.Network/virtualNetworks', parameters('vnetname'))]",  
        "[resourceId('Microsoft.Network/subnets', parameters('vnetname'), parameters('subnetname'))]"  
      ],  
      "sku": {  
        "name": "Standard"  
      },  
      "properties": {  
        "idleTimeoutInMinutes": 4,  
        "publicIpAddresses": "[if(not(empty(parameters('publicipdns'))), variables('publicIpAddresses'),  
          json('null'))]"  
      }  
    }  
  ],  
  "outputs": {}  
}
```

```

    },
    "metadata": {
        "description": "Address space for virtual network"
    }
},
"vnetsubnetprefix": {
    "defaultValue": "192.168.0.0/24",
    "type": "String",
    "metadata": {
        "description": "Subnet prefix for virtual network"
    }
},
"natgatewayname": {
    "defaultValue": "myNATgateway",
    "type": "String",
    "metadata": {
        "description": "Name of the NAT gateway resource"
    }
},
"publicipdns": {
    "defaultValue": "[concat('gw-', uniqueString(resourceGroup().id))]",
    "type": "String",
    "metadata": {
        "description": "dns of the public ip address, leave blank for no dns"
    }
},
"location": {
    "defaultValue": "[resourceGroup().location]",
    "type": "String",
    "metadata": {
        "description": "Location of resources"
    }
},
"variables": {
    "publicIpName": "[concat(parameters('natgatewayname'), 'ip')]",
    "publicIpAddresses": [
        {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicipname'))]"
        }
    ]
},
"resources": [
{
    "type": "Microsoft.Network/publicIPAddresses",
    "apiVersion": "2019-11-01",
    "name": "[variables('publicIpName')]",
    "location": "[parameters('location')]",
    "sku": {
        "name": "Standard"
    },
    "properties": {
        "publicIPAddressVersion": "IPv4",
        "publicIPAllocationMethod": "Static",
        "idleTimeoutInMinutes": 4,
        "dnsSettings": {
            "domainNameLabel": "[parameters('publicipdns')]"
        }
    }
},
{
    "type": "Microsoft.Network/natGateways",
    "apiVersion": "2019-11-01",
    "name": "[parameters('natgatewayname')]",
    "location": "[parameters('location')]",
    "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicipname'))]"
    ],
    "sku": {
        "name": "Standard"
    }
}
]
}

```

```

    "name": "Standard",
  },
  "properties": {
    "idleTimeoutInMinutes": 4,
    "publicIpAddresses": "[if(not(empty(parameters('publicipdns'))), variables('publicIpAddresses'), json('null'))]"
  }
},
{
  "type": "Microsoft.Network/virtualNetworks",
  "apiVersion": "2019-11-01",
  "name": "[parameters('vnetname')]",
  "location": "[parameters('location')]",
  "dependsOn": [
    "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
  ],
  "properties": {
    "addressSpace": {
      "addressPrefixes": [
        "[parameters('vnetaddressspace')]"
      ]
    },
    "subnets": [
      {
        "name": "[parameters('subnetname')]",
        "properties": {
          "addressPrefix": "[parameters('vnetsubnetprefix')]",
          "natGateway": {
            "id": "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
          },
          "privateEndpointNetworkPolicies": "Enabled",
          "privateLinkServiceNetworkPolicies": "Enabled"
        }
      }
    ],
    "enableDdosProtection": false,
    "enableVmProtection": false
  }
},
{
  "type": "Microsoft.Network/virtualNetworks/subnets",
  "apiVersion": "2019-11-01",
  "name": "[concat(parameters('vnetname'), '/mySubnet')]",
  "dependsOn": [
    "[resourceId('Microsoft.Network/virtualNetworks', parameters('vnetname'))]",
    "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
  ],
  "properties": {
    "addressPrefix": "[parameters('vnetsubnetprefix')]",
    "natGateway": {
      "id": "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
    },
    "privateEndpointNetworkPolicies": "Enabled",
    "privateLinkServiceNetworkPolicies": "Enabled"
  }
}
]
}

```

NAT gateways are defined with a property on a subnet within a virtual network. Flows created by virtual machines on subnet **subnetname** of virtual network **vnetname** will use the NAT gateway. All outbound connectivity will use the IP addresses associated with **natgatewayname** as the source IP address.

For more information on the Azure Resource Manager template used in this example, see:

- [Quickstart: Create a NAT gateway - Resource Manager template](#)
- [Virtual Network NAT](#)

Design Guidance

Review this section to familiarize yourself with considerations for designing virtual networks with NAT.

1. [Cost optimization](#)
2. [Coexistence of inbound and outbound](#)
3. [Managing Basic resources](#)
4. [Availability Zones](#)

Cost optimization

[Service endpoints](#) and [private link](#) are options to consider for optimizing cost. NAT isn't needed for these services. Traffic directed to service endpoints or private link is not processed by the virtual network's NAT.

Service endpoints tie Azure service resources to your virtual network and control access to your Azure service resources. For example, when you access Azure storage, use a service endpoint for storage to avoid data processed NAT charges. Service endpoints are free.

Private link exposes Azure PaaS service (or other services hosted with private link) as a private endpoint inside a virtual network. Private link is billed based on duration and data processed.

Evaluate if either or both of these approaches are a good fit for your scenario and use as needed.

Coexistence of inbound and outbound

NAT gateway is compatible with:

- Standard load balancer
- Standard public IP
- Standard public IP prefix

When developing a new deployment, start with standard SKUs.

Figure: Virtual Network NAT for outbound to Internet

The Internet outbound only scenario provided by NAT gateway can be expanded with inbound from Internet functionality. Each resource is aware of the direction in which a flow is originated. On a subnet with a NAT gateway, all outbound to Internet scenarios are superseded by the NAT gateway. Inbound from Internet scenarios are provided by the respective resource.

NAT and VM with instance-level Public IP

Figure: Virtual Network NAT and VM with instance-level Public IP

DIRECTION	RESOURCE
Inbound	VM with instance-level Public IP
Outbound	NAT gateway

VM will use NAT gateway for outbound. Inbound originated isn't affected.

NAT and VM with public Load Balancer

Figure: Virtual Network NAT and VM with public Load Balancer

DIRECTION	RESOURCE
Inbound	public Load Balancer
Outbound	NAT gateway

Any outbound configuration from a load-balancing rule or outbound rules is superseded by NAT gateway. Inbound originated isn't affected.

NAT and VM with instance-level public IP and public Load Balancer

Figure: Virtual Network NAT and VM with instance-level public IP and public Load Balancer

DIRECTION	RESOURCE
Inbound	VM with instance-level public IP and public Load Balancer
Outbound	NAT gateway

Any outbound configuration from a load-balancing rule or outbound rules is superseded by NAT gateway. The VM will also use NAT gateway for outbound. Inbound originated isn't affected.

Managing Basic resources

Standard load balancer, public IP, and public IP prefix are compatible with NAT gateway. NAT gateways operate in the scope of a subnet. The basic SKU of these services must be deployed on a subnet without a NAT gateway. This separation allows both SKU variants to coexist in the same virtual network.

NAT gateways take precedence over outbound scenarios of the subnet. Basic load balancer or public IP (and any managed service built with them) is unable to be adjusted with the correct translations. NAT gateway takes control over outbound to Internet traffic on a subnet. Inbound traffic to basic load balancer and public ip is unavailable. Inbound traffic to a basic load balancer and, or a public ip configured on a VM won't be available.

Availability Zones

Zone isolation with zonal stacks

Figure: Virtual Network NAT with zone isolation, creating multiple "zonal stacks"

Even without availability zones, NAT is resilient and can survive multiple infrastructure component failures. Availability zones build on this resiliency with zone isolation scenarios for NAT.

Virtual networks and their subnets are regional constructs. Subnets aren't restricted to a zone.

A zonal promise for zone isolation exists when a virtual machine instance using a NAT gateway resource is in the same zone as the NAT gateway resource and its public IP addresses. The pattern you want to use for zone isolation is creating a "zonal stack" per availability zone. This "zonal stack" consists of virtual machine instances, NAT gateway resources, public IP address and/or prefix resources on a subnet that is assumed to be serving only the same zone. The control plane operations and data plane are then aligned with and constrained to the specified zone.

Failure in a zone other than where your scenario exists is expected to be without impact to NAT. Outbound traffic from virtual machines in the same zone will fail because of zone isolation.

Integrating inbound endpoints

If your scenario requires inbound endpoints, you have two options:

OPTION	PATTERN	EXAMPLE	PRO	CON
(1)	Align the inbound endpoints with the respective zonal stacks you're creating for outbound.	Create a standard load balancer with zonal frontend.	Same health model and failure mode for inbound and outbound. Simpler to operate.	Individual IP addresses per zone may need to be masked by a common DNS name.
(2)	Overlay the zonal stacks with a cross-zone inbound endpoint.	Create a standard load balancer with zone-redundant frontend.	Single IP address for inbound endpoint.	Varying health model and failure modes for inbound and outbound. More complex to operate.

NOTE

A zone-isolated NAT gateway requires IP addresses to match the zone of the NAT gateway. NAT gateway resources with IP addresses from a different zone or without a zone aren't allowed.

Cross-zone outbound scenarios not supported

Figure: Virtual Network NAT not compatible with zone-spanning subnet

You can't achieve a zonal promise with NAT gateway resources when virtual machine instances are deployed in multiple zones within the same subnet. And even if there were multiple zonal NAT gateways attached to a subnet, the virtual machine instance wouldn't know which NAT gateway resource to select.

A zonal promise doesn't exist when a) the zone of a virtual machine instance and the zone of a zonal NAT gateway are not aligned, or b) a regional NAT gateway resource is used with zonal virtual machine instances.

While the scenario will appear to work, its health model and failure mode is undefined from an availability zone point of view. Consider going with zonal stacks or all regional instead.

NOTE

The zones property of a NAT gateway resource isn't mutable. Redeploy NAT gateway resource with the intended regional or zone preference.

NOTE

IP addresses by themselves aren't zone-redundant if no zone is specified. The frontend of a [Standard Load Balancer is zone-redundant](#) if an IP address isn't created in a specific zone. This doesn't apply to NAT. Only regional or zone-isolation is supported.

Performance

Each NAT gateway resource can provide up to 50 Gbps of throughput. You can split your deployments into multiple subnets and assign each subnet or groups of subnets a NAT gateway to scale out.

Each NAT gateway can support 64,000 flows for TCP and UDP respectively per assigned outbound IP address. Review the following section on Source Network Address Translation (SNAT) for details as well as the [troubleshooting article](#) for specific problem resolution guidance.

Source Network Address Translation

Source network address translation (SNAT) rewrites the source of a flow to originate from a different IP address. NAT gateway resources use a variant of SNAT commonly referred to port address translation (PAT). PAT rewrites the source address and source port. With SNAT, there's no fixed relationship between the number of private addresses and their translated public addresses.

Fundamentals

Let's look at an example of four flows to explain the basic concept. The NAT gateway is using public IP address resource 65.52.1.1 and the VM is making connections to 65.52.0.1.

FLOW	SOURCE TUPLE	DESTINATION TUPLE
1	192.168.0.16:4283	65.52.0.1:80
2	192.168.0.16:4284	65.52.0.1:80
3	192.168.0.17.5768	65.52.0.1:80

These flows might look like this after PAT has taken place:

FLOW	SOURCE TUPLE	SNAT'ED SOURCE TUPLE	DESTINATION TUPLE
1	192.168.0.16:4283	65.52.1.1:1234	65.52.0.1:80
2	192.168.0.16:4284	65.52.1.1:1235	65.52.0.1:80
3	192.168.0.17.5768	65.52.1.1:1236	65.52.0.1:80

The destination will see the source of the flow as 65.52.0.1 (SNAT source tuple) with the assigned port shown. PAT as shown in the preceding table is also called port masquerading SNAT. Multiple private sources are masqueraded behind an IP and port.

source (SNAT) port reuse

NAT gateways opportunistically reuse source (SNAT) ports. The following illustrates this concept as an additional flow for the preceding set of flows. The VM in the example is a flow to 65.52.0.2.

FLOW	SOURCE TUPLE	DESTINATION TUPLE
4	192.168.0.16:4285	65.52.0.2:80

A NAT gateway will likely translate flow 4 to a port that may be used for other destinations as well. See [Scaling](#) for additional discussion on correctly sizing your IP address provisioning.

FLOW	SOURCE TUPLE	SNAT'ED SOURCE TUPLE	DESTINATION TUPLE
4	192.168.0.16:4285	65.52.1.1:1234	65.52.0.2:80

Don't take a dependency on the specific way source ports are assigned in the above example. The preceding is an illustration of the fundamental concept only.

SNAT provided by NAT is different from [Load Balancer](#) in several aspects.

On-demand

NAT provides on-demand SNAT ports for new outbound traffic flows. All available SNAT ports in inventory are

used by any virtual machine on subnets configured with NAT.

Figure: Virtual Network NAT on-demand outbound SNAT

Any IP configuration of a virtual machine can create outbound flows on-demand as needed. Pre-allocation, per instance planning including per instance worst case overprovisioning, isn't required.

Figure: Differences in exhaustion scenarios

Once a SNAT port releases, it's available for use by any virtual machine on subnets configured with NAT. On-demand allocation allows dynamic and divergent workloads on subnet(s) to use SNAT ports as they need. As long as there's SNAT port inventory available, SNAT flows will succeed. SNAT port hot spots benefit from the larger inventory instead. SNAT ports aren't left unused for virtual machines not actively needing them.

Scaling

Scaling NAT is primarily a function of managing the shared, available SNAT port inventory. NAT needs sufficient SNAT port inventory for expected peak outbound flows for all subnets attached to a NAT gateway resource. You can use public IP address resources, public IP prefix resources, or both to create SNAT port inventory.

NOTE

If you are assigning a public IP prefix resource, the entire public IP prefix will be used. You can't assign a public IP prefix resource and then break out individual IP addresses to assign to other resources. If you want to assign individual IP addresses from a public IP prefix to multiple resources, you need to create individual public IP addresses from the public IP prefix resource and assign them as needed instead of the public IP prefix resource itself.

SNAT maps private addresses to one or more public IP addresses, rewriting source address and source port in the process. A NAT gateway resource will use 64,000 ports (SNAT ports) per configured public IP address for this translation. NAT gateway resources can scale up to 16 IP addresses and 1M SNAT ports. If a public IP prefix resource is provided, each IP address within the prefix is providing SNAT port inventory. And adding more public IP addresses increases the available inventory SNAT ports. TCP and UDP are separate SNAT port inventories and unrelated.

NAT gateway resources opportunistically reuse source (SNAT) ports. As design guidance for scaling purposes, you should assume each flow requires a new SNAT port and scale the total number of available IP addresses for outbound traffic. You should carefully consider the scale you are designing for and provision IP addresses quantities accordingly.

SNAT ports to different destinations are most likely to be reused when possible. And as SNAT port exhaustion approaches, flows may not succeed.

See [SNAT fundamentals](#) for example.

Protocols

NAT gateway resources interact with IP and IP transport headers of UDP and TCP flows and are agnostic to application layer payloads. Other IP protocols aren't supported.

Timers

IMPORTANT

Long idle timer can unnecessarily increase likelihood of SNAT exhaustion. The longer of a timer you specify, the longer NAT will hold on to SNAT ports until they eventually idle timeout. If your flows are idle timed out, they will fail eventually anyway and unnecessarily consume SNAT port inventory. Flows that fail at 2 hours would have failed at the default 4 minutes as well. Increasing the idle timeout is a last resort option that should be used sparingly. If a flow never does go idle, it will not be impacted by the idle timer.

TCP idle timeout can be adjusted from 4 minutes (default) to 120 minutes (2 hours) for all flows. Additionally, you can reset the idle timer with traffic on the flow. A recommended pattern for refreshing long idle connections and endpoint liveness detection is TCP keepalives. TCP keepalives appear as duplicate ACKs to the endpoints, are low overhead, and invisible to the application layer.

The following timers are used for SNAT port release:

TIMER	VALUE
TCP FIN	60 seconds
TCP RST	10 seconds
TCP half open	30 seconds

A SNAT port is available for reuse to the same destination IP address and destination port after 5 seconds.

NOTE

These timer settings are subject to change. The values are provided to help troubleshooting and you shouldn't take a dependency on specific timers at this time.

Limitations

- NAT is compatible with standard SKU public IP, public IP prefix, and load balancer resources. Basic resources (for example basic load balancer) and any products derived from them aren't compatible with NAT. Basic resources must be placed on a subnet not configured with NAT.
- IPv4 address family is supported. NAT doesn't interact with IPv6 address family. NAT can't be deployed on a subnet with an IPv6 prefix.
- NAT can't span multiple virtual networks.
- IP fragmentation is not supported.

Suggestions

We want to know how we can improve the service. Are missing a capability? Make your case for what we should build next at [UserVoice for NAT](#).

Next steps

- Learn about [virtual network NAT](#).
- Learn about [metrics and alerts for NAT gateway resources](#).
- Learn about [troubleshooting NAT gateway resources](#).
- Tutorial for validating NAT Gateway
 - [Azure CLI](#)

- [PowerShell](#)
- [Portal](#)
- Quickstart for deploying a NAT gateway resource
 - [Azure CLI](#)
 - [PowerShell](#)
 - [Portal](#)
 - [Template](#)
- Learn about NAT gateway resource API
 - [REST API](#)
 - [Azure CLI](#)
 - [PowerShell](#)
- Learn about [availability zones](#).
- Learn about [standard load balancer](#).
- Learn about [availability zones and standard load balancer](#).
- [Tell us what to build next for Virtual Network NAT in UserVoice](#).

Azure Virtual Network NAT metrics

11/2/2020 • 2 minutes to read • [Edit Online](#)

Azure Virtual Network NAT gateway resources provide multi-dimensional metrics. You can use these metrics to observe the operation and for [troubleshooting](#). Alerts can be configured for critical issues such as SNAT exhaustion.

Figure: Virtual Network NAT for outbound to Internet

Metrics

NAT gateway resources provide the following multi-dimensional metrics in Azure Monitor:

METRIC	DESCRIPTION	RECOMMENDED AGGREGATION	DIMENSIONS
Bytes	Bytes processed inbound and outbound	Sum	Direction (In; Out), Protocol (6 TCP; 17 UDP)
Packets	Packets processed inbound and outbound	Sum	Direction (In; Out), Protocol (6 TCP; 17 UDP)
Dropped packets	Packets dropped by the NAT gateway	Sum	/
SNAT Connection Count	State transitions per interval	Sum	Connection State, Protocol (6 TCP; 17 UDP)
Total SNAT connection count	Current active SNAT connections (~ SNAT ports in use)	Sum	Protocol (6 TCP; 17 UDP)

Alerts

Alerts for metrics can be configured in Azure Monitor for each of the preceding [metrics](#).

Limitations

Resource Health isn't supported.

Next steps

- Learn about [Virtual Network NAT](#)
- Learn about [NAT gateway resource](#)
- Learn about [Azure Monitor](#)
- Learn about [troubleshooting NAT gateway resources](#).
- [Tell us what to build next for Virtual Network NAT in UserVoice](#).

Troubleshoot Azure Virtual Network NAT connectivity

2/18/2021 • 10 minutes to read • [Edit Online](#)

This article helps administrators diagnose and resolve connectivity problems when using Virtual Network NAT.

Problems

- [SNAT exhaustion](#)
- [ICMP ping is failing](#)
- [Connectivity failures](#)
- [IPv6 coexistence](#)
- [Connection doesn't originate from NAT gateway IP\(s\)](#)

To resolve these problems, follow the steps in the following section.

Resolution

SNAT exhaustion

A single [NAT gateway resource](#) supports from 64,000 up to 1 million concurrent flows. Each IP address provides 64,000 SNAT ports to the available inventory. You can use up to 16 IP addresses per NAT gateway resource. The SNAT mechanism is described [here](#) in more detail.

Frequently the root cause of SNAT exhaustion is an anti-pattern for how outbound connectivity is established, managed, or configurable timers changed from their default values. Review this section carefully.

Steps

1. Check if you have modified the default idle timeout to a value higher than 4 minutes.
2. Investigate how your application is creating outbound connectivity (for example, code review or packet capture).
3. Determine if this activity is expected behavior or whether the application is misbehaving. Use [metrics](#) in Azure Monitor to substantiate your findings. Use "Failed" category for SNAT Connections metric.
4. Evaluate if appropriate patterns are followed.
5. Evaluate if SNAT port exhaustion should be mitigated with additional IP addresses assigned to NAT gateway resource.

Design patterns

Always take advantage of connection reuse and connection pooling whenever possible. These patterns will avoid resource exhaustion problems and result in predictable behavior. Primitives for these patterns can be found in many development libraries and frameworks.

Solution: Use appropriate patterns and best practices

- NAT gateway resources have a default TCP idle timeout of 4 minutes. If this setting is changed to a higher value, NAT will hold on to flows longer and can cause [unnecessary pressure on SNAT port inventory](#).
- Atomic requests (one request per connection) are a poor design choice. Such anti-pattern limits scale, reduces performance, and decreases reliability. Instead, reuse HTTP/S connections to reduce the numbers of connections and associated SNAT ports. The application scale will increase and performance improve due to reduced handshakes, overhead, and cryptographic operation cost when using TLS.

- DNS can introduce many individual flows at volume when the client is not caching the DNS resolvers result. Use caching.
- UDP flows (for example DNS lookups) allocate SNAT ports for the duration of the idle timeout. The longer the idle timeout, the higher the pressure on SNAT ports. Use short idle timeout (for example 4 minutes).
- Use connection pools to shape your connection volume.
- Never silently abandon a TCP flow and rely on TCP timers to clean up flow. If you don't let TCP explicitly close the connection, state remains allocated at intermediate systems and endpoints and makes SNAT ports unavailable for other connections. This pattern can trigger application failures and SNAT exhaustion.
- Don't change OS-level TCP close related timer values without expert knowledge of impact. While the TCP stack will recover, your application performance can be negatively impacted when the endpoints of a connection have mismatched expectations. The desire to change timers is usually a sign of an underlying design problem. Review following recommendations.

SNAT exhaustion can also be amplified with other anti-patterns in the underlying application. Review these additional patterns and best practices to improve the scale and reliability of your service.

- Explore impact of reducing [TCP idle timeout](#) to lower values including default idle timeout of 4 minutes to free up SNAT port inventory earlier.
- Consider [asynchronous polling patterns](#) for long-running operations to free up connection resources for other operations.
- Long-lived flows (for example reused TCP connections) should use TCP keepalives or application layer keepalives to avoid intermediate systems timing out. Increasing the idle timeout is a last resort and may not resolve the root cause. A long timeout can cause low rate failures when timeout expires and introduce delay and unnecessary failures.
- Graceful [retry patterns](#) should be used to avoid aggressive retries/bursts during transient failure or failure recovery. Creating a new TCP connection for every HTTP operation (also known as "atomic connections") is an anti-pattern. Atomic connections will prevent your application from scaling well and waste resources. Always pipeline multiple operations into the same connection. Your application will benefit in transaction speed and resource costs. When your application uses transport layer encryption (for example TLS), there's a significant cost associated with the processing of new connections. Review [Azure Cloud Design Patterns](#) for additional best practice patterns.

Additional possible mitigations

Solution: Scale outbound connectivity as follows:

SCENARIO	EVIDENCE	MITIGATION
You're experiencing contention for SNAT ports and SNAT port exhaustion during periods of high usage.	"Failed" category for SNAT Connections metric in Azure Monitor shows transient or persistent failures over time and high connection volume.	Determine if you can add additional public IP address resources or public IP prefix resources. This addition will allow for up to 16 IP addresses in total to your NAT gateway. This addition will provide more inventory for available SNAT ports (64,000 per IP address) and allow you to scale your scenario further.
You've already given 16 IP addresses and still are experiencing SNAT port exhaustion.	Attempt to add additional IP address fails. Total number of IP addresses from public IP address resources or public IP prefix resources exceeds a total of 16.	Distribute your application environment across multiple subnets and provide a NAT gateway resource for each subnet. Reevaluate your design pattern(s) to optimize based on preceding guidance .

NOTE

It is important to understand why SNAT exhaustion occurs. Make sure you are using the right patterns for scalable and reliable scenarios. Adding more SNAT ports to a scenario without understanding the cause of the demand should be a last resort. If you do not understand why your scenario is applying pressure on SNAT port inventory, adding more SNAT ports to the inventory by adding more IP addresses will only delay the same exhaustion failure as your application scales. You may be masking other inefficiencies and anti-patterns.

ICMP ping is failing

[Virtual Network NAT](#) supports IPv4 UDP and TCP protocols. ICMP isn't supported and expected to fail.

Solution: Instead, use TCP connection tests (for example "TCP ping") and UDP-specific application layer tests to validate end to end connectivity.

The following table can be used a starting point for which tools to use to start tests.

OPERATING SYSTEM	GENERIC TCP CONNECTION TEST	TCP APPLICATION LAYER TEST	UDP
Linux	nc (generic connection test)	curl (TCP application layer test)	application specific
Windows	PsPing	PowerShell Invoke-WebRequest	application specific

Connectivity failures

Connectivity issues with [Virtual Network NAT](#) can be caused by several different issues:

- permanent failures due to configuration mistakes,
- transient or persistent [SNAT exhaustion](#) of the NAT gateway,
- transient failures in the Azure infrastructure,
- transient failures in the path between Azure and the public Internet destination,
- transient or persistent failures at the public Internet destination.

Use tools like the following to validation connectivity. [ICMP ping isn't supported](#).

OPERATING SYSTEM	GENERIC TCP CONNECTION TEST	TCP APPLICATION LAYER TEST	UDP
Linux	nc (generic connection test)	curl (TCP application layer test)	application specific
Windows	PsPing	PowerShell Invoke-WebRequest	application specific

Configuration

Check your configuration:

1. Does the NAT gateway resource have at least one public IP resource or one public IP prefix resource? You must at least have one IP address associated with the NAT gateway for it to be able to provide outbound connectivity.
2. Is the virtual network's subnet configured to use the NAT gateway?
3. Are you using UDR (user-defined route) and are you overriding the destination? NAT gateway resources become the default route (0/0) on configured subnets.

SNAT exhaustion

Review section on [SNAT exhaustion](#) in this article.

Azure infrastructure

Azure monitors and operates its infrastructure with great care. Transient failures can occur, there's no guarantee that transmissions are lossless. Use design patterns that allow for SYN retransmissions for TCP applications. Use connection timeouts large enough to permit TCP SYN retransmission to reduce transient impacts caused by a lost SYN packet.

Solution:

- Check for [SNAT exhaustion](#).
- The configuration parameter in a TCP stack that controls the SYN retransmission behavior is called RTO ([Retransmission Time-Out](#)). The RTO value is adjustable but typically 1 second or higher by default with exponential back-off. If your application's connection time-out is too short (for example 1 second), you may see sporadic connection timeouts. Increase the application connection time-out.
- If you observe longer, unexpected timeouts with default application behaviors, open a support case for further troubleshooting.

We don't recommend artificially reducing the TCP connection timeout or tuning the RTO parameter.

Public Internet transit

The chances of transient failures increases with a longer path to the destination and more intermediate systems. It's expected that transient failures can increase in frequency over [Azure infrastructure](#).

Follow the same guidance as preceding [Azure infrastructure](#) section.

Internet endpoint

The previous sections apply, along with the Internet endpoint that communication is established with. Other factors that can impact connectivity success are:

- traffic management on destination side, including
- API rate limiting imposed by the destination side
- Volumetric DDoS mitigations or transport layer traffic shaping
- firewall or other components at the destination

Usually packet captures at the source and the destination (if available) are required to determine what is taking place.

Solution:

- Check for [SNAT exhaustion](#).
- Validate connectivity to an endpoint in the same region or elsewhere for comparison.
- If you're creating high volume or transaction rate testing, explore if reducing the rate reduces the occurrence of failures.
- If changing rate impacts the rate of failures, check if API rate limits or other constraints on the destination side might have been reached.
- If your investigation is inconclusive, open a support case for further troubleshooting.

TCP Resets received

The NAT gateway generates TCP resets on the source VM for traffic that isn't recognized as in progress.

One possible reason is the TCP connection has idle timed out. You can adjust the idle timeout from 4 minutes to up to 120 minutes.

TCP Resets aren't generated on the public side of NAT gateway resources. TCP resets on the destination side are

generated by the source VM, not the NAT gateway resource.

Solution:

- Review [design patterns](#) recommendations.
- Open a support case for further troubleshooting if necessary.

IPv6 coexistence

[Virtual Network NAT](#) supports IPv4 UDP and TCP protocols and deployment on a [subnet with an IPv6 prefix isn't supported](#).

Solution: Deploy NAT gateway on a subnet without IPv6 prefix.

You can indicate interest in additional capabilities through [Virtual Network NAT UserVoice](#).

Connection doesn't originate from NAT gateway IP(s)

You configure NAT gateway, IP address(es) to use, and which subnet should use a NAT gateway resource. However, connections from virtual machine instances that existed before the NAT gateway was deployed don't use the IP address(es). They appear to be using IP address(es) not used with the NAT gateway resource.

Solution:

[Virtual Network NAT](#) replaces the outbound connectivity for the subnet it is configured on. When transitioning from default SNAT or load balancer outbound SNAT to using NAT gateways, new connections will immediately begin using the IP address(es) associated with the NAT gateway resource. However, if a virtual machine still has an established connection during the switch to NAT gateway resource, the connection will continue using the old SNAT IP address that was assigned when the connection was established. Make sure you are really establishing a new connection rather than reusing a connection that already existed because the OS or the browser was caching the connections in a connection pool. For example, when using `curl` in PowerShell, make sure to specify the `-DisableKeepalive` parameter to force a new connection. If you're using a browser, connections may also be pooled.

It's not necessary to reboot a virtual machine configuring a subnet for a NAT gateway resource. However, if a virtual machine is rebooted, the connection state is flushed. When the connection state has been flushed, all connections will begin using the NAT gateway resource's IP address(es). However, this is a side effect of the virtual machine being rebooted and not an indicator that a reboot is required.

If you are still having trouble, open a support case for further troubleshooting.

Next steps

- Learn about [Virtual Network NAT](#)
- Learn about [NAT gateway resource](#)
- Learn about [metrics and alerts for NAT gateway resources](#).
- [Tell us what to build next for Virtual Network NAT in UserVoice](#).

Virtual network traffic routing

2/18/2021 • 24 minutes to read • [Edit Online](#)

Learn about how Azure routes traffic between Azure, on-premises, and Internet resources. Azure automatically creates a route table for each subnet within an Azure virtual network and adds system default routes to the table. To learn more about virtual networks and subnets, see [Virtual network overview](#). You can override some of Azure's system routes with [custom routes](#), and add additional custom routes to route tables. Azure routes outbound traffic from a subnet based on the routes in a subnet's route table.

System routes

Azure automatically creates system routes and assigns the routes to each subnet in a virtual network. You can't create system routes, nor can you remove system routes, but you can override some system routes with [custom routes](#). Azure creates default system routes for each subnet, and adds additional [optional default routes](#) to specific subnets, or every subnet, when you use specific Azure capabilities.

Default

Each route contains an address prefix and next hop type. When traffic leaving a subnet is sent to an IP address within the address prefix of a route, the route that contains the prefix is the route Azure uses. Learn more about [how Azure selects a route](#) when multiple routes contain the same prefixes, or overlapping prefixes. Whenever a virtual network is created, Azure automatically creates the following default system routes for each subnet within the virtual network:

SOURCE	ADDRESS PREFIXES	NEXT HOP TYPE
Default	Unique to the virtual network	Virtual network
Default	0.0.0.0/0	Internet
Default	10.0.0.0/8	None
Default	192.168.0.0/16	None
Default	100.64.0.0/10	None

The next hop types listed in the previous table represent how Azure routes traffic destined for the address prefix listed. Explanations for the next hop types follow:

- **Virtual network:** Routes traffic between address ranges within the [address space](#) of a virtual network. Azure creates a route with an address prefix that corresponds to each address range defined within the address space of a virtual network. If the virtual network address space has multiple address ranges defined, Azure creates an individual route for each address range. Azure automatically routes traffic between subnets using the routes created for each address range. You don't need to define gateways for Azure to route traffic between subnets. Though a virtual network contains subnets, and each subnet has a defined address range, Azure does *not* create default routes for subnet address ranges, because each subnet address range is within an address range of the address space of a virtual network.
- **Internet:** Routes traffic specified by the address prefix to the Internet. The system default route specifies the 0.0.0.0/0 address prefix. If you don't override Azure's default routes, Azure routes traffic for any address not specified by an address range within a virtual network, to the Internet, with one exception. If

the destination address is for one of Azure's services, Azure routes the traffic directly to the service over Azure's backbone network, rather than routing the traffic to the Internet. Traffic between Azure services does not traverse the Internet, regardless of which Azure region the virtual network exists in, or which Azure region an instance of the Azure service is deployed in. You can override Azure's default system route for the 0.0.0.0/0 address prefix with a [custom route](#).

- **None:** Traffic routed to the **None** next hop type is dropped, rather than routed outside the subnet. Azure automatically creates default routes for the following address prefixes:

- **10.0.0.0/8 and 192.168.0.0/16:** Reserved for private use in RFC 1918.
- **100.64.0.0/10:** Reserved in RFC 6598.

If you assign any of the previous address ranges within the address space of a virtual network, Azure automatically changes the next hop type for the route from **None** to **Virtual network**. If you assign an address range to the address space of a virtual network that includes, but isn't the same as, one of the four reserved address prefixes, Azure removes the route for the prefix and adds a route for the address prefix you added, with **Virtual network** as the next hop type.

Optional default routes

Azure adds additional default system routes for different Azure capabilities, but only if you enable the capabilities. Depending on the capability, Azure adds optional default routes to either specific subnets within the virtual network, or to all subnets within a virtual network. The additional system routes and next hop types that Azure may add when you enable different capabilities are:

SOURCE	ADDRESS PREFIXES	NEXT HOP TYPE	SUBNET WITHIN VIRTUAL NETWORK THAT ROUTE IS ADDED TO
Default	Unique to the virtual network, for example: 10.1.0.0/16	VNet peering	All
Virtual network gateway	Prefixes advertised from on-premises via BGP, or configured in the local network gateway	Virtual network gateway	All
Default	Multiple	VirtualNetworkServiceEndpoint	Only the subnet a service endpoint is enabled for.

- **Virtual network (VNet) peering:** When you create a virtual network peering between two virtual networks, a route is added for each address range within the address space of each virtual network a peering is created for. Learn more about [virtual network peering](#).
- **Virtual network gateway:** One or more routes with *Virtual network gateway* listed as the next hop type are added when a virtual network gateway is added to a virtual network. The source is also *virtual network gateway*, because the gateway adds the routes to the subnet. If your on-premises network gateway exchanges border gateway protocol ([BGP](#)) routes with an Azure virtual network gateway, a route is added for each route propagated from the on-premises network gateway. It's recommended that you summarize on-premises routes to the largest address ranges possible, so the fewest number of routes are propagated to an Azure virtual network gateway. There are limits to the number of routes you can propagate to an Azure virtual network gateway. For details, see [Azure limits](#).
- **VirtualNetworkServiceEndpoint:** The public IP addresses for certain services are added to the route table by Azure when you enable a service endpoint to the service. Service endpoints are enabled for individual subnets within a virtual network, so the route is only added to the route table of a subnet a service endpoint is enabled for. The public IP addresses of Azure services change periodically. Azure

manages the addresses in the route table automatically when the addresses change. Learn more about [virtual network service endpoints](#), and the services you can create service endpoints for.

NOTE

The **VNet peering** and **VirtualNetworkServiceEndpoint** next hop types are only added to route tables of subnets within virtual networks created through the Azure Resource Manager deployment model. The next hop types are not added to route tables that are associated to virtual network subnets created through the classic deployment model. Learn more about Azure [deployment models](#).

Custom routes

You create custom routes by either creating [user-defined](#) routes, or by exchanging [border gateway protocol](#) (BGP) routes between your on-premises network gateway and an Azure virtual network gateway.

User-defined

You can create custom, or user-defined(static), routes in Azure to override Azure's default system routes, or to add additional routes to a subnet's route table. In Azure, you create a route table, then associate the route table to zero or more virtual network subnets. Each subnet can have zero or one route table associated to it. To learn about the maximum number of routes you can add to a route table and the maximum number of user-defined route tables you can create per Azure subscription, see [Azure limits](#). If you create a route table and associate it to a subnet, the routes within it are combined with, or override, the default routes Azure adds to a subnet by default.

You can specify the following next hop types when creating a user-defined route:

- **Virtual appliance:** A virtual appliance is a virtual machine that typically runs a network application, such as a firewall. To learn about a variety of pre-configured network virtual appliances you can deploy in a virtual network, see the [Azure Marketplace](#). When you create a route with the **virtual appliance** hop type, you also specify a next hop IP address. The IP address can be:
 - The [private IP address](#) of a network interface attached to a virtual machine. Any network interface attached to a virtual machine that forwards network traffic to an address other than its own must have the Azure *Enable IP forwarding* option enabled for it. The setting disables Azure's check of the source and destination for a network interface. Learn more about how to [enable IP forwarding for a network interface](#). Though *Enable IP forwarding* is an Azure setting, you may also need to enable IP forwarding within the virtual machine's operating system for the appliance to forward traffic between private IP addresses assigned to Azure network interfaces. If the appliance must route traffic to a public IP address, it must either proxy the traffic, or network address translate the private IP address of the source's private IP address to its own private IP address, which Azure then network address translates to a public IP address, before sending the traffic to the Internet. To determine required settings within the virtual machine, see the documentation for your operating system or network application. To understand outbound connections in Azure, see [Understanding outbound connections](#).

NOTE

Deploy a virtual appliance into a different subnet than the resources that route through the virtual appliance are deployed in. Deploying the virtual appliance to the same subnet, then applying a route table to the subnet that routes traffic through the virtual appliance, can result in routing loops, where traffic never leaves the subnet.

- The private IP address of an Azure [internal load balancer](#). A load balancer is often used as part of a [high availability strategy for network virtual appliances](#).

You can define a route with 0.0.0.0/0 as the address prefix and a next hop type of virtual appliance, enabling the appliance to inspect the traffic and determine whether to forward or drop the traffic. If you intend to create a user-defined route that contains the 0.0.0.0/0 address prefix, read [0.0.0.0/0 address prefix](#) first.

- **Virtual network gateway:** Specify when you want traffic destined for specific address prefixes routed to a virtual network gateway. The virtual network gateway must be created with type **VPN**. You cannot specify a virtual network gateway created as type **ExpressRoute** in a user-defined route because with ExpressRoute, you must use BGP for custom routes. You can define a route that directs traffic destined for the 0.0.0.0/0 address prefix to a [route-based](#) virtual network gateway. On your premises, you might have a device that inspects the traffic and determines whether to forward or drop the traffic. If you intend to create a user-defined route for the 0.0.0.0/0 address prefix, read [0.0.0.0/0 address prefix](#) first. Instead of configuring a user-defined route for the 0.0.0.0/0 address prefix, you can advertise a route with the 0.0.0.0/0 prefix via BGP, if you've [enabled BGP for a VPN virtual network gateway](#).
- **None:** Specify when you want to drop traffic to an address prefix, rather than forwarding the traffic to a destination. If you haven't fully configured a capability, Azure may list *None* for some of the optional system routes. For example, if you see *None* listed as the **Next hop IP address** with a **Next hop type** of *Virtual network gateway* or *Virtual appliance*, it may be because the device isn't running, or isn't fully configured. Azure creates system [default routes](#) for reserved address prefixes with **None** as the next hop type.
- **Virtual network:** Specify when you want to override the default routing within a virtual network. See [Routing example](#), for an example of why you might create a route with the **Virtual network** hop type.
- **Internet:** Specify when you want to explicitly route traffic destined to an address prefix to the Internet, or if you want traffic destined for Azure services with public IP addresses kept within the Azure backbone network.

You cannot specify **VNet peering** or **VirtualNetworkServiceEndpoint** as the next hop type in user-defined routes. Routes with the **VNet peering** or **VirtualNetworkServiceEndpoint** next hop types are only created by Azure, when you configure a virtual network peering, or a service endpoint.

Next hop types across Azure tools

The name displayed and referenced for next hop types is different between the Azure portal and command-line tools, and the Azure Resource Manager and classic deployment models. The following table lists the names used to refer to each next hop type with the different tools and [deployment models](#):

NEXT HOP TYPE	AZURE CLI AND POWERSHELL (RESOURCE MANAGER)	AZURE CLASSIC CLI AND POWERSHELL (CLASSIC)
Virtual network gateway	VirtualNetworkGateway	VPNGateway
Virtual network	VNetLocal	VNETLocal (not available in the classic CLI in asm mode)
Internet	Internet	Internet (not available in the classic CLI in asm mode)
Virtual appliance	VirtualAppliance	VirtualAppliance
None	None	Null (not available in the classic CLI in asm mode)

NEXT HOP TYPE	AZURE CLI AND POWERSHELL (RESOURCE MANAGER)	AZURE CLASSIC CLI AND POWERSHELL (CLASSIC)
Virtual network peering	VNet peering	Not applicable
Virtual network service endpoint	VirtualNetworkServiceEndpoint	Not applicable

Border gateway protocol

An on-premises network gateway can exchange routes with an Azure virtual network gateway using the border gateway protocol (BGP). Using BGP with an Azure virtual network gateway is dependent on the type you selected when you created the gateway. If the type you selected were:

- **ExpressRoute**: You must use BGP to advertise on-premises routes to the Microsoft Edge router. You cannot create user-defined routes to force traffic to the ExpressRoute virtual network gateway if you deploy a virtual network gateway deployed as type: ExpressRoute. You can use user-defined routes for forcing traffic from the Express Route to, for example, a Network Virtual Appliance.
- **VPN**: You can, optionally use BGP. For details, see [BGP with site-to-site VPN connections](#).

When you exchange routes with Azure using BGP, a separate route is added to the route table of all subnets in a virtual network for each advertised prefix. The route is added with *Virtual network gateway* listed as the source and next hop type.

ER and VPN Gateway route propagation can be disabled on a subnet using a property on a route table. When you exchange routes with Azure using BGP, routes are not added to the route table of all subnets with Virtual network gateway route propagation disabled. Connectivity with VPN connections is achieved using [custom routes](#) with a next hop type of *Virtual network gateway*. **Route propagation should not be disabled on the GatewaySubnet. The gateway will not function with this setting disabled.** For details, see [How to disable Virtual network gateway route propagation](#).

How Azure selects a route

When outbound traffic is sent from a subnet, Azure selects a route based on the destination IP address, using the longest prefix match algorithm. For example, a route table has two routes: One route specifies the 10.0.0.0/24 address prefix, while the other route specifies the 10.0.0.0/16 address prefix. Azure routes traffic destined for 10.0.0.5, to the next hop type specified in the route with the 10.0.0.0/24 address prefix, because 10.0.0.0/24 is a longer prefix than 10.0.0.0/16, even though 10.0.0.5 is within both address prefixes. Azure routes traffic destined to 10.0.1.5, to the next hop type specified in the route with the 10.0.0.0/16 address prefix, because 10.0.1.5 isn't included in the 10.0.0.0/24 address prefix, therefore the route with the 10.0.0.0/16 address prefix is the longest prefix that matches.

If multiple routes contain the same address prefix, Azure selects the route type, based on the following priority:

1. User-defined route
2. BGP route
3. System route

NOTE

System routes for traffic related to virtual network, virtual network peerings, or virtual network service endpoints, are preferred routes, even if BGP routes are more specific.

For example, a route table contains the following routes:

SOURCE	ADDRESS PREFIXES	NEXT HOP TYPE
Default	0.0.0.0/0	Internet
User	0.0.0.0/0	Virtual network gateway

When traffic is destined for an IP address outside the address prefixes of any other routes in the route table, Azure selects the route with the **User** source, because user-defined routes are higher priority than system default routes.

See [Routing example](#) for a comprehensive routing table with explanations of the routes in the table.

0.0.0.0/0 address prefix

A route with the 0.0.0.0/0 address prefix instructs Azure how to route traffic destined for an IP address that is not within the address prefix of any other route in a subnet's route table. When a subnet is created, Azure creates a **default** route to the 0.0.0.0/0 address prefix, with the **Internet** next hop type. If you don't override this route, Azure routes all traffic destined to IP addresses not included in the address prefix of any other route, to the Internet. The exception is that traffic to the public IP addresses of Azure services remains on the Azure backbone network, and is not routed to the Internet. If you override this route, with a **custom** route, traffic destined to addresses not within the address prefixes of any other route in the route table is sent to a network virtual appliance or virtual network gateway, depending on which you specify in a custom route.

When you override the 0.0.0.0/0 address prefix, in addition to outbound traffic from the subnet flowing through the virtual network gateway or virtual appliance, the following changes occur with Azure's default routing:

- Azure sends all traffic to the next hop type specified in the route, including traffic destined for public IP addresses of Azure services. When the next hop type for the route with the 0.0.0.0/0 address prefix is **Internet**, traffic from the subnet destined to the public IP addresses of Azure services never leaves Azure's backbone network, regardless of the Azure region the virtual network or Azure service resource exist in. When you create a user-defined or BGP route with a **Virtual network gateway** or **Virtual appliance** next hop type however, all traffic, including traffic sent to public IP addresses of Azure services you haven't enabled **service endpoints** for, is sent to the next hop type specified in the route. If you've enabled a service endpoint for a service, traffic to the service is not routed to the next hop type in a route with the 0.0.0.0/0 address prefix, because address prefixes for the service are specified in the route that Azure creates when you enable the service endpoint, and the address prefixes for the service are longer than 0.0.0.0/0.
- You are no longer able to directly access resources in the subnet from the Internet. You can indirectly access resources in the subnet from the Internet, if inbound traffic passes through the device specified by the next hop type for a route with the 0.0.0.0/0 address prefix before reaching the resource in the virtual network. If the route contains the following values for next hop type:
 - **Virtual appliance**: The appliance must:
 - Be accessible from the Internet
 - Have a public IP address assigned to it,
 - Not have a network security group rule associated to it that prevents communication to the device
 - Not deny the communication
 - Be able to network address translate and forward, or proxy the traffic to the destination resource in the subnet, and return the traffic back to the Internet.
 - **Virtual network gateway**: If the gateway is an ExpressRoute virtual network gateway, an Internet-connected device on-premises can network address translate and forward, or proxy the

traffic to the destination resource in the subnet, via ExpressRoute's [private peering](#).

If your virtual network is connected to an Azure VPN gateway, do not associate a route table to the [gateway subnet](#) that includes a route with a destination of 0.0.0.0/0. Doing so can prevent the gateway from functioning properly. For details, see the *Why are certain ports opened on my VPN gateway?* question in the [VPN Gateway FAQ](#).

See [DMZ between Azure and your on-premises datacenter](#) for implementation details when using virtual network gateways between the Internet and Azure.

Routing example

To illustrate the concepts in this article, the sections that follow describe:

- A scenario, with requirements
- The custom routes necessary to meet the requirements
- The route table that exists for one subnet that includes the default and custom routes necessary to meet the requirements

NOTE

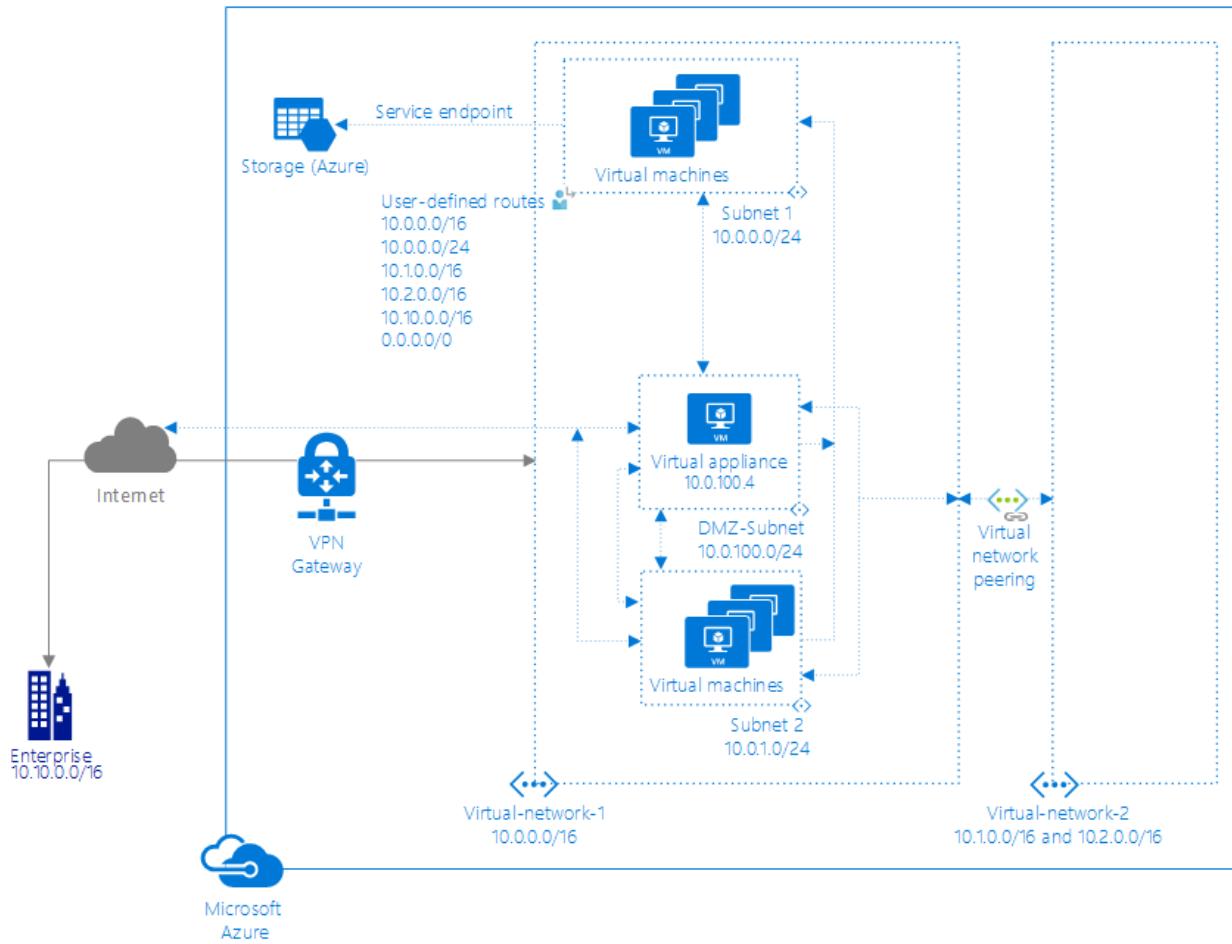
This example is not intended to be a recommended or best practice implementation. Rather, it is provided only to illustrate concepts in this article.

Requirements

1. Implement two virtual networks in the same Azure region and enable resources to communicate between the virtual networks.
2. Enable an on-premises network to communicate securely with both virtual networks through a VPN tunnel over the Internet. *Alternatively, an ExpressRoute connection could be used, but in this example, a VPN connection is used.*
3. For one subnet in one virtual network:
 - Force all outbound traffic from the subnet, except to Azure Storage and within the subnet, to flow through a network virtual appliance, for inspection and logging.
 - Do not inspect traffic between private IP addresses within the subnet; allow traffic to flow directly between all resources.
 - Drop any outbound traffic destined for the other virtual network.
 - Enable outbound traffic to Azure storage to flow directly to storage, without forcing it through a network virtual appliance.
4. Allow all traffic between all other subnets and virtual networks.

Implementation

The following picture shows an implementation through the Azure Resource Manager deployment model that meets the previous requirements:



Arrows show the flow of traffic.

Route tables

Subnet1

The route table for **Subnet1** in the picture contains the following routes:

ID	SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP IP ADDRESS	USER-DEFINED ROUTE NAME
1	Default	Invalid	10.0.0.0/16	Virtual network		
2	User	Active	10.0.0.0/16	Virtual appliance	10.0.100.4	Within-VNet1
3	User	Active	10.0.0.0/24	Virtual network		Within-Subnet1
4	Default	Invalid	10.1.0.0/16	VNet peering		
5	Default	Invalid	10.2.0.0/16	VNet peering		
6	User	Active	10.1.0.0/16	None		ToVNet2-1-Drop
7	User	Active	10.2.0.0/16	None		ToVNet2-2-Drop

ID	SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP IP ADDRESS	USER-DEFINED ROUTE NAME
8	Default	Invalid	10.10.0.0/16	Virtual network gateway	[X.X.X.X]	
9	User	Active	10.10.0.0/16	Virtual appliance	10.0.100.4	To-On-Prem
10	Default	Active	[X.X.X.X]	VirtualNetworkServiceEndpoint		
11	Default	Invalid	0.0.0.0/0	Internet		
12	User	Active	0.0.0.0/0	Virtual appliance	10.0.100.4	Default-NVA

An explanation of each route ID follows:

1. Azure automatically added this route for all subnets within *Virtual-network-1*, because 10.0.0.0/16 is the only address range defined in the address space for the virtual network. If the user-defined route in route ID2 weren't created, traffic sent to any address between 10.0.0.1 and 10.0.255.254 would be routed within the virtual network, because the prefix is longer than 0.0.0.0/0, and not within the address prefixes of any of the other routes. Azure automatically changed the state from *Active* to *Invalid*, when ID2, a user-defined route, was added, since it has the same prefix as the default route, and user-defined routes override default routes. The state of this route is still *Active* for *Subnet2*, because the route table that user-defined route, ID2 is in, isn't associated to *Subnet2*.
2. Azure added this route when a user-defined route for the 10.0.0.0/16 address prefix was associated to the *Subnet1* subnet in the *Virtual-network-1* virtual network. The user-defined route specifies 10.0.100.4 as the IP address of the virtual appliance, because the address is the private IP address assigned to the virtual appliance virtual machine. The route table this route exists in is not associated to *Subnet2*, so doesn't appear in the route table for *Subnet2*. This route overrides the default route for the 10.0.0.0/16 prefix (ID1), which automatically routed traffic addressed to 10.0.0.1 and 10.0.255.254 within the virtual network through the virtual network next hop type. This route exists to meet [requirement 3](#), to force all outbound traffic through a virtual appliance.
3. Azure added this route when a user-defined route for the 10.0.0.0/24 address prefix was associated to the *Subnet1* subnet. Traffic destined for addresses between 10.0.0.1 and 10.0.0.254 remains within the subnet, rather than being routed to the virtual appliance specified in the previous rule (ID2), because it has a longer prefix than the ID2 route. This route was not associated to *Subnet2*, so the route does not appear in the route table for *Subnet2*. This route effectively overrides the ID2 route for traffic within *Subnet1*. This route exists to meet [requirement 3](#).
4. Azure automatically added the routes in IDs 4 and 5 for all subnets within *Virtual-network-1*, when the virtual network was peered with *Virtual-network-2*. *Virtual-network-2* has two address ranges in its address space: 10.1.0.0/16 and 10.2.0.0/16, so Azure added a route for each range. If the user-defined routes in route IDs 6 and 7 weren't created, traffic sent to any address between 10.1.0.1-10.1.255.254 and 10.2.0.1-10.2.255.254 would be routed to the peered virtual network, because the prefix is longer than 0.0.0.0/0, and not within the address prefixes of any of the other routes. Azure automatically changed the state from *Active* to *Invalid*, when the routes in IDs 6 and 7 were added, since they have the same prefixes as the routes in IDs 4 and 5, and user-defined routes override default routes. The state of the routes in IDs 4 and 5 are still *Active* for *Subnet2*, because the route table that the user-defined routes in IDs 6 and 7 are in, isn't associated to *Subnet2*. A virtual network peering was created to meet [requirement 1](#).

5. Same explanation as ID4.
6. Azure added this route and the route in ID7, when user-defined routes for the 10.1.0.0/16 and 10.2.0.0/16 address prefixes were associated to the *Subnet1* subnet. Traffic destined for addresses between 10.1.0.1-10.1.255.254 and 10.2.0.1-10.2.255.254 is dropped by Azure, rather than being routed to the peered virtual network, because user-defined routes override default routes. The routes are not associated to *Subnet2*, so the routes do not appear in the route table for *Subnet2*. The routes override the ID4 and ID5 routes for traffic leaving *Subnet1*. The ID6 and ID7 routes exist to meet [requirement 3](#) to drop traffic destined to the other virtual network.
7. Same explanation as ID6.
8. Azure automatically added this route for all subnets within *Virtual-network-1* when a VPN type virtual network gateway was created within the virtual network. Azure added the public IP address of the virtual network gateway to the route table. Traffic sent to any address between 10.10.0.1 and 10.10.255.254 is routed to the virtual network gateway. The prefix is longer than 0.0.0.0/0 and not within the address prefixes of any of the other routes. A virtual network gateway was created to meet [requirement 2](#).
9. Azure added this route when a user-defined route for the 10.10.0.0/16 address prefix was added to the route table associated to *Subnet1*. This route overrides ID8. The route sends all traffic destined for the on-premises network to an NVA for inspection, rather than routing traffic directly on-premises. This route was created to meet [requirement 3](#).
10. Azure automatically added this route to the subnet when a service endpoint to an Azure service was enabled for the subnet. Azure routes traffic from the subnet to a public IP address of the service, over the Azure infrastructure network. The prefix is longer than 0.0.0.0/0 and not within the address prefixes of any of the other routes. A service endpoint was created to meet [requirement 3](#), to enable traffic destined for Azure Storage to flow directly to Azure Storage.
11. Azure automatically added this route to the route table of all subnets within *Virtual-network-1* and *Virtual-network-2*. The 0.0.0.0/0 address prefix is the shortest prefix. Any traffic sent to addresses within a longer address prefix are routed based on other routes. By default, Azure routes all traffic destined for addresses other than the addresses specified in one of the other routes to the Internet. Azure automatically changed the state from *Active* to *Invalid* for the *Subnet1* subnet when a user-defined route for the 0.0.0.0/0 address prefix (ID12) was associated to the subnet. The state of this route is still *Active* for all other subnets within both virtual networks, because the route isn't associated to any other subnets within any other virtual networks.
12. Azure added this route when a user-defined route for the 0.0.0.0/0 address prefix was associated to the *Subnet1* subnet. The user-defined route specifies 10.0.100.4 as the IP address of the virtual appliance. This route is not associated to *Subnet2*, so the route does not appear in the route table for *Subnet2*. All traffic for any address not included in the address prefixes of any of the other routes is sent to the virtual appliance. The addition of this route changed the state of the default route for the 0.0.0.0/0 address prefix (ID11) from *Active* to *Invalid* for *Subnet1*, because a user-defined route overrides a default route. This route exists to meet the third [requirement](#).

Subnet2

The route table for *Subnet2* in the picture contains the following routes:

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP IP ADDRESS
Default	Active	10.0.0.0/16	Virtual network	
Default	Active	10.1.0.0/16	VNet peering	
Default	Active	10.2.0.0/16	VNet peering	
Default	Active	10.10.0.0/16	Virtual network gateway	[X.X.X.X]

Source	State	Address Prefixes	Next Hop Type	Next Hop IP Address
Default	Active	0.0.0.0/0	Internet	
Default	Active	10.0.0.0/8	None	
Default	Active	100.64.0.0/10	None	
Default	Active	192.168.0.0/16	None	

The route table for *Subnet2* contains all Azure-created default routes and the optional VNet peering and Virtual network gateway optional routes. Azure added the optional routes to all subnets in the virtual network when the gateway and peering were added to the virtual network. Azure removed the routes for the 10.0.0.0/8, 192.168.0.0/16, and 100.64.0.0/10 address prefixes from the *Subnet1* route table when the user-defined route for the 0.0.0.0/0 address prefix was added to *Subnet1*.

Next steps

- [Create a user-defined route table with routes and a network virtual appliance](#)
- [Configure BGP for an Azure VPN Gateway](#)
- [Use BGP with ExpressRoute](#)
- [View all routes for a subnet](#). A user-defined route table only shows you the user-defined routes, not the default, and BGP routes for a subnet. Viewing all routes shows you the default, BGP, and user-defined routes for the subnet a network interface is in.
- [Determine the next hop type](#) between a virtual machine and a destination IP address. The Azure Network Watcher next hop feature enables you to determine whether traffic is leaving a subnet and being routed to where you think it should be.

Interoperability in Azure : Test setup

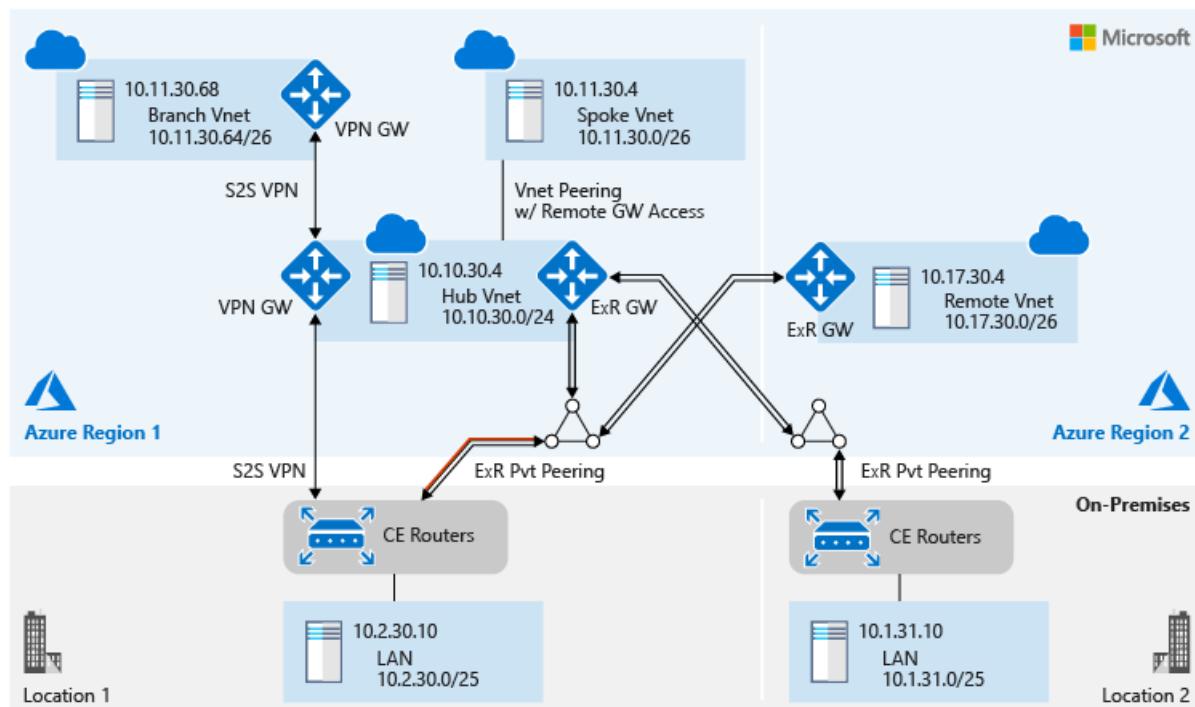
2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes a test setup you can use to analyze how Azure networking services interoperate at the control plane level and data plane level. Let's look briefly at the Azure networking components:

- **Azure ExpressRoute:** Use private peering in Azure ExpressRoute to directly connect private IP spaces in your on-premises network to your Azure Virtual Network deployments. ExpressRoute can help you achieve higher bandwidth and a private connection. Many ExpressRoute eco partners offer ExpressRoute connectivity with SLAs. To learn more about ExpressRoute and to learn how to configure ExpressRoute, see [Introduction to ExpressRoute](#).
- **Site-to-site VPN:** You can use Azure VPN Gateway as a site-to-site VPN to securely connect an on-premises network to Azure over the internet or by using ExpressRoute. To learn how to configure a site-to-site VPN to connect to Azure, see [Configure VPN Gateway](#).
- **VNet peering:** Use virtual network (VNet) peering to establish connectivity between VNets in Azure Virtual Network. To learn more about VNet peering, see the [tutorial on VNet peering](#).

Test setup

The following figure illustrates the test setup:



The centerpiece of the test setup is the hub VNet in Azure Region 1. The hub VNet is connected to different networks in the following ways:

- The hub VNet is connected to the spoke VNet by using VNet peering. The spoke VNet has remote access to both gateways in the hub VNet.
- The hub VNet is connected to the branch VNet by using site-to-site VPN. The connectivity uses eBGP to exchange routes.
- The hub VNet is connected to the on-premises Location 1 network by using ExpressRoute private peering as the primary path. It uses site-to-site VPN connectivity as the backup path. In the rest of this article, we refer to this ExpressRoute circuit as ExpressRoute 1. By default, ExpressRoute circuits provide redundant

connectivity for high availability. On ExpressRoute 1, the secondary customer edge (CE) router's subinterface that faces the secondary Microsoft Enterprise Edge Router (MSEE) is disabled. A red line over the double-line arrow in the preceding figure represents the disabled CE router subinterface.

- The hub VNet is connected to the on-premises Location 2 network by using another ExpressRoute private peering. In the rest of this article, we refer to this second ExpressRoute circuit as ExpressRoute 2.
- ExpressRoute 1 also connects both the hub VNet and the on-premises Location 1 network to a remote VNet in Azure Region 2.

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [configuration details](#) for the test topology.

Learn about [control plane analysis](#) of the test setup and the views of different VNets or VLANs in the topology.

Learn about the [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure back-end connectivity features: Test configuration details

2/18/2021 • 6 minutes to read • [Edit Online](#)

This article describes the configuration details of the [test setup](#). The test setup helps you analyze how Azure networking services interoperate at the control plane level and data plane level.

Spoke VNet connectivity by using VNet peering

The following figure shows the Azure Virtual Network peering details of a spoke virtual network (VNet). To learn how to set up peering between two VNets, see [Manage VNet peering](#). If you want the spoke VNet to use the gateways that are connected to the hub VNet, select **Use remote gateways**.

The screenshot shows the configuration details for a VNet peering named "Spoke01-VNet01-peering".

Name: Spoke01-VNet01-peering

Peering status: Connected

Provisioning state: Succeeded

Peer details:

- Address space:** 10.10.30.0/24
- Virtual network:** VNet01

Configuration:

- Allow virtual network access:** Enabled
- Allow forwarded traffic:** Checked
- Allow gateway transit:** Unchecked
- Use remote gateways:** Checked

The following figure shows the VNet peering details of the hub VNet. If you want the hub VNet to permit the spoke VNet to use the hub's gateways, select **Allow gateway transit**.

VNet01-Spoke01-Peering

VNet01

Save Discard Delete

Name
VNet01-Spoke01-Peering

Peering status
Connected

Provisioning state
Succeeded

Peer details

Address space
10.11.30.0/26

Virtual network
Spoke01-VNet

Configuration

Allow virtual network access ⓘ

Disabled Enabled

Allow forwarded traffic ⓘ

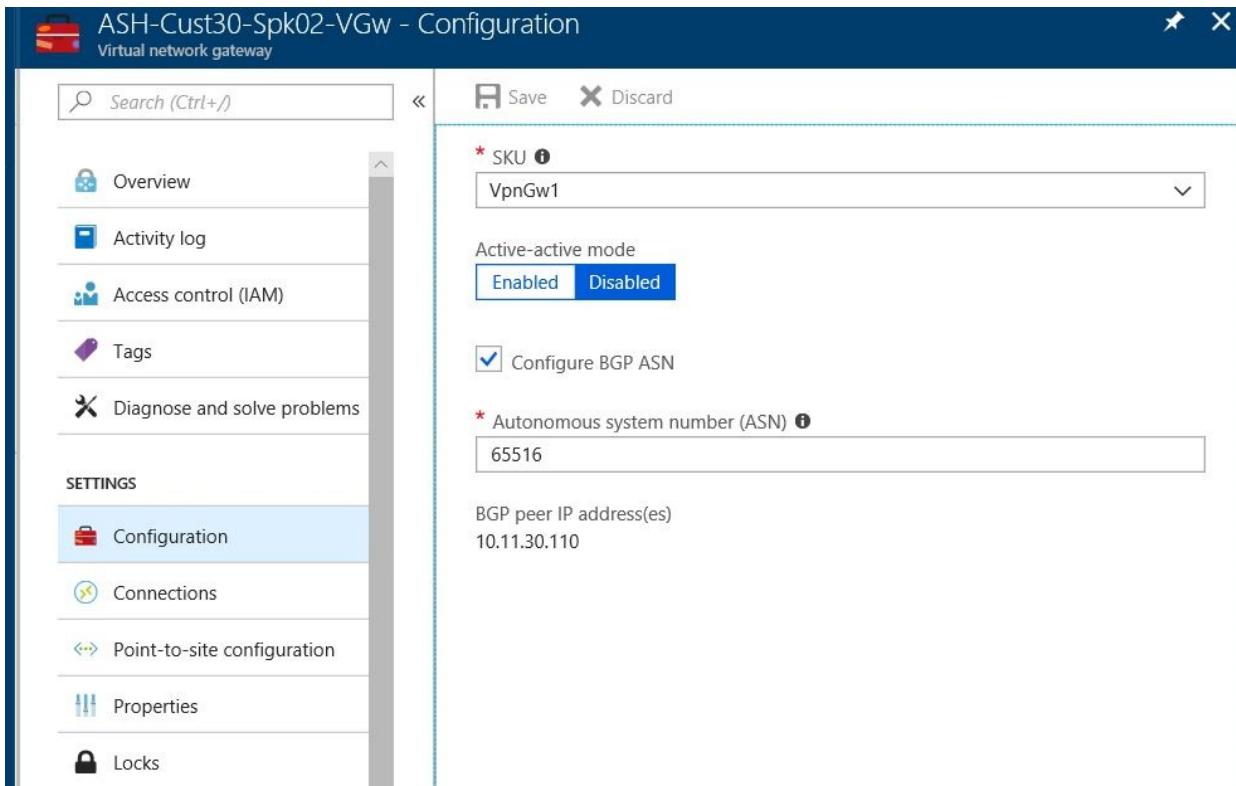
Allow gateway transit ⓘ

Use remote gateways ⓘ

The screenshot shows the 'VNet01-Spoke01-Peering' configuration page in the Azure portal. At the top, there are 'Save', 'Discard', and 'Delete' buttons. Below that, the 'Name' is listed as 'VNet01-Spoke01-Peering'. Under 'Peering status', it says 'Connected'. The 'Provisioning state' is 'Succeeded'. In the 'Peer details' section, the 'Address space' is '10.11.30.0/26'. The 'Virtual network' dropdown is set to 'Spoke01-VNet'. In the 'Configuration' section, 'Allow virtual network access' is set to 'Enabled'. Under 'Allow virtual network access', there are four checkboxes: 'Allow forwarded traffic' (checked), 'Allow gateway transit' (checked), and 'Use remote gateways' (unchecked). The 'Use remote gateways' checkbox has an information icon next to it.

Branch VNet connectivity by using a site-to-site VPN

Set up site-to-site VPN connectivity between the hub and branch VNets by using VPN gateways in Azure VPN Gateway. By default, VPN gateways and Azure ExpressRoute gateways use a private autonomous system number (ASN) value of **65515**. You can change the ASN value in VPN Gateway. In the test setup, the ASN value of the branch VNet VPN gateway is changed to **65516** to support eBGP routing between the hub and branch VNets.



On-premises Location 1 connectivity by using ExpressRoute and a site-to-site VPN

ExpressRoute 1 configuration details

The following figure shows the Azure Region 1 ExpressRoute circuit configuration toward on-premises Location 1 customer edge (CE) routers:

Move		Delete	Refresh	
Resource group (change)	ASH-Cust30	Provider	Equinix	
Circuit status	Enabled	Provider status	Provisioned	
Location	East US	Peering location	Washington DC	
Subscription (change)	ExpressRoute-Lab	Bandwidth	50 Mbps	
Subscription ID		Service key		
Tags (change)				
Click here to add tags				
Peering				
Type	Status	Primary Subnet	Secondary Subnet	Last Modified By
Azure private	Provisioned	192.168.30.16/30	192.168.30.20/30	Customer

The following figure shows the connection configuration between the ExpressRoute 1 circuit and the hub VNet:

 Move  Delete

Resource group (change)	Data in
ASH-Cust30	0 B
Status	Data out
Succeeded	0 B
Location	Virtual network
East US	VNet01
Subscription (change)	Virtual network gateway
ExpressRoute-Lab	ASH-Cust30-gw (13.90.87.1)
Subscription ID	Circuit
	ASH-Cust30-ER
Tags (change)	
Click here to add tags	

The following list shows the primary CE router configuration for ExpressRoute private peering connectivity. (Cisco ASR1000 routers are used as CE routers in the test setup.) When site-to-site VPN and ExpressRoute circuits are configured in parallel to connect an on-premises network to Azure, Azure prioritizes the ExpressRoute circuit by default. To avoid asymmetrical routing, the on-premises network also should prioritize ExpressRoute connectivity over site-to-site VPN connectivity. The following configuration establishes prioritization by using the BGP **local-preference** attribute:

```
interface TenGigabitEthernet0/0/0.300
description Customer 30 private peering to Azure
encapsulation dot1Q 30 second-dot1q 300
ip vrf forwarding 30
ip address 192.168.30.17 255.255.255.252
!
interface TenGigabitEthernet1/0/0.30
description Customer 30 to south bound LAN switch
encapsulation dot1Q 30
ip vrf forwarding 30
ip address 192.168.30.0 255.255.255.254
ip ospf network point-to-point
!
router ospf 30 vrf 30
router-id 10.2.30.253
redistribute bgp 65021 subnets route-map BGP2OSPF
network 192.168.30.0 0.0.0.1 area 0.0.0.0
default-information originate always
default-metric 10
!
router bgp 65021
!
address-family ipv4 vrf 30
network 10.2.30.0 mask 255.255.255.128
neighbor 192.168.30.18 remote-as 12076
neighbor 192.168.30.18 activate
neighbor 192.168.30.18 next-hop-self
neighbor 192.168.30.18 soft-reconfiguration inbound
neighbor 192.168.30.18 route-map prefer-ER-over-VPN in
neighbor 192.168.30.18 prefix-list Cust30_to_Private out
exit-address-family
!
route-map prefer-ER-over-VPN permit 10
set local-preference 200
!
ip prefix-list Cust30_to_Private seq 10 permit 10.2.30.0/25
!
```

Site-to-site VPN configuration details

The following list shows the primary CE router configuration for site-to-site VPN connectivity:

```

crypto ikev2 proposal Cust30-azure-proposal
  encryption aes-cbc-256 aes-cbc-128 3des
  integrity sha1
  group 2
!
crypto ikev2 policy Cust30-azure-policy
  match address local 66.198.12.106
  proposal Cust30-azure-proposal
!
crypto ikev2 keyring Cust30-azure-keyring
  peer azure
    address 52.168.162.84
    pre-shared-key local IamSecure123
    pre-shared-key remote IamSecure123
!
crypto ikev2 profile Cust30-azure-profile
  match identity remote address 52.168.162.84 255.255.255.255
  identity local address 66.198.12.106
  authentication local pre-share
  authentication remote pre-share
  keyring local Cust30-azure-keyring
!
crypto ipsec transform-set Cust30-azure-ipsec-proposal-set esp-aes 256 esp-sha-hmac
  mode tunnel
!
crypto ipsec profile Cust30-azure-ipsec-profile
  set transform-set Cust30-azure-ipsec-proposal-set
  set ikev2-profile Cust30-azure-profile
!
interface Loopback30
  ip address 66.198.12.106 255.255.255.255
!
interface Tunnel30
  ip vrf forwarding 30
  ip address 10.2.30.125 255.255.255.255
  tunnel source Loopback30
  tunnel mode ipsec ipv4
  tunnel destination 52.168.162.84
  tunnel protection ipsec profile Cust30-azure-ipsec-profile
!
router bgp 65021
!
address-family ipv4 vrf 30
  network 10.2.30.0 mask 255.255.255.128
  neighbor 10.10.30.254 remote-as 65515
  neighbor 10.10.30.254 ebgp-multihop 5
  neighbor 10.10.30.254 update-source Tunnel30
  neighbor 10.10.30.254 activate
  neighbor 10.10.30.254 soft-reconfiguration inbound
exit-address-family
!
ip route vrf 30 10.10.30.254 255.255.255.255 Tunnel30

```

On-premises Location 2 connectivity by using ExpressRoute

A second ExpressRoute circuit, in closer proximity to on-premises Location 2, connects on-premises Location 2 to the hub VNet. The following figure shows the second ExpressRoute configuration:

Resource group (change) ASH-Cust30		Provider Equinix			
Circuit status	Enabled	Provider status Provisioned			
Location	East US	Peering location Seattle			
Subscription (change)	ExpressRoute-Lab	Bandwidth 50 Mbps			
Subscription ID		Service key			
Tags (change) Click here to add tags		▲			
Peerings					
TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY	...
Azure private	Provisioned	192.168.31.16/30	192.168.31.20/30	Customer	...

The following figure shows the connection configuration between the second ExpressRoute circuit and the hub VNet:

Resource group (change) ASH-Cust30	Data in 0 B
Status Succeeded	Data out 0 B
Location East US	Virtual network VNet01
Subscription (change) ExpressRoute-Lab	Virtual network gateway ASH-Cust30-gw (13.90.87.1)
Subscription ID	Circuit SEA-Cust31-ER
Tags (change) Click here to add tags	

ExpressRoute 1 connects both the hub VNet and on-premises Location 1 to a remote VNet in a different Azure region:

Move Delete	
Resource group (change) ASH-Cust30	Data in 0 B
Status Succeeded	Data out 0 B
Location West US 2	Virtual network USWst2-VNet
Subscription (change) ExpressRoute-Lab	Virtual network gateway ASH30-USWst2-ERGW (52.175.245.182)
Subscription ID	Circuit ASH-Cust30-ER
Tags (change) Click here to add tags	

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than

ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [control plane analysis](#) of the test setup and the views of different VNets or VLANs in the topology.

Learn about [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure : Control plane analysis

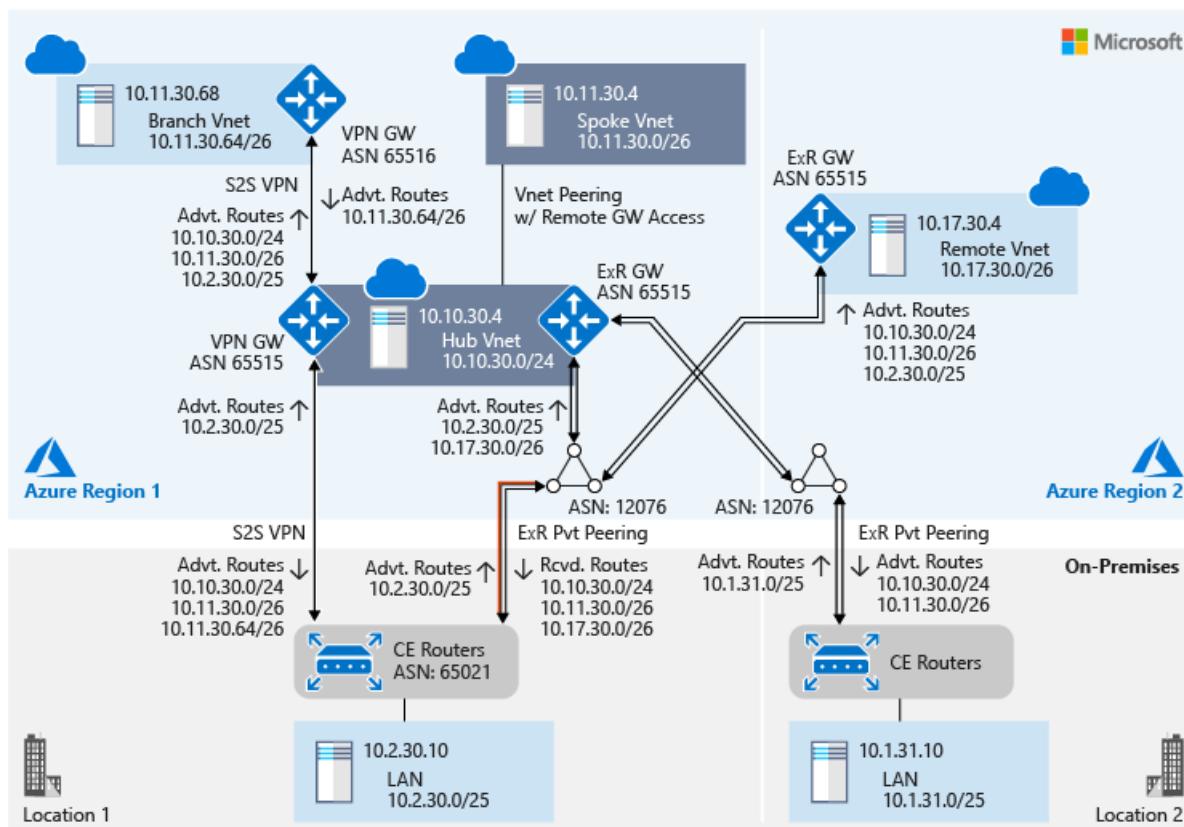
2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes the control plane analysis of the [test setup](#). You can also review the [test setup configuration](#) and the [data plane analysis](#) of the test setup.

Control plane analysis essentially examines routes that are exchanged between networks within a topology. Control plane analysis can help you understand how different networks view the topology.

Hub and spoke VNet perspective

The following figure illustrates the network from the perspective of a hub virtual network (VNet) and a spoke VNet (highlighted in blue). The figure also shows the autonomous system number (ASN) of different networks and routes that are exchanged between different networks:



The ASN of the VNet's Azure ExpressRoute gateway is different from the ASN of Microsoft Enterprise Edge Routers (MSEEs). An ExpressRoute gateway uses a private ASN (a value of 65515) and MSEEs use public ASN (a value of 12076) globally. When you configure ExpressRoute peering, because MSEE is the peer, you use 12076 as the peer ASN. On the Azure side, MSEE establishes eBGP peering with the ExpressRoute gateway. The dual eBGP peering that the MSEE establishes for each ExpressRoute peering is transparent at the control plane level. Therefore, when you view an ExpressRoute route table, you see the VNet's ExpressRoute gateway ASN for the VNet's prefixes.

The following figure shows a sample ExpressRoute route table:

Route table (Primary)
AzurePrivatePeering - ASH-Cust30-ER

[Download](#) [Show secondary](#)

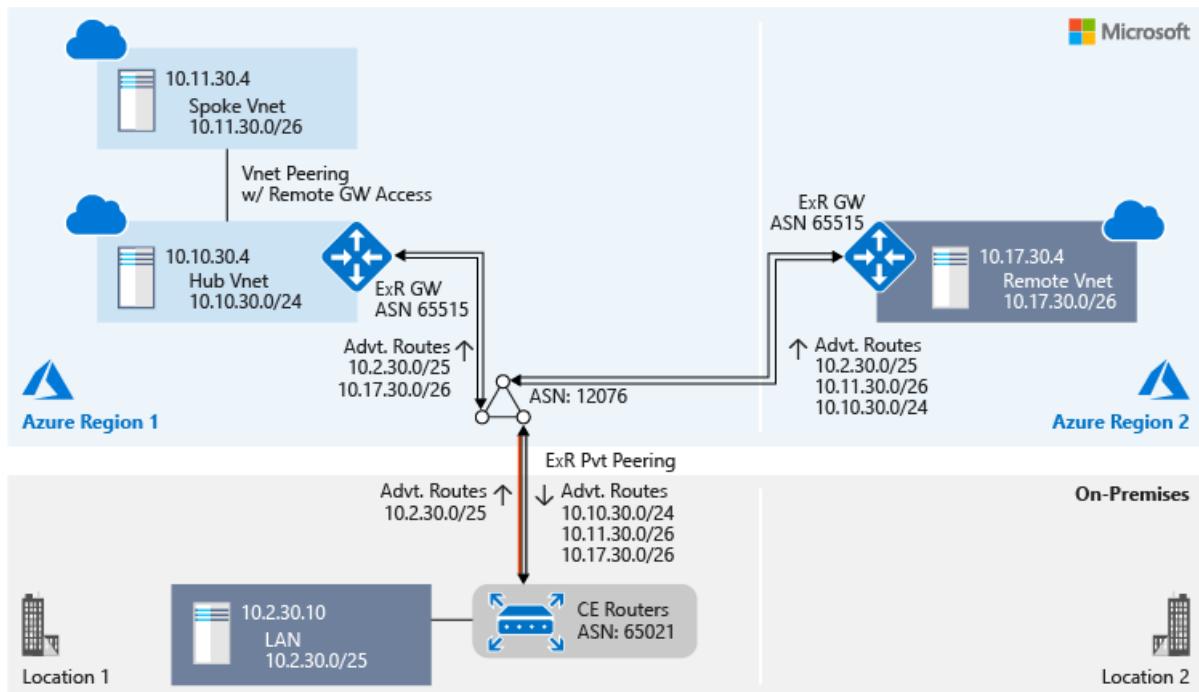
Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.2.30.0/25	192.168.30.17		0	65021
10.2.30.125/32	192.168.30.17		0	65021 65515
10.10.30.0/24	10.10.30.141		0	65515
	10.10.30.140		0	65515
10.11.30.0/26	10.10.30.141		0	65515
	10.10.30.140		0	65515

Within Azure, the ASN is significant only from a peering perspective. By default, the ASN of both the ExpressRoute gateway and the VPN gateway in Azure VPN Gateway is **65515**.

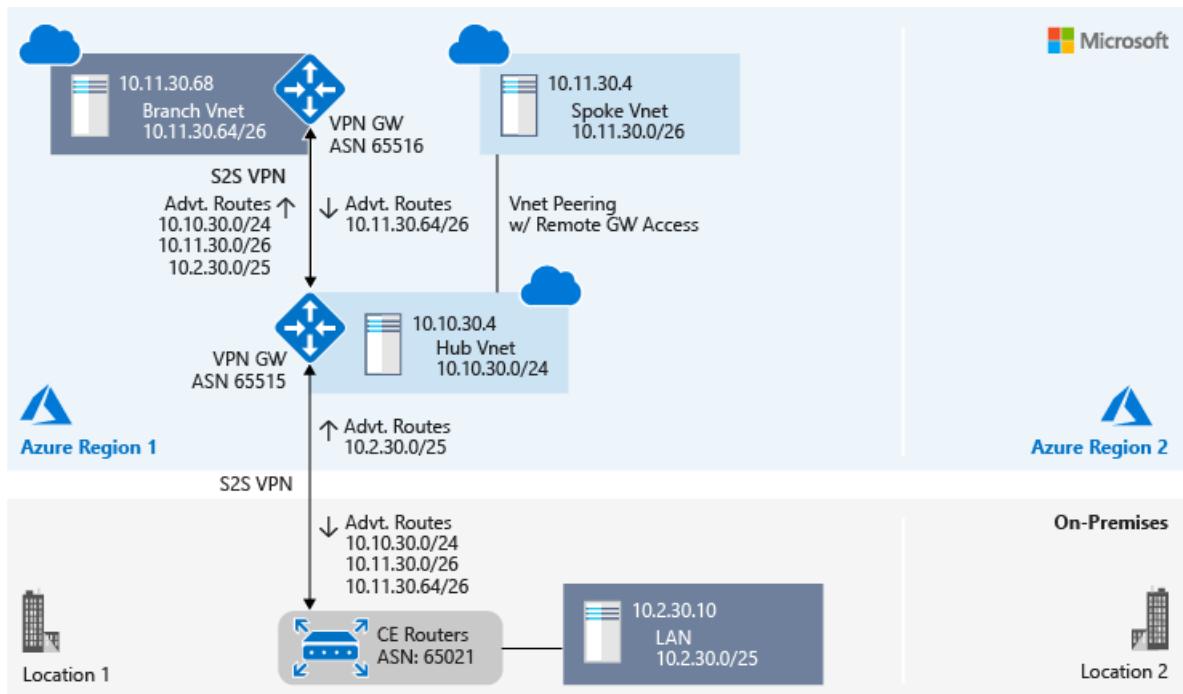
On-premises Location 1 and the remote VNet perspective via ExpressRoute 1

Both on-premises Location 1 and the remote VNet are connected to the hub VNet via ExpressRoute 1. They share the same perspective of the topology, as shown in the following diagram:



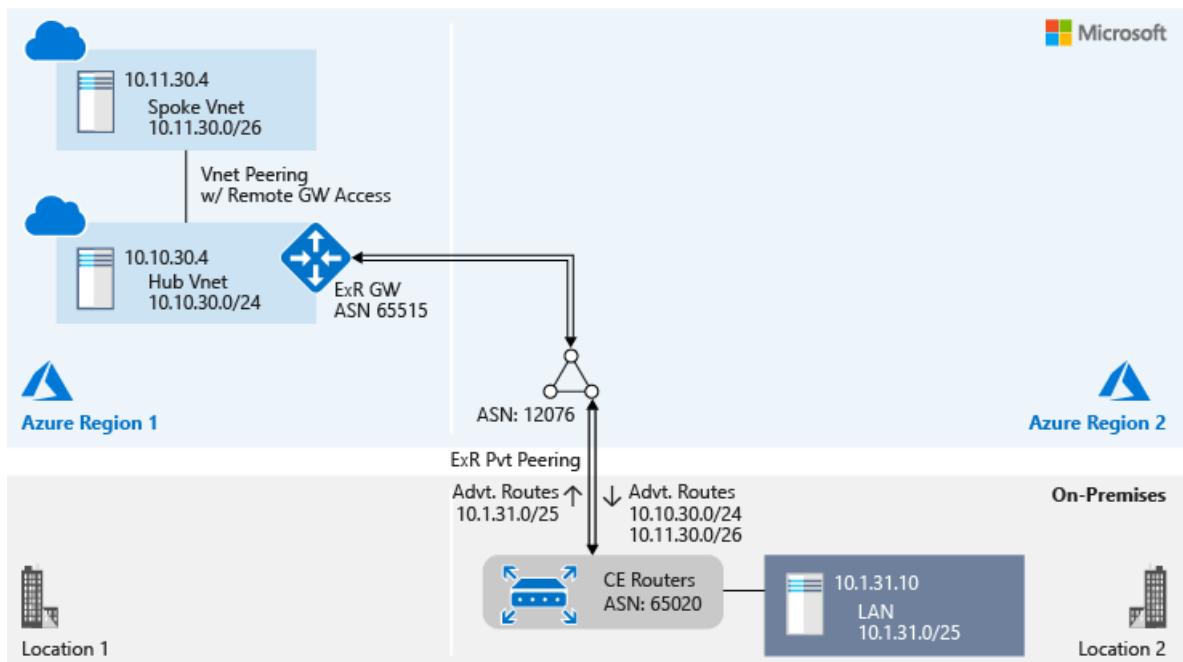
On-premises Location 1 and the branch VNet perspective via a site-to-site VPN

Both on-premises Location 1 and the branch VNet are connected to a hub VNet's VPN gateway via a site-to-site VPN connection. They share the same perspective of the topology, as shown in the following diagram:



On-premises Location 2 perspective

On-premises Location 2 is connected to a hub VNet via private peering of ExpressRoute 2:



ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure : Data plane analysis

2/18/2021 • 16 minutes to read • [Edit Online](#)

This article describes the data plane analysis of the [test setup](#). You can also review the [test setup configuration](#) and the [control plane analysis](#) of the test setup.

Data plane analysis examines the path taken by packets that traverse from one local network (LAN or virtual network) to another within a topology. The data path between two local networks isn't necessarily symmetrical. Therefore, in this article, we analyze a forwarding path from a local network to another network that's separate from the reverse path.

Data path from the hub VNet

Path to the spoke VNet

Virtual network (VNet) peering emulates network bridge functionality between the two VNets that are peered. Traceroute output from a hub VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1      2 ms      1 ms      1 ms  10.11.30.4

Trace complete.
```

The following figure shows the graphical connection view of the hub VNet and the spoke VNet from the perspective of Azure Network Watcher:



Path to the branch VNet

Traceroute output from a hub VNet to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1      1 ms      1 ms      1 ms  10.10.30.142
 2      *          *          *      Request timed out.
 3      2 ms      2 ms      2 ms  10.11.30.68

Trace complete.
```

In this traceroute, the first hop is the VPN gateway in Azure VPN Gateway of the hub VNet. The second hop is the VPN gateway of the branch VNet. The IP address of the VPN gateway of the branch VNet isn't advertised in the hub VNet. The third hop is the VM on the branch VNet.

The following figure shows the graphical connection view of the hub VNet and the branch VNet from the

perspective of Network Watcher:



For the same connection, the following figure shows the grid view in Network Watcher:

Hops

NAME	IP ADDRESS	STATUS	NEXT HOP IP ADDRESS
ash-cust30-vm1195	10.10.30.4	✓	13.90.87.1
ASH-Cust30-gw	13.90.87.1	✓	10.11.30.68
ash-c30-sk2-vm1619	10.11.30.68	✓	-

Average Latency in milliseconds

3

Minimum Latency in milliseconds

2

Maximum Latency in milliseconds

6

Probes Sent

66

Probes Failed

0

Path to on-premises Location 1

Traceroute output from a hub VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1    2 ms      2 ms      2 ms  10.10.30.132
 2      *        *        *      Request timed out.
 3      *        *        *      Request timed out.
 4    2 ms      2 ms      2 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the Azure ExpressRoute gateway tunnel endpoint to a Microsoft Enterprise Edge Router (MSEE). The second and third hops are the customer edge (CE) router and the on-premises Location 1 LAN IPs. These IP addresses aren't advertised in the hub VNet. The fourth hop is the VM in the on-premises Location 1.

Path to on-premises Location 2

Traceroute output from a hub VNet to a VM in on-premises Location 2 is shown here:

```
C:\Users\rb>tracert 10.1.31.10

Tracing route to 10.1.31.10 over a maximum of 30 hops

 1  76 ms    75 ms    75 ms  10.10.30.134
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  75 ms    75 ms    75 ms  10.1.31.10

Trace complete.
```

In this traceroute, the first hop is the ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 2 LAN IPs. These IP addresses aren't advertised in the hub VNet. The fourth hop is the VM on the on-premises Location 2.

Path to the remote VNet

Traceroute output from a hub VNet to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1  2 ms    2 ms    2 ms  10.10.30.132
 2  *         *         *      Request timed out.
 3  69 ms   68 ms   69 ms  10.17.30.4

Trace complete.
```

In this traceroute, the first hop is the ExpressRoute gateway tunnel endpoint to an MSEE. The second hop is the remote VNet's gateway IP. The second hop IP range isn't advertised in the hub VNet. The third hop is the VM on the remote VNet.

Data path from the spoke VNet

The spoke VNet shares the network view of the hub VNet. Through VNet peering, the spoke VNet uses the remote gateway connectivity of the hub VNet as if it's directly connected to the spoke VNet.

Path to the hub VNet

Traceroute output from the spoke VNet to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms   <1 ms   <1 ms  10.10.30.4

Trace complete.
```

Path to the branch VNet

Traceroute output from the spoke VNet to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1  1 ms    <1 ms    <1 ms  10.10.30.142
 2  *        *        *        Request timed out.
 3  3 ms    2 ms    2 ms  10.11.30.68

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the hub VNet. The second hop is the VPN gateway of the branch VNet. The IP address of the VPN gateway of the branch VNet isn't advertised within the hub/spoke VNet. The third hop is the VM on the branch VNet.

Path to on-premises Location 1

Traceroute output from the spoke VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  24 ms    2 ms    3 ms  10.10.30.132
 2  *        *        *        Request timed out.
 3  *        *        *        Request timed out.
 4  3 ms    2 ms    2 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 1 LAN IPs. These IP addresses aren't advertised in the hub/spoke VNet. The fourth hop is the VM in the on-premises Location 1.

Path to on-premises Location 2

Traceroute output from the spoke VNet to a VM in on-premises Location 2 is shown here:

```
C:\Users\rb>tracert 10.1.31.10

Tracing route to 10.1.31.10 over a maximum of 30 hops

 1  76 ms    75 ms    76 ms  10.10.30.134
 2  *        *        *        Request timed out.
 3  *        *        *        Request timed out.
 4  75 ms    75 ms    75 ms  10.1.31.10

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 2 LAN IPs. These IP addresses aren't advertised in the hub/spoke VNets. The fourth hop is the VM in the on-premises Location 2.

Path to the remote VNet

Traceroute output from the spoke VNet to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1      2 ms      1 ms      1 ms  10.10.30.133
 2      *         *         *      Request timed out.
 3     71 ms     70 ms     70 ms  10.17.30.4

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second hop is the remote VNet's gateway IP. The second hop IP range isn't advertised in the hub/spoke VNet. The third hop is the VM on the remote VNet.

Data path from the branch VNet

Path to the hub VNet

Traceroute output from the branch VNet to a VM in the hub VNet is shown here:

```
C:\Windows\system32>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1    <1 ms    <1 ms    <1 ms  10.11.30.100
 2      *        *        *      Request timed out.
 3     4 ms     3 ms     3 ms  10.10.30.4

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VM on the hub VNet.

Path to the spoke VNet

Traceroute output from the branch VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1      1 ms    <1 ms    1 ms  10.11.30.100
 2      *         *         *      Request timed out.
 3     4 ms     3 ms     2 ms  10.11.30.4

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VM on the spoke VNet.

Path to on-premises Location 1

Traceroute output from the branch VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1      1 ms     <1 ms    <1 ms  10.11.30.100
 2      *         *         *      Request timed out.
 3      3 ms     2 ms     2 ms  10.2.30.125
 4      *         *         *      Request timed out.
 5      3 ms     3 ms     3 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VPN tunnel termination point on the primary CE router. The fourth hop is an internal IP address of on-premises Location 1. This LAN IP address isn't advertised outside the CE router. The fifth hop is the destination VM in the on-premises Location 1.

Path to on-premises Location 2 and the remote VNet

As we discussed in the control plane analysis, the branch VNet has no visibility either to on-premises Location 2 or to the remote VNet per the network configuration. The following ping results confirm:

```
C:\Users\rb>ping 10.1.31.10

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.1.31.10:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\rb>ping 10.17.30.4

Pinging 10.17.30.4 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.17.30.4:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Data path from on-premises Location 1

Path to the hub VNet

Traceroute output from on-premises Location 1 to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  <1 ms    <1 ms    <1 ms  192.168.30.18
 4  *         *         *      Request timed out.
 5  2 ms     2 ms     2 ms   10.10.30.4

Trace complete.
```

In this traceroute, the first two hops are part of the on-premises network. The third hop is the primary MSEE interface that faces the CE router. The fourth hop is the ExpressRoute gateway of the hub VNet. The IP range of the ExpressRoute gateway of the hub VNet isn't advertised to the on-premises network. The fifth hop is the destination VM.

Network Watcher provides only an Azure-centric view. For an on-premises perspective, we use Azure Network Performance Monitor. Network Performance Monitor provides agents that you can install on servers in networks outside Azure for data path analysis.

The following figure shows the topology view of the on-premises Location 1 VM connectivity to the VM on the hub VNet via ExpressRoute:



As discussed earlier, the test setup uses a site-to-site VPN as backup connectivity for ExpressRoute between the on-premises Location 1 and the hub VNet. To test the backup data path, let's induce an ExpressRoute link failure between the on-premises Location 1 primary CE router and the corresponding MSEE. To induce an ExpressRoute link failure, shut down the CE interface that faces the MSEE:

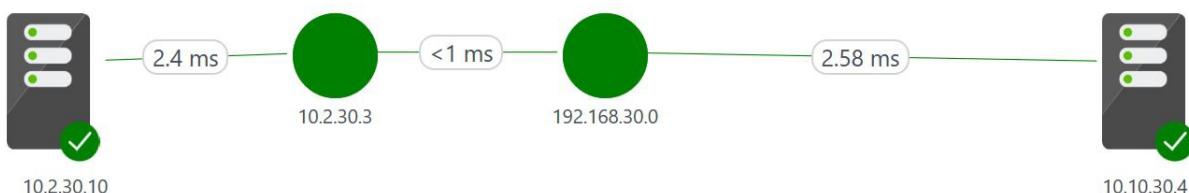
```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  3 ms     2 ms     3 ms   10.10.30.4

Trace complete.
```

The following figure shows the topology view of the on-premises Location 1 VM connectivity to the VM on the hub VNet via site-to-site VPN connectivity when ExpressRoute connectivity is down:



Path to the spoke VNet

Traceroute output from on-premises Location 1 to a VM in the spoke VNet is shown here:

Let's bring back the ExpressRoute primary connectivity to do the data path analysis toward the spoke VNet:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  <1 ms    <1 ms    <1 ms  192.168.30.18
 4  *         *         *      Request timed out.
 5  3 ms    2 ms    2 ms  10.11.30.4

Trace complete.
```

Bring up the primary ExpressRoute 1 connectivity for the remainder of the data path analysis.

Path to the branch VNet

Traceroute output from on-premises Location 1 to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  3 ms    2 ms    2 ms  10.11.30.68

Trace complete.
```

Path to on-premises Location 2

As we discuss in the [control plane analysis](#), the on-premises Location 1 has no visibility to on-premises Location 2 per the network configuration. The following ping results confirm:

```
C:\Users\rb>ping 10.1.31.10

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.1.31.10:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Path to the remote VNet

Traceroute output from on-premises Location 1 to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  2 ms     5 ms     7 ms  192.168.30.0
 3  <1 ms    <1 ms    <1 ms  192.168.30.18
 4  *         *         *      Request timed out.
 5  69 ms    70 ms    69 ms  10.17.30.4

Trace complete.
```

Data path from on-premises Location 2

Path to the hub VNet

Traceroute output from on-premises Location 2 to a VM in the hub VNet is shown here:

```
C:\Windows\system32>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.1.31.3
 2  <1 ms    <1 ms    <1 ms  192.168.31.4
 3  <1 ms    <1 ms    <1 ms  192.168.31.22
 4  *         *         *      Request timed out.
 5  75 ms    74 ms    74 ms  10.10.30.4

Trace complete.
```

Path to the spoke VNet

Traceroute output from on-premises Location 2 to a VM in the spoke VNet is shown here:

```
C:\Windows\system32>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    1 ms   10.1.31.3
 2  <1 ms    <1 ms    <1 ms  192.168.31.0
 3  <1 ms    <1 ms    <1 ms  192.168.31.18
 4  *         *         *      Request timed out.
 5  75 ms    74 ms    74 ms  10.11.30.4

Trace complete.
```

Path to the branch VNet, on-premises Location 1, and the remote VNet

As we discuss in the [control plane analysis](#), the on-premises Location 1 has no visibility to the branch VNet, to on-premises Location 1, or to the remote VNet per the network configuration.

Data path from the remote VNet

Path to the hub VNet

Traceroute output from the remote VNet to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  65 ms    65 ms    65 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  69 ms    68 ms    68 ms  10.10.30.4

Trace complete.
```

Path to the spoke VNet

Traceroute output from the remote VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  67 ms    67 ms    67 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  71 ms    69 ms    69 ms  10.11.30.4

Trace complete.
```

Path to the branch VNet and on-premises Location 2

As we discuss in the [control plane analysis](#), the remote VNet has no visibility to the branch VNet or to on-premises Location 2 per the network configuration.

Path to on-premises Location 1

Traceroute output from the remote VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  67 ms    67 ms    67 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  69 ms    69 ms    69 ms  10.2.30.10

Trace complete.
```

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

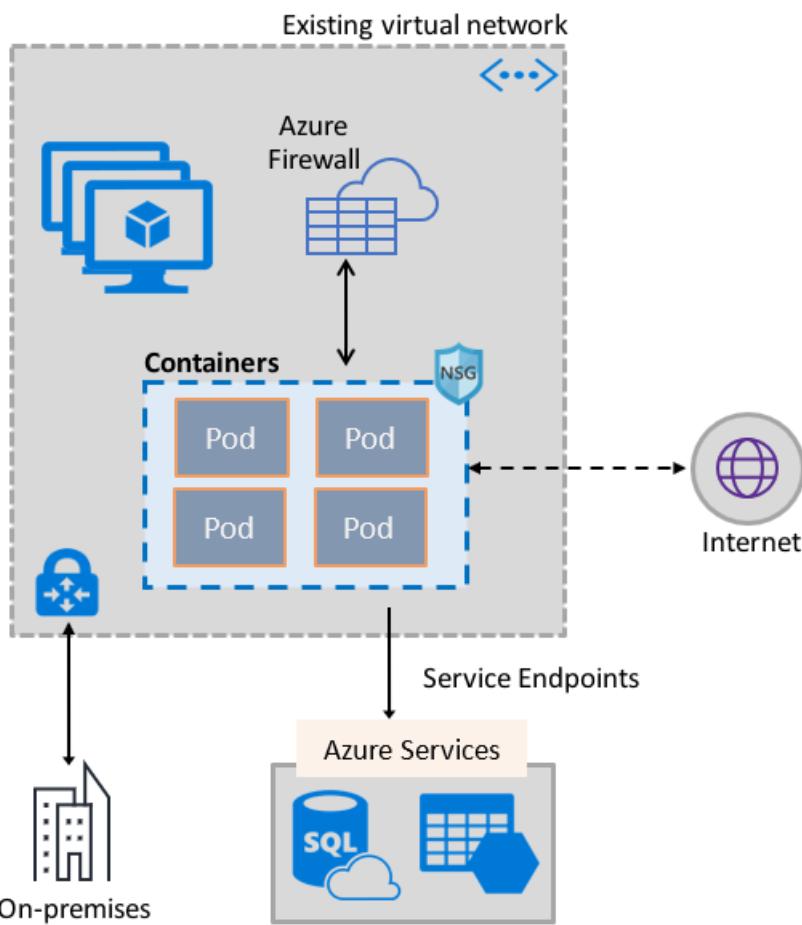
Enable containers to use Azure Virtual Network capabilities

2/18/2021 • 3 minutes to read • [Edit Online](#)

Bring the rich set of Azure network capabilities to containers, by utilizing the same software defined networking stack that powers virtual machines. The Azure Virtual Network container network interface (CNI) plug-in installs in an Azure Virtual Machine. The plug-in assigns IP addresses from a virtual network to containers brought up in the virtual machine, attaching them to the virtual network, and connecting them directly to other containers and virtual network resources. The plug-in doesn't rely on overlay networks, or routes, for connectivity, and provides the same performance as virtual machines. At a high level, the plug-in provides the following capabilities:

- A virtual network IP address is assigned to every Pod, which could consist of one or more containers.
- Pods can connect to peered virtual networks and to on-premises over ExpressRoute or a site-to-site VPN. Pods are also reachable from peered and on-premises networks.
- Pods can access services such as Azure Storage and Azure SQL Database, that are protected by virtual network service endpoints.
- Network security groups and routes can be applied directly to Pods.
- Pods can be placed directly behind an Azure internal or public Load Balancer, just like virtual machines
- Pods can be assigned a public IP address, which makes them directly accessible from the internet. Pods can also access the internet themselves.
- Works seamlessly with Kubernetes resources such as Services, Ingress controllers, and Kube DNS. A Kubernetes Service can also be exposed internally or externally through the Azure Load Balancer.

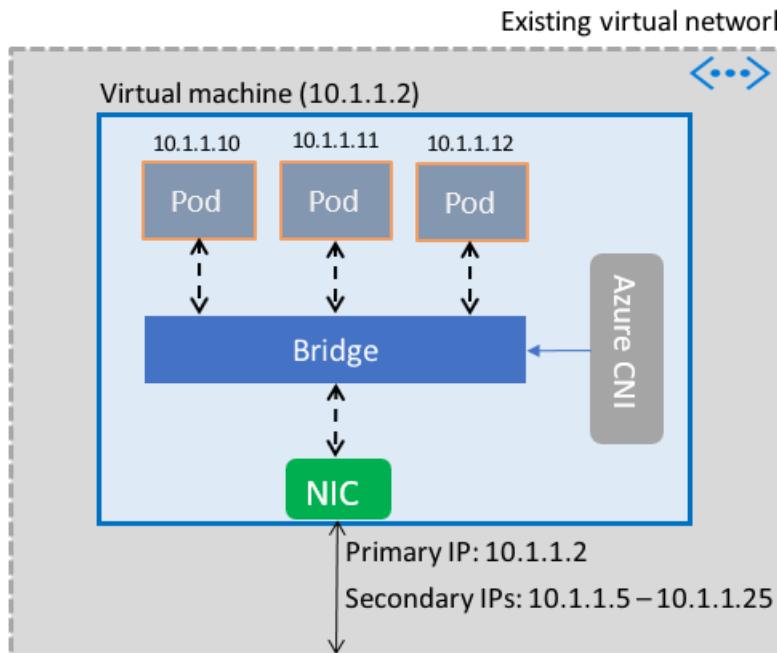
The following picture shows how the plug-in provides Azure Virtual Network capabilities to Pods:



The plug-in supports both Linux and Windows platforms.

Connecting Pods to a virtual network

Pods are brought up in a virtual machine that is part of a virtual network. A pool of IP addresses for the Pods is configured as secondary addresses on a virtual machine's network interface. Azure CNI sets up the basic Network connectivity for Pods and manages the utilization of the IP addresses in the pool. When a Pod comes up in the virtual machine, Azure CNI assigns an available IP address from the pool and connects the Pod to a software bridge in the virtual machine. When the Pod terminates, the IP address is added back to the pool. The following picture shows how Pods connect to a virtual network:



Internet access

To enable Pods to access the internet, the plug-in configures *iptables* rules to network address translate (NAT) the internet bound traffic from Pods. The source IP address of the packet is translated to the primary IP address on the virtual machine's network interface. Windows virtual machines automatically source NAT (SNAT) traffic destined to IP addresses outside the subnet the virtual machine is in. Typically, all traffic destined to an IP address outside of the IP range of the virtual network is translated.

Limits

The plug-in supports up to 250 Pods per virtual machine and up to 16,000 Pods in a virtual network. These limits are different for the [Azure Kubernetes Service](#).

Using the plug-in

The plug-in can be used in the following ways, to provide basic virtual network attach for Pods or Docker containers:

- **Azure Kubernetes Service:** The plug-in is integrated into the Azure Kubernetes Service (AKS), and can be used by choosing the *Advanced Networking* option. Advanced Networking lets you deploy a Kubernetes cluster in an existing, or a new, virtual network. To learn more about Advanced Networking and the steps to set it up, see [Network configuration in AKS](#).
- **AKS-Engine:** AKS-Engine is a tool that generates an Azure Resource Manager template for the deployment of a Kubernetes cluster in Azure. For detailed instructions, see [Deploy the plug-in for AKS-Engine Kubernetes clusters](#).
- **Creating your own Kubernetes cluster in Azure:** The plug-in can be used to provide basic networking for Pods in Kubernetes clusters that you deploy yourself, without relying on AKS, or tools like the AKS-Engine. In this case, the plug-in is installed and enabled on every virtual machine in a cluster. For detailed instructions, see [Deploy the plug-in for a Kubernetes cluster that you deploy yourself](#).
- **Virtual network attach for Docker containers in Azure:** The plug-in can be used in cases where you don't want to create a Kubernetes cluster, and would like to create Docker containers with virtual network attach, in virtual machines. For detailed instructions, see [Deploy the plug-in for Docker](#).

Next steps

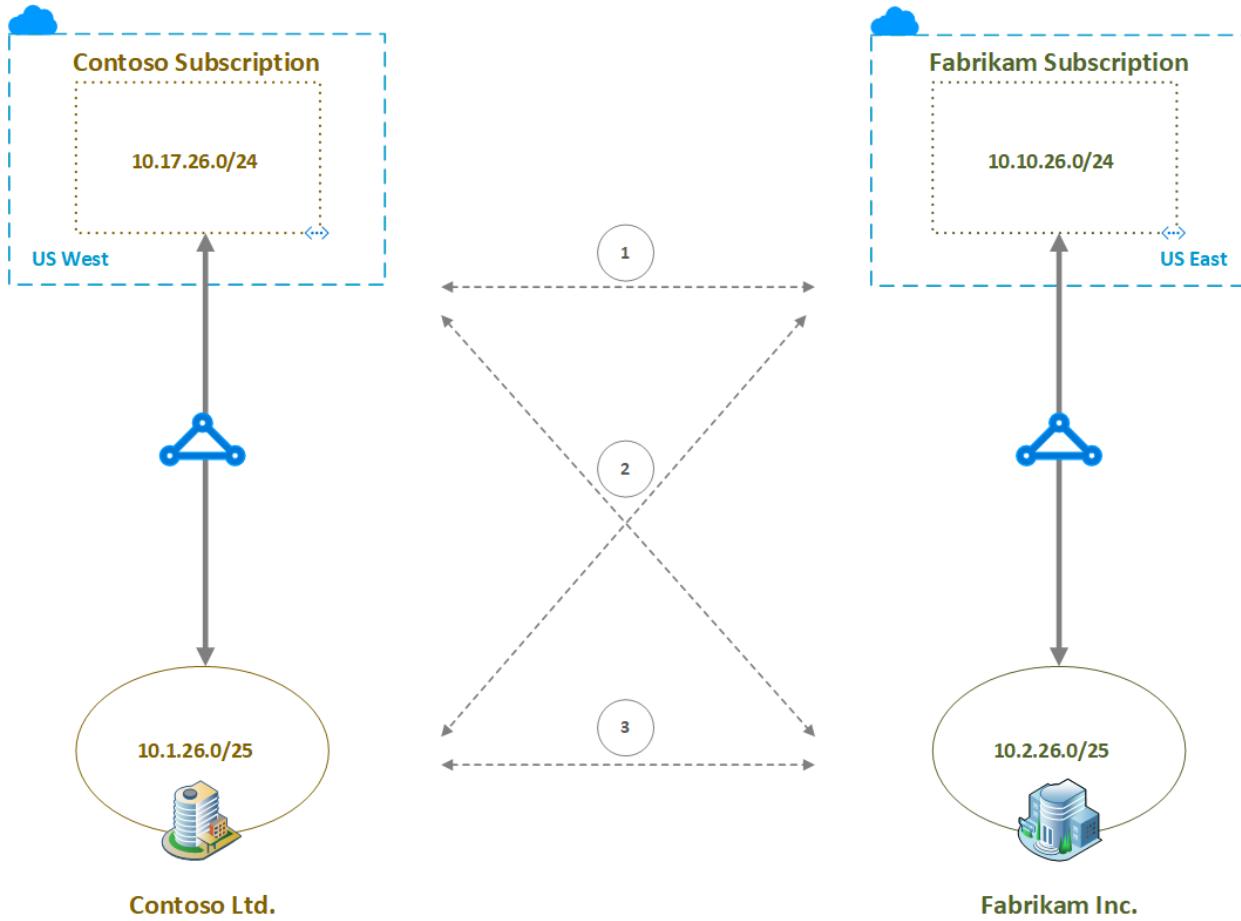
[Deploy the plug-in](#) for Kubernetes clusters or Docker containers

Cross-network connectivity

11/2/2020 • 4 minutes to read • [Edit Online](#)

Fabrikam Inc. has a large physical presence and Azure deployment in East US. Fabrikam has back-end connectivity between its on-premises and Azure deployments via ExpressRoute. Similarly, Contoso Ltd. has a presence and Azure deployment in West US. Contoso has back-end connectivity between its on-premises and Azure deployments via ExpressRoute.

Fabrikam Inc. acquires Contoso Ltd. Following the merger, Fabrikam wants to interconnect the networks. The following figure illustrates the scenario:



The dashed arrows in the middle of the above figure indicate the desired network interconnections. Specifically, there are three types cross connections desired: 1) Fabrikam and Contoso VNets cross connect, 2) Cross regional on-premises and VNets cross connects (that is, connecting Fabrikam on-premises network to Contoso VNet and connecting Contoso on-premises network to Fabrikam VNet), and 3) Fabrikam and Contoso on-premises network cross connect.

The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., before the merger.

Route table (Primary)

AzurePrivatePeering - Contoso-ER



[Download](#) [Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.1.26.0/25	192.168.26.17		0	65020
10.17.26.0/24	10.17.26.140		0	65515
	10.17.26.141		0	65515

The following table shows the effective routes of a VM in the Contoso subscription, before the merger. Per the table, the VM on the VNet is aware of the VNet address space and the Contoso on-premises network, apart from the default ones.

[Download](#) [Refresh](#)

i Showing only top 200 records, click Download above to see all.

Scope Network interface (Contoso-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.17.26.0/24	Virtual network	-
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., before the merger.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER



[Download](#) [Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.2.26.0/25	192.168.26.17		0	65021
10.10.26.0/24	10.10.26.140		0	65515
	10.10.26.141		0	65515

The following table shows the effective routes of a VM in the Fabrikam subscription, before the merger. Per the table, the VM on the VNet is aware of the VNet address space and the Fabrikam on-premises network, apart from the default ones.

[Download](#) [Refresh](#)

i Showing only top 200 records, click Download above to see all.

Scope Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-

In this article, let's go through step by step and discuss how to achieve the desired cross connections using the following Azure network features:

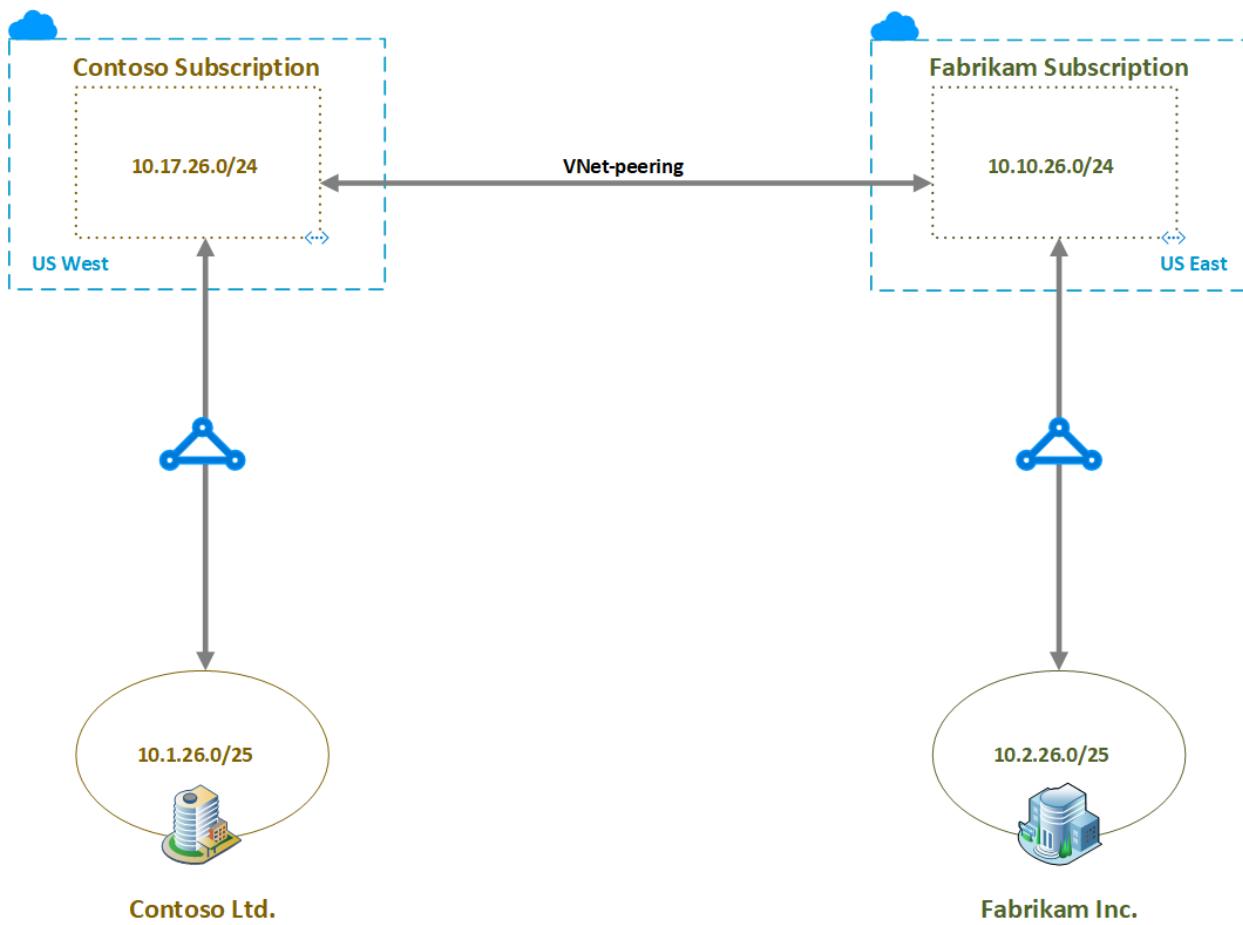
- [Virtual network peering](#)
- [Virtual network ExpressRoute connection](#)
- [Global Reach](#)

Cross connecting VNets

Virtual network peering (VNet peering) provides the most optimal and the best network performance when connecting two virtual networks. VNet peering supports peering two VNets both within the same Azure region (commonly called VNet peering) and in two different Azure regions (commonly called Global VNet peering).

Let's configure Global VNet peering between the VNets in Contoso and Fabrikam Azure subscriptions. For how to create the virtual network peering between two the virtual networks, see [Create a virtual network peering](#) article.

The following picture shows the network architecture after configuring Global VNet peering.



The following table shows the routes known to the Contoso subscription VM. Pay attention to the last entry of the table. This entry is the result of cross connecting the virtual networks.

Effective routes					
SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS	
Default	Active	10.17.26.0/24	Virtual network	-	
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53	
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52	
Default	Active	0.0.0.0/0	Internet	-	
Default	Active	10.0.0.0/8	None	-	
Default	Active	100.64.0.0/10	None	-	
Default	Active	192.168.0.0/16	None	-	
Default	Active	10.10.26.0/24	VNetGlobalPeering	-	

The following table shows the routes known to the Fabrikam subscription VM. Pay attention to the last entry of the table. This entry is the result of cross connecting the virtual networks.

[Download](#) [Refresh](#)

Showing only top 200 records, click Download above to see all.

Scope

Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.17.26.0/24	VNetGlobalPeering	-

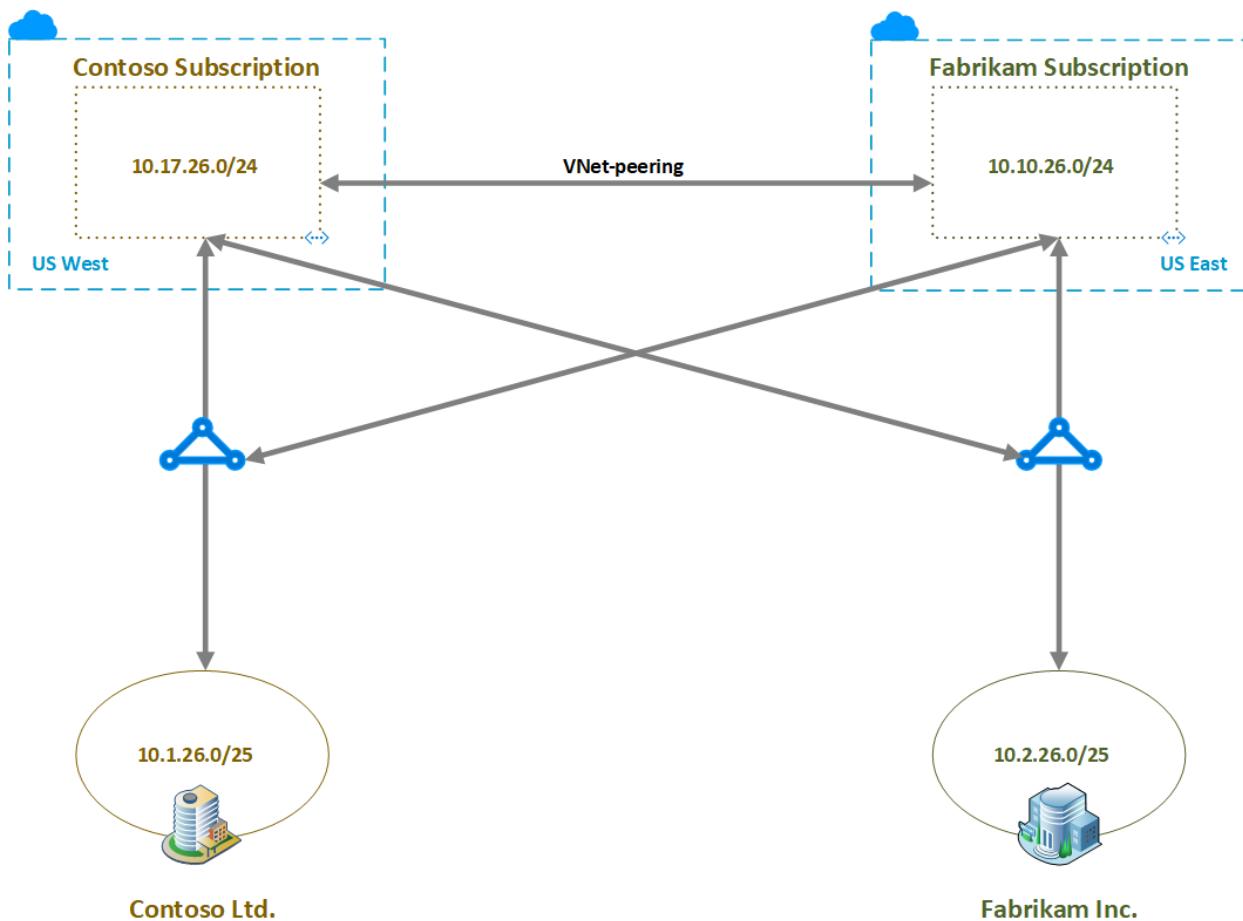
VNet peering directly links two virtual networks (see there are no next hop for *VNetGlobalPeering* entry in the above two tables)

Cross connecting VNets to the on-premises networks

We can connect an ExpressRoute circuit to multiple virtual networks. See [Subscription and service limits](#) for the maximum number of virtual networks that can be connected to an ExpressRoute circuit.

Let's connect Fabrikam ExpressRoute circuit to Contoso subscription VNet and similarly Contoso ExpressRoute circuit to Fabrikam subscription VNet to enable cross connectivity between virtual networks and the on-premises networks. To connect a virtual network to an ExpressRoute circuit in a different subscription, we need to create and use an authorization. See the article: [Connect a virtual network to an ExpressRoute circuit](#).

The following picture shows the network architecture after configuring the ExpressRoute cross connectivity to the virtual networks.



The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., after cross connecting virtual networks to the on-premises networks via ExpressRoute. See that the route table has routes belonging to both the virtual networks.

Route table (Primary)						
AzurePrivatePeering - Contoso-ER						
	Download	Show secondary				
i Showing only top 200 primary records, click Download above to see all.						
NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH		
10.1.26.0/25	192.168.26.17	0	65020			
10.10.26.0/24	10.10.26.141	0	65515			
	10.10.26.140	0	65515			
10.17.26.0/24	10.17.26.140	0	65515			
	10.17.26.141	0	65515			

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., after cross connecting virtual networks to the on-premises networks via ExpressRoute. See that the route table has routes belonging to both the virtual networks.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER



[Download](#)

[Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.2.26.0/25	192.168.26.17		0	65021
10.10.26.0/24	10.10.26.140		0	65515
	10.10.26.141		0	65515
10.17.26.0/24	10.17.26.140		0	65515
	10.17.26.141		0	65515

The following table shows the routes known to the Contoso subscription VM. Pay attention to *Virtual network gateway* entries of the table. The VM sees routes for both the on-premises networks.

[Download](#) [Refresh](#)

i Showing only top 200 records, click Download above to see all.

Scope

Network interface (Contoso-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.17.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.10.26.0/24	VNetGlobalPeering	-

The following table shows the routes known to the Fabrikam subscription VM. Pay attention to *Virtual network gateway* entries of the table. The VM sees routes for both the on-premises networks.

[Download](#) [Refresh](#)

Showing only top 200 records, click Download above to see all.

Scope

Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.17.26.0/24	VNetGlobalPeering	-

NOTE

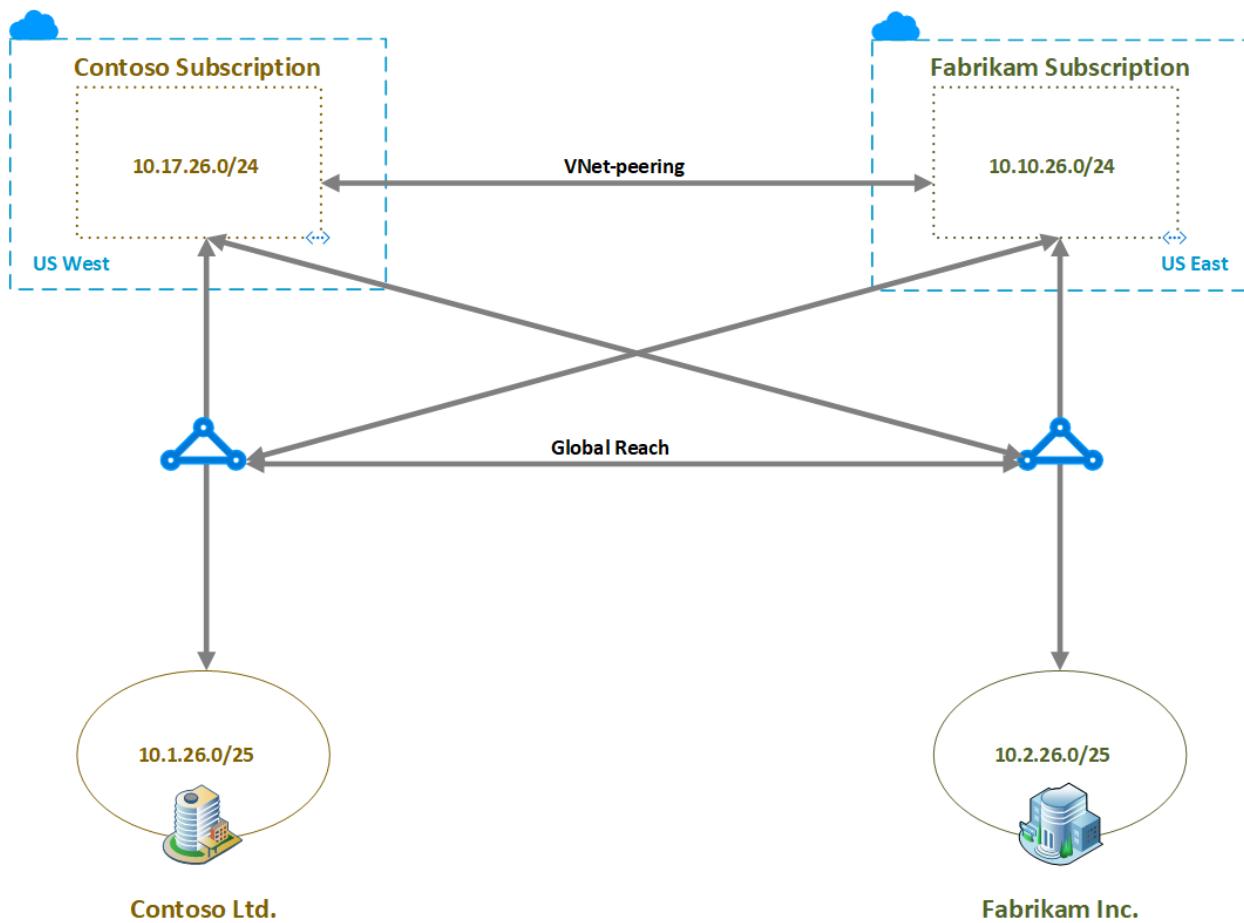
In either the Fabrikam and/or Contoso subscriptions you can also have spoke VNets to the respective hub VNet (a hub and spoke design is not illustrated in the architecture diagrams in this article). The cross connections between the hub VNet gateways to ExpressRoute will also allow communication between East and West hubs and spokes.

Cross connecting on-premises networks

ExpressRoute Global Reach provides connectivity between on-premises networks that are connected to different ExpressRoute circuits. Let's configure Global Reach between Contoso and Fabrikam ExpressRoute circuits.

Because the ExpressRoute circuits are in different subscriptions, we need to create and use an authorization. See [Configure ExpressRoute Global Reach](#) article for step by step guidance.

The following picture shows the network architecture after configuring Global Reach.



The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., after configuring Global Reach. See that the route table has routes belonging to both the on-premises networks.

Route table (Primary)							
AzurePrivatePeering - Contoso-ER							
NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH			
10.1.26.0/25	192.168.26.17		0	65020			
10.2.26.0/25	192.168.26.51	10	0	65021			
10.10.26.0/24	192.168.26.51	10	0	65515			
	10.10.26.141		0	65515			
	10.10.26.140		0	65515			
10.17.26.0/24	192.168.26.51	10	0	65515			
	10.17.26.140		0	65515			
	10.17.26.141		0	65515			

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., after configuring Global Reach. See that the route table has routes belonging to both the on-premises networks.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER



[Download](#)

[Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.1.26.0/25	192.168.26.49	10	0	65020
10.2.26.0/25	192.168.26.17		0	65021
10.10.26.0/24	192.168.26.49	10	0	65515
	10.10.26.140		0	65515
	10.10.26.141		0	65515
10.17.26.0/24	192.168.26.49	10	0	65515
	10.17.26.140		0	65515
	10.17.26.141		0	65515

Next steps

See [virtual network FAQ](#), for any further questions on VNet and VNet-peering. See [ExpressRoute FAQ](#) for any further questions on ExpressRoute and virtual network connectivity.

Global Reach is rolled out on a country/region by country/region basis. To see if Global Reach is available in the countries/regions that you want, see [ExpressRoute Global Reach](#).

Virtual network peering

2/18/2021 • 6 minutes to read • [Edit Online](#)

Virtual network peering enables you to seamlessly connect two or more [Virtual Networks](#) in Azure. The virtual networks appear as one for connectivity purposes. The traffic between virtual machines in peered virtual networks uses the Microsoft backbone infrastructure. Like traffic between virtual machines in the same network, traffic is routed through Microsoft's *private* network only.

Azure supports the following types of peering:

- Virtual network peering: Connect virtual networks within the same Azure region.
- Global virtual network peering: Connecting virtual networks across Azure regions.

The benefits of using virtual network peering, whether local or global, include:

- A low-latency, high-bandwidth connection between resources in different virtual networks.
- The ability for resources in one virtual network to communicate with resources in a different virtual network.
- The ability to transfer data between virtual networks across Azure subscriptions, Azure Active Directory tenants, deployment models, and Azure regions.
- The ability to peer virtual networks created through the Azure Resource Manager.
- The ability to peer a virtual network created through Resource Manager to one created through the classic deployment model. To learn more about Azure deployment models, see [Understand Azure deployment models](#).
- No downtime to resources in either virtual network when creating the peering, or after the peering is created.

Network traffic between peered virtual networks is private. Traffic between the virtual networks is kept on the Microsoft backbone network. No public Internet, gateways, or encryption is required in the communication between the virtual networks.

Connectivity

For peered virtual networks, resources in either virtual network can directly connect with resources in the peered virtual network.

The network latency between virtual machines in peered virtual networks in the same region is the same as the latency within a single virtual network. The network throughput is based on the bandwidth that's allowed for the virtual machine, proportionate to its size. There isn't any additional restriction on bandwidth within the peering.

The traffic between virtual machines in peered virtual networks is routed directly through the Microsoft backbone infrastructure, not through a gateway or over the public Internet.

You can apply network security groups in either virtual network to block access to other virtual networks or subnets. When configuring virtual network peering, either open or close the network security group rules between the virtual networks. If you open full connectivity between peered virtual networks, you can apply network security groups to block or deny specific access. Full connectivity is the default option. To learn more about network security groups, see [Security groups](#).

Service chaining

Service chaining enables you to direct traffic from one virtual network to a virtual appliance or gateway in a

peered network through user-defined routes.

To enable service chaining, configure user-defined routes that point to virtual machines in peered virtual networks as the *next hop* IP address. User-defined routes could also point to virtual network gateways to enable service chaining.

You can deploy hub-and-spoke networks, where the hub virtual network hosts infrastructure components such as a network virtual appliance or VPN gateway. All the spoke virtual networks can then peer with the hub virtual network. Traffic flows through network virtual appliances or VPN gateways in the hub virtual network.

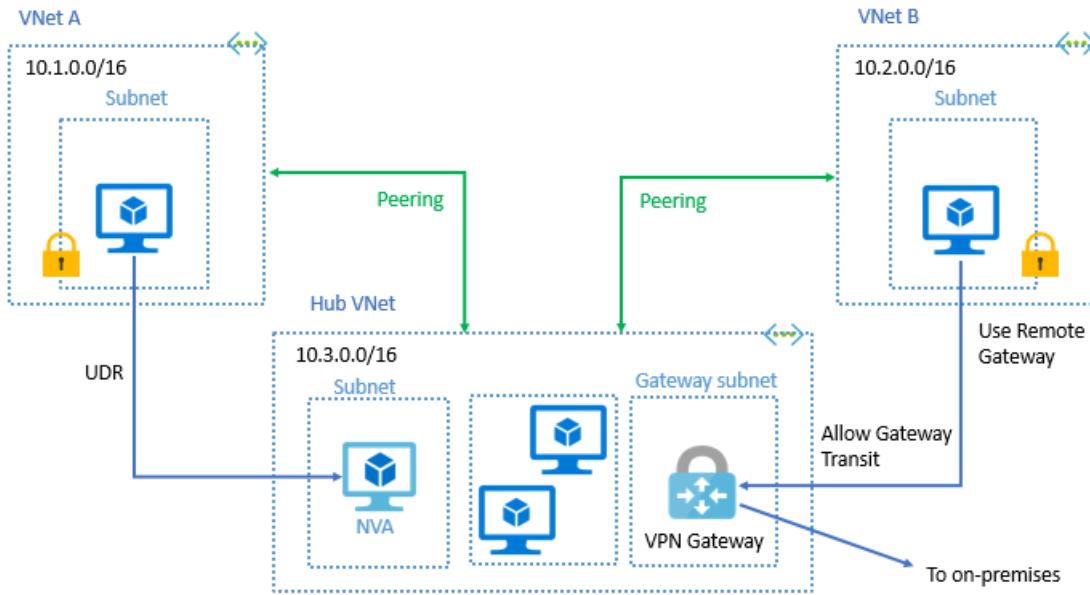
Virtual network peering enables the next hop in a user-defined route to be the IP address of a virtual machine in the peered virtual network, or a VPN gateway. You can't route between virtual networks with a user-defined route that specifies an Azure ExpressRoute gateway as the next hop type. To learn more about user-defined routes, see [User-defined routes overview](#). To learn how to create a hub and spoke network topology, see [Hub-spoke network topology in Azure](#).

Gateways and on-premises connectivity

Each virtual network, including a peered virtual network, can have its own gateway. A virtual network can use its gateway to connect to an on-premises network. You can also configure [virtual network-to-virtual network connections](#) by using gateways, even for peered virtual networks.

When you configure both options for virtual network interconnectivity, the traffic between the virtual networks flows through the peering configuration. The traffic uses the Azure backbone.

You can also configure the gateway in the peered virtual network as a transit point to an on-premises network. In this case, the virtual network that is using a remote gateway can't have its own gateway. A virtual network has only one gateway. The gateway is either a local or remote gateway in the peered virtual network, as shown in the following diagram:



Both virtual network peering and global virtual network peering support gateway transit.

Gateway transit between virtual networks created through different deployment models is supported. The gateway must be in the virtual network in the Resource Manager model. To learn more about using a gateway for transit, see [Configure a VPN gateway for transit in a virtual network peering](#).

When you peer virtual networks that share a single Azure ExpressRoute connection, the traffic between them goes through the peering relationship. That traffic uses the Azure backbone network. You can still use local gateways in each virtual network to connect to the on-premises circuit. Otherwise, you can use a shared

gateway and configure transit for on-premises connectivity.

Troubleshoot

To confirm that virtual networks are peered, you can check effective routes. Check routes for a network interface in any subnet in a virtual network. If a virtual network peering exists, all subnets within the virtual network have routes with next hop type *VNet peering*, for each address space in each peered virtual network. For more information, see [Diagnose a virtual machine routing problem](#).

You can also troubleshoot connectivity to a virtual machine in a peered virtual network using Azure Network Watcher. A connectivity check lets you see how traffic is routed from a source virtual machine's network interface to a destination virtual machine's network interface. For more information, see [Troubleshoot connections with Azure Network Watcher using the Azure portal](#).

You can also try the [Troubleshoot virtual network peering issues](#).

Constraints for peered virtual networks

The following constraints apply only when virtual networks are globally peered:

- Resources in one virtual network can't communicate with the front-end IP address of a Basic Internal Load Balancer (ILB) in a globally peered virtual network.
- Some services that use a Basic load balancer don't work over global virtual network peering. For more information, see [What are the constraints related to Global VNet Peering and Load Balancers?](#).

For more information, see [Requirements and constraints](#). To learn more about the supported number of peerings, see [Networking limits](#).

Permissions

To learn about permissions required to create a virtual network peering, see [Permissions](#).

Pricing

There's a nominal charge for ingress and egress traffic that uses a virtual network peering connection. For more information, see [Virtual Network pricing](#).

Gateway Transit is a peering property that enables a virtual network to utilize a VPN/ExpressRoute gateway in a peered virtual network. Gateway transit works for both cross premises and network-to-network connectivity. Traffic to the gateway (ingress or egress) in the peered virtual network incurs virtual network peering charges on the spoke VNet (or non-gateway VNet). For more information, see [VPN Gateway pricing](#) for VPN gateway charges and ExpressRoute Gateway pricing for ExpressRoute gateway charges.

NOTE

A previous version of this document stated that virtual network peering charges would not apply on the spoke VNet (or non-gateway VNet) with Gateway Transit. It now reflects accurate pricing per the pricing page.

Next steps

- You can create a peering between two virtual networks. The networks can belong to the same subscription, different deployment models in the same subscription, or different subscriptions. Complete a tutorial for one of the following scenarios:

AZURE DEPLOYMENT MODEL	SUBSCRIPTION
Both Resource Manager	Same
	Different
One Resource Manager, one classic	Same
	Different

- To learn how to create a hub and spoke network topology, see [Hub-spoke network topology in Azure](#).
- To learn about all virtual network peering settings, see [Create, change, or delete a virtual network peering](#).
- For answers to common virtual network peering and global virtual network peering questions, see [VNet Peering](#).

Integrate Azure services with virtual networks for network isolation

2/18/2021 • 4 minutes to read • [Edit Online](#)

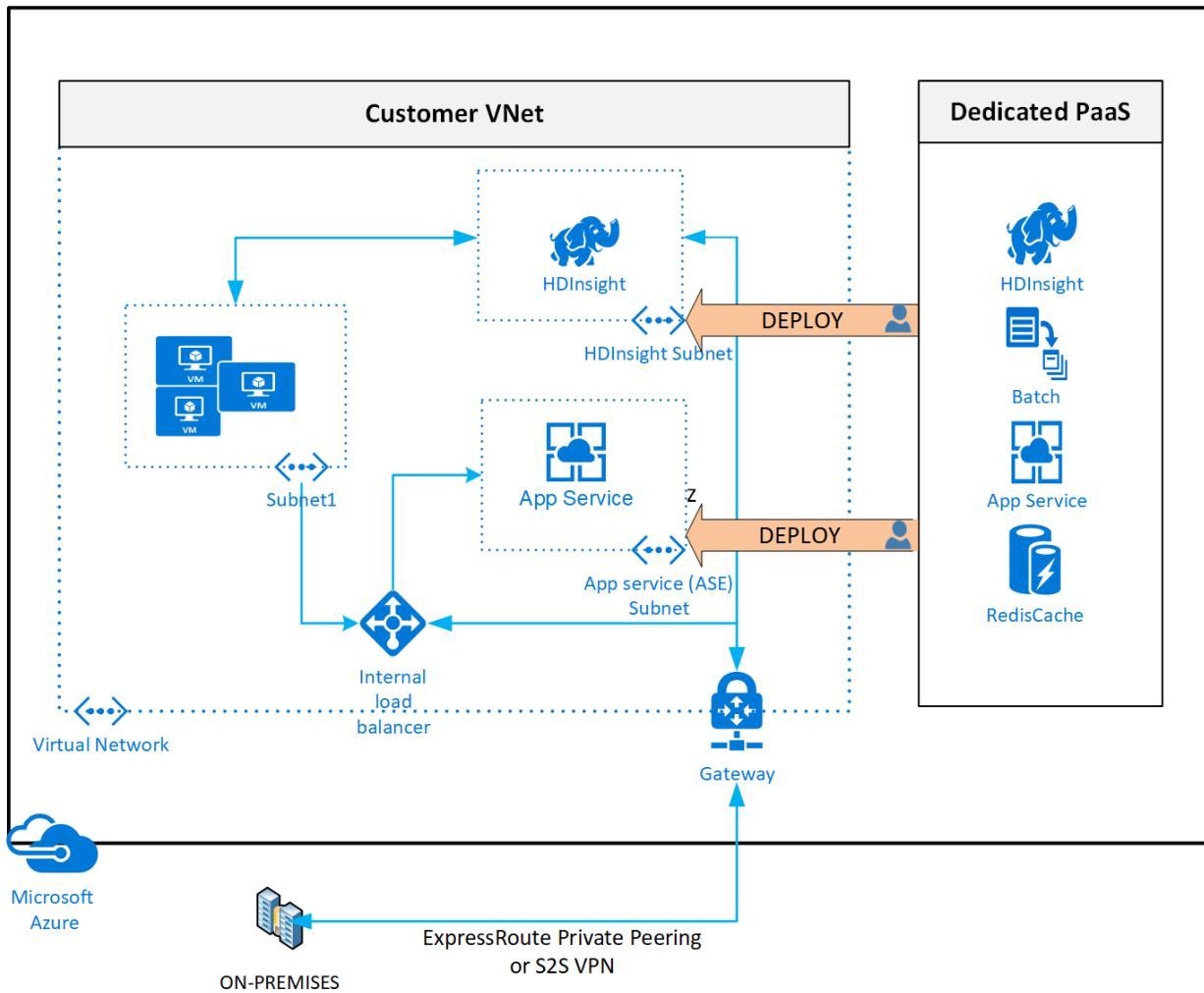
Virtual Network (VNet) integration for an Azure service enables you to lock down access to the service to only your virtual network infrastructure. The VNet infrastructure also includes peered virtual networks and on-premises networks.

VNet integration provides Azure services the benefits of network isolation and can be accomplished by one or more of the following methods:

- [Deploying dedicated instances of the service into a virtual network](#). The services can then be privately accessed within the virtual network and from on-premises networks.
- Using [Private Endpoint](#) that connects you privately and securely to a service powered by [Azure Private Link](#). Private Endpoint uses a private IP address from your VNet, effectively bringing the service into your virtual network.
- Accessing the service using public endpoints by extending a virtual network to the service, through [service endpoints](#). Service endpoints allow service resources to be secured to the virtual network.
- Using [service tags](#) to allow or deny traffic to your Azure resources to and from public IP endpoints.

Deploy dedicated Azure services into virtual networks

When you deploy dedicated Azure services in a virtual network, you can communicate with the service resources privately, through private IP addresses.



Deploying an dedicated Azure service into your virtual network provides the following capabilities:

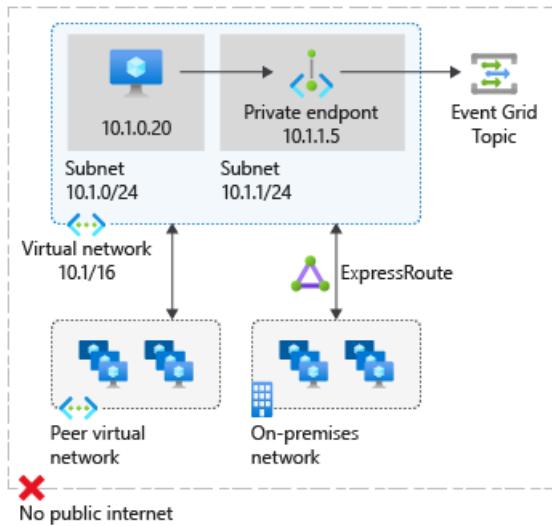
- Resources within the virtual network can communicate with each other privately, through private IP addresses. Example, directly transferring data between HDInsight and SQL Server running on a virtual machine, in the virtual network.
- On-premises resources can access resources in a virtual network using private IP addresses over a Site-to-Site VPN (VPN Gateway) or ExpressRoute.
- Virtual networks can be peered to enable resources in the virtual networks to communicate with each other, using private IP addresses.
- Service instances in a virtual network are typically fully managed by the Azure service. This includes monitoring the health of the resources and scaling with load.
- Service instances are deployed into a subnet in a virtual network. Inbound and outbound network access for the subnet must be opened through network security groups, per guidance provided by the service.
- Certain services also impose restrictions on the subnet they are deployed in, limiting the application of policies, routes or combining VMs and service resources within the same subnet. Check with each service on the specific restrictions as they may change over time. Examples of such services are Azure NetApp Files, Dedicated HSM, Azure Container Instances, App Service.
- Optionally, services might require a delegated subnet as an explicit identifier that a subnet can host a particular service. By delegating, services get explicit permissions to create service-specific resources in the delegated subnet.
- See an example of a REST API response on a virtual network with a delegated subnet. A comprehensive list of services that are using the delegated subnet model can be obtained via the Available Delegations API.

For a list of services that can be deployed into a virtual network, see [Deploy dedicated Azure services into virtual networks](#).

Private Link and Private Endpoints

You can use private endpoints to allow ingress of events directly from your virtual network to Azure resource securely over a private link without going through the public internet. A private endpoint is a special network interface for an Azure service in your virtual network. When you create a private endpoint for your Azure resource, it provides secure connectivity between clients on your virtual network and your Azure resource. The private endpoint is assigned an IP address from the IP address range of your virtual network. The connection between the private endpoint and the Azure service uses a secure private link.

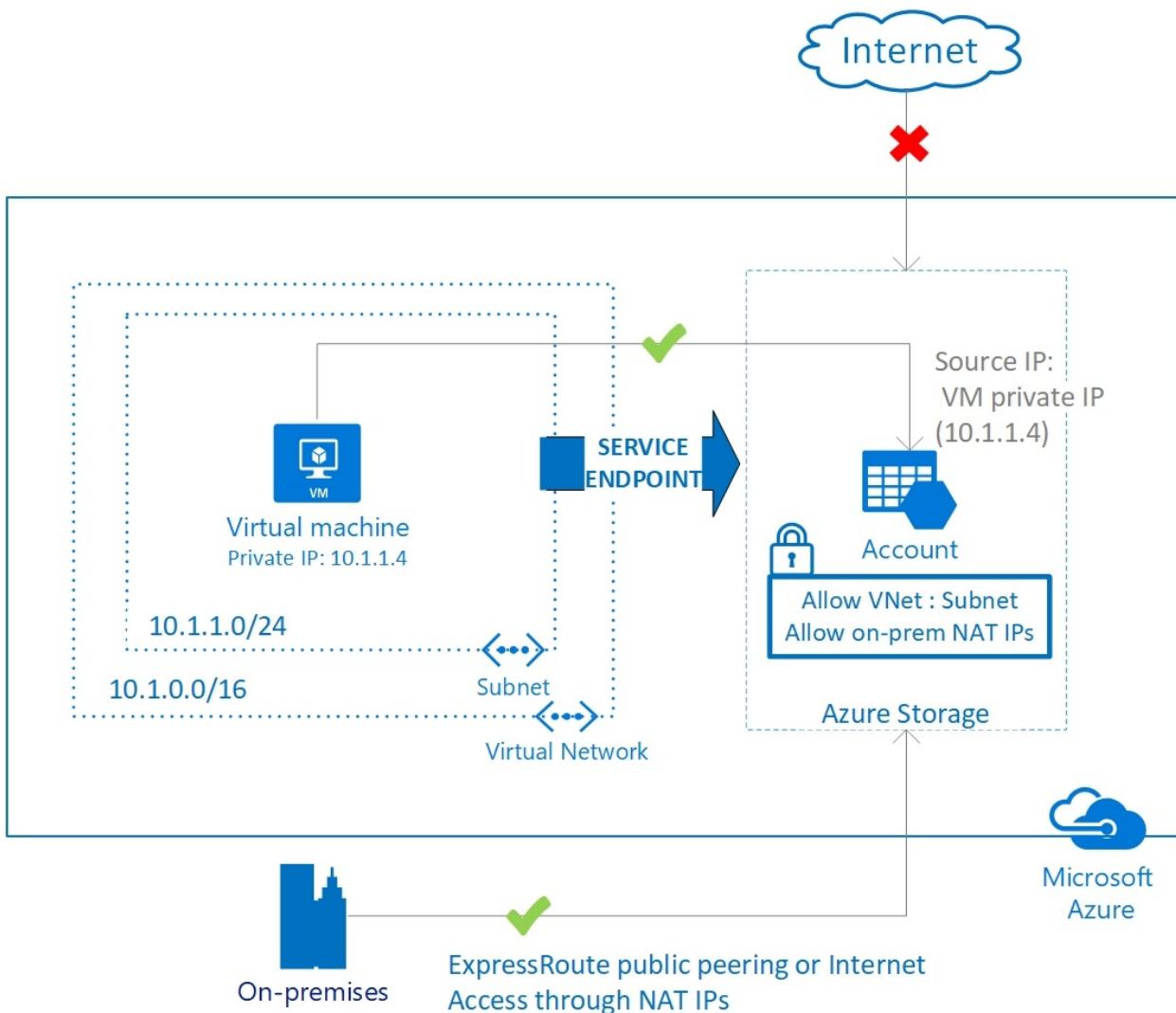
The following example shows private access of an Event Grid resource private endpoint that provides secure connectivity between clients on a virtual network and Event Grid resource.



For more information on Private Link and a list of Azure services that are supported, see [What is Private Link?](#)

Service endpoints

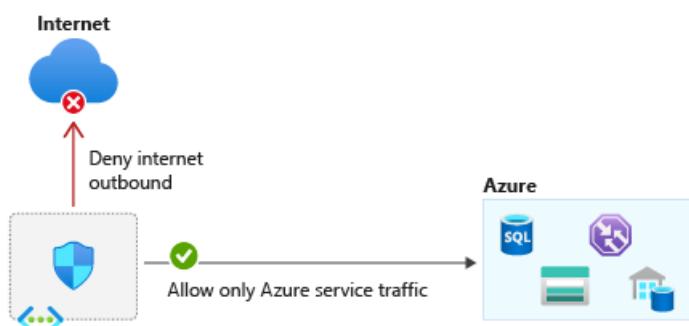
VNet service endpoint provides secure and direct connectivity to Azure services over an optimized route over the Azure backbone network. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Service Endpoints enables private IP addresses in the VNet to reach the endpoint of an Azure service without needing a public IP address on the VNet.



For more information, see [Virtual network service endpoints](#)

Service tags

A service tag represents a group of IP address prefixes from a given Azure service. Using service tags, you can define network access controls on [network security groups](#) or [Azure Firewall](#). By specifying the service tag name (for example, AzureEventGrid) in the appropriate source or destination field of a rule, you can allow or deny the traffic for the corresponding service.



Service Tags in a network security group

Action	Name	Source	Destination	Destination service tag	Protocol
Allow	AllowStorage	VirtualNetwork	Service Tag	Storage	Any
Allow	AllowSQL	VirtualNetwork	Service Tag	Sql.EastUS	Any
Deny	DenyAllOutBound	Any	Any	Any	Any

You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. Create inbound/outbound network security group rules to deny traffic to/from **Internet** and allow traffic to/from **AzureCloud** or other [available service tags](#) of specific Azure services.

For more information about Service Tags and Azure services that support them, see [Service Tags Overview](#)

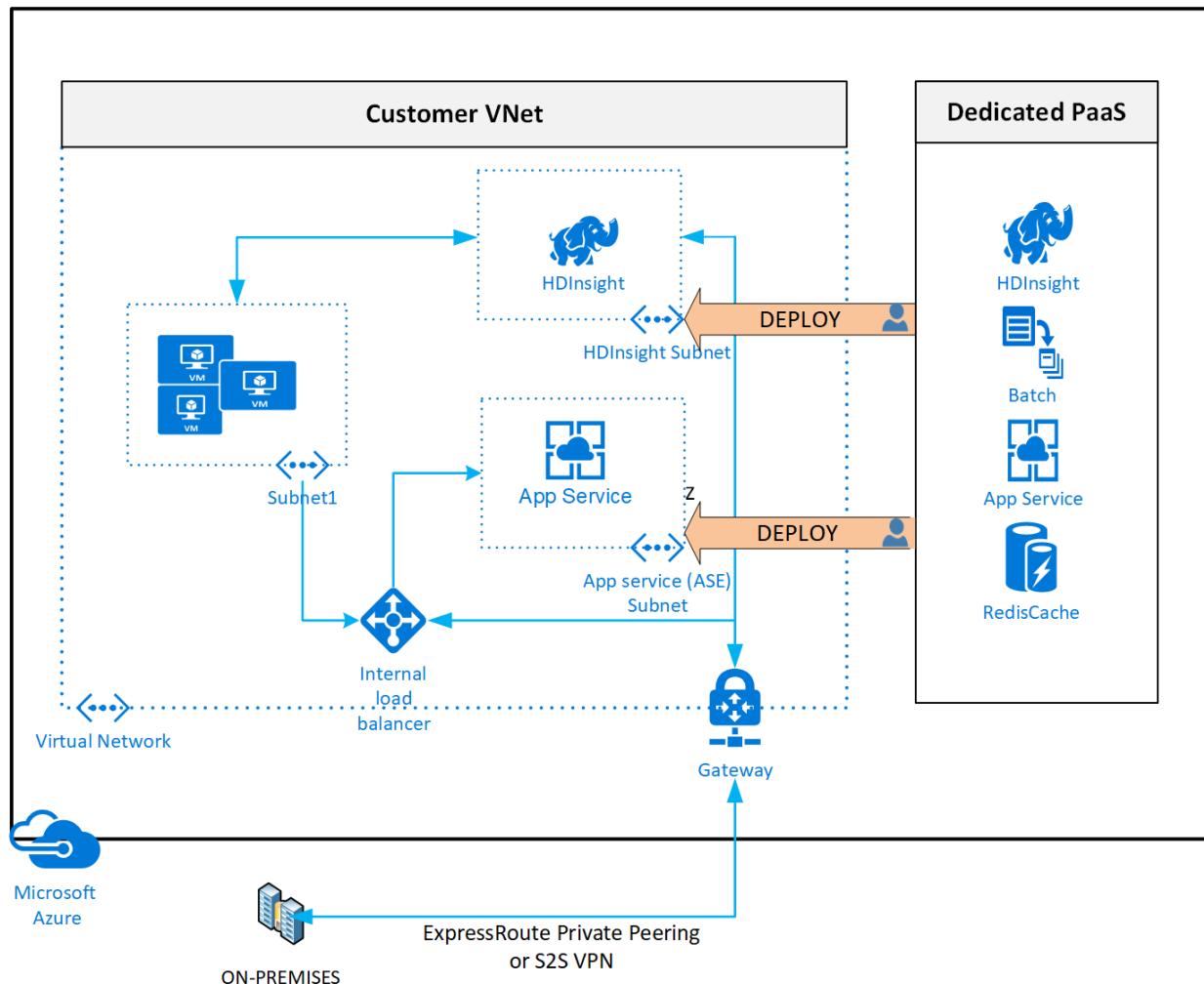
Next steps

- Learn how to [integrate your app with an Azure network](#).
- Learn how to [restrict access to resources using Service Tags](#).
- Learn how to [connect privately to an Azure Cosmos account using Azure Private Link](#).

Deploy dedicated Azure services into virtual networks

2/18/2021 • 2 minutes to read • [Edit Online](#)

When you deploy dedicated Azure services in a [virtual network](#), you can communicate with the service resources privately, through private IP addresses.



Deploying services within a virtual network provides the following capabilities:

- Resources within the virtual network can communicate with each other privately, through private IP addresses. Example, directly transferring data between HDInsight and SQL Server running on a virtual machine, in the virtual network.
- On-premises resources can access resources in a virtual network using private IP addresses over a [Site-to-Site VPN \(VPN Gateway\)](#) or [ExpressRoute](#).
- Virtual networks can be [peered](#) to enable resources in the virtual networks to communicate with each other, using private IP addresses.
- Service instances in a virtual network are typically fully managed by the Azure service. This includes monitoring the health of the resources and scaling with load.
- Service instances are deployed into a subnet in a virtual network. Inbound and outbound network access for the subnet must be opened through [network security groups](#), per guidance provided by the service.
- Certain services also impose restrictions on the subnet they are deployed in, limiting the application of policies, routes or combining VMs and service resources within the same subnet. Check with each service on

the specific restrictions as they may change over time. Examples of such services are Azure NetApp Files, Dedicated HSM, Azure Container Instances, App Service.

- Optionally, services might require a [delegated subnet](#) as an explicit identifier that a subnet can host a particular service. By delegating, services get explicit permissions to create service-specific resources in the delegated subnet.
- See an example of a REST API response on a [virtual network with a delegated subnet](#). A comprehensive list of services that are using the delegated subnet model can be obtained via the [Available Delegations API](#).

Services that can be deployed into a virtual network

CATEGORY	SERVICE	DEDICATED ¹ SUBNET
Compute	Virtual machines: Linux or Windows Virtual machine scale sets Cloud Service : Virtual network (classic) only Azure Batch	No No No No ²
Network	Application Gateway - WAF VPN Gateway Azure Firewall Azure Bastion Network Virtual Appliances	Yes Yes Yes Yes No
Data	RedisCache Azure SQL Managed Instance	Yes Yes
Analytics	Azure HDInsight Azure Databricks	No ² No ²
Identity	Azure Active Directory Domain Services	No
Containers	Azure Kubernetes Service (AKS) Azure Container Instance (ACI) Azure Container Service Engine with Azure Virtual Network CNI plug-in Azure Functions	No ² Yes No Yes
Web	API Management Web Apps App Service Environment Azure Logic Apps	Yes Yes Yes Yes
Hosted	Azure Dedicated HSM Azure NetApp Files	Yes Yes
Azure Spring Cloud	Deploy in Azure virtual network (VNet injection)	Yes

¹ 'Dedicated' implies that only service specific resources can be deployed in this subnet and cannot be combined with customer VM/VMSSs

² It is recommended as a best practice to have these services in a dedicated subnet, but not a mandatory requirement imposed by the service.

What is Azure Private Link?

2/18/2021 • 6 minutes to read • [Edit Online](#)

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a [private endpoint](#) in your virtual network.

Traffic between your virtual network and the service travels the Microsoft backbone network. Exposing your service to the public internet is no longer necessary. You can create your own [private link service](#) in your virtual network and deliver it to your customers. Setup and consumption using Azure Private Link is consistent across Azure PaaS, customer-owned, and shared partner services.

IMPORTANT

Azure Private Link is now generally available. Both Private Endpoint and Private Link service (service behind standard load balancer) are generally available. Different Azure PaaS will onboard to Azure Private Link at different schedules. Check [availability](#) section in this article for accurate status of Azure PaaS on Private Link. For known limitations, see [Private Endpoint](#) and [Private Link Service](#).

The screenshot shows the Azure Private Link Center interface. On the left, there's a sidebar with navigation links like 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES', 'Resource groups', 'All resources', 'Recent', 'App Services', 'SQL databases', 'Virtual machines (classic)', 'Virtual machines', 'Cloud services (classic)', 'Subscriptions', 'Azure Active Directory', 'Monitor', 'Security Center', 'Help + support', 'Advisor', 'Cost Management + Billing', and 'Virtual networks (classic)'. The main area has a title 'Private Link Center' and a sub-section 'Overview'. It includes a search bar, a 'Pending connections' section, and a 'Private endpoints' section. Below these are sections for 'Resources' (Active connections, Supported resources), 'Build a private connection to a service' (with a 'Create private endpoint' button), and 'Expose your own service so others can connect' (with a 'Create private link service' button). A central diagram illustrates the connection flow from a virtual machine through a private endpoint and private link service to a cloud provider.

Key benefits

Azure Private Link provides the following benefits:

- **Privately access services on the Azure platform:** Connect your virtual network to services in Azure without a public IP address at the source or destination. Service providers can render their services in their own virtual network and consumers can access those services in their local virtual network. The Private Link platform will handle the connectivity between the consumer and services over the Azure backbone network.
- **On-premises and peered networks:** Access services running in Azure from on-premises over ExpressRoute private peering, VPN tunnels, and peered virtual networks using private endpoints. There's no need to configure ExpressRoute Microsoft peering or traverse the internet to reach the service. Private Link provides a secure way to migrate workloads to Azure.
- **Protection against data leakage:** A private endpoint is mapped to an instance of a PaaS resource instead of the entire service. Consumers can only connect to the specific resource. Access to any other resource in the service is blocked. This mechanism provides protection against data leakage risks.
- **Global reach:** Connect privately to services running in other regions. The consumer's virtual network could be in region A and it can connect to services behind Private Link in region B.
- **Extend to your own services:** Enable the same experience and functionality to render your service privately to consumers in Azure. By placing your service behind a standard Azure Load Balancer, you can enable it for Private Link. The consumer can then connect directly to your service using a private endpoint in their own virtual network. You can manage the connection requests using an approval call flow. Azure Private Link works for consumers and services belonging to different Azure Active Directory tenants.

Availability

The following table lists the Private Link services and the regions where they're available.

SUPPORTED SERVICES	AVAILABLE REGIONS	ADDITIONAL CONSIDERATIONS	STATUS
Private Link services behind standard Azure Load Balancer	All public regions All Government regions All China regions	Supported on Standard Load Balancer	GA Learn how to create a private link service.
Azure Blob storage (including Data Lake Storage Gen2)	All public regions All Government regions	Supported on Account Kind General Purpose V2	GA Learn how to create a private endpoint for blob storage.
Azure Files	All public regions All Government regions		GA Learn how to create Azure Files network endpoints.
Azure File Sync	All public regions		GA Learn how to create Azure Files network endpoints.

Supported Services	Available Regions	Additional Considerations	Status	
Azure Queue storage	All public regions All Government regions	Supported on Account Kind General Purpose V2	GA Learn how to create a private endpoint for queue storage.	
Azure Table storage	All public regions All Government regions	Supported on Account Kind General Purpose V2	GA Learn how to create a private endpoint for table storage.	
Azure SQL Database	All public regions All Government regions All China regions	Supported for Proxy connection policy	GA Learn how to create a private endpoint for Azure SQL	
Azure Synapse Analytics	All public regions All Government regions	Supported for Proxy connection policy	GA Learn how to create a private endpoint for Azure Synapse Analytics.	
Azure Cosmos DB	All public regions All Government regions All China regions		GA Learn how to create a private endpoint for Cosmos DB.	
Azure Database for PostgreSQL - Single server	All public regions All Government regions All China regions	Supported for General Purpose and Memory Optimized pricing tiers	GA Learn how to create a private endpoint for Azure Database for PostgreSQL.	
Azure Database for MySQL	All public regions All Government regions All China regions		GA Learn how to create a private endpoint for Azure Database for MySQL.	
Azure Database for MariaDB	All public regions All Government regions All China regions		GA Learn how to create a private endpoint for Azure Database for MariaDB.	
Azure Digital Twins	All public regions supported by Azure Digital Twins		Preview Learn how to create a private endpoint for Azure Digital Twins.	
Azure Key Vault	All public regions All Government regions		GA Learn how to create a private endpoint for Azure Key Vault.	

Supported Services	Available Regions	Additional Considerations	Status	
Azure Kubernetes Service - Kubernetes API	All public regions		GA Learn how to create a private endpoint for Azure Kubernetes Service.	
Azure Search	All public regions All Government regions	Supported with service in Private Mode	GA Learn how to create a private endpoint for Azure Search.	
Azure Container Registry	All public regions All Government regions	Supported with premium tier of container registry. Select for tiers	GA Learn how to create a private endpoint for Azure Container Registry.	
Azure App Configuration	All public regions		Preview Learn how to create a private endpoint for Azure App Configuration	
Azure Backup	All public regions All Government regions		GA Learn how to create a private endpoint for Azure Backup.	
Azure Event Hub	All public regions All Government regions		GA Learn how to create a private endpoint for Azure Event Hub.	
Azure Service Bus	All public region All Government regions	Supported with premium tier of Azure Service Bus. Select for tiers	GA Learn how to create a private endpoint for Azure Service Bus.	
Azure Relay	All public regions		Preview Learn how to create a private endpoint for Azure Relay.	
Azure Event Grid	All public regions All Government regions		GA Learn how to create a private endpoint for Azure Event Grid.	
Azure Web Apps	All public regions	Supported with PremiumV2, PremiumV3, or Function Premium plan	GA Learn how to create a private endpoint for Azure Web Apps.	

Supported Services	Available Regions	Additional Considerations	Status	
Azure Machine Learning	All public regions		GA Learn how to create a private endpoint for Azure Machine Learning.	
Azure Automation	All public regions All Government regions		Preview Learn how to create a private endpoint for Azure Automation.	
Azure IoT Hub	All public regions		GA Learn how to create a private endpoint for Azure IoT Hub.	
Azure SignalR	EAST US, SOUTH CENTRAL US, WEST US 2, All China regions		Preview Learn how to create a private endpoint for Azure SignalR.	
Azure Monitor (Log Analytics & Application Insights)	All public regions		GA Learn how to create a private endpoint for Azure Monitor.	
Azure Batch	All public regions except: Germany CENTRAL, Germany NORTHEAST All Government regions		GA Learn how to create a private endpoint for Azure Batch.	
Azure Data Factory	All public regions All Government regions All China regions	Credentials need to be stored in an Azure key vault	GA Learn how to create a private endpoint for Azure Data Factory.	
Azure Managed Disks	All public regions All Government regions All China regions	Click here for known limitations	GA Learn how to create a private endpoint for Azure Managed Disks.	

For the most up-to-date notifications, check the [Azure Private Link updates page](#).

Logging and monitoring

Azure Private Link has integration with Azure Monitor. This combination allows:

- Archival of logs to a storage account.
- Streaming of events to your Event Hub.
- Azure Monitor logging.

You can access the following information on Azure Monitor:

- **Private endpoint:**
 - Data processed by the Private Endpoint (IN/OUT)
- **Private Link service:**
 - Data processed by the Private Link service (IN/OUT)
 - NAT port availability

Pricing

For pricing details, see [Azure Private Link pricing](#).

FAQs

For FAQs, see [Azure Private Link FAQs](#).

Limits

For limits, see [Azure Private Link limits](#).

Service Level Agreement

For SLA, see [SLA for Azure Private Link](#).

Next steps

- [Quickstart: Create a Private Endpoint using Azure portal](#)
- [Quickstart: Create a Private Link service by using the Azure portal](#)

Virtual Network service endpoints

2/18/2021 • 11 minutes to read • [Edit Online](#)

Virtual Network (VNet) service endpoint provides secure and direct connectivity to Azure services over an optimized route over the Azure backbone network. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Service Endpoints enables private IP addresses in the VNet to reach the endpoint of an Azure service without needing a public IP address on the VNet.

This feature is available for the following Azure services and regions. The *Microsoft.** resource is in parenthesis. Enable this resource from the subnet side while configuring service endpoints for your service:

Generally available

- [Azure Storage \(Microsoft.Storage\)](#): Generally available in all Azure regions.
- [Azure SQL Database \(Microsoft.Sql\)](#): Generally available in all Azure regions.
- [Azure Synapse Analytics \(Microsoft.Sql\)](#): Generally available in all Azure regions for dedicated SQL pools (formerly SQL DW).
- [Azure Database for PostgreSQL server \(Microsoft.Sql\)](#): Generally available in Azure regions where database service is available.
- [Azure Database for MySQL server \(Microsoft.Sql\)](#): Generally available in Azure regions where database service is available.
- [Azure Database for MariaDB \(Microsoft.Sql\)](#): Generally available in Azure regions where database service is available.
- [Azure Cosmos DB \(Microsoft.AzureCosmosDB\)](#): Generally available in all Azure regions.
- [Azure Key Vault \(Microsoft.KeyVault\)](#): Generally available in all Azure regions.
- [Azure Service Bus \(Microsoft.ServiceBus\)](#): Generally available in all Azure regions.
- [Azure Event Hubs \(Microsoft.EventHub\)](#): Generally available in all Azure regions.
- [Azure Data Lake Store Gen 1 \(Microsoft.AzureActiveDirectory\)](#): Generally available in all Azure regions where ADLS Gen1 is available.
- [Azure App Service \(Microsoft.Web\)](#): Generally available in all Azure regions where App service is available.
- [Azure Cognitive Services \(Microsoft.CognitiveServices\)](#): Generally available in all Azure regions where Cognitive services are available.

Public Preview

- [Azure Container Registry \(Microsoft.ContainerRegistry\)](#): Preview available in limited Azure regions where Azure Container Registry is available.

For the most up-to-date notifications, check the [Azure Virtual Network updates](#) page.

Key benefits

Service endpoints provide the following benefits:

- **Improved security for your Azure service resources:** VNet private address spaces can overlap. You can't use overlapping spaces to uniquely identify traffic that originates from your VNet. Service endpoints provide the ability to secure Azure service resources to your virtual network by extending VNet identity to the service. Once you enable service endpoints in your virtual network, you can add a virtual network rule to secure the Azure service resources to your virtual network. The rule addition provides improved security by fully removing public internet access to resources and allowing traffic only from your virtual

network.

- **Optimal routing for Azure service traffic from your virtual network:** Today, any routes in your virtual network that force internet traffic to your on-premises and/or virtual appliances also force Azure service traffic to take the same route as the internet traffic. Service endpoints provide optimal routing for Azure traffic.

Endpoints always take service traffic directly from your virtual network to the service on the Microsoft Azure backbone network. Keeping traffic on the Azure backbone network allows you to continue auditing and monitoring outbound Internet traffic from your virtual networks, through forced-tunneling, without impacting service traffic. For more information about user-defined routes and forced-tunneling, see [Azure virtual network traffic routing](#).

- **Simple to set up with less management overhead:** You no longer need reserved, public IP addresses in your virtual networks to secure Azure resources through IP firewall. There are no Network Address Translation (NAT) or gateway devices required to set up the service endpoints. You can configure service endpoints through a simple click on a subnet. There's no additional overhead to maintaining the endpoints.

Limitations

- The feature is available only to virtual networks deployed through the Azure Resource Manager deployment model.
- Endpoints are enabled on subnets configured in Azure virtual networks. Endpoints can't be used for traffic from your premises to Azure services. For more information, see [Secure Azure service access from on-premises](#)
- For Azure SQL, a service endpoint applies only to Azure service traffic within a virtual network's region. For Azure Storage, endpoints also extend to include paired regions where you deploy the virtual network to support Read-Access Geo-Redundant Storage (RA-GRS) and Geo-Redundant Storage (GRS) traffic. For more information, see [Azure paired regions](#).
- For Azure Data Lake Storage (ADLS) Gen 1, the VNet Integration capability is only available for virtual networks within the same region. Also note that virtual network integration for ADLS Gen1 uses the virtual network service endpoint security between your virtual network and Azure Active Directory (Azure AD) to generate additional security claims in the access token. These claims are then used to authenticate your virtual network to your Data Lake Storage Gen1 account and allow access. The *Microsoft.AzureActiveDirectory* tag listed under services supporting service endpoints is used only for supporting service endpoints to ADLS Gen 1. Azure AD doesn't support service endpoints natively. For more information about Azure Data Lake Store Gen 1 VNet integration, see [Network security in Azure Data Lake Storage Gen1](#).

Secure Azure services to virtual networks

- A virtual network service endpoint provides the identity of your virtual network to the Azure service. Once you enable service endpoints in your virtual network, you can add a virtual network rule to secure the Azure service resources to your virtual network.
- Today, Azure service traffic from a virtual network uses public IP addresses as source IP addresses. With service endpoints, service traffic switches to use virtual network private addresses as the source IP addresses when accessing the Azure service from a virtual network. This switch allows you to access the services without the need for reserved, public IP addresses used in IP firewalls.

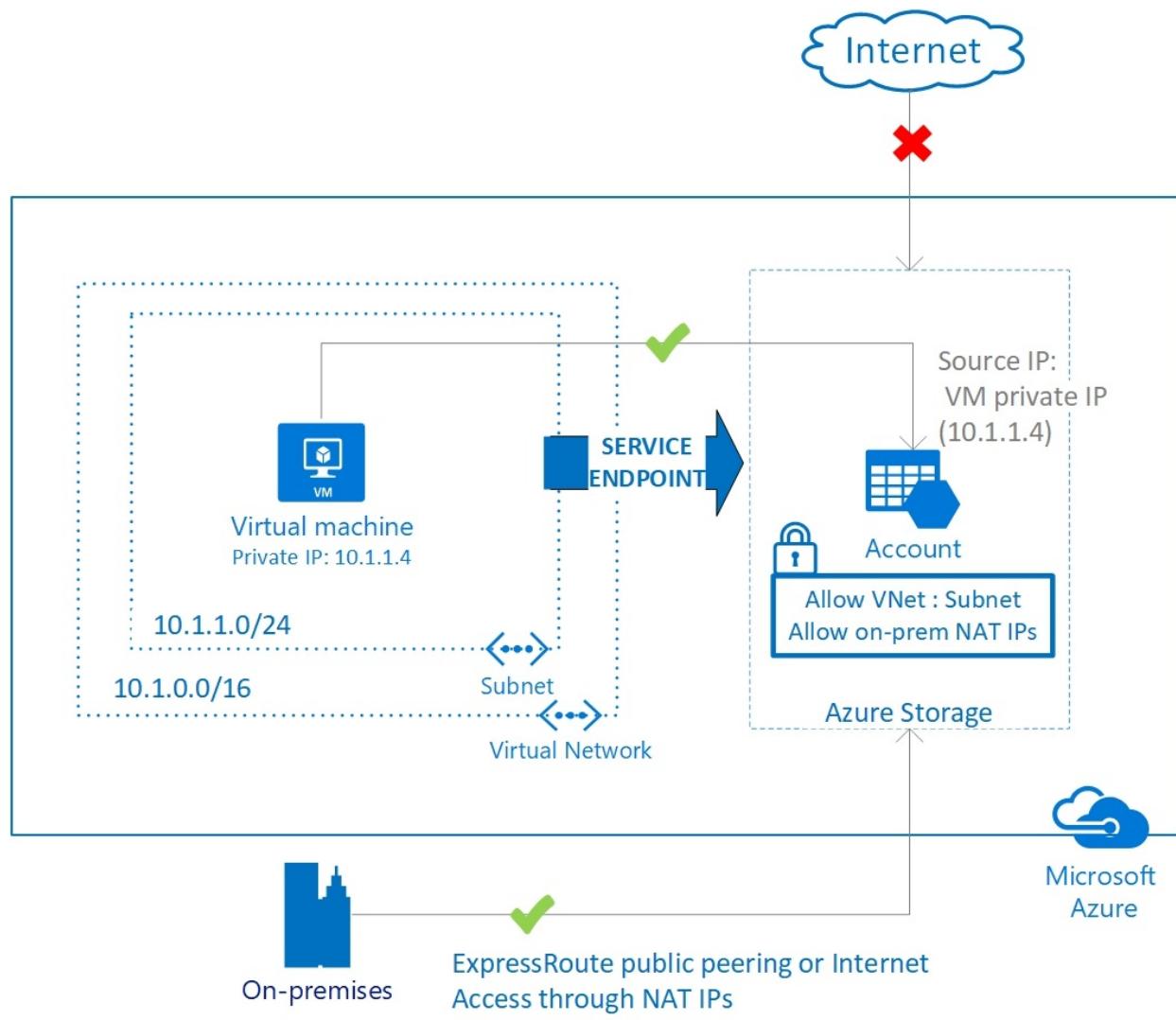
NOTE

With service endpoints, the source IP addresses of the virtual machines in the subnet for service traffic switches from using public IPv4 addresses to using private IPv4 addresses. Existing Azure service firewall rules using Azure public IP addresses will stop working with this switch. Please ensure Azure service firewall rules allow for this switch before setting up service endpoints. You may also experience temporary interruption to service traffic from this subnet while configuring service endpoints.

Secure Azure service access from on-premises

By default, Azure service resources secured to virtual networks aren't reachable from on-premises networks. If you want to allow traffic from on-premises, you must also allow public (typically, NAT) IP addresses from your on-premises or ExpressRoute. You can add these IP addresses through the IP firewall configuration for Azure service resources.

ExpressRoute: If you're using [ExpressRoute](#) for public peering or Microsoft peering from your premises, you'll need to identify the NAT IP addresses that you're using. For public peering, each ExpressRoute circuit uses two NAT IP addresses, by default, applied to Azure service traffic when the traffic enters the Microsoft Azure network backbone. For Microsoft peering, the NAT IP addresses are either customer provided or provided by the service provider. To allow access to your service resources, you must allow these public IP addresses in the resource IP firewall setting. To find your public peering ExpressRoute circuit IP addresses, [open a support ticket with ExpressRoute](#) via the Azure portal. For more information about NAT for ExpressRoute public and Microsoft peering, see [ExpressRoute NAT requirements](#).



- Configure service endpoints on a subnet in a virtual network. Endpoints work with any type of compute instances running within that subnet.
- You can configure multiple service endpoints for all supported Azure services (Azure Storage or Azure SQL Database, for example) on a subnet.
- For Azure SQL Database, virtual networks must be in the same region as the Azure service resource. If using GRS and RA-GRS Azure Storage accounts, the primary account must be in the same region as the virtual network. For all other services, you can secure Azure service resources to virtual networks in any region.
- The virtual network where the endpoint is configured can be in the same or different subscription than the Azure service resource. For more information on permissions required for setting up endpoints and securing Azure services, see [Provisioning](#).
- For supported services, you can secure new or existing resources to virtual networks using service endpoints.

Considerations

- After enabling a service endpoint, the source IP addresses switch from using public IPv4 addresses to using their private IPv4 address when communicating with the service from that subnet. Any existing open TCP connections to the service are closed during this switch. Ensure that no critical tasks are running when enabling or disabling a service endpoint to a service for a subnet. Also, ensure that your applications can automatically connect to Azure services after the IP address switch.

The IP address switch only impacts service traffic from your virtual network. There's no impact to any other traffic addressed to or from the public IPv4 addresses assigned to your virtual machines. For Azure services, if you have existing firewall rules using Azure public IP addresses, these rules stop working with the switch to virtual network private addresses.

- With service endpoints, DNS entries for Azure services remain as-is today and continue to resolve to public IP addresses assigned to the Azure service.
- Network security groups (NSGs) with service endpoints:
 - By default, NSGs allow outbound internet traffic and also allow traffic from your VNet to Azure services. This traffic continues to work with service endpoints as is.
 - If you want to deny all outbound internet traffic and allow only traffic to specific Azure services, you can do so using [service tags](#) in your NSGs. You can specify supported Azure services as destination in your NSG rules and Azure also provides the maintenance of IP addresses underlying each tag. For more information, see [Azure Service tags for NSGs](#).

Scenarios

- Peered, connected, or multiple virtual networks:** To secure Azure services to multiple subnets within a virtual network or across multiple virtual networks, you can enable service endpoints on each of the subnets independently, and secure Azure service resources to all of the subnets.
- Filtering outbound traffic from a virtual network to Azure services:** If you want to inspect or filter the traffic sent to an Azure service from a virtual network, you can deploy a network virtual appliance within the virtual network. You can then apply service endpoints to the subnet where the network virtual appliance is deployed, and secure Azure service resources only to this subnet. This scenario might be helpful if you want use network virtual appliance filtering to restrict Azure service access from your virtual network only to specific Azure resources. For more information, see [egress with network virtual appliances](#).
- Securing Azure resources to services deployed directly into virtual networks:** You can directly deploy various Azure services into specific subnets in a virtual network. You can secure Azure service resources to [managed service](#) subnets by setting up a service endpoint on the managed service subnet.
- Disk traffic from an Azure virtual machine:** Virtual Machine Disk traffic for managed and unmanaged disks isn't affected by service endpoints routing changes for Azure Storage. This traffic includes diskIO as well as mount and unmount. You can limit REST access to page blobs to select networks through service

endpoints and [Azure Storage network rules](#).

Logging and troubleshooting

Once you configure service endpoints to a specific service, validate that the service endpoint route is in effect by:

- Validating the source IP address of any service request in the service diagnostics. All new requests with service endpoints show the source IP address for the request as the virtual network private IP address, assigned to the client making the request from your virtual network. Without the endpoint, the address is an Azure public IP address.
- Viewing the effective routes on any network interface in a subnet. The route to the service:
 - Shows a more specific default route to address prefix ranges of each service
 - Has a nextHopType of *VirtualNetworkServiceEndpoint*
 - Indicates that a more direct connection to the service is in effect compared to any forced-tunneling routes

NOTE

Service endpoint routes override any BGP or UDR routes for the address prefix match of an Azure service. For more information, see [troubleshooting with effective routes](#).

Provisioning

Service endpoints can be configured on virtual networks independently by a user with write access to a virtual network. To secure Azure service resources to a VNet, the user must have permission to `Microsoft.Network/virtualNetworks/subnets/joinViaServiceEndpoint/action` for the added subnets. The built-in service administrator roles include this permission by default. You can modify the permission by creating custom roles.

For more information about built-in roles, see [Azure built-in roles](#). For more information about assigning specific permissions to custom roles, see [Azure custom roles](#).

Virtual networks and Azure service resources can be in the same or different subscriptions. Certain Azure Services (not all) such as Azure Storage and Azure Key Vault also support service endpoints across different Active Directory(AD) tenants i.e., the virtual network and Azure service resource can be in different Active Directory (AD) tenants. Please check individual service documentation for more details.

Pricing and limits

There's no additional charge for using service endpoints. The current pricing model for Azure services (Azure Storage, Azure SQL Database, etc.) applies as-is today.

There's no limit on the total number of service endpoints in a virtual network.

Certain Azure services, such as Azure Storage Accounts, may enforce limits on the number of subnets used for securing the resource. Refer to the documentation for various services in the [Next steps](#) section for details.

VNet service endpoint policies

VNet service endpoint policies allow you to filter virtual network traffic to Azure services. This filter allows only specific Azure service resources over service endpoints. Service endpoint policies provide granular access control for virtual network traffic to Azure services. For more information, see [Virtual Network Service Endpoint Policies](#).

FAQs

For FAQs, see [Virtual Network Service Endpoint FAQs](#).

Next steps

- [Configure virtual network service endpoints](#)
- [Secure an Azure Storage account to a virtual network](#)
- [Secure an Azure SQL Database to a virtual network](#)
- [Secure an Azure Synapse Analytics to a virtual network](#)
- [Azure service integration in virtual networks](#)
- [Virtual Network Service Endpoint Policies](#)
- [Azure Resource Manager template](#)

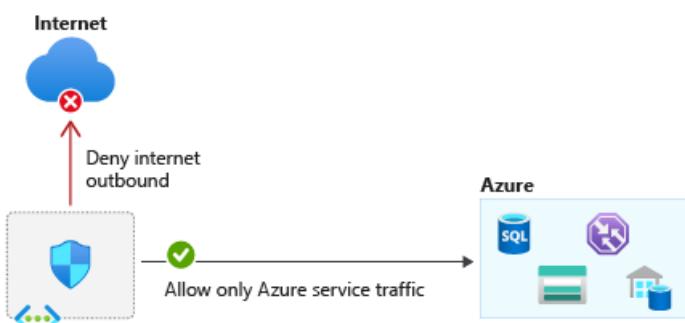
Virtual network service tags

2/18/2021 • 9 minutes to read • [Edit Online](#)

A service tag represents a group of IP address prefixes from a given Azure service. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change, minimizing the complexity of frequent updates to network security rules.

You can use service tags to define network access controls on [network security groups](#) or [Azure Firewall](#). Use service tags in place of specific IP addresses when you create security rules. By specifying the service tag name, such as **ApiManagement**, in the appropriate *source* or *destination* field of a rule, you can allow or deny the traffic for the corresponding service.

You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. Create inbound/outbound network security group rules to deny traffic to/from **Internet** and allow traffic to/from **AzureCloud** or other [available service tags](#) of specific Azure services.



Network Security Group (NSG)				
Action	Name	Source	Destination	Port
✓ Allow	AllowStorage	VirtualNetwork	Storage	Any
✓ Allow	AllowSQL	VirtualNetwork	Sql.EastUS	Any
✗ Deny	DenyAllOutBound	Any	Any	Any

Available service tags

The following table includes all the service tags available for use in [network security group](#) rules.

The columns indicate whether the tag:

- Is suitable for rules that cover inbound or outbound traffic.
- Supports [regional](#) scope.
- Is usable in [Azure Firewall](#) rules.

By default, service tags reflect the ranges for the entire cloud. Some service tags also allow more granular control by restricting the corresponding IP ranges to a specified region. For example, the service tag **Storage** represents Azure Storage for the entire cloud, but **Storage.WestUS** narrows the range to only the storage IP address ranges from the WestUS region. The following table indicates whether each service tag supports such regional scope.

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
ActionGroup	Action Group.	Inbound	No	No
ApiManagement	<p>Management traffic for Azure API Management - dedicated deployments.</p> <p><i>Note:</i> This tag represents the Azure API Management service endpoint for control plane per region. This enables customers to perform management operations on the APIs, Operations, Policies, NamedValues configured on the API Management service.</p>	Inbound	Yes	Yes
ApplicationInsightsAvailability	Application Insights Availability.	Inbound	No	No
AppConfiguration	App Configuration.	Outbound	No	No
AppService	Azure App Service. This tag is recommended for outbound security rules to web apps and Function apps.	Outbound	Yes	Yes
AppServiceManagement	Management traffic for deployments dedicated to App Service Environment.	Both	No	Yes
AzureActiveDirectory	Azure Active Directory.	Outbound	No	Yes
AzureActiveDirectoryDomainServices	Management traffic for deployments dedicated to Azure Active Directory Domain Services.	Both	No	Yes
AzureAdvancedThreatProtection	Azure Advanced Threat Protection.	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureArcInfrastructure	Azure Arc enabled servers, Azure Arc enabled Kubernetes, and Guest Configuration traffic. <i>Note:</i> This tag has a dependency on the AzureActiveDirectory , AzureTrafficManager , and AzureResourceManager tags. <i>This tag is not currently configurable via Azure Portal.</i>	Outbound	No	Yes
AzureBackup	Azure Backup. <i>Note:</i> This tag has a dependency on the Storage and AzureActiveDirectory tags.	Outbound	No	Yes
AzureBotService	Azure Bot Service.	Outbound	No	No
AzureCloud	All datacenter public IP addresses .	Outbound	Yes	Yes
AzureCognitiveSearch	Azure Cognitive Search. This tag or the IP addresses covered by this tag can be used to grant indexers secure access to data sources. Refer to the indexer connection documentation for more details. <i>Note:</i> The IP of the search service is not included in the list of IP ranges for this service tag and also needs to be added to the IP firewall of data sources.	Inbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureConnectors	This tag represents the IP addresses used for managed connectors that make inbound webhook callbacks to the Azure Logic Apps service and outbound calls to their respective services, for example, Azure Storage or Azure Event Hubs.	Inbound / Outbound	Yes	Yes
AzureContainerRegistry	Azure Container Registry.	Outbound	Yes	Yes
AzureCosmosDB	Azure Cosmos DB.	Outbound	Yes	Yes
AzureDatabricks	Azure Databricks.	Both	No	No
AzureDataExplorerManagement	Azure Data Explorer Management.	Inbound	No	No
AzureDataLake	Azure Data Lake Storage Gen1.	Outbound	No	Yes
AzureDevSpaces	Azure Dev Spaces.	Outbound	No	No
AzureDevOps	Azure Dev Ops. <i>Note: This tag is not currently configurable via Azure Portal</i>	Inbound	No	Yes
AzureDigitalTwins	Azure Digital Twins. <i>Note: This tag or the IP addresses covered by this tag can be used to restrict access to endpoints configured for event routes. This tag is not currently configurable via Azure Portal</i>	Inbound	No	Yes
AzureEventGrid	Azure Event Grid.	Both	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureFrontDoor.Frontend AzureFrontDoor.Backend AzureFrontDoor.FirstParty	Azure Front Door.	Both	No	No
AzureInformationProtection	Azure Information Protection. <i>Note:</i> This tag has a dependency on the AzureActiveDirectory , AzureFrontDoor.Frontend and AzureFrontDoor.FirstParty tags.	Outbound	No	No
AzureIoTHub	Azure IoT Hub.	Outbound	No	No
AzureKeyVault	Azure Key Vault. <i>Note:</i> This tag has a dependency on the AzureActiveDirectory tag.	Outbound	Yes	Yes
AzureLoadBalancer	The Azure infrastructure load balancer. The tag translates to the virtual IP address of the host (168.63.129.16) where the Azure health probes originate. This only includes probe traffic, not real traffic to your backend resource. If you're not using Azure Load Balancer, you can override this rule.	Both	No	No
AzureMachineLearning	Azure Machine Learning.	Both	No	Yes

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureMonitor	<p>Log Analytics, Application Insights, AzMon, and custom metrics (GiG endpoints).</p> <p><i>Note:</i> For Log Analytics, this tag has a dependency on the Storage tag.</p>	Outbound	No	Yes
AzureOpenDatasets	<p>Azure Open Datasets.</p> <p><i>Note:</i> This tag has a dependency on the AzureFrontDoor.Frontend and Storage tag.</p>	Outbound	No	No
AzurePlatformDNS	<p>The basic infrastructure (default) DNS service.</p> <p>You can use this tag to disable the default DNS. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No
AzurePlatformIMDS	<p>Azure Instance Metadata Service (IMDS), which is a basic infrastructure service.</p> <p>You can use this tag to disable the default IMDS. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzurePlatformLKM	<p>Windows licensing or key management service.</p> <p>You can use this tag to disable the defaults for licensing. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No
AzureResourceManager	Azure Resource Manager.	Outbound	No	No
AzureSignalR	Azure SignalR.	Outbound	No	No
AzureSiteRecovery	Azure Site Recovery.	<p>Outbound</p> <p><i>Note:</i> This tag has a dependency on the AzureActiveDirectory, AzureKeyVault, EventHub, GuestAndHybridManagement and Storage tags.</p>	No	No
AzureTrafficManager	<p>Azure Traffic Manager probe IP addresses.</p> <p>For more information on Traffic Manager probe IP addresses, see Azure Traffic Manager FAQ.</p>	Inbound	No	Yes
BatchNodeManagement	Management traffic for deployments dedicated to Azure Batch.	Both	No	Yes
CognitiveServicesManagement	The address ranges for traffic for Azure Cognitive Services.	Both	No	No
DataFactory	Azure Data Factory	Both	No	No
DataFactoryManagement	Management traffic for Azure Data Factory.	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
Dynamics365ForMarketingEmail	The address ranges for the marketing email service of Dynamics 365.	Outbound	Yes	No
EventHub	Azure Event Hubs.	Outbound	Yes	Yes
GatewayManager	Management traffic for deployments dedicated to Azure VPN Gateway and Application Gateway.	Inbound	No	No
GuestAndHybridManagement	Azure Automation and Guest Configuration.	Outbound	No	Yes
HDInsight	Azure HDInsight.	Inbound	Yes	No
Internet	The IP address space that's outside the virtual network and reachable by the public internet. The address range includes the Azure-owned public IP address space .	Both	No	No
LogicApps	Logic Apps.	Both	No	No
LogicAppsManagement	Management traffic for Logic Apps.	Inbound	No	No
MicrosoftCloudAppSecurity	Microsoft Cloud App Security.	Outbound	No	No
MicrosoftContainerRegistry	Container registry for Microsoft container images. <i>Note:</i> This tag has a dependency on the AzureFrontDoor.FirstParty tag.	Outbound	Yes	Yes
PowerBI	PowerBi. <i>Note: This tag is not currently configurable via Azure Portal.</i>	Both	No	No
PowerQueryOnline	Power Query Online.	Both	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
ServiceBus	Azure Service Bus traffic that uses the Premium service tier.	Outbound	Yes	Yes
ServiceFabric	Azure Service Fabric. <i>Note:</i> This tag represents the Service Fabric service endpoint for control plane per region. This enables customers to perform management operations for their Service Fabric clusters from their VNET (endpoint eg. https://westus.servicefabric.azure.com)	Both	No	No
Sql	Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Synapse Analytics. <i>Note:</i> This tag represents the service, but not specific instances of the service. For example, the tag represents the Azure SQL Database service, but not a specific SQL database or server. This tag does not apply to SQL managed instance.	Outbound	Yes	Yes
SqlManagement	Management traffic for SQL-dedicated deployments.	Both	No	Yes

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
Storage	Azure Storage. <i>Note:</i> This tag represents the service, but not specific instances of the service. For example, the tag represents the Azure Storage service, but not a specific Azure Storage account.	Outbound	Yes	Yes
StorageSyncService	Storage Sync Service.	Both	No	No
WindowsVirtualDesktop	Windows Virtual Desktop.	Both	No	Yes
VirtualNetwork	The virtual network address space (all IP address ranges defined for the virtual network), all connected on-premises address spaces, peered virtual networks, virtual networks connected to a virtual network gateway , the virtual IP address of the host , and address prefixes used on user-defined routes . This tag might also contain default routes.	Both	No	No

NOTE

In the classic deployment model (before Azure Resource Manager), a subset of the tags listed in the previous table are supported. These tags are spelled differently:

CLASSIC SPELLING	EQUIVALENT RESOURCE MANAGER TAG
AZURE_LOADBALANCER	AzureLoadBalancer
INTERNET	Internet
VIRTUAL_NETWORK	VirtualNetwork

NOTE

Service tags of Azure services denote the address prefixes from the specific cloud being used. For example, the underlying IP ranges that correspond to the `Sql` tag value on the Azure Public cloud will be different from the underlying ranges on the Azure China cloud.

NOTE

If you implement a [virtual network service endpoint](#) for a service, such as Azure Storage or Azure SQL Database, Azure adds a [route](#) to a virtual network subnet for the service. The address prefixes in the route are the same address prefixes, or CIDR ranges, as those of the corresponding service tag.

Service tags on-premises

You can obtain the current service tag and range information to include as part of your on-premises firewall configurations. This information is the current point-in-time list of the IP ranges that correspond to each service tag. You can obtain the information programmatically or via a JSON file download, as described in the following sections.

Use the Service Tag Discovery API (public preview)

You can programmatically retrieve the current list of service tags together with IP address range details:

- [REST](#)
- [Azure PowerShell](#)
- [Azure CLI](#)

NOTE

While it's in public preview, the Discovery API might return information that's less current than information returned by the JSON downloads. (See the next section.)

Discover service tags by using downloadable JSON files

You can download JSON files that contain the current list of service tags together with IP address range details. These lists are updated and published weekly. Locations for each cloud are:

- [Azure Public](#)
- [Azure US Government](#)
- [Azure China](#)
- [Azure Germany](#)

The IP address ranges in these files are in CIDR notation.

NOTE

A subset of this information has been published in XML files for [Azure Public](#), [Azure China](#), and [Azure Germany](#). These XML downloads will be deprecated by June 30, 2020 and will no longer be available after that date. You should migrate to using the Discovery API or JSON file downloads as described in the previous sections.

Tips

- You can detect updates from one publication to the next by noting increased `changeNumber` values in the JSON file. Each subsection (for example, `Storage.WestUS`) has its own `changeNumber` that's incremented

as changes occur. The top level of the file's *changeNumber* is incremented when any of the subsections is changed.

- For examples of how to parse the service tag information (for example, get all address ranges for Storage in WestUS), see the [Service Tag Discovery API PowerShell](#) documentation.
- When new IP addresses are added to service tags, they will not be used in Azure for at least one week. This gives you time to update any systems that might need to track the IP addresses associated with service tags.

Next steps

- Learn how to [create a network security group](#).

Public IP addresses

2/18/2021 • 7 minutes to read • [Edit Online](#)

Public IP addresses allow Internet resources to communicate inbound to Azure resources. Public IP addresses enable Azure resources to communicate to Internet and public-facing Azure services. The address is dedicated to the resource, until it's unassigned by you. A resource without a public IP assigned can communicate outbound. Azure dynamically assigns an available IP address that isn't dedicated to the resource. For more information about outbound connections in Azure, see [Understand outbound connections](#).

In Azure Resource Manager, a [public IP](#) address is a resource that has its own properties. Some of the resources you can associate a public IP address resource with:

- Virtual machine network interfaces
- Internet-facing load balancers
- VPN gateways
- Application gateways
- Azure Firewall

IP address version

Public IP addresses are created with an IPv4 or IPv6 address.

SKU

To learn about SKU upgrade, refer to [Public IP upgrade](#).

Public IP addresses are created with one of the following SKUs:

IMPORTANT

Matching SKUs are required for load balancer and public IP resources. You can't have a mixture of basic SKU resources and standard SKU resources. You can't attach standalone virtual machines, virtual machines in an availability set resource, or a virtual machine scale set resources to both SKUs simultaneously. New designs should consider using Standard SKU resources. Please review [Standard Load Balancer](#) for details.

Standard

Standard SKU public IP addresses:

- Always use static allocation method.
- Have an adjustable inbound originated flow idle timeout of 4-30 minutes, with a default of 4 minutes, and fixed outbound originated flow idle timeout of 4 minutes.
- Secure by default and closed to inbound traffic. Allow list inbound traffic with a [network security group](#).
- Assigned to network interfaces, standard public load balancers, or Application Gateways. For more information about Standard load balancer, see [Azure Standard Load Balancer](#).
- Can be zone-redundant (advertized from all 3 zones), zonal (guaranteed in a specific pre-selected availability zone), or no-zone (not associated with a specific pre-selected availability zone). To learn more about availability zones, see [Availability zones overview](#) and [Standard Load Balancer and Availability Zones](#). **Zone redundant IPs can only be created in regions where 3 availability zones are live.** IPs created before zones are live will not be zone redundant.

- Can be used as anycast frontend IPs for [cross-region load balancers](#) (preview functionality).

NOTE

Inbound communication with a Standard SKU resource fails until you create and associate a [network security group](#) and explicitly allow the desired inbound traffic.

NOTE

Only Public IP addresses with basic SKU are available when using [instance metadata service IMDS](#). Standard SKU is not supported.

NOTE

Diagnostic settings does not appear under the resource blade when using a Standard SKU Public IP address. To enable logging on your Standard Public IP address resource navigate to diagnostic settings under the Azure Monitor blade and select your IP address resource.

Basic

All public IP addresses created before the introduction of SKUs are Basic SKU public IP addresses.

With the introduction of SKUs, specify which SKU you would like the public IP address to be.

Basic SKU addresses:

- Assigned with the static or dynamic allocation method.
- Have an adjustable inbound originated flow idle timeout of 4-30 minutes, with a default of 4 minutes, and fixed outbound originated flow idle timeout of 4 minutes.
- Are open by default. Network security groups are recommended but optional for restricting inbound or outbound traffic.
- Assigned to any Azure resource that can be assigned a public IP address, such as:
 - Network interfaces
 - VPN Gateways
 - Application Gateways
 - Public load balancers
- Don't support Availability Zone scenarios. Use Standard SKU public IP for Availability Zone scenarios. To learn more about availability zones, see [Availability zones overview](#) and [Standard Load Balancer and Availability Zones](#).

Allocation method

Basic and standard public IPs support **static** assignment. The resource is assigned an IP address at the time it's created. The IP address is released when the resource is deleted.

Basic SKU public IP addresses support a **dynamic** assignment. Dynamic is the default assignment method. The IP address **isn't** given to the resource at the time of creation when selecting dynamic.

The IP is assigned when you associate the public IP address resource with a:

- Virtual machine
- The first virtual machine is associated with the backend pool of a load balancer.

The IP address is released when you stop (or delete) the resource.

For example, a public IP resource is released from a resource named **Resource A**. **Resource A** receives a different IP on start-up if the public IP resource is reassigned.

The IP address is released when the allocation method is changed from **static** to **dynamic**. To ensure the IP address for the associated resource remains the same, set the allocation method explicitly to **static**. A static IP address is assigned immediately.

NOTE

Even when you set the allocation method to **static**, you cannot specify the actual IP address assigned to the public IP address resource. Azure assigns the IP address from a pool of available IP addresses in the Azure location the resource is created in.

Static public IP addresses are commonly used in the following scenarios:

- When you must update firewall rules to communicate with your Azure resources.
- DNS name resolution, where a change in IP address would require updating A records.
- Your Azure resources communicate with other apps or services that use an IP address-based security model.
- You use TLS/SSL certificates linked to an IP address.

NOTE

Azure allocates public IP addresses from a range unique to each region in each Azure cloud. You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds.

DNS hostname resolution

Select the option to specify a DNS domain name label for a public IP resource.

This selection creates a mapping for **domainnamelabel.location.cloudapp.azure.com** to the public IP in the Azure-managed DNS.

For instance, creation of a public IP with:

- **contoso** as a **domainnamelabel**
- **West US** **Azure location**

The fully qualified domain name (FQDN) **contoso.westus.cloudapp.azure.com** resolves to the public IP address of the resource.

IMPORTANT

Each domain name label created must be unique within its Azure location.

DNS Recommendations

If a region move is needed, you can't migrate the FQDN of your public IP. Use the FQDN to create a custom CNAME record pointing to the public IP address.

If a move to a different public IP is required, update the CNAME record instead of updating the FQDN.

You can use [Azure DNS](#) or an external DNS provider for your DNS Record.

Virtual machines

You can associate a public IP address with a [Windows](#) or [Linux](#) virtual machine by assigning it to its **network interface**.

Choose **dynamic** or **static** for the public IP address. Learn more about [assigning IP addresses to network interfaces](#).

Internet-facing load balancers

You can associate a public IP address of either [SKU](#) with an [Azure Load Balancer](#), by assigning it to the load balancer **frontend** configuration. The public IP serves as a load-balanced IP.

You can assign either a dynamic or a static public IP address to a load balancer front end. You can assign multiple public IP addresses to a load balancer front end. This configuration enables [multi-VIP](#) scenarios like a multi-tenant environment with TLS-based websites.

For more information about Azure load balancer SKUs, see [Azure load balancer standard SKU](#).

VPN gateways

[Azure VPN Gateway](#) connects an Azure virtual network to:

- Azure virtual networks
- On-premises network(s).

A public IP address is assigned to the VPN Gateway to enable communication with the remote network. You can only assign a **dynamic** basic public IP address to a VPN gateway.

Application gateways

You can associate a public IP address with an Azure [Application Gateway](#), by assigning it to the gateway's **frontend** configuration.

- Assign a **dynamic** basic public IP to an application gateway V1 front-end configuration.
- Assign a **static** standard SKU address to a V2 front-end configuration.

Azure Firewall

[Azure Firewall](#) allows you to create, enforce, and log application and network connectivity policies across subscriptions and virtual networks.

You can only associate **static** standard public IP addresses with a firewall. This allows outside firewalls to identify traffic originating from your virtual network.

At-a-glance

The following table shows the property through which a public IP can be associated to a top-level resource and the possible allocation methods.

TOP-LEVEL RESOURCE	IP ADDRESS ASSOCIATION	DYNAMIC	STATIC
Virtual machine	Network interface	Yes	Yes
Internet-facing Load balancer	Front-end configuration	Yes	Yes
VPN gateway	Gateway IP configuration	Yes	No

TOP-LEVEL RESOURCE	IP ADDRESS ASSOCIATION	DYNAMIC	STATIC
Application gateway	Front-end configuration	Yes (V1 only)	Yes (V2 only)
Azure Firewall	Front-end configuration	No	Yes

Limits

The limits for IP addressing are listed in the full set of [limits for networking](#) in Azure.

The limits are per region and per subscription. [Contact support](#) to increase the default limits up to the maximum limits based on your business needs.

Pricing

Public IP addresses may have a nominal charge. To learn more about IP address pricing in Azure, review the [IP address pricing](#) page.

Next steps

- Learn about [Private IP Addresses in Azure](#)
- [Deploy a VM with a static public IP using the Azure portal](#)

Private IP addresses

2/18/2021 • 3 minutes to read • [Edit Online](#)

Private IPs allow communication between resources in Azure.

Resources can be:

- Azure Services such as:
 - Virtual machine network interfaces
 - Internal load balancers (ILBs)
 - Application gateways
- In a [virtual network](#).
- On-premises network through a VPN gateway or ExpressRoute circuit.

Private IPs allow communication to these resources without the use of a public IP address.

Allocation method

Azure assigns private IP addresses to resources from the address range of the virtual network subnet where the resource is.

Azure reserves the first four addresses in each subnet address range. The addresses can't be assigned to resources. For example, if the subnet's address range is 10.0.0.0/16, addresses 10.0.0.0-10.0.0.3 and 10.0.255.255 are unavailable. IP addresses within the subnet's address range can only be assigned to one resource at a time.

There are two methods in which a private IP address is given:

- **Dynamic:** Azure assigns the next available unassigned or unreserved IP address in the subnet's address range. For example, Azure assigns 10.0.0.10 to a new resource, if addresses 10.0.0.4-10.0.0.9 are already assigned to other resources.

Dynamic is the default allocation method. Once assigned, dynamic IP addresses are released if a network interface is:

- Deleted
- Reassigned to a different subnet within the same virtual network.
- The allocation method is changed to static, and a different IP address is specified.

By default, Azure assigns the previous dynamically assigned address as the static address when you change the allocation method from dynamic to static.

- **Static:** You select and assign any unassigned or unreserved IP address in the subnet's address range.

For example, a subnet's address range is 10.0.0.0/16 and addresses 10.0.0.4-10.0.0.9 are assigned to other resources. You can assign any address between 10.0.0.10 - 10.0.255.254. Static addresses are only released if a network interface is deleted.

Azure assigns the static IP as the dynamic IP when the allocation method is changed. The reassignment occurs even if the address isn't the next available in the subnet. The address changes when the network interface is assigned to a different subnet.

To assign the network interface to a different subnet, you change the allocation method from static to dynamic. Assign the network interface to a different subnet, then change the allocation method back to

static. Assign an IP address from the new subnet's address range.

Virtual machines

One or more private IP addresses are assigned to one or more **network interfaces**. The network interfaces are assigned to a [Windows](#) or [Linux](#) virtual machine. You can specify the allocation method as either dynamic or static for each private IP address.

Internal DNS hostname resolution (for virtual machines)

Azure virtual machines are configured with [Azure-managed DNS servers](#) by default. You can explicitly configure custom DNS servers. These DNS servers provide internal name resolution for virtual machines that are within the same virtual network.

A mapping for the hostname to a virtual machine's private IP address is added to the Azure-managed DNS servers.

A hostname is mapped to the primary IP of the main network interface when a VM has:

- Multiple network interfaces
- Multiple IP addresses
- Both

VMs configured with Azure-managed DNS resolve the hostnames within the same virtual network. Use a custom DNS server to resolve host names of VMs in connected virtual networks.

Internal load balancers (ILB) & Application gateways

You can assign a private IP address to the **front-end** configuration of an:

- [Azure internal load balancer](#) (ILB)
- [Azure Application Gateway](#)

This private IP address serves as an internal endpoint. The internal endpoint is accessible only to the resources within its virtual network and the remote networks connected to it. A dynamic or static IP can be assigned.

At-a-glance

The following table shows the property through which a private IP can be associated to a resource.

The possible allocation methods that can be used are also displayed:

- Dynamic
- Static

TOP-LEVEL RESOURCE	IP ADDRESS ASSOCIATION	DYNAMIC	STATIC
Virtual machine	Network interface	Yes	Yes
Load balancer	Front-end configuration	Yes	Yes
Application gateway	Front-end configuration	Yes	Yes

Limits

The limits on IP addressing are found in the full set of [limits for networking](#) in Azure. The limits are per region and per subscription. [Contact support](#) to increase the default limits up to the maximum limits based on your

business needs.

Next steps

Learn about [Public IP Addresses in Azure](#)

- [Deploy a VM with a static private IP address using the Azure portal](#)

Public IP address prefix

2/18/2021 • 4 minutes to read • [Edit Online](#)

A public IP address prefix is a reserved range of IP addresses in Azure. Azure gives a contiguous range of addresses to your subscription based on how many you specify.

If you're not familiar with public addresses, see [Public IP addresses](#).

Public IP addresses are assigned from a pool of addresses in each Azure region. You can [download](#) the list of ranges Azure uses for each region. For example, 40.121.0.0/16 is one of over 100 ranges Azure uses in the East US region. The range includes the usable addresses of 40.121.0.1 - 40.121.255.254.

You create a public IP address prefix in an Azure region and subscription by specifying a name, and how many addresses you want the prefix to include.

Public IP address ranges are assigned with a prefix of your choosing. If you create a prefix of /28, Azure gives 16 ip addresses from one of its ranges.

You don't know which range Azure will assign until you create the range, but the addresses are contiguous.

Public IP address prefixes have a fee, for more information, see [public IP address pricing](#).

Why create a public IP address prefix?

When you create public IP address resources, Azure assigns an available public IP address from any of the ranges used in that region.

Until Azure assigns the IP address, you won't know the exact IP. This process can be problematic when you create firewall rules that allow specific IP addresses. For every IP address added, a corresponding firewall rule must be added.

When you assign addresses to your resources from a public IP address prefix, firewall rule updates aren't required. The entire range is added to the rule.

Benefits

- Creation of public IP address resources from a known range.
- Firewall rule configuration with ranges that include public IP addresses you've currently assigned, and addresses you haven't assigned yet. This configuration eliminates the need to change firewall rules as you assign IP addresses to new resources.
- The default size of a range you can create is /28 or 16 IP addresses.
- There aren't limits as to how many ranges you can create. There are limits on the maximum number of static public IP addresses you can have in an Azure subscription. The number of ranges you create can't encompass more static public IP addresses than you can have in your subscription. For more information, see [Azure limits](#).
- The addresses that you create using addresses from the prefix can be assigned to any Azure resource that you can assign a public IP address to.
- You can easily see which IP addresses that are given and not given within the range.

Scenarios

You can associate the following resources to a static public IP address from a prefix:

RESOURCE	SCENARIO	STEPS
Virtual machines	Associating public IPs from a prefix to your virtual machines in Azure reduces management overhead when adding IP addresses to an allow list in the firewall. You can add an entire prefix with a single firewall rule. As you scale with virtual machines in Azure, you can associate IPs from the same prefix saving cost, time, and management overhead.	To associate IPs from a prefix to your virtual machine: 1. Create a prefix . 2. Create an IP from the prefix . 3. Associate the IP to your virtual machine's network interface . You can also associate the IPs to a Virtual Machine Scale Set .
Standard load balancers	Associating public IPs from a prefix to your frontend IP configuration or outbound rule of a load balancer ensures simplification of your Azure public IP address space. Simplify your scenario by grooming outbound connections from a range of contiguous IP addresses.	To associate IPs from a prefix to your load balancer: 1. Create a prefix . 2. Create an IP from the prefix . 3. When creating the load balancer, select or update the IP created in step 2 above as the frontend IP of your load balancer.
Azure Firewall	You can use a public IP from a prefix for outbound SNAT. All outbound virtual network traffic is translated to the Azure Firewall public IP.	To associate an IP from a prefix to your firewall: 1. Create a prefix . 2. Create an IP from the prefix . 3. When you deploy the Azure firewall , be sure to select the IP you previously gave from the prefix.
Application Gateway v2	You can use a public IP from a prefix for your autoscaling and zone-redundant Application gateway v2.	To associate an IP from a prefix to your gateway: 1. Create a prefix . 2. Create an IP from the prefix . 3. When you deploy the Application Gateway , be sure to select the IP you previously gave from the prefix.

Constraints

- You can't specify the IP addresses for the prefix. Azure gives the IP addresses for the prefix, based on the size that you specify.
- You can create a prefix of up to 16 IP addresses or a /28 by default. Review [Network limits increase requests](#) and [Azure limits](#) for more information.
- You can't change the range, once you've created the prefix.
- Only static public IP addresses created with the Standard SKU can be assigned from the prefix's range. To learn more about public IP address SKUs, see [public IP address](#).
- Addresses from the range can only be assigned to Azure Resource Manager resources. Addresses can't be assigned to resources in the classic deployment model.
- All public IP addresses created from the prefix must exist in the same Azure region and subscription as the prefix. Addresses must be assigned to resources in the same region and subscription.
- You can't delete a prefix if any addresses within it are assigned to public IP address resources associated to a resource. Dissociate all public IP address resources that are assigned IP addresses from the prefix first.

Next steps

- [Create](#) a public IP address prefix

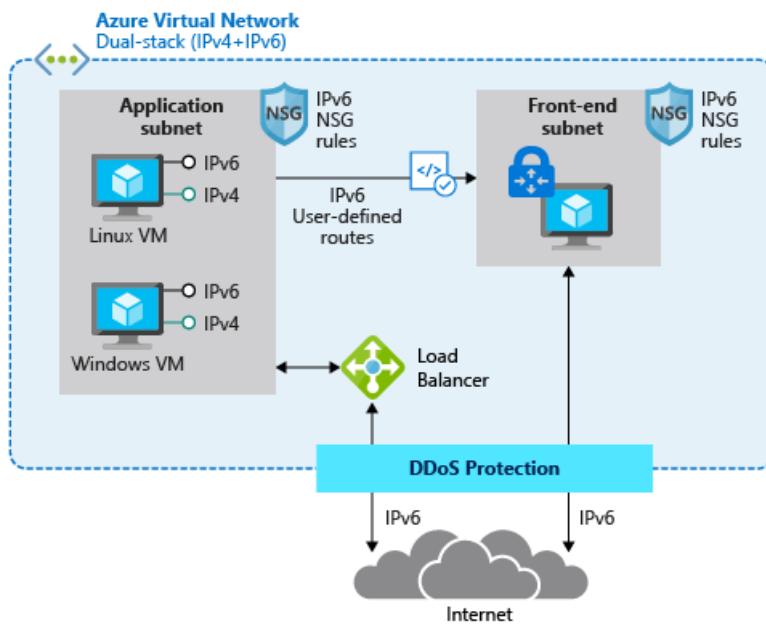
What is IPv6 for Azure Virtual Network?

2/18/2021 • 5 minutes to read • [Edit Online](#)

IPv6 for Azure Virtual Network (VNet) enables you to host applications in Azure with IPv6 and IPv4 connectivity both within a virtual network and to and from the Internet. Due to the exhaustion of public IPv4 addresses, new networks for mobility and Internet of Things (IoT) are often built on IPv6. Even long established ISP and mobile networks are being transformed to IPv6. IPv4-only services can find themselves at a real disadvantage in both existing and emerging markets. Dual stack IPv4/IPv6 connectivity enables Azure-hosted services to traverse this technology gap with globally available, dual-stacked services that readily connect with both the existing IPv4 and these new IPv6 devices and networks.

Azure's original IPv6 connectivity makes it easy to provide dual stack (IPv4/IPv6) Internet connectivity for applications hosted in Azure. It allows for simple deployment of VMs with load balanced IPv6 connectivity for both inbound and outbound initiated connections. This feature is still available and more information is available [here](#). IPv6 for Azure virtual network is much more full featured- enabling full IPv6 solution architectures to be deployed in Azure.

The following diagram depicts a simple dual stack (IPv4/IPv6) deployment in Azure:



Benefits

IPv6 for Azure VNET benefits:

- Helps expand the reach of your Azure-hosted applications into the growing mobile and Internet of Things markets.
- Dual stacked IPv4/IPv6 VMs provide maximum service deployment flexibility. A single service instance can connect with both IPv4 and IPv6-capable Internet clients.
- Builds on long-established, stable Azure VM-to-Internet IPv6 connectivity.
- Secure by default since IPv6 connectivity to the Internet is only established when you explicitly request it in your deployment.

Capabilities

IPv6 for Azure VNet includes the following capabilities:

- Azure customers can define their own IPv6 virtual network address space to meet the needs of their applications, customers, or seamlessly integrate into their on-premises IP space.
- Dual stack (IPv4 and IPv6) virtual networks with dual stack subnets enable applications to connect with both IPv4 and IPv6 resources in their virtual network or - the Internet.

IMPORTANT

The subnets for IPv6 must be exactly /64 in size. This ensures future compatibility should you decide to enable routing of the subnet to an on-premises network since some routers can only accept /64 IPv6 routes.

- Protect your resources with IPv6 rules for Network Security Groups.
 - And the Azure platform's Distributed Denial of Service (DDoS) protections are extended to Internet-facing Public IP's
- Customize the routing of IPv6 traffic in your virtual network with User-Defined Routes- especially when leveraging Network Virtual Appliances to augment your application.
- Linux and Windows Virtual Machines can all use IPv6 for Azure VNET
- [Standard IPv6 public Load Balancer](#) support to create resilient, scalable applications, which include:
 - Optional IPv6 health probe to determine which backend pool instances are health and thus can receive new flows.
 - Optional outbound rules which provide full declarative control over outbound connectivity to scale and tune this ability to your specific needs.
 - Optional multiple front-end configurations which enable a single load balancer to use multiple IPv6 public IP addresses- the same frontend protocol and port can be reused across frontend addresses.
 - Optional IPv6 ports can be reused on backend instances using the *Floating IP* feature of load-balancing rules
 - Note: Load balancing does not perform any protocol translation (no NAT64).
 - Note: IPv6 can be load balanced only to the primary network interface (NIC) on Azure VMs.
- [Standard IPv6 internal Load Balancer](#) support to create resilient multi-tier applications within Azure VNETs.
- Basic IPv6 public Load Balancer support for compatibility with legacy deployments
- [Reserved IPv6 Public IP addresses and address ranges](#) provide stable, predictable IPv6 addresses which ease filtering of your azure-hosted applications for your company and your customers.
- Instance-level Public IP provides IPv6 Internet connectivity directly to individual VMs.
- [Add IPv6 to Existing IPv4-only deployments](#)- this feature enables you to easily add IPv6 connectivity to existing IPv4-only deployments without the need to recreate deployments. The IPv4 network traffic is unaffected during this process so depending on your application and OS you may be able to add IPv6 even to live services.
- Let Internet clients seamlessly access your dual stack application using their protocol of choice with Azure DNS support for IPv6 (AAAA) records.
- Create dual stack applications that automatically scale to your load with virtual machine scale sets with IPv6.
- [Virtual Network \(VNET\) Peering](#) - both within-regional and global peering - enables you to seamlessly connect dual stack VNETs- both the IPv4 and IPv6 endpoints on VMs in the peered networks will be able to communicate with each other. You can even peer dual stack with IPv4-only VNETs as you are transitioning your deployments to dual stack.
- IPv6 Troubleshooting and Diagnostics are available with load balancer metrics/alerting and Network Watcher features such as packet capture, NSG flow logs, connection troubleshooting and connection monitoring.

Scope

IPv6 for Azure VNET is a foundational feature set which enables customers to host dual stack (IPv4+IPv6) applications in Azure. We intend to add IPv6 support to more Azure networking features over time and eventually to offer dual stack versions of Azure PaaS services but in the meantime all Azure PaaS services can be accessed via the IPv4 endpoints on dual stack Virtual Machines.

Limitations

The current IPv6 for Azure virtual network release has the following limitations:

- IPv6 for Azure virtual network is available in all global Azure Commercial and US Government regions using all deployment methods.
- ExpressRoute gateways CAN be used for IPv4-only traffic in a VNET with IPv6 enabled. Support for IPv6 traffic is on our roadmap.
- VPN gateways CANNOT be used in a VNET with IPv6 enabled, either directly or peered with "UseRemoteGateway".
- The Azure platform (AKS, etc.) does not support IPv6 communication for Containers.
- IPv6 can be load balanced only to the primary network interface (NIC) on Azure VMs. Load balancing IPv6 traffic to secondary NICs is not supported.
- IPv6-only Virtual Machines or Virtual Machines Scale Sets are not supported, each NIC must include at least one IPv4 IP configuration.
- When adding IPv6 to existing IPv4 deployments, IPv6 ranges can not be added to a VNET with existing resource navigation links.
- Forward DNS for IPv6 is supported for Azure public DNS today but Reverse DNS is not yet supported.

Pricing

IPv6 Azure resources and bandwidth are charged at the same rates as IPv4. There are no additional or different charges for IPv6. You can find details about pricing for [public IP addresses](#), [network bandwidth](#), or [Load Balancer](#).

Next steps

- Learn how to [deploy an IPv6 dual stack application using Azure PowerShell](#).
- Learn how to [deploy an IPv6 dual stack application using Azure CLI](#).
- Learn how to [deploy an IPv6 dual stack application using Resource Manager Templates \(JSON\)](#)

Reserved public IPv6 address prefix

2/18/2021 • 2 minutes to read • [Edit Online](#)

In Azure, dual stack (IPv4+IPv6) virtual networks (VNet) and virtual machines (VMs) are secure by default since they have no Internet connectivity. You can easily add an IPv6 Internet connectivity to your Azure Load Balancers and VMs with public IPv6 addresses that you obtain from Azure.

Any public IPs that you reserve are associated with an Azure region of your choice and with your Azure subscription. You may move a reserved (static) IPv6 public IP between any of the Azure Load Balancers or VMs in your subscription. You may dissociate the IPv6 public IP entirely and it will be held for your use when you're ready.

WARNING

Use caution to not delete your public IP addresses accidentally. Deleting a public IP removes it from your subscription and you will not be able to recover it (not even with the help of Azure support).

In addition to reserving individual IPv6 addresses, you can reserve contiguous ranges of Azure IPv6 addresses (known as IP prefix) for your use. Similar to individual IP addresses, reserved prefixes are associated with an Azure region of your choice and with your Azure subscription. Reserving a predictable, contiguous range of addresses has many uses. For example, you can greatly simplify IP *filtering* of your Azure-hosted applications by your company and your customers as your static IP ranges can be readily programmed into on-premises firewalls. You can create individual public IPs from your IP prefix as needed and when you delete those individual Public IPs they are *returned* to your reserved range so that you can reuse them later. All the IP addresses in your IP Prefix are reserved for your exclusive use until such time as you delete your Prefix.

IPv6 prefix sizes

The following public IP prefix sizes are available:

- Minimum IPv6 Prefix size: /127 = 2 addresses
- Maximum IPv6 Prefix size: /124 = 16 addresses

Prefix size is specified as a Classless Inter-Domain Routing (CIDR) mask size. For example, a mask of /128 represents an individual IPv6 address as IPv6 addresses are composed of 128 bits.

Pricing

For costs associated with using Azure Public IPs, both individual IP addresses and IP ranges, see [Public IP Address pricing](#).

Limitations

IPv6 is supported on Basic Public IPs only with "dynamic" allocation that means that the IPv6 address will change if you delete and redeploy your application (VM's or load balancers) in Azure. Standard IPv6 Public IP's support solely static (reserved) allocation though Standard INTERNAL load balancers can also support dynamic allocation from within the subnet to which they are assigned.

As a best practice, we recommend that you use Standard Public IPs and Standard Load Balancers for your IPv6 applications.

Next steps

- Reserve a public [IPv6 address prefix](#).
- Learn more about [IPv6 addresses](#).
- Learn about [how to create and use public IPs](#) (both IPv4 and IPv6) in Azure.

What is routing preference (preview)?

2/18/2021 • 3 minutes to read • [Edit Online](#)

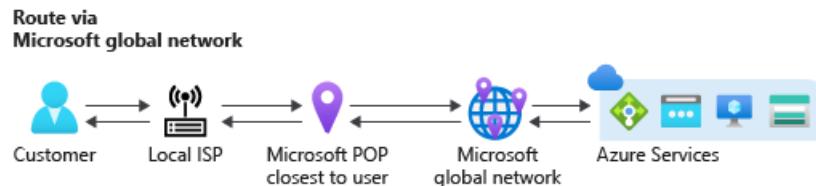
Azure routing preference enables you to choose how your traffic routes between Azure and the Internet. You can choose to route traffic either via the Microsoft network, or via the ISP network (public internet). These options are also referred to as *cold potato routing* and *hot potato routing* respectively. Egress data transfer price varies based on the routing selection. You can choose the routing option while creating a public IP address. The public IP address can be associated with resources such as virtual machine, virtual machine scale sets, internet-facing load balancer, etc. You can also set the routing preference for Azure storage resources such as blobs, files, web, and Azure DataLake. By default, traffic is routed via the Microsoft global network for all Azure services.

IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Routing via Microsoft global network

When you route your traffic via the *Microsoft global network*, traffic is delivered over one of the largest networks on the globe spanning over 160,000 miles of fiber with over 165 edge Point of Presence (POP). The network is well provisioned with multiple redundant fiber paths to ensure exceptionally high reliability and availability. The traffic engineering is managed by a software defined WAN controller that ensures low latency path selection for your traffic and offers the premium network performance.



Ingress traffic: The global BGP Anycast announcement ensures ingress traffic enters Microsoft network closest to the user. For example, if a user from Singapore accesses Azure resources hosted in Chicago, USA then traffic is entered into Microsoft global network in Singapore Edge POP and travels on Microsoft network to the service hosted in Chicago.

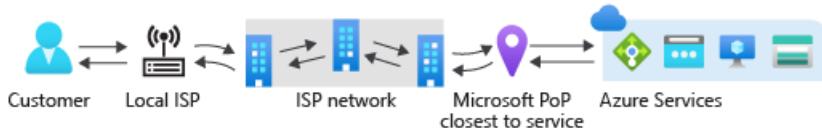
Egress traffic: The egress traffic follows the same principle. Traffic travels majority of its journey on Microsoft global network and exits closest to the user. For example, if traffic from Azure Chicago is destined to a user from Singapore, then traffic travels on Microsoft network from Chicago to Singapore, and exits the Microsoft network in Singapore Edge POP.

Both ingress and egress traffic stays bulk of the travel on the Microsoft global network. This is also known as *cold potato routing*.

Routing over public Internet (ISP network)

The new routing choice *Internet routing* minimizes travel on the Microsoft global network, and uses the transit ISP network to route your traffic. This cost-optimized routing option offers network performance that is comparable to other cloud providers.

Route via ISP network



Ingress traffic: The ingress path uses *hot potato routing* which means that traffic enters the Microsoft network that is closest to the hosted service region. For example, if a user from Singapore accesses Azure resources hosted in Chicago then traffic travels over the public internet and enters the Microsoft global network in Chicago.

Egress traffic: The egress traffic follows the same principle. Traffic exits Microsoft network in the same region that the service is hosted. For example, if traffic from your service in Azure Chicago is destined to a user from Singapore, then traffic exits the Microsoft network in Chicago and travels over the public internet to the user in Singapore.

Supported services

Public IP with Routing preference choice "Microsoft Global Network" can be associated with any Azure services. However, Public IP with Routing preference choice **Internet** can be associated with the following Azure resources:

- Virtual machine
- Virtual machine scale set
- Azure Kubernetes Service (AKS)
- Internet-facing load balancer
- Application Gateway
- Azure Firewall

For storage, primary endpoints always use the **Microsoft global network**. You can enable secondary endpoints with **Internet** as your choice for traffic routing. Supported storage services are:

- Blobs
- Files
- Web
- Azure DataLake

Pricing

The price difference between both options is reflected in the internet egress data transfer pricing. Routing via **Microsoft global network** data transfer price is same as current internet egress price. Visit [Azure bandwidth pricing page](#) for the latest pricing information.

Limitations

- Routing preference is only compatible with zone-redundant standard SKU of public IP address. Basic SKU of public IP address is not supported.
- Routing preference currently supports only IPv4 public IP addresses. IPv6 public IP addresses are not supported.
- Virtual machines with multiple NICs can have only one type of routing preference.

Next steps

- [Configure routing preference for a VM using the Azure PowerShell](#)

- Configure routing preference for a VM using the Azure CLI

Azure security baseline for Azure Public IP

2/18/2021 • 12 minutes to read • [Edit Online](#)

This security baseline applies guidance from the [Azure Security Benchmark version 1.0](#) to Azure Public IP. The Azure Security Benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the **security controls** defined by the Azure Security Benchmark and the related guidance applicable to Azure Public IP. **Controls** not applicable to Azure Public IP have been excluded. Note Azure Public IPs do not store customer data.

To see how Azure Public IP completely maps to the Azure Security Benchmark, see the [full Azure Public IP security baseline mapping file](#).

Network security

For more information, see the [Azure Security Benchmark: Network security](#).

1.10: Document traffic configuration rules

Guidance: Azure Public IPs can be assigned tags. Use resource tags for network security groups and other resources related to network security. Use any of the built-in Azure Policy definitions related to tagging, such as "Require tag and its value" that ensure that all resources are created with tags and to notify you of existing untagged resources.

Azure PowerShell or Azure CLI can be used to look up or perform actions on resources based on their tags.

- [How to create and use tags](#)
- [How to create an Azure Virtual Network](#)
- [How to filter network traffic with network security group rules](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Logging and monitoring

For more information, see the [Azure Security Benchmark: Logging and monitoring](#).

2.2: Configure central security log management

Guidance: Use Azure Activity Log to monitor configurations and detect changes to your Public IP instances. Other than at the control plane (for example, Azure portal), Public IP itself does not generate logs related to network traffic.

Public IP provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network.

Instead, you can enable and on-board data to Azure Sentinel or a third-party SIEM.

- [How to collect platform logs and metrics with Azure Monitor](#)
- [How to onboard Azure Sentinel](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.3: Enable audit logging for Azure resources

Guidance: Use Azure Activity Log to monitor configurations and detect changes for your Public IP instances. Other than at the control plane (for example, Azure portal), Public IP itself does not generate audit logs. Public IP provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network.

- [How to view and retrieve Azure Activity Log events](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.5: Configure security log storage retention

Guidance: Use Azure Monitor to set log retention period for the Log Analytics workspaces associated with Public IP instances per your organization's compliance obligations.

- [How to set log retention parameters](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

2.6: Monitor and review Logs

Guidance: Public IP provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network.

Use Azure Activity Log to monitor configurations and detect changes for your Public IP instances.

Public IP itself does not generate logs related to network traffic other than at the control plane (for example, Azure portal).

- [How to view and retrieve Azure Activity Log events](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

2.7: Enable alerts for anomalous activities

Guidance: Configure your alerts based on activity logs related to Public IP. Use Azure Monitor to configure an alert to send an email notification, call a webhook, or invoke an Azure Logic App.

- [How to manage alerts in Azure Security Center](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

Identity and access control

For more information, see the [Azure Security Benchmark: Identity and access control](#).

3.1: Maintain an inventory of administrative accounts

Guidance: Use Azure role-based access control (Azure RBAC) to manage access to Azure resources like Public IP instances with role assignments. Assign these roles to users, groups, service principals, and managed identities.

Inventoried or query pre-defined Azure built-in roles exist for certain resources through tools such as Azure CLI, Azure PowerShell, or the Azure portal.

- [How to get a directory role in Azure AD with PowerShell](#)

- [How to get members of a directory role in Azure AD with PowerShell](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.3: Use dedicated administrative accounts

Guidance: Create standard operating procedures around the use of dedicated administrative accounts.

Enabled Just-In-Time access by using Azure Active Directory (Azure AD) Privileged Identity Management (PIM) and Azure Resource Manager.

- [Learn more about Privileged Identity Management](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.5: Use multi-factor authentication for all Azure Active Directory based access

Guidance: Enable Azure Active Directory Multi-Factor Authentication and follow Security Center Identity and Access Management recommendations.

- [How to enable MFA in Azure](#)
- [How to monitor identity and access within Azure Security Center](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.6: Use dedicated machines (Privileged Access Workstations) for all administrative tasks

Guidance: Use a Privileged Access Workstation (PAW) with Azure AD Multi-Factor Authentication (MFA) enabled to log into and configure your Azure Sentinel-related resources.

- [Privileged Access Workstations](#)
- [Planning a cloud-based Azure AD Multi-Factor Authentication deployment](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.7: Log and alert on suspicious activities from administrative accounts

Guidance: Use Azure Active Directory (Azure AD) Privileged Identity Management (PIM) for generation of logs and alerts when suspicious or unsafe activity occurs in the environment.

Review and action Azure AD risk detections for alerts and reports on risky user behavior.

- [How to deploy Privileged Identity Management \(PIM\)](#)
- [Understand Azure AD risk detections](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.8: Manage Azure resources only from approved locations

Guidance: Use Conditional Access Named Locations to allow access to the Azure portal only from specific logical groupings of IP address ranges or countries/regions.

- [How to configure Named Locations in Azure](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.9: Use Azure Active Directory

Guidance: Use Azure Active Directory (Azure AD) as the central authentication and authorization system. Azure AD protects data by using strong encryption for data at rest and in transit. Azure AD also salts, hashes, and securely stores user credentials.

- [How to create and configure an Azure AD instance](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.10: Regularly review and reconcile user access

Guidance: Discover stale accounts with logs in Azure Active Directory (Azure AD).

Use Azure Identity Access Reviews to efficiently manage group memberships, access to enterprise applications, and role assignments. User access can be reviewed on a regular basis to make sure users have approved and continued access.

- [Understand Azure AD reporting](#)
- [How to use Azure Identity Access Reviews](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.11: Monitor attempts to access deactivated credentials

Guidance: Implement integration with any SIEM/monitoring tool based on your access to Azure Active Directory (Azure AD) sign-in activity, audit, and risk event log sources. Streamline this process by creating diagnostic settings for Azure AD user accounts and sending the audit logs and sign-in logs to a Log Analytics Workspace. Configure desired alerts within the Log Analytics workspace.

- [How to integrate Azure activity logs with Azure Monitor](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.12: Alert on account login behavior deviation

Guidance: Use Azure Active Directory (Azure AD) Identity Protection features to configure automated responses to detected suspicious actions related to user identities. Ingest data into Azure Sentinel for further investigation as desired and based on business requirements.

- [How to view Azure AD risky sign-ins](#)
- [How to configure and enable Identity Protection risk policies](#)
- [How to onboard Azure Sentinel](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Inventory and asset management

For more information, see the [Azure Security Benchmark: Inventory and asset management](#).

6.1: Use automated asset discovery solution

Guidance: Use Azure Resource Graph to query/discover all resources (such as compute, storage, network, ports, and protocols and so on) within your subscriptions. Ensure appropriate (read) permissions in your tenant and enumerate all Azure subscriptions as well as resources within your subscriptions.

Although classic Azure resources may be discovered via Resource Graph, it is highly recommended that you create and use Azure Resource Manager resources going forward.

- [How to create queries with Azure Resource Graph](#)
- [How to view your Azure Subscriptions](#)
- [Understand Azure RBAC](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.2: Maintain asset metadata

Guidance: Apply tags to Azure resources giving metadata to logically organize them into a taxonomy.

- [How to create and use tags](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.3: Delete unauthorized Azure resources

Guidance: Use tagging, management groups, and separate subscriptions, where appropriate, to organize and track Azure resources. Reconcile inventory on a regular basis and ensure unauthorized resources are deleted from the subscription in a timely manner.

In addition, use Azure Policy to put restrictions on the type of resources that can be created in customer subscriptions using the following built-in policy definitions:

- Not allowed resource types
- Allowed resource types
- [How to create additional Azure subscriptions](#)
- [How to create management groups](#)
- [How to create and use tags](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.5: Monitor for unapproved Azure resources

Guidance: Use Azure Policy to put restrictions on the type of resources that can be created in your subscriptions.

Use Azure Resource Graph to query for and discover resources within their subscriptions. Ensure that all Azure resources present in the environment are approved.

- [How to configure and manage Azure Policy](#)
- [How to create queries with Azure Resource Graph](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.9: Use only approved Azure services

Guidance: Use Azure Policy to put restrictions on the type of resources that can be created in customer subscriptions using the following built-in policy definitions:

- Not allowed resource types
- Allowed resource types
- [How to configure and manage Azure Policy](#)
- [How to deny a specific resource type with Azure Policy](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.11: Limit users' ability to interact with Azure Resource Manager

Guidance: Configure Azure Conditional Access to limit users' ability to interact with Azure Resource Manager by configuring "Block access" for the "Microsoft Azure Management" App.

- [How to configure Conditional Access to block access to Azure Resource Manager](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Secure configuration

For more information, see the [Azure Security Benchmark: Secure configuration](#).

7.1: Establish secure configurations for all Azure resources

Guidance: Define and implement standard security configurations for Azure Public IP with Azure Policy. Use Azure Policy aliases in the "Microsoft.Network" namespace to create custom policies to audit or enforce the network configuration of your Azure Public IP instances. You may also make use of built-in policy definitions.

- [How to configure and manage Azure Policy](#)
- [How to create a custom policy with policy aliases](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.3: Maintain secure Azure resource configurations

Guidance: Use Azure Policy [deny] and [deploy if not exist] to enforce secure settings across your Azure resources.

- [How to configure and manage Azure Policy](#)
- [Understand Azure Policy Effects](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.5: Securely store configuration of Azure resources

Guidance: If using custom Azure Policy definitions, use Azure DevOps or Azure Repos to securely store and manage your code.

- [How to store code in Azure DevOps](#)
- [Azure Repos Documentation](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.7: Deploy configuration management tools for Azure resources

Guidance: Define and implement standard security configurations for Azure Public IP with Azure Policy. Use Azure Policy aliases in the "Microsoft.Network" namespace to create custom policies to audit or enforce the network configuration of your Azure Public IP instances.

- [How to configure and manage Azure Policy](#)
- [How to create a custom policy with policy aliases](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.9: Implement automated configuration monitoring for Azure resources

Guidance: Use built-in Azure Policy definitions as well as Azure Policy aliases in the "Microsoft.Network" namespace to create custom Azure Policy definitions to alert, audit, and enforce system configurations. Use Azure Policy [audit], [deny], and [deploy if not exist] to automatically enforce configurations for your Azure resources.

- [How to configure and manage Azure Policy](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Incident response

For more information, see the [Azure Security Benchmark: Incident response](#).

10.1: Create an incident response guide

Guidance: Build out an incident response guide for your organization. Ensure that there are written incident response plans that define all roles of personnel as well as phases of incident handling/management from detection to post-incident review.

- [How to configure Workflow Automation within Azure Security Center](#)
- [Guidance on building your own security incident response process](#)
- [Microsoft Security Response Center's Anatomy of an Incident](#)
- [Customers may also use NIST's Computer Security Incident Handling Guide to aid in the creation of their own incident response plan](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.2: Create an incident scoring and prioritization procedure

Guidance: Security Center assigns a severity to each alert to help you prioritize which alerts should be investigated first. The severity is based on how confident Security Center is in the finding or the analytic used to issue the alert as well as the confidence level that there was malicious intent behind the activity that led to the alert.

Additionally, clearly mark subscriptions (for ex. production, non-prod) and create a naming system to clearly identify and categorize Azure resources.

- [Security alerts in Azure Security Center](#)
- [Use tags to organize your Azure resources](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

10.3: Test security response procedures

Guidance: Conduct exercises to test your systems' incident response capabilities on a regular cadence. Identify weak points and gaps and revise plan as needed.

- [Refer to NIST's publication: Guide to Test, Training, and Exercise Programs for IT Plans and Capabilities](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.4: Provide security incident contact details and configure alert notifications for security incidents

Guidance: Security incident contact information will be used by Microsoft to contact you if the Microsoft Security Response Center (MSRC) discovers that the customer's data has been accessed by an unlawful or unauthorized party. Review incidents after the fact to ensure that issues are resolved.

- [How to set the Azure Security Center Security Contact](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.5: Incorporate security alerts into your incident response system

Guidance: Export your Security Center alerts and recommendations using the Continuous Export feature. Continuous Export allows you to export alerts and recommendations either manually or in an ongoing, continuous fashion. You may use the Azure Security Center data connector to stream the alerts to Azure Sentinel.

- [How to configure continuous export](#)
- [How to stream alerts into Azure Sentinel](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.6: Automate the response to security alerts

Guidance: Use the Workflow Automation feature in Azure Security Center to automatically trigger responses via "Logic Apps" on security alerts and recommendations.

- [How to configure Workflow Automation and Logic Apps](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Penetration tests and red team exercises

For more information, see the [Azure Security Benchmark: Penetration tests and red team exercises](#).

11.1: Conduct regular penetration testing of your Azure resources and ensure remediation of all critical

security findings

Guidance: Follow the Microsoft Cloud Penetration Testing Rules of Engagement to ensure your penetration tests are not in violation of Microsoft policies. Use Microsoft's strategy and execution of Red Teaming and live site penetration testing against Microsoft-managed cloud infrastructure, services, and applications.

- [Penetration Testing Rules of Engagement](#)
- [Microsoft Cloud Red Teaming](#)

Azure Security Center monitoring: Not applicable

Responsibility: Shared

Next steps

- See the [Azure security benchmark](#)
- Learn more about [Azure security baselines](#)

Azure security baseline for Virtual Network

2/18/2021 • 23 minutes to read • [Edit Online](#)

This security baseline applies guidance from the [Azure Security Benchmark version 1.0](#) to Azure Virtual Network. The Azure Security Benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the **security controls** defined by the Azure Security Benchmark and the related guidance applicable to Azure Virtual Network. **Controls** not applicable to Azure Virtual Network have been excluded.

To see how Azure Virtual Network completely maps to the Azure Security Benchmark, see the [full Azure Virtual Network security baseline mapping file](#).

Network security

For more information, see the [Azure Security Benchmark: Network security](#).

1.2: Monitor and log the configuration and traffic of virtual networks, subnets, and network interfaces

Guidance: Use Security Center and follow network protection recommendations to help secure your network resources in Azure.

Send network security group flow logs to a Log Analytics Workspace and use Traffic Analytics to provide insights into traffic flow into your Azure cloud. Traffic Analytics offers the ability to visualize network activity and identify hot spots, identify security threats, understand traffic flow patterns, and pinpoint network misconfigurations.

Use Azure Monitor logs to provide insights into your environment. A workspace should be used to collate and analyze the data, and can integrate with other Azure services such as Application Insights and Security Center.

Choose resource logs to send to an Azure storage account or an event hub. A different platform can also be used to analyze the logs.

- [How to Enable NSG Flow Logs](#)
- [How to Enable and use Traffic Analytics](#)
- [Understand Network Security provided by Security Center](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

1.4: Deny communications with known-malicious IP addresses

Guidance: Enable distributed denial of service (DDoS) Standard protection on your Azure Virtual Network to guard against DDoS attacks.

Deploy Azure Firewall at each of the organization's network boundaries with threat intelligence-based filtering enabled and configured to "Alert and deny" for malicious network traffic.

Use Security Center's threat protection features to detect communications with known malicious IP addresses.

Apply Security Center's Adaptive Network Hardening recommendations for network security group configurations that limit ports and source IPs based on actual traffic and threat intelligence.

- [Manage Azure DDoS Protection Standard using the Azure portal](#)

- [Azure Firewall threat intelligence-based filtering](#)
- [Threat protection in Security Center](#)
- [Adaptive Network Hardening in Azure Security Center](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

1.5: Record network packets

Guidance: Use VPN Gateway's packet capture in addition to commonly available packet capture tools to record network packets.

You can also review agent based or NVA solutions that provide Terminal Access Point (TAP) or Network Visibility functionality through Packet Broker partner solutions available in Azure Marketplace Offerings.

- [Configure packet captures for VPN gateways](#)
- [Network Virtual Appliance Partner](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

1.6: Deploy network-based intrusion detection/intrusion prevention systems (IDS/IPS)

Guidance: Use an Azure Firewall deployed on your virtual network with Threat Intelligence enabled. Use Azure Firewall Threat intelligence-based filtering to alert or to deny traffic to and from known malicious IP addresses and domains. The IP addresses and domains are sourced from the Microsoft Threat Intelligence feed.

You can also select an appropriate offer from the Azure Marketplace that supports IDS/IPS functionality with payload inspection capabilities.

Deploy the firewall solution of your choice at each of your organization's network boundaries to detect and/or deny malicious traffic.

- [How to deploy Azure Firewall](#)
- [How to configure alerts with Azure Firewall](#)
- [Azure Marketplace](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

1.8: Minimize complexity and administrative overhead of network security rules

Guidance: Use Virtual Network service tags to define network access controls on network security groups or Azure Firewall. Service tags can be used in place of specific IP addresses when creating security rules. Allow or deny the traffic for the corresponding service by specifying the service tag name (for example, ApiManagement) in the appropriate source or destination field of a rule. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change.

Use Application Security Groups to help simplify complex security configuration. Application security groups allow you to configure network security as a natural extension of an application's structure. This enables you to group virtual machines and define network security policies based on those groups.

- [Understand and use Service tags](#)
- [Understand and use Application Security Groups](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

1.9: Maintain standard security configurations for network devices

Guidance: Define and implement standard security configurations for network resources with Azure Policy and review the built-in network policy definitions for implementation.

Refer to the default policy for Security Center which contains available security recommendations related to your virtual networks.

Use Azure Blueprints to simplify large-scale Azure deployments by packaging key environment artifacts, such as Azure Resource Manager templates, Azure role-based access control (Azure RBAC) assignments, and policies, in a single blueprint definition. Azure Blueprint can be applied to new subscriptions for fine-tuned control and management through versioning.

- [How to configure and manage Azure Policy](#)
- [Azure Policy samples for networking](#)
- [How to create an Azure Blueprint](#)
- [Enable monitoring in Azure Security Center](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

1.10: Document traffic configuration rules

Guidance: Use tags for network security groups and other resources related to network security and traffic flow. Use the "Description" field to specify business need, duration, and other information for any rules that allow traffic to/from a network for individual network security group rules. Use any of the built-in Azure Policy definitions related to tagging, such as "Require tag and its value" to ensure that all resources are created with tags and to notify you of existing untagged resources.

Choose Azure PowerShell or Azure CLI to look up or perform actions on resources based on their tags.

- [How to create and use Tags](#)
- [How to create a Virtual Network](#)
- [How to create an NSG with a Security Config](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

1.11: Use automated tools to monitor network resource configurations and detect changes

Guidance: Use Azure Activity Log to monitor resource configurations and detect changes to your virtual network. Create alerts within Azure Monitor which will trigger when changes to critical resources take place.

- [How to view and retrieve Azure Activity Log events](#)
- [How to create alerts in Azure Monitor](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

Logging and monitoring

For more information, see the [Azure Security Benchmark: Logging and monitoring](#).

2.2: Configure central security log management

Guidance: Enable Azure Monitor for access to your audit and activity logs which includes event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

In Azure Monitor, use Log Analytics Workspaces to query and perform analytics, and use Azure Storage Accounts for long-term/archival storage. Alternatively, you may enable and on-board data to Azure Sentinel or a third-party SIEM.

- [How to collect platform logs and metrics with Azure Monitor](#)
- [View and retrieve Azure Activity log events](#)
- [How to onboard Azure Sentinel](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.3: Enable audit logging for Azure resources

Guidance: Enable Azure Monitor for access to your audit and activity logs which includes event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

- [How to collect platform logs and metrics with Azure Monitor](#)
- [View and retrieve Azure Activity log events](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.5: Configure security log storage retention

Guidance: Within Azure Monitor, set your Log Analytics Workspace retention period according to your organization's compliance regulations. Use Azure Storage accounts for long-term/archival storage of security log storage retention.

- [Change the data retention period in Log Analytics](#)
- [How to configure retention policy for Azure Storage account logs](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

2.6: Monitor and review logs

Guidance: Analyze and monitor logs for anomalous behavior and regularly review results. Use Azure Monitor's Log Analytics Workspace to query and perform analytics, and use Azure Storage Accounts for long-term/archival storage.

Alternatively, you may enable and on-board data to Azure Sentinel or a third-party SIEM.

- [Understand Log Analytics Workspace](#)
- [How to perform custom queries in Azure Monitor](#)
- [How to onboard Azure Sentinel](#)
- [How to get started with Azure Monitor and third-party SIEM integration](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

2.7: Enable alerts for anomalous activities

Guidance: Use Security Center with Log Analytics Workspace for monitoring and alerting on anomalous activity found in security logs and events.

Alternatively, you may enable and onboard data to Azure Sentinel or a third-party SIEM for alerting.

- [How to manage alerts in Security Center](#)
- [How to alert on log analytics log data](#)
- [How to onboard Azure Sentinel](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.9: Enable DNS query logging

Guidance: Implement a third-party solution from Azure Marketplace for DNS logging solution as per your organizational need.

Azure Security Center monitoring: Currently not available

Responsibility: Customer

Identity and access control

For more information, see the [Azure Security Benchmark: Identity and access control](#).

3.1: Maintain an inventory of administrative accounts

Guidance: Use Azure Active Directory (Azure AD) built-in administrator roles that can be explicitly assigned and are queryable.

Use the Azure AD PowerShell module to perform ad hoc queries to discover accounts that are members of administrative groups.

- [How to get a directory role in Azure AD with PowerShell](#)
- [How to get members of a directory role in Azure AD with PowerShell](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.3: Use dedicated administrative accounts

Guidance: Create standard operating procedures around the use of dedicated administrative accounts. Use Security Center's Identity and Access Management to monitor the number of administrative accounts.

Enable Just-In-Time / Just-Enough-Access by using Azure AD Privileged Identity Management Privileged Roles for Microsoft Services and Azure Resource Manager.

- [Learn more about Privileged Identity Management](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.4: Use Azure Active Directory single sign-on (SSO)

Guidance: Use SSO with Azure Active Directory (Azure AD) rather than configuring individual stand-alone credentials per-service. Use Security Center's Identity and Access Management recommendations.

- [Single sign-on to applications in Azure Active Directory](#)
- [How to monitor identity and access within Azure Security Center](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

3.5: Use multi-factor authentication for all Azure Active Directory-based access

Guidance: Enable Azure Active Directory (Azure AD) Multi-Factor Authentication (MFA) and follow Security Center's Identity and Access Management recommendations.

- [How to enable MFA in Azure](#)
- [How to monitor identity and access within Azure Security Center](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.6: Use secure, Azure-managed workstations for administrative tasks

Guidance: Use Privileged Access Workstations (PAW) with Multi-Factor Authentication (MFA) configured to log into and access Azure network resources.

- [Learn about Privileged Access Workstations](#)
- [How to enable MFA in Azure](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

3.7: Log and alert on suspicious activities from administrative accounts

Guidance: Utilize Azure Active Directory (Azure AD) Risk Detections to view alerts and reports on risky user behavior.

Ingest Security Center Risk Detection alerts into Azure Monitor and configure custom alerting/notifications using Action Groups.

- [Understanding Security Center risk detections \(suspicious activity\)](#)
- [How to integrate Azure Activity Logs into Azure Monitor](#)
- [How to configure action groups for custom alerting and notification](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.8: Manage Azure resources from only approved locations

Guidance: Use Conditional Access named locations to allow access from only specific logical groupings of IP address ranges or countries/regions.

- [How to configure named locations in Azure](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

3.9: Use Azure Active Directory

Guidance: Use Azure Active Directory (Azure AD) as a central authentication and authorization system for your

services. Azure AD protects data by using strong encryption for data at rest and in transit and also salts, hashes, and securely stores user credentials.

- [How to create and configure an Azure AD instance](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.10: Regularly review and reconcile user access

Guidance: Use Azure Active Directory (Azure AD) to provide logs to help discover stale accounts.

Azure Identity Access Reviews can be performed to efficiently manage group memberships, access to enterprise applications, and role assignments. User access should be reviewed on a regular basis to make sure only the active users have continued access.

- [Understand Azure AD reporting](#)
- [How to use Azure Identity Access Reviews](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

3.11: Monitor attempts to access deactivated credentials

Guidance: Integrate Azure Active Directory (Azure AD) Sign-in Activity, Audit and Risk Event log sources, with any SIEM or Monitoring tool based on your access.

Streamline this process by creating Diagnostic Settings for Azure Active Directory user accounts and sending the audit logs and sign-in logs to a Log Analytics Workspace. Any desired alerts can be configured within Log Analytics Workspace.

- [How to integrate Azure Activity Logs into Azure Monitor](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

3.12: Alert on account sign-in behavior deviation

Guidance: Use Azure Active Directory's (Azure AD) Risk and Identity Protection features to configure automated responses to detected suspicious actions related to user identities for your Virtual network. Ingest data into Azure Sentinel for any further investigations.

- [How to view Azure AD risky sign-ins](#)
- [How to configure and enable Identity Protection risk policies](#)
- [How to onboard Azure Sentinel](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

Data protection

For more information, see the [Azure Security Benchmark: Data protection](#).

4.4: Encrypt all sensitive information in transit

Guidance: Encrypt all sensitive information in transit. Ensure that any clients connecting to your Azure resources in your virtual networks are able to negotiate TLS 1.2 or greater. Follow Security Center

recommendations for encryption at rest and encryption in transit.

Microsoft provides several options which can be utilized by customers for securing data in transit internally within the Azure network and externally across the Internet to the end user. These include communication through Virtual Private Networks (utilizing IPsec/IKE encryption), Transport Layer Security (TLS) 1.2 or later (via Azure components such as Application Gateway or Azure Front Door), protocols directly on the Azure virtual machines (such as Windows IPsec or SMB), and more.

Additionally, "encryption by default" using MACsec (an IEEE standard at the data-link layer) is enabled for all Azure traffic traveling between Azure datacenters to ensure confidentiality and integrity of customer data.

- [Understand encryption in transit with Azure](#)

Azure Security Center monitoring: Yes

Responsibility: Shared

4.6: Use Azure RBAC to manage access to resources

Guidance: Use Azure role-based access control (Azure RBAC) to manage access to data and resources. Otherwise use service-specific access-control methods.

Choose built-in roles like Owner, Contributor, or Network contributor and assign the role to the appropriate scope. For example, you can assign a subset of virtual network capabilities with the specific permissions required for virtual networks to any of these roles.

- [How to configure Azure RBAC](#)
- [Plan virtual networks](#)
- [Review the Azure built-in roles](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

4.9: Log and alert on changes to critical Azure resources

Guidance: Use Azure Monitor with the Azure Activity Logs to create alerts for when changes take place to critical Azure resources like Virtual Networks and network security groups.

- [Diagnostic logging for a network security group](#)
- [How to create alerts for Azure Activity Log events](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

Inventory and asset management

For more information, see the [Azure Security Benchmark: Inventory and asset management](#).

6.1: Use automated asset discovery solution

Guidance: Use Azure Resource Graph to query and discover all networking resources like Virtual Networks, subnets within your subscriptions. Ensure you have appropriate (read) permissions in your tenant and are able to enumerate all Azure subscriptions as well as resources within your subscriptions.

- [How to create queries with Azure Graph](#)
- [How to view your Azure Subscriptions](#)
- [Understand Azure RBAC](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.2: Maintain asset metadata

Guidance: Apply tags to Azure resources giving metadata to logically organize them into a taxonomy.

- [How to create and use Tags](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.3: Delete unauthorized Azure resources

Guidance: Use tagging, management groups, and separate subscriptions, where appropriate, to organize and track Virtual network and related resources. Reconcile inventory on a regular basis and ensure unauthorized resources are deleted from the subscription in a timely manner.

- [How to create additional Azure subscriptions](#)
- [How to create Management Groups](#)
- [How to create and use Tags](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.4: Define and maintain inventory of approved Azure resources

Guidance: You will need to create an inventory of approved Azure resources and approved software for compute resources as per your organizational needs.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.5: Monitor for unapproved Azure resources

Guidance: Use Azure Policy to put restrictions on the type of resources that can be created in customer subscriptions using the following built-in policy definitions:

- Not allowed resource types
- Allowed resource types

Query or discover resources within the subscriptions with Azure Resource Graph in high security-based environments, such as those with Azure Storage accounts.

- [How to configure and manage Azure Policy](#)
- [How to create queries with Azure Graph](#)
- [Azure policy sample built-ins for virtual network](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.7: Remove unapproved Azure resources and software applications

Guidance: Prevent resource creation or usage with Azure Policy as required by the organization's policies. Implement processes for removing unauthorized resources.

- [How to configure and manage Azure Policy](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.9: Use only approved Azure services

Guidance: Use Azure Policy to put restrictions on the type of resources that can be created in customer subscriptions using the following built-in policy definitions:

- Not allowed resource types
- Allowed resource types
- [How to configure and manage Azure Policy](#)
- [How to deny a specific resource type with Azure Policy](#)
- [Azure policy sample built-ins for virtual network](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.11: Limit users' ability to interact with Azure Resource Manager

Guidance: Use Azure Conditional Access to limit user's ability to interact with Azure Resource Manager by configuring "Block access" for the "Microsoft Azure Management" App.

- [How to configure Conditional Access to block access to Azure Resource Manager](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

Secure configuration

For more information, see the [Azure Security Benchmark: Secure configuration](#).

7.1: Establish secure configurations for all Azure resources

Guidance: Use Azure Policy aliases to create custom policies to audit or enforce the configuration of your Azure network resources and also use built-in Azure Policy definitions.

Export any of your build templates with Azure Resource Manager in JavaScript Object Notation (JSON) form and review it to ensure that the configurations meet or exceed the security requirements for your organization.

Implement recommendations from Security Center as a secure configuration baseline for your Azure resources.

- [How to view available Azure Policy Aliases](#)
- [Tutorial: Create and manage policies to enforce compliance](#)
- [Azure policy sample built-ins for virtual network](#)
- [Single and multi-resource export to a template in Azure portal](#)
- [Security recommendations - a reference guide](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.3: Maintain secure Azure resource configurations

Guidance: Use Azure Resource Manager templates and Azure Policies to securely configure Azure resources associated with the Virtual network and related resources. Azure Resource Manager templates are JSON

(JavaScript Object Notation) based files used to deploy virtual machines along with Azure resources. Microsoft performs the maintenance on the base templates.

Use Azure Policy [deny] and [deploy if not exist] effects to enforce secure settings across your Azure resources.

- [Information on creating Azure Resource Manager templates](#)
- [How to configure and manage Azure Policy](#)
- [Understanding Azure Policy Effects](#)
- [Azure Resource Manager template samples for virtual network](#)
- [Azure policy sample built-ins for virtual network](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.5: Securely store configuration of Azure resources

Guidance: Use Azure DevOps to securely store and manage your code like custom Azure policies, Azure Resource Manager templates, desired state configuration scripts. and so on.

You must have permissions to access the resources you wish to manage in Azure DevOps, such as your code, builds, and work tracking. Most permissions are granted through built-in security groups. You can grant or deny permissions to specific users, built-in security groups, or groups defined in Azure Active Directory (Azure AD) if integrated with Azure DevOps, or Active Directory if integrated with Team Foundation Server.

- [How to store code in Azure DevOps](#)
- [About permissions and groups in Azure DevOps](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.7: Deploy configuration management tools for Azure resources

Guidance: Define and implement standard security configurations for Azure resources using Azure Policy. Use Azure Policy aliases to create custom policies to audit or enforce the network configuration of your Azure resources and any built-in policy definitions related to specific resources.

- [How to configure and manage Azure Policy](#)
- [How to use Aliases](#)
- [Azure policy sample built-ins for virtual network](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.9: Implement automated configuration monitoring for Azure resources

Guidance: Use Security Center to perform baseline scans for your Azure Virtual Network and related resources. Use Azure Policy to alert and audit Azure resource configurations.

- [How to remediate recommendations in Security Center](#)
- [How to configure and manage Azure Policy](#)
- [Azure policy sample built-ins for virtual network](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.11: Manage Azure secrets securely

Guidance: Use Managed Service Identity in conjunction with Azure Key Vault to simplify and secure secret management for your Azure resources hosted in an Azure Virtual Network.

- [How to integrate with Azure Managed Identities](#)
- [How to create a Key Vault](#)
- [How to provide Key Vault authentication with a managed identity](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.13: Eliminate unintended credential exposure

Guidance: Implement Credential Scanner to identify credentials within code. Credential Scanner will also encourage moving discovered credentials to more secure locations such as Azure Key Vault.

- [How to setup Credential Scanner](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

Data recovery

For more information, see the [Azure Security Benchmark: Data recovery](#).

9.1: Ensure regular automated back-ups

Guidance: Use Azure Resource Manager to deploy a virtual network and related resources. Azure Resource Manager provides ability to export templates which can be used as backups to restore Virtual network and related resources. Use Azure Automation to call the Azure Resource Manager template export API on a regular basis.

- [Overview of Azure Resource Manager](#)
- [Azure Resource Manager template samples for virtual network](#)
- [Single and multi-resource export to a template in Azure portal](#)
- [Resource Groups - Export Template](#)
- [Introduction to Azure Automation](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

9.2: Perform complete system backups and backup any customer-managed keys

Guidance: Use Azure Resource Manager to deploy a virtual network and related resources. Azure Resource Manager provides ability to export templates which can be used as backups to restore Virtual network and related resources. Use Azure Automation to call the Azure Resource Manager template export API on a regular basis. Back up customer-managed keys within Azure Key Vault.

- [Overview of Azure Resource Manager](#)
- [Azure Resource Manager template samples for virtual network](#)
- [Single and multi-resource export to a template in Azure portal](#)

- [Resource Groups - Export Template](#)
- [Introduction to Azure Automation](#)
- [How to backup key vault keys in Azure](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

9.3: Validate all backups including customer-managed keys

Guidance: Periodically perform deployment of Azure Resource Manager templates to an isolated subscription and test restoration of backed up customer-managed keys.

- [Deploy resources with ARM templates and Azure portal](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

9.4: Ensure protection of backups and customer-managed keys

Guidance: Use Azure DevOps to securely store and manage your code like custom Azure Policy definitions and Azure Resource Manager templates.

Grant or deny permissions to specific users, built-in security groups, or groups defined in Azure Active Directory (Azure AD) if integrated with Azure DevOps, or Active Directory if integrated with Team Foundation Server.

Use Azure role-based access control (Azure RBAC) to protect customer-managed keys.

Enable Soft-Delete and purge protection in Key Vault to protect keys against accidental or malicious deletion.

- [How to store code in Azure DevOps](#)
- [About permissions and groups in Azure DevOps](#)
- [How to enable Soft-Delete and Purge protection in Key Vault](#)
- [Soft delete for Azure Storage blobs](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Incident response

For more information, see the [Azure Security Benchmark: Incident response](#).

10.1: Create an incident response guide

Guidance: Build out an incident response guide for your organization. Ensure that there are written incident response plans that define all roles of personnel as well as phases of incident handling/management from detection to post-incident review.

- [Guidance on building your own security incident response process](#)
- [Microsoft Security Response Center's Anatomy of an Incident](#)
- [Leverage NIST's Computer Security Incident Handling Guide to aid in the creation of your own incident response plan](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

10.2: Create an incident scoring and prioritization procedure

Guidance: Security Center assigns a severity to each alert to help you prioritize which alerts should be investigated first. The severity is based on how confident Security Center is in the finding or the analytic used to issue the alert as well as the confidence level that there was malicious intent behind the activity that led to the alert.

Clearly mark subscriptions (for example, production or non-production) using tags and create a naming system to clearly identify and categorize Azure resources, especially those processing sensitive data. It is your responsibility to prioritize the remediation of alerts based on the criticality of the Azure resources and environment where the incident occurred.

- [Security alerts in Security Center](#)
- [Use tags to organize your Azure resources](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

10.3: Test security response procedures

Guidance: Conduct exercises to test your systems' incident response capabilities on a regular cadence to help protect your Azure resources. Identify weak points and gaps and revise plan as needed.

- [NIST's publication - Guide to Test, Training, and Exercise Programs for IT Plans and Capabilities](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.4: Provide security incident contact details and configure alert notifications for security incidents

Guidance: Security incident contact information will be used by Microsoft to contact you if the Microsoft Security Response Center (MSRC) discovers that your data has been accessed by an unlawful or unauthorized party. Review incidents after the fact to ensure that issues are resolved.

- [How to set the Security Center Security Contact](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

10.5: Incorporate security alerts into your incident response system

Guidance: Export your Security Center alerts and recommendations using the Continuous Export feature to help identify risks to Azure resources. Continuous Export allows you to export alerts and recommendations either manually or in an ongoing, continuous fashion.

You can also use the Security Center data connector to stream the alerts to Azure Sentinel.

- [How to configure continuous export](#)
- [How to stream alerts into Azure Sentinel](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

10.6: Automate the response to security alerts

Guidance: Use the Workflow Automation feature in Security Center to automatically trigger responses via "Logic Apps" on security alerts and recommendations to protect your Azure resources.

- [How to configure Workflow Automation and Logic Apps](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Penetration tests and red team exercises

For more information, see the [Azure Security Benchmark: Penetration tests and red team exercises](#).

11.1: Conduct regular penetration testing of your Azure resources and ensure remediation of all critical security findings

Guidance: Follow the Microsoft Rules of Engagement to ensure your Penetration Tests are not in violation of Microsoft policies. Use Microsoft's strategy and execution of Red Teaming and live site penetration testing against Microsoft-managed cloud infrastructure, services, and applications.

- [Penetration Testing Rules of Engagement](#)
- [Microsoft Cloud Red Teaming](#)

Azure Security Center monitoring: Not applicable

Responsibility: Shared

Next steps

- See the [Azure security benchmark](#)
- Learn more about [Azure security baselines](#)

Azure security baseline for Virtual Network NAT

2/18/2021 • 8 minutes to read • [Edit Online](#)

This security baseline applies guidance from the [Azure Security Benchmark version 1.0](#) to Microsoft Virtual Network NAT. The Azure Security Benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the **security controls** defined by the Azure Security Benchmark and the related guidance applicable to Virtual Network NAT. Controls not applicable to Virtual Network NAT have been excluded.

To see how Virtual Network NAT completely maps to the Azure Security Benchmark, see the [full Virtual Network NAT security baseline mapping file](#).

Network Security

For more information, see the [Azure Security Benchmark: Network Security](#).

1.2: Monitor and log the configuration and traffic of virtual networks, subnets, and NICs

Guidance: Virtual Network NAT doesn't support controlling outbound traffic with network security groups (NSG). Inbound traffic is only allowed in response to an outbound originated flow.

However, network security group (NSG) flow logs can be used with NAT gateway resources to monitor outbound originated traffic.

Use Azure Security Center and follow the network protection recommendations to help secure your Azure network resources. Enable network security group flow logs and send the logs to an Azure Storage account for auditing. You can also send the flow logs to a Log Analytics workspace and then use Traffic Analytics to provide insights into traffic patterns in your Azure cloud. Some advantages of Traffic Analytics are the ability to visualize network activity, identify hot spots and security threats, understand traffic flow patterns, and pinpoint network misconfigurations.

- [Virtual Network NAT overview](#)
- [NAT gateway resource](#)
- [How to enable network security group flow logs](#)
- [How to enable and use Traffic Analytics](#)
- [Understand network security provided by Azure Security Center](#)

Azure Security Center monitoring: Currently not available

Responsibility: Customer

1.5: Record network packets

Guidance: Enable Network Watcher packet capture to investigate anomalous activities.

- [How to create a Network Watcher instance](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

1.9: Maintain standard security configurations for network devices

Guidance: Define and implement standard security configurations for subnets configured with NAT gateway

resources using built-in or custom Azure Policy definitions and assignments.

- [How to configure and manage Azure Policy](#)
- [Azure Policy samples for networking](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

1.11: Use automated tools to monitor network resource configurations and detect changes

Guidance: Use Azure Activity log to monitor resource configurations and detect changes to your NAT gateway resources and virtual network resources. Create alerts in Azure Monitor to notify you when critical resources are changed.

- [How to view and retrieve Azure Activity log events](#)
- [How to create alerts in Azure Monitor](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Logging and Monitoring

For more information, see the [Azure Security Benchmark: Logging and Monitoring](#).

2.2: Configure central security log management

Guidance: Ingest logs related to Virtual Network NAT via Azure Monitor to aggregate security data generated by endpoint devices, network resources, and other security systems. In Azure Monitor, use Log Analytics workspaces to query and perform analytics, and use Azure Storage accounts for long term and archival storage.

Alternatively, you can enable and on-board this data to Azure Sentinel or a third-party SIEM.

- [How to onboard Azure Sentinel](#)
- [How to collect platform logs and metrics with Azure Monitor](#)
- [How to get started with Azure Monitor and third-party SIEM integration](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

2.3: Enable audit logging for Azure resources

Guidance: Activity logs, which are automatically available, contain all write operations (PUT, POST, DELETE) for your NAT gateway resources except read operations (GET). Activity logs can be used to find an error when troubleshooting or to monitor how a user in your organization modified a resource.

- [How to collect platform logs and metrics with Azure Monitor](#)
- [Understand logging and different log types in Azure](#)

Virtual Network NAT currently does not produce any additional diagnostic logs which are configurable by customers.

Azure Security Center monitoring: Yes

Responsibility: Customer

2.7: Enable alerts for anomalous activities

Guidance: Use Azure Security Center with Log Analytics workspace for monitoring and alerting on anomalous activity found in security logs and events. Instead, you can enable and on-board data to Azure Sentinel.

- [How to onboard Azure Sentinel](#)
- [How to manage alerts in Azure Security Center](#)
- [How to alert on log analytics log data](#)

Azure Security Center monitoring: Yes

Responsibility: Customer

Inventory and Asset Management

For more information, see the [Azure Security Benchmark: Inventory and Asset Management](#).

6.1: Use automated asset discovery solution

Guidance: Use Azure Resource Graph to query/discover all resources (such as compute, storage, network, ports, and protocols and so on) within your subscriptions.

Ensure appropriate (read) permissions in your tenant and enumerate all Azure subscriptions and resources within your subscriptions.

Although classic Azure resources may be discovered via Resource Graph, it is highly recommended to create and use Azure Resource Manager based resources going forward.

- [Azure Resource Graph queries](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.2: Maintain asset metadata

Guidance: Apply tags to your Azure resources, resource groups, and subscriptions to logically organize them into a taxonomy. Each tag consists of a name and a value pair. For example, you can apply the name "Environment" and the value "Production" to all the resources in production.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.3: Delete unauthorized Azure resources

Guidance: Apply tags for management of Azure resources.

- [How to create additional Azure subscriptions](#)
- [How to create Management Groups](#)
- [How to create and use Tags](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.4: Define and maintain an inventory of approved Azure resources

Guidance: Currently not available

Azure Security Center monitoring: Currently not available

Responsibility: Customer

6.5: Monitor for unapproved Azure resources

Guidance: Use Azure Policy to put restrictions on the type of resources that can be created in your subscription(s). Use Azure Resource Graph to query/discover resources within their subscription(s). Ensure that all Azure resources present in the environment are approved.

- [How to configure and manage Azure Policy](#)
- [How to create queries with Azure Graph](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

6.9: Use only approved Azure services

Guidance: Use Azure Policy to restrict which services you can provision in your environment.

- [How to configure and manage Azure Policy](#)
- [How to deny a specific resource type with Azure Policy](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Secure Configuration

For more information, see the [Azure Security Benchmark: Secure Configuration](#).

7.3: Maintain secure Azure resource configurations

Guidance: Not applicable; Virtual Network NAT doesn't have any security configurations.

Azure Security Center monitoring: Currently not available

Responsibility: Customer

7.5: Securely store configuration of Azure resources

Guidance: Use Azure DevOps to securely store and manage your code like custom Azure Policy definitions, Azure Resource Manager templates and desired state configuration scripts. To access the resources you manage in Azure DevOps, you can grant or deny permissions to specific users, built-in security groups, or groups defined in Azure Active Directory (Azure AD) if integrated with Azure DevOps, or Active Directory if integrated with TFS.

- [How to store code in Azure DevOps](#)
- [About permissions and groups in Azure DevOps](#)

Azure Security Center monitoring: Not applicable

Responsibility: Customer

7.7: Deploy configuration management tools for Azure resources

Guidance: Not applicable; Virtual Network NAT doesn't have any security configurations.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Incident Response

For more information, see the [Azure Security Benchmark: Incident Response](#).

10.1: Create an incident response guide

Guidance: An incident response procedure should be created for the offering to ensure appropriate incident response processes can occur so that it receives the appropriate level of prioritization till the incidents resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.2: Create an incident scoring and prioritization procedure

Guidance: An incident scoring and prioritization procedure should be created for the offering to ensure proper risk or threat scoring can occur so that it receives the appropriate level of prioritization till the incidents resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.3: Test security response procedures

Guidance: An incident security response procedure should be created and tested for the offering to ensure proper mitigation can occur with the appropriate level of prioritization till the incidents forecasted resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.4: Provide security incident contact details and configure alert notifications for security incidents

Guidance: An incident security response procedure should be created and tested for the offering to ensure proper mitigation can occur with the appropriate level of prioritization till the incidents forecasted resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.5: Incorporate security alerts into your incident response system

Guidance: An incident security response procedure should be created and tested for the offering to ensure proper mitigation can occur with the appropriate level of prioritization till the incidents forecasted resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

10.6: Automate the response to security alerts

Guidance: An incident security response procedure should be created and tested for the offering to ensure proper mitigation can occur with the appropriate level of prioritization till the incidents forecasted resolution.

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Penetration Tests and Red Team Exercises

For more information, see the [Azure Security Benchmark: Penetration Tests and Red Team Exercises](#).

11.1: Conduct regular penetration testing of your Azure resources and ensure remediation of all critical security findings

Guidance: Follow the Microsoft Cloud Penetration Testing Rules of Engagement to ensure your penetration tests are not in violation of Microsoft policies. Use Microsoft's strategy and execution of Red Teaming and live site penetration testing against Microsoft-managed cloud infrastructure, services, and applications.

- Penetration Testing Rules of Engagement
- Microsoft Cloud Red Teaming

Azure Security Center monitoring: Not applicable

Responsibility: Customer

Next steps

- See the [Azure Security Benchmark V2 overview](#)
- Learn more about [Azure security baselines](#)

Integrate Azure services with virtual networks for network isolation

2/18/2021 • 4 minutes to read • [Edit Online](#)

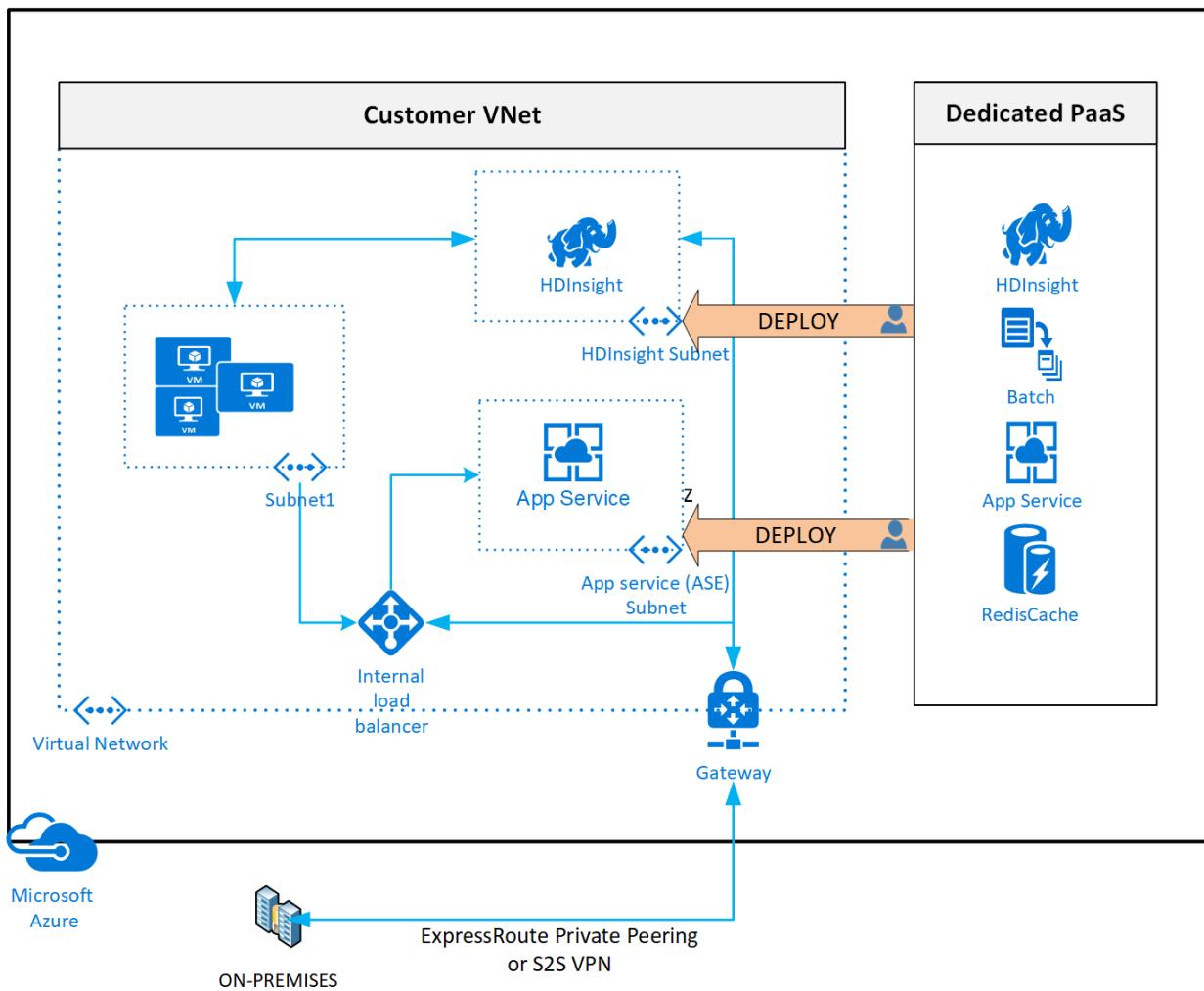
Virtual Network (VNet) integration for an Azure service enables you to lock down access to the service to only your virtual network infrastructure. The VNet infrastructure also includes peered virtual networks and on-premises networks.

VNet integration provides Azure services the benefits of network isolation and can be accomplished by one or more of the following methods:

- [Deploying dedicated instances of the service into a virtual network](#). The services can then be privately accessed within the virtual network and from on-premises networks.
- Using [Private Endpoint](#) that connects you privately and securely to a service powered by [Azure Private Link](#). Private Endpoint uses a private IP address from your VNet, effectively bringing the service into your virtual network.
- Accessing the service using public endpoints by extending a virtual network to the service, through [service endpoints](#). Service endpoints allow service resources to be secured to the virtual network.
- Using [service tags](#) to allow or deny traffic to your Azure resources to and from public IP endpoints.

Deploy dedicated Azure services into virtual networks

When you deploy dedicated Azure services in a virtual network, you can communicate with the service resources privately, through private IP addresses.



Deploying an dedicated Azure service into your virtual network provides the following capabilities:

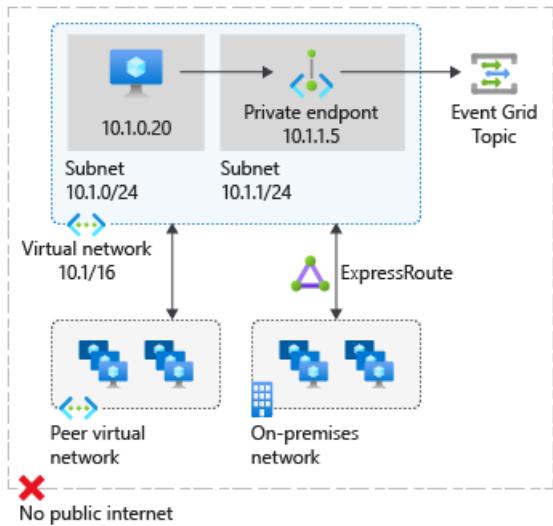
- Resources within the virtual network can communicate with each other privately, through private IP addresses. Example, directly transferring data between HDInsight and SQL Server running on a virtual machine, in the virtual network.
- On-premises resources can access resources in a virtual network using private IP addresses over a Site-to-Site VPN (VPN Gateway) or ExpressRoute.
- Virtual networks can be peered to enable resources in the virtual networks to communicate with each other, using private IP addresses.
- Service instances in a virtual network are typically fully managed by the Azure service. This includes monitoring the health of the resources and scaling with load.
- Service instances are deployed into a subnet in a virtual network. Inbound and outbound network access for the subnet must be opened through network security groups, per guidance provided by the service.
- Certain services also impose restrictions on the subnet they are deployed in, limiting the application of policies, routes or combining VMs and service resources within the same subnet. Check with each service on the specific restrictions as they may change over time. Examples of such services are Azure NetApp Files, Dedicated HSM, Azure Container Instances, App Service.
- Optionally, services might require a delegated subnet as an explicit identifier that a subnet can host a particular service. By delegating, services get explicit permissions to create service-specific resources in the delegated subnet.
- See an example of a REST API response on a virtual network with a delegated subnet. A comprehensive list of services that are using the delegated subnet model can be obtained via the Available Delegations API.

For a list of services that can be deployed into a virtual network, see [Deploy dedicated Azure services into virtual networks](#).

Private Link and Private Endpoints

You can use private endpoints to allow ingress of events directly from your virtual network to Azure resource securely over a private link without going through the public internet. A private endpoint is a special network interface for an Azure service in your virtual network. When you create a private endpoint for your Azure resource, it provides secure connectivity between clients on your virtual network and your Azure resource. The private endpoint is assigned an IP address from the IP address range of your virtual network. The connection between the private endpoint and the Azure service uses a secure private link.

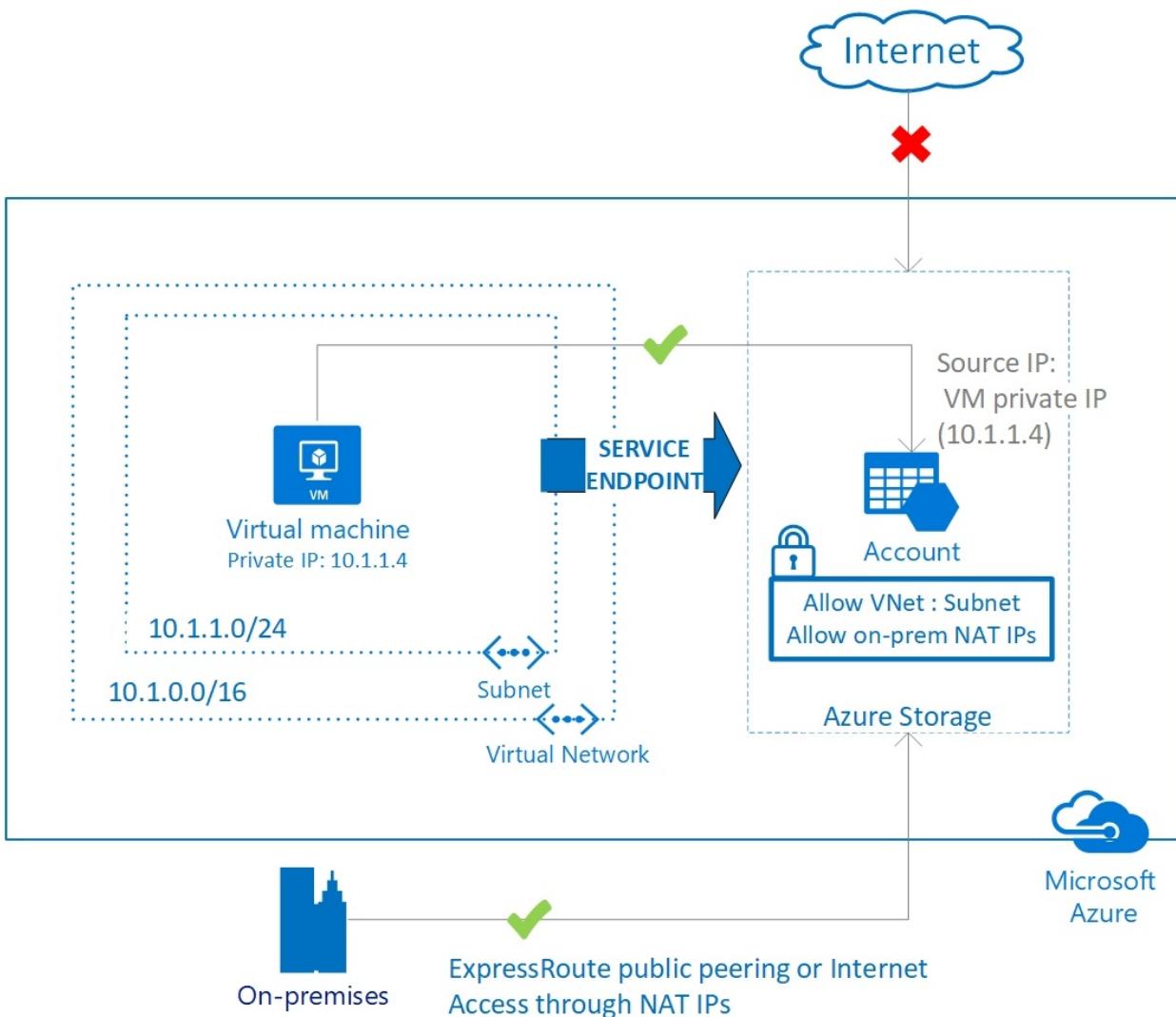
The following example shows private access of an Event Grid resource private endpoint that provides secure connectivity between clients on a virtual network and Event Grid resource.



For more information on Private Link and a list of Azure services that are supported, see [What is Private Link?](#)

Service endpoints

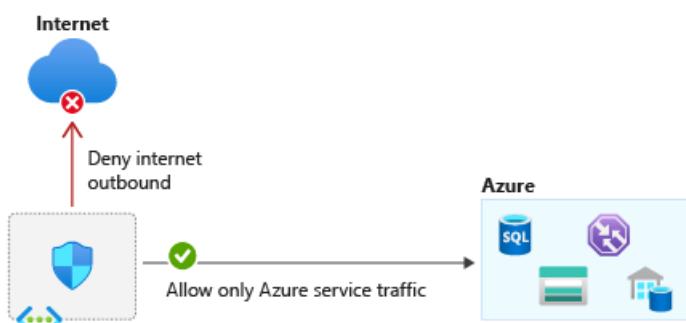
VNet service endpoint provides secure and direct connectivity to Azure services over an optimized route over the Azure backbone network. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Service Endpoints enables private IP addresses in the VNet to reach the endpoint of an Azure service without needing a public IP address on the VNet.



For more information, see [Virtual network service endpoints](#)

Service tags

A service tag represents a group of IP address prefixes from a given Azure service. Using service tags, you can define network access controls on [network security groups](#) or [Azure Firewall](#). By specifying the service tag name (for example, AzureEventGrid) in the appropriate source or destination field of a rule, you can allow or deny the traffic for the corresponding service.



Service Tags in a network security group

Action	Name	Source	Destination	Destination service tag	Protocol
Allow	AllowStorage	VirtualNetwork	Service Tag	Storage	Any
Allow	AllowSQL	VirtualNetwork	Service Tag	Sql.EastUS	Any
Deny	DenyAllOutBound	Any	Any	Any	Any

You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. Create inbound/outbound network security group rules to deny traffic to/from **Internet** and allow traffic to/from **AzureCloud** or other [available service tags](#) of specific Azure services.

For more information about Service Tags and Azure services that support them, see [Service Tags Overview](#)

Next steps

- Learn how to [integrate your app with an Azure network](#).
- Learn how to [restrict access to resources using Service Tags](#).
- Learn how to [connect privately to an Azure Cosmos account using Azure Private Link](#).

Network security groups

2/18/2021 • 9 minutes to read • [Edit Online](#)

You can use an Azure network security group to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains [security rules](#) that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

This article describes properties of a network security group rule, the [default security rules](#) that are applied, and the rule properties that you can modify to create an [augmented security rule](#).

Security rules

A network security group contains zero, or as many rules as desired, within Azure subscription [limits](#). Each rule specifies the following properties:

PROPERTY	EXPLANATION
Name	A unique name within the network security group.
Priority	A number between 100 and 4096. Rules are processed in priority order, with lower numbers processed before higher numbers, because lower numbers have higher priority. Once traffic matches a rule, processing stops. As a result, any rules that exist with lower priorities (higher numbers) that have the same attributes as rules with higher priorities are not processed.
Source or destination	Any, or an individual IP address, classless inter-domain routing (CIDR) block (10.0.0.0/24, for example), service tag, or application security group. If you specify an address for an Azure resource, specify the private IP address assigned to the resource. Network security groups are processed after Azure translates a public IP address to a private IP address for inbound traffic, and before Azure translates a private IP address to a public IP address for outbound traffic. Specifying a range, a service tag, or application security group, enables you to create fewer security rules. The ability to specify multiple individual IP addresses and ranges (you cannot specify multiple service tags or application groups) in a rule is referred to as augmented security rules . Augmented security rules can only be created in network security groups created through the Resource Manager deployment model. You cannot specify multiple IP addresses and IP address ranges in network security groups created through the classic deployment model.
Protocol	TCP, UDP, ICMP or Any.
Direction	Whether the rule applies to inbound, or outbound traffic.

PROPERTY	EXPLANATION
Port range	You can specify an individual or range of ports. For example, you could specify 80 or 10000-10005. Specifying ranges enables you to create fewer security rules. Augmented security rules can only be created in network security groups created through the Resource Manager deployment model. You cannot specify multiple ports or port ranges in the same security rule in network security groups created through the classic deployment model.
Action	Allow or deny

Network security group security rules are evaluated by priority using the 5-tuple information (source, source port, destination, destination port, and protocol) to allow or deny the traffic. You may not create two security rules with the same priority and direction. A flow record is created for existing connections. Communication is allowed or denied based on the connection state of the flow record. The flow record allows a network security group to be stateful. If you specify an outbound security rule to any address over port 80, for example, it's not necessary to specify an inbound security rule for the response to the outbound traffic. You only need to specify an inbound security rule if communication is initiated externally. The opposite is also true. If inbound traffic is allowed over a port, it's not necessary to specify an outbound security rule to respond to traffic over the port.

Existing connections may not be interrupted when you remove a security rule that enabled the flow. Traffic flows are interrupted when connections are stopped and no traffic is flowing in either direction, for at least a few minutes.

There are limits to the number of security rules you can create in a network security group. For details, see [Azure limits](#).

Default security rules

Azure creates the following default rules in each network security group that you create:

Inbound

AllowVNetInBound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow

AllowAzureLoadBalancerInBound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65001	AzureLoadBalancer	0-65535	0.0.0.0/0	0-65535	Any	Allow

DenyAllInbound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny

Outbound

AllowVnetOutBound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow

AllowInternetOutBound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65001	0.0.0.0/0	0-65535	Internet	0-65535	Any	Allow

DenyAllOutBound

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny

In the **Source** and **Destination** columns, *VirtualNetwork*, *AzureLoadBalancer*, and *Internet* are [service tags](#), rather than IP addresses. In the protocol column, Any encompasses TCP, UDP, and ICMP. When creating a rule, you can specify TCP, UDP, ICMP or Any. *0.0.0.0/0* in the **Source** and **Destination** columns represents all addresses. Clients like Azure portal, Azure CLI, or PowerShell can use * or any for this expression.

You cannot remove the default rules, but you can override them by creating rules with higher priorities.

Augmented security rules

Augmented security rules simplify security definition for virtual networks, allowing you to define larger and complex network security policies, with fewer rules. You can combine multiple ports and multiple explicit IP addresses and ranges into a single, easily understood security rule. Use augmented rules in the source, destination, and port fields of a rule. To simplify maintenance of your security rule definition, combine augmented security rules with [service tags](#) or [application security groups](#). There are limits to the number of addresses, ranges, and ports that you can specify in a rule. For details, see [Azure limits](#).

Service tags

A service tag represents a group of IP address prefixes from a given Azure service. It helps to minimize the complexity of frequent updates on network security rules.

For more information, see [Azure service tags](#). For an example on how to use the Storage service tag to restrict network access, see [Restrict network access to PaaS resources](#).

Application security groups

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. You can reuse your security policy at scale without manual maintenance of explicit IP addresses. To learn more, see [Application security groups](#).

Azure platform considerations

- **Virtual IP of the host node:** Basic infrastructure services like DHCP, DNS, IMDS, and health monitoring are provided through the virtualized host IP addresses 168.63.129.16 and 169.254.169.254. These IP addresses belong to Microsoft and are the only virtualized IP addresses used in all regions for this purpose. Effective security rules and effective routes will not include these platform rules. To override this basic infrastructure communication, you can create a security rule to deny traffic by using the following [service tags](#) on your Network Security Group rules: AzurePlatformDNS, AzurePlatformIMDS, AzurePlatformLKM. Learn how to [diagnose network traffic filtering](#) and [diagnose network routing](#).

- **Licensing (Key Management Service):** Windows images running in virtual machines must be licensed. To ensure licensing, a request is sent to the Key Management Service host servers that handle such queries. The request is made outbound through port 1688. For deployments using [default route 0.0.0.0/0](#) configuration, this platform rule will be disabled.
- **Virtual machines in load-balanced pools:** The source port and address range applied are from the originating computer, not the load balancer. The destination port and address range are for the destination computer, not the load balancer.
- **Azure service instances:** Instances of several Azure services, such as HDInsight, Application Service Environments, and Virtual Machine Scale Sets are deployed in virtual network subnets. For a complete list of services you can deploy into virtual networks, see [Virtual network for Azure services](#). Ensure you familiarize yourself with the port requirements for each service before applying a network security group to the subnet the resource is deployed in. If you deny ports required by the service, the service doesn't function properly.
- **Sending outbound email:** Microsoft recommends that you utilize authenticated SMTP relay services (typically connected via TCP port 587, but often others, as well) to send email from Azure Virtual Machines. SMTP relay services specialize in sender reputation, to minimize the possibility that third-party email providers reject messages. Such SMTP relay services include, but are not limited to, Exchange Online Protection and SendGrid. Use of SMTP relay services is in no way restricted in Azure, regardless of your subscription type.

If you created your Azure subscription prior to November 15, 2017, in addition to being able to use SMTP relay services, you can send email directly over TCP port 25. If you created your subscription after November 15, 2017, you may not be able to send email directly over port 25. The behavior of outbound communication over port 25 depends on the type of subscription you have, as follows:

- **Enterprise Agreement:** Outbound port 25 communication is allowed. You are able to send an outbound email directly from virtual machines to external email providers, with no restrictions from the Azure platform.
- **Pay-as-you-go:** Outbound port 25 communication is blocked from all resources. If you need to send email from a virtual machine directly to external email providers (not using an authenticated SMTP relay), you can make a request to remove the restriction. Requests are reviewed and approved at Microsoft's discretion and are only granted after anti-fraud checks are performed. To make a request, open a support case with the issue type *Technical, Virtual Network Connectivity, Cannot send e-mail (SMTP/Port 25)*. In your support case, include details about why your subscription needs to send email directly to mail providers, instead of going through an authenticated SMTP relay. If your subscription is exempted, only virtual machines created after the exemption date are able to communicate outbound over port 25.
- **MSDN, Azure Pass, Azure in Open, Education, BizSpark, and Free trial:** Outbound port 25 communication is blocked from all resources. No requests to remove the restriction can be made, because requests are not granted. If you need to send email from your virtual machine, you have to use an SMTP relay service.
- **Cloud service provider:** Customers that are consuming Azure resources via a cloud service provider can create a support case with their cloud service provider, and request that the provider create an unblock case on their behalf, if a secure SMTP relay cannot be used.

If Azure allows you to send email over port 25, Microsoft cannot guarantee email providers will accept inbound email from your virtual machine. If a specific provider rejects mail from your virtual machine, work directly with the provider to resolve any message delivery or spam filtering issues, or use an authenticated SMTP relay service.

Next steps

- To learn about which Azure resources can be deployed into a virtual network and have network security groups associated to them, see [Virtual network integration for Azure services](#)
- To learn how traffic is evaluated with network security groups, see [How network security groups work](#).
- If you've never created a network security group, you can complete a quick [tutorial](#) to get some experience creating one.
- If you're familiar with network security groups and need to manage them, see [Manage a network security group](#).
- If you're having communication problems and need to troubleshoot network security groups, see [Diagnose a virtual machine network traffic filter problem](#).
- Learn how to enable [network security group flow logs](#) to analyze network traffic to and from resources that have an associated network security group.

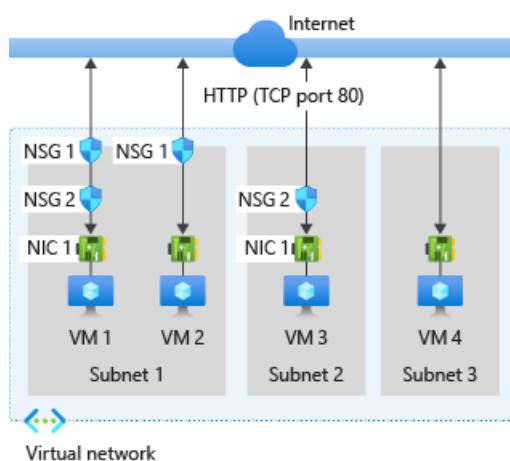
How network security groups filter network traffic

2/18/2021 • 5 minutes to read • [Edit Online](#)

You can use an Azure network security group to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains [security rules](#) that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

You can deploy resources from several Azure services into an Azure virtual network. For a complete list, see [Services that can be deployed into a virtual network](#). You can associate zero, or one, network security group to each virtual network [subnet](#) and [network interface](#) in a virtual machine. The same network security group can be associated to as many subnets and network interfaces as you choose.

The following picture illustrates different scenarios for how network security groups might be deployed to allow network traffic to and from the internet over TCP port 80:



Reference the previous picture, along with the following text, to understand how Azure processes inbound and outbound rules for network security groups:

Inbound traffic

For inbound traffic, Azure processes the rules in a network security group associated to a subnet first, if there is one, and then the rules in a network security group associated to the network interface, if there is one.

- **VM1:** The security rules in *NSG1* are processed, since it is associated to *Subnet1* and *VM1* is in *Subnet1*. Unless you've created a rule that allows port 80 inbound, the traffic is denied by the [DenyAllInbound](#) default security rule, and never evaluated by *NSG2*, since *NSG2* is associated to the network interface. If *NSG1* has a security rule that allows port 80, the traffic is then processed by *NSG2*. To allow port 80 to the virtual machine, both *NSG1* and *NSG2* must have a rule that allows port 80 from the internet.
- **VM2:** The rules in *NSG1* are processed because *VM2* is also in *Subnet1*. Since *VM2* does not have a network security group associated to its network interface, it receives all traffic allowed through *NSG1* or is denied all traffic denied by *NSG1*. Traffic is either allowed or denied to all resources in the same subnet when a network security group is associated to a subnet.
- **VM3:** Since there is no network security group associated to *Subnet2*, traffic is allowed into the subnet and processed by *NSG2*, because *NSG2* is associated to the network interface attached to *VM3*.
- **VM4:** Traffic is allowed to *VM4*, because a network security group isn't associated to *Subnet3*, or the network interface in the virtual machine. All network traffic is allowed through a subnet and network interface if they don't have a network security group associated to them.

Outbound traffic

For outbound traffic, Azure processes the rules in a network security group associated to a network interface first, if there is one, and then the rules in a network security group associated to the subnet, if there is one.

- **VM1:** The security rules in *NSG2* are processed. Unless you create a security rule that denies port 80 outbound to the internet, the traffic is allowed by the [AllowInternetOutbound](#) default security rule in both *NSG1* and *NSG2*. If *NSG2* has a security rule that denies port 80, the traffic is denied, and never evaluated by *NSG1*. To deny port 80 from the virtual machine, either, or both of the network security groups must have a rule that denies port 80 to the internet.
- **VM2:** All traffic is sent through the network interface to the subnet, since the network interface attached to *VM2* does not have a network security group associated to it. The rules in *NSG1* are processed.
- **VM3:** If *NSG2* has a security rule that denies port 80, the traffic is denied. If *NSG2* has a security rule that allows port 80, then port 80 is allowed outbound to the internet, since a network security group is not associated to *Subnet2*.
- **VM4:** All network traffic is allowed from *VM4*, because a network security group isn't associated to the network interface attached to the virtual machine, or to *Subnet3*.

Intra-Subnet traffic

It's important to note that security rules in an NSG associated to a subnet can affect connectivity between VM's within it. For example, if a rule is added to *NSG1* which denies all inbound and outbound traffic, *VM1* and *VM2* will no longer be able to communicate with each other. Another rule would have to be added specifically to allow this.

You can easily view the aggregate rules applied to a network interface by viewing the [effective security rules](#) for a network interface. You can also use the [IP flow verify](#) capability in Azure Network Watcher to determine whether communication is allowed to or from a network interface. IP flow verify tells you whether a communication is allowed or denied, and which network security rule allows or denies the traffic.

NOTE

Network security groups are associated to subnets or to virtual machines and cloud services deployed in the classic deployment model, and to subnets or network interfaces in the Resource Manager deployment model. To learn more about Azure deployment models, see [Understand Azure deployment models](#).

TIP

Unless you have a specific reason to, we recommend that you associate a network security group to a subnet, or a network interface, but not both. Since rules in a network security group associated to a subnet can conflict with rules in a network security group associated to a network interface, you can have unexpected communication problems that require troubleshooting.

Next steps

- To learn about which Azure resources can be deployed into a virtual network and have network security groups associated to them, see [Virtual network integration for Azure services](#).
- If you've never created a network security group, you can complete a quick [tutorial](#) to get some experience creating one.
- If you're familiar with network security groups and need to manage them, see [Manage a network security group](#).
- If you're having communication problems and need to troubleshoot network security groups, see [Diagnose a](#)

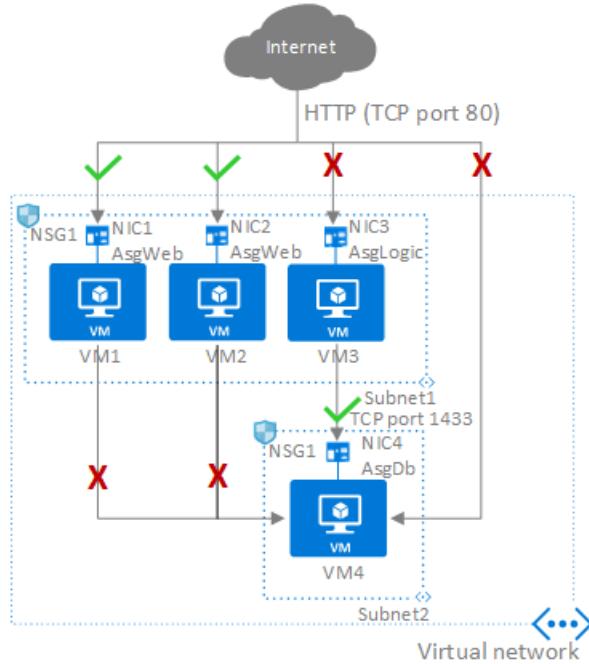
virtual machine network traffic filter problem.

- Learn how to enable [network security group flow logs](#) to analyze network traffic to and from resources that have an associated network security group.

Application security groups

2/18/2021 • 3 minutes to read • [Edit Online](#)

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. You can reuse your security policy at scale without manual maintenance of explicit IP addresses. The platform handles the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic. To better understand application security groups, consider the following example:



In the previous picture, *NIC1* and *NIC2* are members of the *AsgWeb* application security group. *NIC3* is a member of the *AsgLogic* application security group. *NIC4* is a member of the *AsgDb* application security group. Though each network interface in this example is a member of only one network security group, a network interface can be a member of multiple application security groups, up to the [Azure limits](#). None of the network interfaces have an associated network security group. *NSG1* is associated to both subnets and contains the following rules:

Allow-HTTP-Inbound-Internet

This rule is needed to allow traffic from the internet to the web servers. Because inbound traffic from the internet is denied by the **DenyAllInbound** default security rule, no additional rule is needed for the *AsgLogic* or *AsgDb* application security groups.

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
100	Internet	*	AsgWeb	80	TCP	Allow

Deny-Database-All

Because the **AllowVNetInBound** default security rule allows all communication between resources in the same virtual network, this rule is needed to deny traffic from all resources.

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
120	*	*	AsgDb	1433	Any	Deny

Allow-Database-BusinessLogic

This rule allows traffic from the *AsgLogic* application security group to the *AsgDb* application security group. The priority for this rule is higher than the priority for the *Deny-Database-All* rule. As a result, this rule is processed before the *Deny-Database-All* rule, so traffic from the *AsgLogic* application security group is allowed, whereas all other traffic is blocked.

PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
110	AsgLogic	*	AsgDb	1433	TCP	Allow

The rules that specify an application security group as the source or destination are only applied to the network interfaces that are members of the application security group. If the network interface is not a member of an application security group, the rule is not applied to the network interface, even though the network security group is associated to the subnet.

Application security groups have the following constraints:

- There are limits to the number of application security groups you can have in a subscription, as well as other limits related to application security groups. For details, see [Azure limits](#).
- You can specify one application security group as the source and destination in a security rule. You cannot specify multiple application security groups in the source or destination.
- All network interfaces assigned to an application security group have to exist in the same virtual network that the first network interface assigned to the application security group is in. For example, if the first network interface assigned to an application security group named *AsgWeb* is in the virtual network named *VNet1*, then all subsequent network interfaces assigned to *ASGWeb* must exist in *VNet1*. You cannot add network interfaces from different virtual networks to the same application security group.
- If you specify an application security group as the source and destination in a security rule, the network interfaces in both application security groups must exist in the same virtual network. For example, if *AsgLogic* contained network interfaces from *VNet1*, and *AsgDb* contained network interfaces from *VNet2*, you could not assign *AsgLogic* as the source and *AsgDb* as the destination in a rule. All network interfaces for both the source and destination application security groups need to exist in the same virtual network.

TIP

To minimize the number of security rules you need, and the need to change the rules, plan out the application security groups you need and create rules using service tags or application security groups, rather than individual IP addresses, or ranges of IP addresses, whenever possible.

Next steps

- Learn how to [Create a network security group](#).

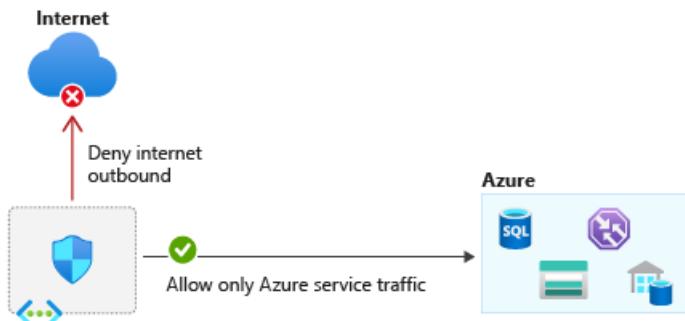
Virtual network service tags

2/18/2021 • 9 minutes to read • [Edit Online](#)

A service tag represents a group of IP address prefixes from a given Azure service. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change, minimizing the complexity of frequent updates to network security rules.

You can use service tags to define network access controls on [network security groups](#) or [Azure Firewall](#). Use service tags in place of specific IP addresses when you create security rules. By specifying the service tag name, such as **ApiManagement**, in the appropriate *source* or *destination* field of a rule, you can allow or deny the traffic for the corresponding service.

You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. Create inbound/outbound network security group rules to deny traffic to/from **Internet** and allow traffic to/from **AzureCloud** or other [available service tags](#) of specific Azure services.



Network Security Group (NSG)				
Action	Name	Source	Destination	Port
Allow	AllowStorage	VirtualNetwork	Storage	Any
Allow	AllowSQL	VirtualNetwork	Sql.EastUS	Any
Deny	DenyAllOutBound	Any	Any	Any

Available service tags

The following table includes all the service tags available for use in [network security group](#) rules.

The columns indicate whether the tag:

- Is suitable for rules that cover inbound or outbound traffic.
- Supports [regional](#) scope.
- Is usable in [Azure Firewall](#) rules.

By default, service tags reflect the ranges for the entire cloud. Some service tags also allow more granular control by restricting the corresponding IP ranges to a specified region. For example, the service tag **Storage** represents Azure Storage for the entire cloud, but **Storage.WestUS** narrows the range to only the storage IP address ranges from the WestUS region. The following table indicates whether each service tag supports such regional scope.

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
ActionGroup	Action Group.	Inbound	No	No
ApiManagement	<p>Management traffic for Azure API Management-dedicated deployments.</p> <p><i>Note:</i> This tag represents the Azure API Management service endpoint for control plane per region. This enables customers to perform management operations on the APIs, Operations, Policies, NamedValues configured on the API Management service.</p>	Inbound	Yes	Yes
ApplicationInsightsAvailability	Application Insights Availability.	Inbound	No	No
AppConfiguration	App Configuration.	Outbound	No	No
AppService	Azure App Service. This tag is recommended for outbound security rules to web apps and Function apps.	Outbound	Yes	Yes
AppServiceManagement	Management traffic for deployments dedicated to App Service Environment.	Both	No	Yes
AzureActiveDirectory	Azure Active Directory.	Outbound	No	Yes
AzureActiveDirectoryDomainServices	Management traffic for deployments dedicated to Azure Active Directory Domain Services.	Both	No	Yes
AzureAdvancedThreatProtection	Azure Advanced Threat Protection.	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureArcInfrastructure	<p>Azure Arc enabled servers, Azure Arc enabled Kubernetes, and Guest Configuration traffic.</p> <p><i>Note:</i> This tag has a dependency on the AzureActiveDirectory, AzureTrafficManager, and AzureResourceManager tags. <i>This tag is not currently configurable via Azure Portal.</i></p>	Outbound	No	Yes
AzureBackup	<p>Azure Backup.</p> <p><i>Note:</i> This tag has a dependency on the Storage and AzureActiveDirectory tags.</p>	Outbound	No	Yes
AzureBotService	Azure Bot Service.	Outbound	No	No
AzureCloud	All datacenter public IP addresses .	Outbound	Yes	Yes
AzureCognitiveSearch	<p>Azure Cognitive Search.</p> <p>This tag or the IP addresses covered by this tag can be used to grant indexers secure access to data sources. Refer to the indexer connection documentation for more details.</p> <p><i>Note:</i> The IP of the search service is not included in the list of IP ranges for this service tag and also needs to be added to the IP firewall of data sources.</p>	Inbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureConnectors	This tag represents the IP addresses used for managed connectors that make inbound webhook callbacks to the Azure Logic Apps service and outbound calls to their respective services, for example, Azure Storage or Azure Event Hubs.	Inbound / Outbound	Yes	Yes
AzureContainerRegistry	Azure Container Registry.	Outbound	Yes	Yes
AzureCosmosDB	Azure Cosmos DB.	Outbound	Yes	Yes
AzureDatabricks	Azure Databricks.	Both	No	No
AzureDataExplorerManagement	Azure Data Explorer Management.	Inbound	No	No
AzureDataLake	Azure Data Lake Storage Gen1.	Outbound	No	Yes
AzureDevSpaces	Azure Dev Spaces.	Outbound	No	No
AzureDevOps	Azure Dev Ops. <i>Note: This tag is not currently configurable via Azure Portal</i>	Inbound	No	Yes
AzureDigitalTwins	Azure Digital Twins. <i>Note: This tag or the IP addresses covered by this tag can be used to restrict access to endpoints configured for event routes. This tag is not currently configurable via Azure Portal</i>	Inbound	No	Yes
AzureEventGrid	Azure Event Grid.	Both	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureFrontDoor.Frontend AzureFrontDoor.Backend AzureFrontDoor.FirstParty	Azure Front Door.	Both	No	No
AzureInformationProtection	Azure Information Protection. <i>Note:</i> This tag has a dependency on the AzureActiveDirectory, AzureFrontDoor.Frontend and AzureFrontDoor.FirstParty tags.	Outbound	No	No
AzureIoTHub	Azure IoT Hub.	Outbound	No	No
AzureKeyVault	Azure Key Vault. <i>Note:</i> This tag has a dependency on the AzureActiveDirectory tag.	Outbound	Yes	Yes
AzureLoadBalancer	The Azure infrastructure load balancer. The tag translates to the virtual IP address of the host (168.63.129.16) where the Azure health probes originate. This only includes probe traffic, not real traffic to your backend resource. If you're not using Azure Load Balancer, you can override this rule.	Both	No	No
AzureMachineLearning	Azure Machine Learning.	Both	No	Yes

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzureMonitor	<p>Log Analytics, Application Insights, AzMon, and custom metrics (GiG endpoints).</p> <p><i>Note:</i> For Log Analytics, this tag has a dependency on the Storage tag.</p>	Outbound	No	Yes
AzureOpenDatasets	<p>Azure Open Datasets.</p> <p><i>Note:</i> This tag has a dependency on the AzureFrontDoor.Frontend and Storage tag.</p>	Outbound	No	No
AzurePlatformDNS	<p>The basic infrastructure (default) DNS service.</p> <p>You can use this tag to disable the default DNS. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No
AzurePlatformIMDS	<p>Azure Instance Metadata Service (IMDS), which is a basic infrastructure service.</p> <p>You can use this tag to disable the default IMDS. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
AzurePlatformLKM	<p>Windows licensing or key management service.</p> <p>You can use this tag to disable the defaults for licensing. Be cautious when you use this tag. We recommend that you read Azure platform considerations. We also recommend that you perform testing before you use this tag.</p>	Outbound	No	No
AzureResourceManager	Azure Resource Manager.	Outbound	No	No
AzureSignalR	Azure SignalR.	Outbound	No	No
AzureSiteRecovery	<p>Azure Site Recovery.</p> <p><i>Note:</i> This tag has a dependency on the AzureActiveDirectory, AzureKeyVault, EventHub, GuestAndHybridManagement and Storage tags.</p>	Outbound	No	No
AzureTrafficManager	<p>Azure Traffic Manager probe IP addresses.</p> <p>For more information on Traffic Manager probe IP addresses, see Azure Traffic Manager FAQ.</p>	Inbound	No	Yes
BatchNodeManagement	Management traffic for deployments dedicated to Azure Batch.	Both	No	Yes
CognitiveServicesManagement	The address ranges for traffic for Azure Cognitive Services.	Both	No	No
DataFactory	Azure Data Factory	Both	No	No
DataFactoryManagement	Management traffic for Azure Data Factory.	Outbound	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
Dynamics365ForMarketingEmail	The address ranges for the marketing email service of Dynamics 365.	Outbound	Yes	No
EventHub	Azure Event Hubs.	Outbound	Yes	Yes
GatewayManager	Management traffic for deployments dedicated to Azure VPN Gateway and Application Gateway.	Inbound	No	No
GuestAndHybridManagement	Azure Automation and Guest Configuration.	Outbound	No	Yes
HDInsight	Azure HDInsight.	Inbound	Yes	No
Internet	The IP address space that's outside the virtual network and reachable by the public internet. The address range includes the Azure-owned public IP address space .	Both	No	No
LogicApps	Logic Apps.	Both	No	No
LogicAppsManagement	Management traffic for Logic Apps.	Inbound	No	No
MicrosoftCloudAppSecurity	Microsoft Cloud App Security.	Outbound	No	No
MicrosoftContainerRegistry	Container registry for Microsoft container images. <i>Note:</i> This tag has a dependency on the AzureFrontDoor.FirstParty tag.	Outbound	Yes	Yes
PowerBI	PowerBi. <i>Note: This tag is not currently configurable via Azure Portal.</i>	Both	No	No
PowerQueryOnline	Power Query Online.	Both	No	No

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
ServiceBus	Azure Service Bus traffic that uses the Premium service tier.	Outbound	Yes	Yes
ServiceFabric	Azure Service Fabric. <i>Note:</i> This tag represents the Service Fabric service endpoint for control plane per region. This enables customers to perform management operations for their Service Fabric clusters from their VNET (endpoint eg. https://westus.servicefabric.azure.com)	Both	No	No
Sql	Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Synapse Analytics. <i>Note:</i> This tag represents the service, but not specific instances of the service. For example, the tag represents the Azure SQL Database service, but not a specific SQL database or server. This tag does not apply to SQL managed instance.	Outbound	Yes	Yes
SqlManagement	Management traffic for SQL-dedicated deployments.	Both	No	Yes

TAG	PURPOSE	CAN USE INBOUND OR OUTBOUND?	CAN BE REGIONAL?	CAN USE WITH AZURE FIREWALL?
Storage	Azure Storage. <i>Note:</i> This tag represents the service, but not specific instances of the service. For example, the tag represents the Azure Storage service, but not a specific Azure Storage account.	Outbound	Yes	Yes
StorageSyncService	Storage Sync Service.	Both	No	No
WindowsVirtualDesktop	Windows Virtual Desktop.	Both	No	Yes
VirtualNetwork	The virtual network address space (all IP address ranges defined for the virtual network), all connected on-premises address spaces, peered virtual networks, virtual networks connected to a virtual network gateway , the virtual IP address of the host , and address prefixes used on user-defined routes . This tag might also contain default routes.	Both	No	No

NOTE

In the classic deployment model (before Azure Resource Manager), a subset of the tags listed in the previous table are supported. These tags are spelled differently:

CLASSIC SPELLING	EQUIVALENT RESOURCE MANAGER TAG
AZURE_LOADBALANCER	AzureLoadBalancer
INTERNET	Internet
VIRTUAL_NETWORK	VirtualNetwork

NOTE

Service tags of Azure services denote the address prefixes from the specific cloud being used. For example, the underlying IP ranges that correspond to the `Sql` tag value on the Azure Public cloud will be different from the underlying ranges on the Azure China cloud.

NOTE

If you implement a [virtual network service endpoint](#) for a service, such as Azure Storage or Azure SQL Database, Azure adds a [route](#) to a virtual network subnet for the service. The address prefixes in the route are the same address prefixes, or CIDR ranges, as those of the corresponding service tag.

Service tags on-premises

You can obtain the current service tag and range information to include as part of your on-premises firewall configurations. This information is the current point-in-time list of the IP ranges that correspond to each service tag. You can obtain the information programmatically or via a JSON file download, as described in the following sections.

Use the Service Tag Discovery API (public preview)

You can programmatically retrieve the current list of service tags together with IP address range details:

- [REST](#)
- [Azure PowerShell](#)
- [Azure CLI](#)

NOTE

While it's in public preview, the Discovery API might return information that's less current than information returned by the JSON downloads. (See the next section.)

Discover service tags by using downloadable JSON files

You can download JSON files that contain the current list of service tags together with IP address range details. These lists are updated and published weekly. Locations for each cloud are:

- [Azure Public](#)
- [Azure US Government](#)
- [Azure China](#)
- [Azure Germany](#)

The IP address ranges in these files are in CIDR notation.

NOTE

A subset of this information has been published in XML files for [Azure Public](#), [Azure China](#), and [Azure Germany](#). These XML downloads will be deprecated by June 30, 2020 and will no longer be available after that date. You should migrate to using the Discovery API or JSON file downloads as described in the previous sections.

Tips

- You can detect updates from one publication to the next by noting increased `changeNumber` values in the JSON file. Each subsection (for example, `Storage.WestUS`) has its own `changeNumber` that's incremented

as changes occur. The top level of the file's *changeNumber* is incremented when any of the subsections is changed.

- For examples of how to parse the service tag information (for example, get all address ranges for Storage in WestUS), see the [Service Tag Discovery API PowerShell](#) documentation.
- When new IP addresses are added to service tags, they will not be used in Azure for at least one week. This gives you time to update any systems that might need to track the IP addresses associated with service tags.

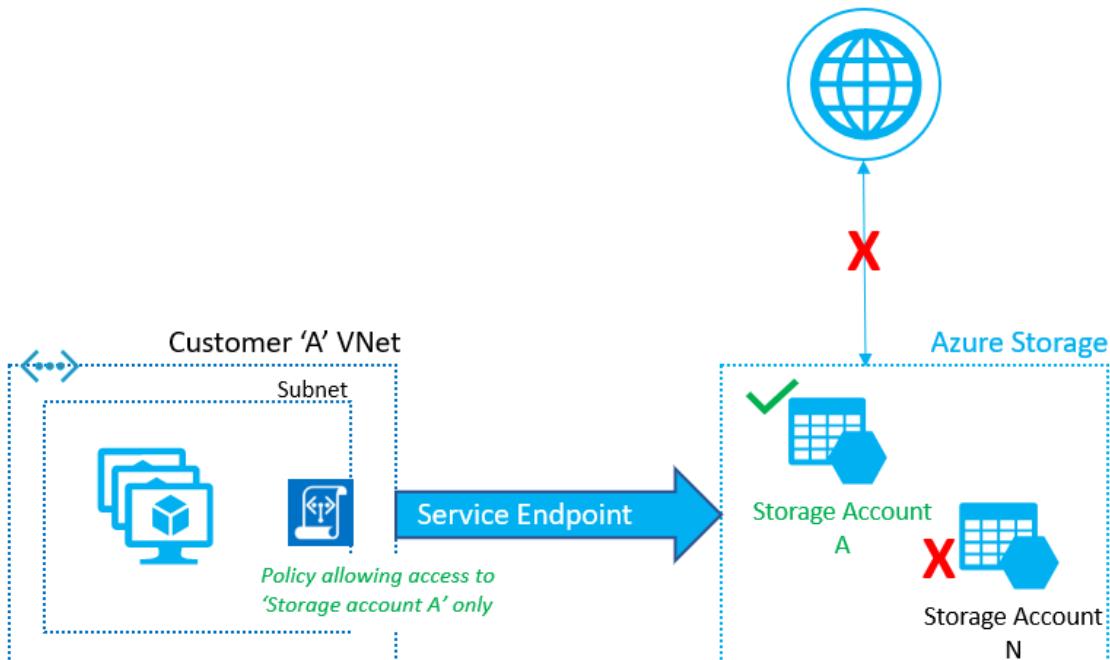
Next steps

- Learn how to [create a network security group](#).

Virtual network service endpoint policies for Azure Storage

2/18/2021 • 6 minutes to read • [Edit Online](#)

Virtual Network (VNet) service endpoint policies allow you to filter egress virtual network traffic to Azure Storage accounts over service endpoint, and allow data exfiltration to only specific Azure Storage accounts. Endpoint policies provide granular access control for virtual network traffic to Azure Storage when connecting over service endpoint.



This feature is generally available for Azure Storage in all global Azure regions.

Key benefits

Virtual network service endpoint policies provide following benefits:

- **Improved security for your Virtual Network traffic to Azure Storage**

Azure service tags for network security groups allow you to restrict virtual network outbound traffic to specific Azure Storage regions. However, this allows traffic to any account within selected Azure Storage region.

Endpoint policies allow you to specify the Azure Storage accounts that are allowed virtual network outbound access and restricts access to all the other storage accounts. This gives much more granular security control for protecting data exfiltration from your virtual network.

- **Scalable, highly available policies to filter Azure service traffic**

Endpoint policies provide horizontally scalable, highly available solution to filter Azure service traffic from virtual networks, over service endpoints. No additional overhead is required to maintain central network appliances for this traffic in your virtual networks.

JSON Object for Service Endpoint policies

Let's take a quick look at the Service Endpoint Policy object.

```

"serviceEndpointPolicyDefinitions": [
    {
        "description": null,
        "name": "MySEP-Definition",
        "resourceGroup": "MySEPD deployment",
        "service": "Microsoft.Storage",
        "serviceResources": [
            "/subscriptions/subscriptionID/resourceGroups/MySEPD deployment/providers/Microsoft.Storage/storageAccounts/my
stgacc"
        ],
        "type": "Microsoft.Network/serviceEndpointPolicies/serviceEndpointPolicyDefinitions"
    }
]

```

Configuration

- You can configure the endpoint policies to restrict virtual network traffic to specific Azure Storage accounts.
- Endpoint policy is configured on a subnet in a virtual network. Service endpoints for Azure Storage should be enabled on the subnet to apply the policy.
- Endpoint policy allows you to add specific Azure Storage accounts to allow list, using the resourceId format. You can restrict access to
 - all storage accounts in a subscription
E.g. /subscriptions/subscriptionId
 - all storage accounts in a resource group
E.g. subscriptions/subscriptionId/resourceGroups/resourceGroupName
 - an individual storage account by listing the corresponding Azure Resource Manager resourceId. This covers traffic to blobs, tables, queues, files and Azure Data Lake Storage Gen2.
E.g. /subscriptions/subscriptionId/resourceGroups/resourceGroupName/providers/Microsoft.Storage/storageAccounts/storageAccour
- By default, if no policies are attached to a subnet with endpoints, you can access all storage accounts in the service. Once a policy is configured on that subnet, only the resources specified in the policy can be accessed from compute instances in that subnet. Access to all other storage accounts will be denied.
- When applying Service Endpoint policies on a subnet, the Azure Storage *Service Endpoint scope* gets upgraded from regional to **global**. This means that all the traffic to Azure Storage is secured over service endpoint thereafter. The Service endpoint policies are also applicable globally, so any storage accounts, that are not explicitly allowed, will be denied access.
- You can apply multiple policies to a subnet. When multiple policies are associated to the subnet, virtual network traffic to resources specified across any of these policies will be allowed. Access to all other service resources, not specified in any of the policies, will be denied.

NOTE

Service endpoint policies are **allow policies**, so apart from the specified resources, all other resources are restricted. Please ensure that all service resource dependencies for your applications are identified and listed in the policy.

- Only storage accounts using the Azure Resource Model can be specified in the endpoint policy. Your classic Azure Storage accounts will not support Azure Service Endpoint Policies.
- RA-GRS secondary access will be automatically allowed if the primary account is listed.
- Storage accounts can be in the same or a different subscription or Azure Active Directory tenant as the virtual network.

Scenarios

- **Peered, connected or multiple virtual networks:** To filter traffic in peered virtual networks, endpoint policies should be applied individually to these virtual networks.
- **Filtering Internet traffic with Network Appliances or Azure Firewall:** Filter Azure service traffic with policies, over service endpoints, and filter rest of the Internet or Azure traffic via appliances or Azure Firewall.
- **Filtering traffic on Azure services deployed into Virtual Networks:** At this time, Azure Service Endpoint Policies are not supported for any managed Azure services that are deployed into your virtual network.
- **Filtering traffic to Azure services from on-premises:** Service endpoint policies only apply to the traffic from subnets associated to the policies. To allow access to specific Azure service resources from on-premises, traffic should be filtered using network virtual appliances or firewalls.

Logging and troubleshooting

No centralized logging is available for service endpoint policies. For service resource logs, see [Service endpoints logging](#).

Troubleshooting scenarios

- Access denied to storage accounts that were working in preview (not in geo-paired region)
 - With Azure Storage upgrading to use Global Service Tags, the scope of Service Endpoint and thus Service Endpoint policies is now Global. So any traffic to Azure Storage is encrypted over Service Endpoints and only Storage accounts that are explicitly listed in policy are allowed access.
 - Explicitly allow list all the required Storage accounts to restore access.
 - Contact Azure support.
- Access is denied for accounts listed in the endpoint policies
 - Network security groups or firewall filtering could be blocking access
 - If removing/re-applying the policy results in connectivity loss:
 - Validate whether the Azure service is configured to allow access from the virtual network over endpoints, or that the default policy for the resource is set to *Allow All*.
 - Validate that the service diagnostics show the traffic over endpoints.
 - Check whether network security group flow logs show the access and that storage logs show the access, as expected, over service endpoints.
 - Contact Azure support.
- Access is denied for accounts not listed in the service endpoint policies
 - Validate whether Azure Storage is configured to allow access from the virtual network over endpoints, or whether the default policy for the resource is set to *Allow All*.
 - Ensure the accounts are not **classic storage accounts** with service endpoint policies on the subnet.
- A managed Azure Service stopped working after applying a Service Endpoint Policy over the subnet
 - Managed services are not supported with service endpoint policies at this time. *Watch this space for updates.*

Provisioning

Service endpoint policies can be configured on subnets by a user with write access to a virtual network. Learn more about Azure [built-in roles](#) and assigning specific permissions to [custom roles](#).

Virtual networks and Azure Storage accounts can be in the same or different subscriptions, or Azure Active Directory tenants.

Limitations

- You can only deploy service endpoint policies on virtual networks deployed through the Azure Resource Manager deployment model.
- Virtual networks must be in the same region as the service endpoint policy.
- You can only apply service endpoint policy on a subnet if service endpoints are configured for the Azure

services listed in the policy.

- You can't use service endpoint policies for traffic from your on-premises network to Azure services.
- Azure managed services do not currently support Endpoint policies. This includes managed services deployed into the shared subnets (e.g. *Azure Batch*, *Azure ADDS*, *Azure Application Gateway*, *Azure VPN Gateway*, *Azure Firewall*) or into the dedicated subnets (e.g. *Azure App Service Environment*, *Azure Redis Cache*, *Azure API Management*, *Azure SQL MI*, *classic managed services*).

WARNING

Azure services deployed into your virtual network, such as Azure HDInsight, access other Azure services, such as Azure Storage, for infrastructure requirements. Restricting endpoint policy to specific resources could break access to these infrastructure resources for the Azure services deployed in your virtual network.

- Classic storage accounts are not supported in endpoint policies. Policies will deny access to all classic storage accounts, by default. If your application needs access to Azure Resource Manager and classic storage accounts, endpoint policies should not be used for this traffic.

Pricing and limits

There is no additional charge for using service endpoint policies. The current pricing model for Azure services (such as, Azure Storage) applies as is today, over service endpoints.

Following limits are enforced on service endpoint policies:

RESOURCE	DEFAULT LIMIT
ServiceEndpointPoliciesPerSubscription	500
ServiceEndpointPoliciesPerSubnet	100
ServiceResourcesPerServiceEndpointPolicyDefinition	200

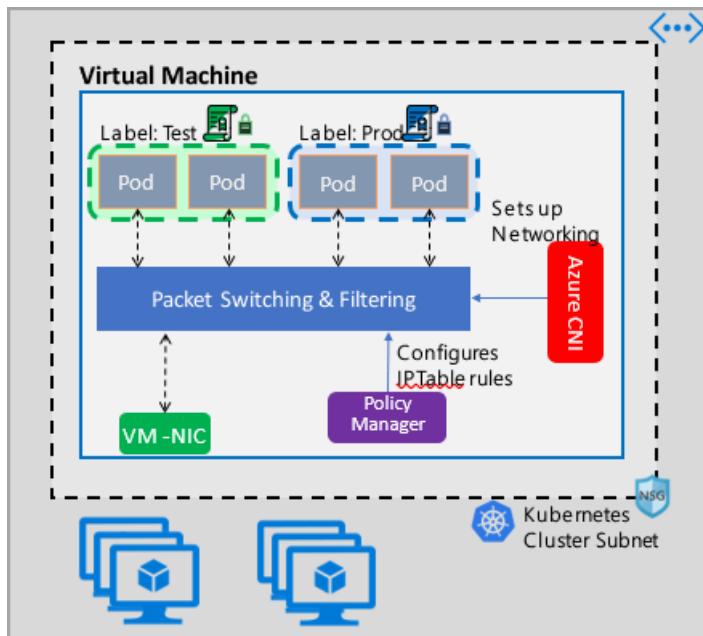
Next Steps

- Learn [how to configure virtual network service endpoint policies](#)
- Learn more about [Virtual network Service Endpoints](#)

Azure Kubernetes Network Policies overview

2/18/2021 • 7 minutes to read • [Edit Online](#)

Network Policies provides micro-segmentation for pods just like Network Security Groups (NSGs) provide micro-segmentation for VMs. The Azure Network Policy Manager (also known as Azure NPM) implementation supports the standard Kubernetes Network Policy specification. You can use labels to select a group of pods and define a list of ingress and egress rules to filter traffic to and from these pods. Learn more about the Kubernetes network policies in the [Kubernetes documentation](#).



Azure NPM implementation works in conjunction with the Azure CNI that provides VNet integration for containers. NPM is supported only on Linux today. The implementation enforces traffic filtering by configuring allow and deny IP rules in Linux IPTables based on the defined policies. These rules are grouped together using Linux IPSets.

Planning security for your Kubernetes cluster

When implementing security for your cluster, use network security groups (NSGs) to filter traffic entering and leaving your cluster subnet (North-South traffic). Use Azure NPM for traffic between pods in your cluster (East-West traffic).

Using Azure NPM

Azure NPM can be used in the following ways to provide micro-segmentation for pods.

Azure Kubernetes Service (AKS)

NPM is available natively in AKS and can be enabled at the time of cluster creation. Learn more about it in [Secure traffic between pods using network policies in Azure Kubernetes Service \(AKS\)](#).

AKS-engine

AKS-Engine is a tool that generates an Azure Resource Manager template for the deployment of a Kubernetes cluster in Azure. The cluster configuration is specified in a JSON file that is passed to the tool when generating the template. To learn more about the entire list of supported cluster settings and their descriptions, see [Microsoft Azure Container Service Engine - Cluster Definition](#).

To enable policies on clusters deployed using acs-engine, specify the value of the networkPolicy setting in the cluster definition file to be "azure".

Example configuration

The below JSON example configuration creates a new virtual network and subnet, and deploys a Kubernetes cluster in it with Azure CNI. We recommend that you use "Notepad" to edit the JSON file.

```
{
  "apiVersion": "vlabs",
  "properties": {
    "orchestratorProfile": {
      "orchestratorType": "Kubernetes",
      "kubernetesConfig": {
        "networkPolicy": "azure"
      }
    },
    "masterProfile": {
      "count": 1,
      "dnsPrefix": "<specify a cluster name>",
      "vmSize": "Standard_D2s_v3"
    },
    "agentPoolProfiles": [
      {
        "name": "agentpool",
        "count": 2,
        "vmSize": "Standard_D2s_v3",
        "availabilityProfile": "AvailabilitySet"
      }
    ],
    "linuxProfile": {
      "adminUsername": "<specify admin username>",
      "ssh": {
        "publicKeys": [
          {
            "keyData": "<cut and paste your ssh key here>"
          }
        ]
      }
    },
    "servicePrincipalProfile": {
      "clientId": "<enter the client ID of your service principal here >",
      "secret": "<enter the password of your service principal here>"
    }
  }
}
```

Do it yourself (DIY) Kubernetes clusters in Azure

For DIY clusters, first install the CNI plug-in and enable it on every virtual machine in a cluster. For detailed instructions, see [Deploy the plug-in for a Kubernetes cluster that you deploy yourself](#).

Once the cluster is deployed run the following `kubectl` command to download and apply the Azure NPM *daemon set* to the cluster.

```
kubectl apply -f https://raw.githubusercontent.com/Azure/acs-engine/master/parts/k8s/addons/kubernetesmasteraddons-azure-npm-daemonset.yaml
```

The solution is also open source and the code is available on the [Azure Container Networking repository](#).

Monitor and Visualize Network Configurations with Azure NPM

Azure NPM includes informative Prometheus metrics that allow you to monitor and better understand your configurations. It provides built-in visualizations in either the Azure portal or Grafana Labs. You can start collecting these metrics using either Azure Monitor or a Prometheus Server.

Benefits of Azure NPM Metrics

Users previously were only able to learn about their Network Configuration with the command `iptables -L` run inside a cluster node, which yields a verbose and difficult to understand output. NPM metrics provide the following benefits related to Network Policies, IPTables Rules, and IP Sets.

- Provides insight into the relationship between the three and a time dimension to debug a configuration.
- Number of entries in all IP Sets and each IP Set.
- Time taken to apply a policy with IPTable/IPSet level granularity.

Supported Metrics

Following is the list of supported metrics:

METRIC NAME	DESCRIPTION	PROMETHEUS METRIC TYPE	LABELS
<code>npm_num_policies</code>	number of network policies	Gauge	-
<code>npm_num_iptables_rules</code>	number of IPTables rules	Gauge	-
<code>npm_num_ipsets</code>	number of IP Sets	Gauge	-
<code>npm_num_ipset_entries</code>	number of IP address entries in all IP Sets	Gauge	-
<code>npm_add_policy_exec_time</code>	runtime for adding a network policy	Summary	quantile (0.5, 0.9, or 0.99)
<code>npm_add_iptables_rule_exec_time</code>	runtime for adding an IPTables rule	Summary	quantile (0.5, 0.9, or 0.99)
<code>npm_add_ipset_exec_time</code>	runtime for adding an IP Set	Summary	quantile (0.5, 0.9, or 0.99)
<code>npm_ipset_counts</code> (advanced)	number of entries within each individual IP Set	GaugeVec	set name & hash

The different quantile levels in "exec_time" metrics help you differentiate between the general and worst case scenarios.

There's also an "exec_time_count" and "exec_time_sum" metric for each "exec_time" Summary metric.

The metrics can be scraped through Azure Monitor for Containers or through Prometheus.

Setup for Azure Monitor

The first step is to enable Azure Monitor for containers for your Kubernetes cluster. Steps can be found in [Azure Monitor for containers Overview](#). Once you have Azure Monitor for containers enabled, configure the [Azure Monitor for containers ConfigMap](#) to enable NPM integration and collection of Prometheus NPM metrics. Azure monitor for containers ConfigMap has an `integrations` section with settings to collect NPM metrics. These settings are disabled by default in the ConfigMap. Enabling the basic setting `collect_basic_metrics = true`, will collect basic NPM metrics. Enabling advanced setting `collect_advanced_metrics = true` will collect advanced metrics in addition to basic metrics.

After editing the ConfigMap, save it locally and apply the ConfigMap to your cluster as follows.

```
kubectl apply -f container-azm-ms-agentconfig.yaml
```

Below is a snippet from the [Azure monitor for containers ConfigMap](#), which shows the NPM integration enabled with advanced metrics collection.

```
integrations: |-
  [integrations.azure_network_policy_manager]
    collect_basic_metrics = false
    collect_advanced_metrics = true
```

Advanced metrics are optional, and turning them on will automatically turn on basic metrics collection.

Advanced metrics currently include only `npm_ipset_counts`

Learn more about [Azure monitor for containers collection settings in config map](#)

Visualization Options for Azure Monitor

Once NPM metrics collection is enabled, you can view the metrics in the Azure portal using Container Insights or in Grafana.

Viewing in Azure portal under Insights for the cluster

Open Azure portal. Once in your cluster's Insights, navigate to "Workbooks" and open "Network Policy Manager (NPM) Configuration".

Besides viewing the workbook (pictures below), you can also directly query the Prometheus metrics in "Logs" under the Insights section. For example, this query will return all the metrics being collected. | where TimeGenerated > ago(5h) | where Name contains "npm_"

You can also query Log Analytics directly for the metrics. Learn more about it with [Getting Started with Log Analytics Queries](#)

Viewing in Grafana Dashboard

Set up your Grafana Server and configure a Log Analytics Data Source as described [here](#). Then, import [Grafana Dashboard with a Log Analytics backend](#) into your Grafana Labs.

The dashboard has visuals similar to the Azure Workbook. You can add panels to chart & visualize NPM metrics from InsightsMetrics table.

Setup for Prometheus Server

Some users may choose to collect metrics with a Prometheus Server instead of Azure Monitor for containers. You merely need to add two jobs to your scrape config to collect NPM metrics.

To install a simple Prometheus Server, add this helm repo on your cluster

```
helm repo add stable https://kubernetes-charts.storage.googleapis.com
helm repo update
```

then add a server

```
helm install prometheus stable/prometheus -n monitoring \
--set pushgateway.enabled=false,alertmanager.enabled=false, \
--set-file extraScrapeConfigs=prometheus-server-scrape-config.yaml
```

where `prometheus-server-scrape-config.yaml` consists of

```

- job_name: "azure-npm-node-metrics"
  metrics_path: /node-metrics
  kubernetes_sd_configs:
    - role: node
  relabel_configs:
    - source_labels: [__address__]
      action: replace
      regex: ([^:]+)(?::\d+)??
      replacement: "$1:10091"
      target_label: __address__
- job_name: "azure-npm-cluster-metrics"
  metrics_path: /cluster-metrics
  kubernetes_sd_configs:
    - role: service
  relabel_configs:
    - source_labels: [__meta_kubernetes_namespace]
      regex: kube-system
      action: keep
    - source_labels: [__meta_kubernetes_service_name]
      regex: npm-metrics-cluster-service
      action: keep
# Comment from here to the end to collect advanced metrics: number of entries for each IPSet
  metric_relabel_configs:
    - source_labels: [__name__]
      regex: npm_ipset_counts
      action: drop

```

You can also replace the `azure-npm-node-metrics` job with the content below or incorporate it into a pre-existing job for Kubernetes pods:

```

- job_name: "azure-npm-node-metrics-from-pod-config"
  metrics_path: /node-metrics
  kubernetes_sd_configs:
    - role: pod
  relabel_configs:
    - source_labels: [__meta_kubernetes_namespace]
      regex: kube-system
      action: keep
    - source_labels: [__meta_kubernetes_pod_annotationpresent_azure_npm_scrapeable]
      action: keep
    - source_labels: [__address__]
      action: replace
      regex: ([^:]+)(?::\d+)??
      replacement: "$1:10091"
      target_label: __address__

```

Visualization Options for Prometheus

When using a Prometheus Server only Grafana Dashboard is supported.

If you haven't already, set up your Grafana Server and configure a Prometheus Data Source. Then, import our [Grafana Dashboard with a Prometheus backend](#) into your Grafana Labs.

The visuals for this dashboard are identical to the dashboard with a Container Insights/Log Analytics backend.

Sample Dashboards

Following are some sample dashboard for NPM metrics in Container Insights (CI) and Grafana

CI Summary Counts

Time Range ⓘ

Last 30 minutes



Network Policy Manager Metrics

Current Counts

Policies

14

IPTables Rules

198

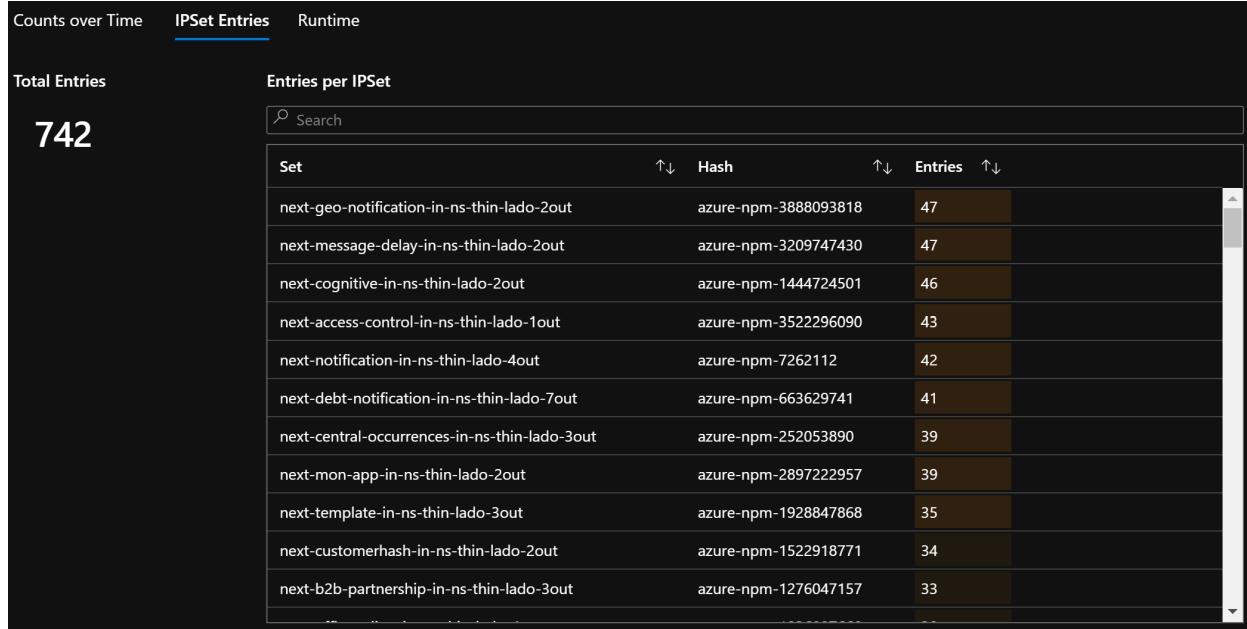
IPSets

162

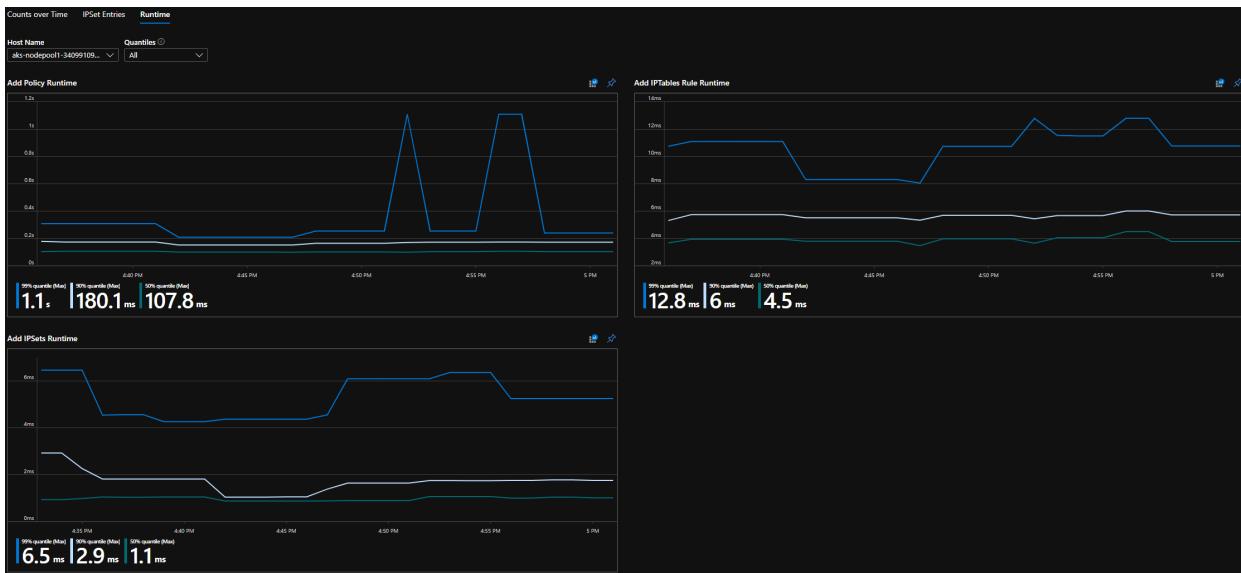
CI Counts over Time



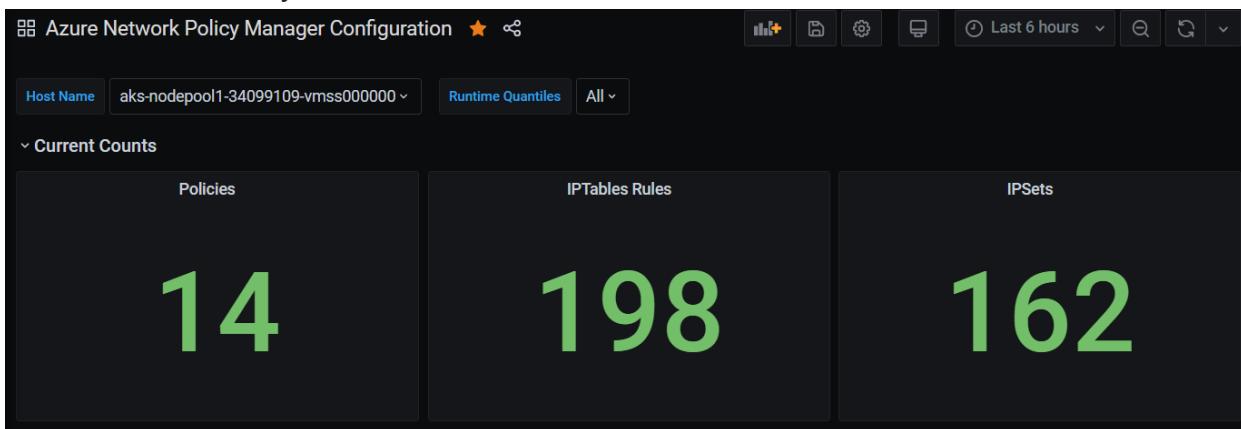
CI IPSet Entries



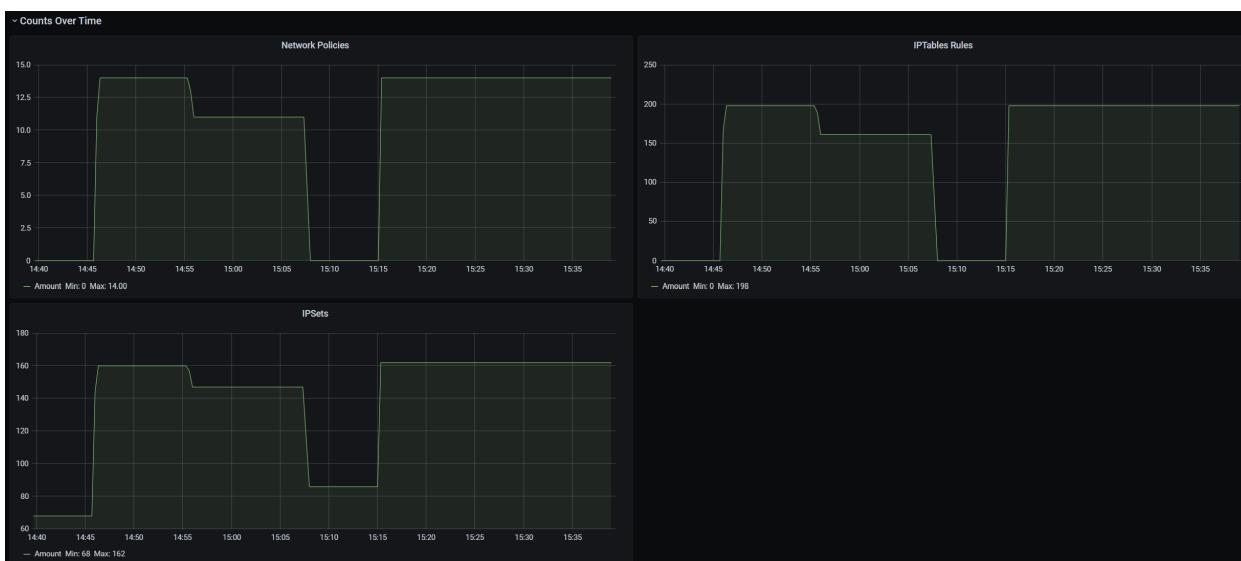
CI Runtime Quantiles



Grafana Dashboard Summary Counts



Grafana Dashboard Counts over Time



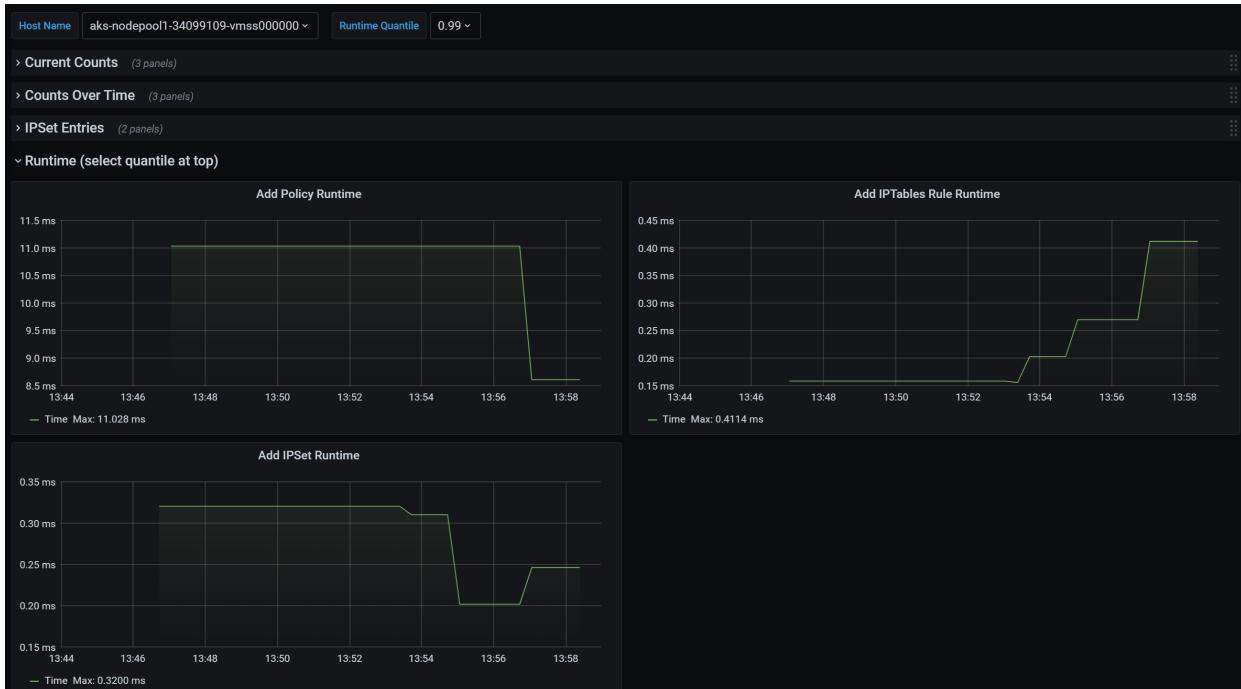
Grafana Dashboard IPSet Entries

IPSet Entries

Total Entries: 742

Entries per IPSet		
Set	Hash	Entries
next-geo-notification-in-ns-thin-lado-2out	azure-npm-3888093818	47
next-message-delay-in-ns-thin-lado-2out	azure-npm-3209747430	47
next-cognitive-in-ns-thin-lado-2out	azure-npm-1444724501	46
next-access-control-in-ns-thin-lado-1out	azure-npm-3522296090	43
next-notification-in-ns-thin-lado-4out	azure-npm-7262112	42
next-debt-notification-in-ns-thin-lado-7out	azure-npm-663629741	41
next-central-occurrences-in-ns-thin-lado-3out	azure-npm-252053890	39
...

Grafana Dashboard Runtime Quantiles



Next steps

- Learn about [Azure Kubernetes Service](#).
- Learn about [container networking](#).
- [Deploy the plug-in](#) for Kubernetes clusters or Docker containers.

Azure DDoS Protection Standard overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

Distributed denial of service (DDoS) attacks are some of the largest availability and security concerns facing customers that are moving their applications to the cloud. A DDoS attack attempts to exhaust an application's resources, making the application unavailable to legitimate users. DDoS attacks can be targeted at any endpoint that is publicly reachable through the internet.

Every property in Azure is protected by Azure's infrastructure DDoS (Basic) Protection at no additional cost. The scale and capacity of the globally deployed Azure network provides defense against common network-layer attacks through always-on traffic monitoring and real-time mitigation. DDoS Protection Basic requires no user configuration or application changes. DDoS Protection Basic helps protect all Azure services, including PaaS services like Azure DNS.

Azure DDoS Protection Standard, combined with application design best practices, provides enhanced DDoS mitigation features to defend against DDoS attacks. It is automatically tuned to help protect your specific Azure resources in a virtual network. Protection is simple to enable on any new or existing virtual network, and it requires no application or resource changes. It has several advantages over the basic service, including logging, alerting, and telemetry.

Feature	DDoS Protection Basic	DDoS Protection Standard
Active traffic monitoring & always on detection	●	●
Automatic attack mitigations	●	●
Availability guarantee	●	●
Cost Protection	●	●
Mitigation policies tuned to customers application	●	●
Metrics & alerts	●	●
Mitigation reports	●	●
Mitigation flow logs	●	●
DDoS rapid response support		●

Azure DDoS protection does not store customer data.

Features

- **Native platform integration:** Natively integrated into Azure. Includes configuration through the Azure portal. DDoS Protection Standard understands your resources and resource configuration.
- **Turnkey protection:** Simplified configuration immediately protects all resources on a virtual network as soon as DDoS Protection Standard is enabled. No intervention or user definition is required.
- **Always-on traffic monitoring:** Your application traffic patterns are monitored 24 hours a day, 7 days a week, looking for indicators of DDoS attacks. DDoS Protection Standard instantly and automatically mitigates the attack, once it is detected.
- **Adaptive tuning:** Intelligent traffic profiling learns your application's traffic over time, and selects and updates the profile that is the most suitable for your service. The profile adjusts as traffic changes over time.
- **Multi-Layered protection:** When deployed with a web application firewall (WAF), DDoS Protection Standard protects both at the network layer (Layer 3 and 4, offered by Azure DDoS Protection Standard) and at the application layer (Layer 7, offered by a WAF). WAF offerings include Azure [Application Gateway WAF](#)

[SKU](#) as well as third-party web application firewall offerings available in the [Azure Marketplace](#).

- **Extensive mitigation scale:** Over 60 different attack types can be mitigated, with global capacity, to protect against the largest known DDoS attacks.
- **Attack analytics:** Get detailed reports in five-minute increments during an attack, and a complete summary after the attack ends. Stream mitigation flow logs to [Azure Sentinel](#) or an offline security information and event management (SIEM) system for near real-time monitoring during an attack.
- **Attack metrics:** Summarized metrics from each attack are accessible through Azure Monitor.
- **Attack alerting:** Alerts can be configured at the start and stop of an attack, and over the attack's duration, using built-in attack metrics. Alerts integrate into your operational software like Microsoft Azure Monitor logs, Splunk, Azure Storage, Email, and the Azure portal.
- **DDoS Rapid Response:** Engage the DDoS Protection Rapid Response (DRR) team for help with attack investigation and analysis. To learn more, see [DDoS Rapid Response](#).
- **Cost guarantee:** Receive data-transfer and application scale-out service credit for resource costs incurred as a result of documented DDoS attacks.

Pricing

DDoS protection plans have a fixed monthly charge of \$2,944 per month which covers up to 100 public IP addresses. Protection for additional resources will cost an additional \$30 per resource per month.

Under a tenant, a single DDoS protection plan can be used across multiple subscriptions, so there is no need to create more than one DDoS protection plan.

To learn about Azure DDoS Protection Standard pricing, see [Azure DDoS Protection Standard pricing](#).

Reference architectures

DDoS Protection Standard is designed for [services that are deployed in a virtual network](#). For other services, the default DDoS Protection Basic service applies. To learn more about supported architectures, see [DDoS Protection reference architectures](#).

Next steps

[Create a DDoS Protection Plan](#)

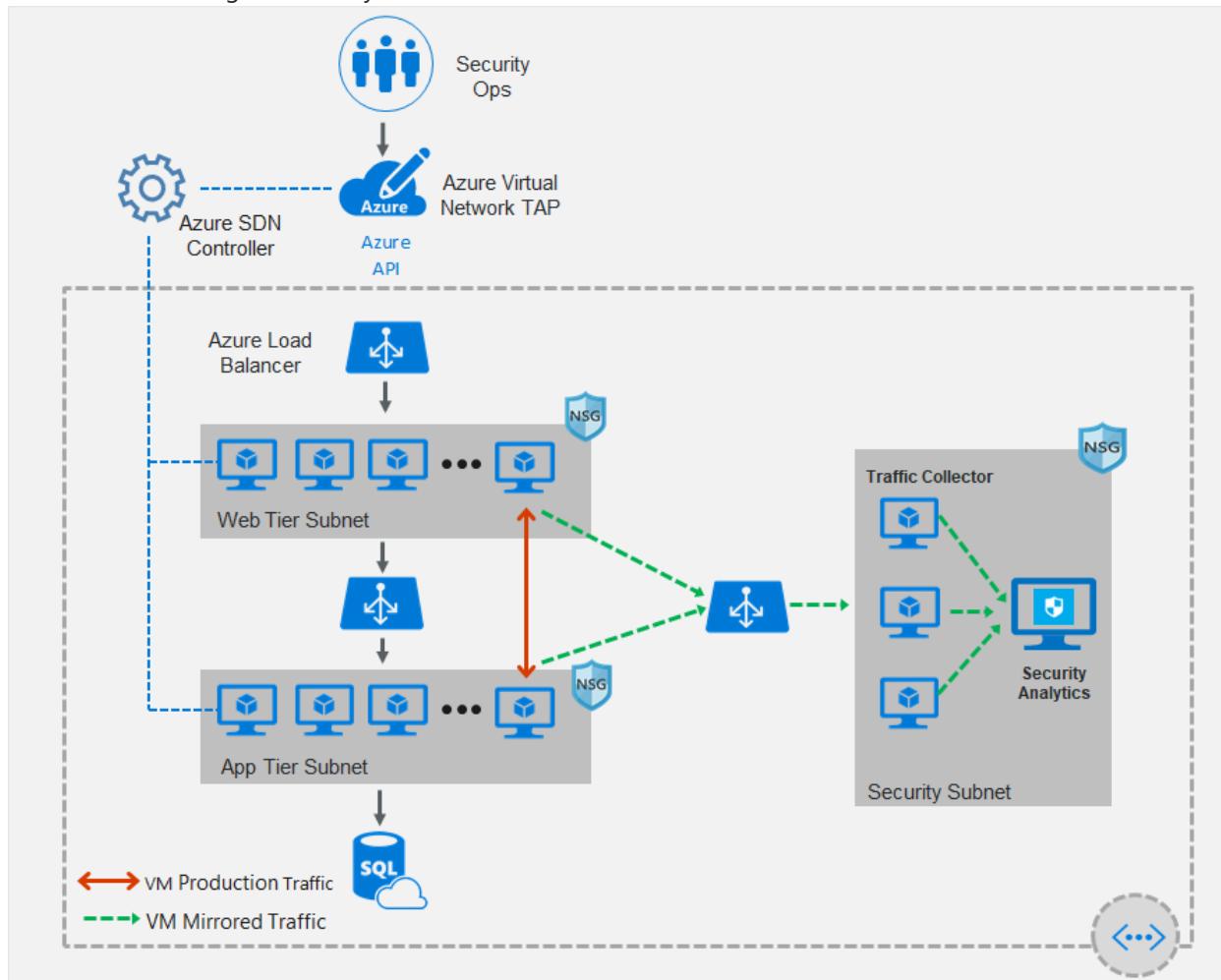
Virtual network TAP

2/18/2021 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Virtual network TAP Preview is currently on hold in all Azure regions. You can email us at azurevnettap@microsoft.com with your subscription ID and we will notify you with future updates about the preview. In the interim, you can use agent based or NVA solutions that provide TAP/Network Visibility functionality through our [Packet Broker partner solutions](#) available in [Azure Marketplace Offerings](#).

Azure virtual network TAP (Terminal Access Point) allows you to continuously stream your virtual machine network traffic to a network packet collector or analytics tool. The collector or analytics tool is provided by a [network virtual appliance](#) partner. For a list of partner solutions that are validated to work with virtual network TAP, see [partner solutions](#). The following picture shows how virtual network TAP works. You can add a TAP configuration on a [network interface](#) that is attached to a virtual machine deployed in your virtual network. The destination is a virtual network IP address in the same virtual network as the monitored network interface or a [peered virtual](#) network. The collector solution for virtual network TAP can be deployed behind an Azure Internal Load balancer for high availability.



Prerequisites

Before you create a virtual network TAP, you must have received a confirmation mail that you are enrolled in the preview, and have one or more virtual machines created using [Azure Resource Manager](#) deployment model and

a partner solution for aggregating the TAP traffic in the same Azure region. If you don't have a partner solution in your virtual network, see [partner solutions](#) to deploy one. You can use the same virtual network TAP resource to aggregate traffic from multiple network interfaces in the same or different subscriptions. If the monitored network interfaces are in different subscriptions, the subscriptions must be associated to the same Azure Active Directory tenant. Additionally, the monitored network interfaces and the destination endpoint for aggregating the TAP traffic can be in peered virtual networks in the same region. If you are using this deployment model ensure that the [virtual network peering](#) is enabled before you configure virtual network TAP.

Permissions

The accounts you use to apply TAP configuration on network interfaces must be assigned to the [network contributor](#) role or a [custom role](#) that is assigned the necessary actions from the following table:

ACTION	NAME
Microsoft.Network/virtualNetworkTaps/*	Required to create, update, read and delete a virtual network TAP resource
Microsoft.Network/networkInterfaces/read	Required to read the network interface resource on which the TAP will be configured
Microsoft.Network/tapConfigurations/*	Required to create, update, read and delete the TAP configuration on a network interface

Virtual network TAP partner solutions

Network packet brokers

- [GigaVUE Cloud Suite for Azure](#)
- [Ixia CloudLens](#)
- [Nubeva Prisms](#)
- [Big Switch Big Monitoring Fabric](#)

Security analytics, network/application performance management

- [Awake Security](#)
- [Cisco Stealthwatch Cloud](#)
- [Darktrace](#)
- [ExtraHop Reveal\(x\)](#)
- [Fidelis Cybersecurity](#)
- [Flowmon](#)
- [NetFort LANGuardian](#)
- [Netscout vSTREAM](#)
- [Riverbed SteelCentral AppResponse](#)
- [RSA NetWitness® Platform](#)
- [Vectra Cognito](#)

Next steps

- Learn how to [Create a virtual network TAP](#).

Azure Policy Regulatory Compliance controls for Azure Virtual Network

2/18/2021 • 19 minutes to read • [Edit Online](#)

Regulatory Compliance in Azure Policy provides Microsoft created and managed initiative definitions, known as **built-ins**, for the **compliance domains** and **security controls** related to different compliance standards. This page lists the **compliance domains** and **security controls** for Azure Virtual Network. You can assign the built-ins for a **security control** individually to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the [Azure Policy GitHub repo](#).

IMPORTANT

Each control below is associated with one or more Azure Policy definitions. These policies may help you [assess compliance](#) with the control; however, there often is not a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves; this doesn't ensure you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards may change over time.

Azure Security Benchmark

The [Azure Security Benchmark](#) provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Azure Security Benchmark, see the [Azure Security Benchmark mapping files](#).

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Azure Security Benchmark](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
Network Security	NS-1	Implement security for internal traffic	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Network Security	NS-1	Implement security for internal traffic	Subnets should be associated with a Network Security Group	3.0.0
Network Security	NS-4	Protect applications and services from external network attacks	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network Security	NS-4	Protect applications and services from external network attacks	Azure DDoS Protection Standard should be enabled	3.0.0
Network Security	NS-4	Protect applications and services from external network attacks	RDP access from the Internet should be blocked	2.0.0
Network Security	NS-4	Protect applications and services from external network attacks	SSH access from the Internet should be blocked	2.0.0
Network Security	NS-4	Protect applications and services from external network attacks	Subnets should be associated with a Network Security Group	3.0.0
Network Security	NS-4	Protect applications and services from external network attacks	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
Network Security	NS-4	Protect applications and services from external network attacks	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
Network Security	NS-5	Deploy intrusion detection/intrusion prevention systems (IDS/IPS)	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Logging and Threat Detection	LT-3	Enable logging for Azure network activities	Network Watcher should be enabled	1.1.0

Azure Security Benchmark v1

The [Azure Security Benchmark](#) provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Azure Security Benchmark, see the [Azure Security Benchmark mapping files](#).

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Azure Security Benchmark](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
--------	------------	---------------	-----------------------	-------------------------

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network Security	1.1	Protect resources using Network Security Groups or Azure Firewall on your Virtual Network	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Network Security	1.1	Protect resources using Network Security Groups or Azure Firewall on your Virtual Network	Subnets should be associated with a Network Security Group	3.0.0
Network Security	1.1	Protect resources using Network Security Groups or Azure Firewall on your Virtual Network	Virtual machines should be connected to an approved virtual network	1.0.0
Network Security	1.1	Protect resources using Network Security Groups or Azure Firewall on your Virtual Network	Virtual networks should use specified virtual network gateway	1.0.0
Network Security	1.2	Monitor and log the configuration and traffic of Vnets, Subnets, and NICs	Network Watcher should be enabled	1.1.0
Network Security	1.4	Deny communications with known malicious IP addresses	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Network Security	1.4	Deny communications with known malicious IP addresses	Azure DDoS Protection Standard should be enabled	3.0.0
Network Security	1.5	Record network packets and flow logs	Network Watcher should be enabled	1.1.0

CIS Microsoft Azure Foundations Benchmark

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.1.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
--------	------------	---------------	--------------------------	----------------------------

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Security Center	2.9	Ensure ASC Default policy setting "Enable Next Generation Firewall(NGFW) Monitoring" is not "Disabled"	Subnets should be associated with a Network Security Group	3.0.0
Networking	6.1	Ensure that RDP access is restricted from the internet	RDP access from the Internet should be blocked	2.0.0
Networking	6.2	Ensure that SSH access is restricted from the internet	SSH access from the Internet should be blocked	2.0.0
Networking	6.5	Ensure that Network Watcher is 'Enabled'	Network Watcher should be enabled	1.1.0

CMMC Level 3

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CMMC Level 3](#). For more information about this compliance standard, see [Cybersecurity Maturity Model Certification \(CMMC\)](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	RDP access from the Internet should be blocked	2.0.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	SSH access from the Internet should be blocked	2.0.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	RDP access from the Internet should be blocked	2.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	SSH access from the Internet should be blocked	2.0.0
Access Control	AC.1.003	Verify and control/limit connections to and use of external information systems.	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Access Control	AC.1.003	Verify and control/limit connections to and use of external information systems.	RDP access from the Internet should be blocked	2.0.0
Access Control	AC.1.003	Verify and control/limit connections to and use of external information systems.	SSH access from the Internet should be blocked	2.0.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Network Watcher should be enabled	1.1.0
Access Control	AC.2.015	Route remote access via managed access control points.	RDP access from the Internet should be blocked	2.0.0
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	RDP access from the Internet should be blocked	2.0.0
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Subnets should be associated with a Network Security Group	3.0.0
Incident Response	IR.2.093	Detect and report events.	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Incident Response	IR.2.093	Detect and report events.	Flow log should be configured for every network security group	1.0.0
Incident Response	IR.2.093	Detect and report events.	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Incident Response	IR.2.093	Detect and report events.	Web Application Firewall (WAF) should be enabled for Azure Front Door Service	1.0.1
Incident Response	IR.2.093	Detect and report events.	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
Incident Response	IR.2.093	Detect and report events.	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Flow log should be configured for every network security group	1.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Network Watcher should be enabled	1.1.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
System and Communications Protection	SC.1.176	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Subnets should be associated with a Network Security Group	3.0.0
System and Communications Protection	SC.3.180	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Subnets should be associated with a Network Security Group	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Flow log should be configured for every network security group	1.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Network Watcher should be enabled	1.1.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Subnets should be associated with a Network Security Group	3.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Flow log should be configured for every network security group	1.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Network Watcher should be enabled	1.1.0
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
System and Information Integrity	SI.2.216	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0
System and Information Integrity	SI.2.217	Identify unauthorized use of organizational systems.	Network Watcher should be enabled	1.1.0

HIPAA HITRUST 9.2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - HIPAA HITRUST 9.2](#). For more information about this compliance standard, see [HIPAA HITRUST 9.2](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
Segregation in Networks	0805.01m1Organizational.12 - 01.m	The organization's security gateways (e.g. firewalls) enforce security policies and are configured to filter traffic between domains, block unauthorized access, and are used to maintain segregation between internal wired, internal wireless, and external network segments (e.g., the Internet) including DMZs and enforce access control policies for each of the domains.	Gateway subnets should not be configured with a network security group	1.0.0
Segregation in Networks	0805.01m1Organizational.12 - 01.m	The organization's security gateways (e.g. firewalls) enforce security policies and are configured to filter traffic between domains, block unauthorized access, and are used to maintain segregation between internal wired, internal wireless, and external network segments (e.g., the Internet) including DMZs and enforce access control policies for each of the domains.	Subnets should be associated with a Network Security Group	3.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Segregation in Networks	0805.01m1Organizational.12 - 01.m	The organization's security gateways (e.g. firewalls) enforce security policies and are configured to filter traffic between domains, block unauthorized access, and are used to maintain segregation between internal wired, internal wireless, and external network segments (e.g., the Internet) including DMZs and enforce access control policies for each of the domains.	Virtual machines should be connected to an approved virtual network	1.0.0
Segregation in Networks	0806.01m2Organizational.12356 - 01.m	The organization's network is logically and physically segmented with a defined security perimeter and a graduated set of controls, including subnetworks for publicly accessible system components that are logically separated from the internal network, based on organizational requirements; and traffic is controlled based on functionality required and classification of the data/systems based on a risk assessment and their respective security requirements.	Gateway subnets should not be configured with a network security group	1.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Segregation in Networks	0806.01m2Organizational.12356 - 01.m	The organizations network is logically and physically segmented with a defined security perimeter and a graduated set of controls, including subnetworks for publicly accessible system components that are logically separated from the internal network, based on organizational requirements; and traffic is controlled based on functionality required and classification of the data/systems based on a risk assessment and their respective security requirements.	Subnets should be associated with a Network Security Group	3.0.0
Segregation in Networks	0806.01m2Organizational.12356 - 01.m	The organizations network is logically and physically segmented with a defined security perimeter and a graduated set of controls, including subnetworks for publicly accessible system components that are logically separated from the internal network, based on organizational requirements; and traffic is controlled based on functionality required and classification of the data/systems based on a risk assessment and their respective security requirements.	Virtual machines should be connected to an approved virtual network	1.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Segregation in Networks	0894.01m2Organizational.7 - 01.m	Networks are segregated from production-level networks when migrating physical servers, applications or data to virtualized servers.	Deploy network watcher when virtual networks are created	1.0.0
Segregation in Networks	0894.01m2Organizational.7 - 01.m	Networks are segregated from production-level networks when migrating physical servers, applications or data to virtualized servers.	Gateway subnets should not be configured with a network security group	1.0.0
Segregation in Networks	0894.01m2Organizational.7 - 01.m	Networks are segregated from production-level networks when migrating physical servers, applications or data to virtualized servers.	Subnets should be associated with a Network Security Group	3.0.0
Segregation in Networks	0894.01m2Organizational.7 - 01.m	Networks are segregated from production-level networks when migrating physical servers, applications or data to virtualized servers.	Virtual machines should be connected to an approved virtual network	1.0.0
Network Connection Control	0809.01n2Organizational.1234 - 01.n	Network traffic is controlled in accordance with the organizations access control policy through firewall and other network-related restrictions for each network access point or external telecommunication service's managed interface.	Subnets should be associated with a Network Security Group	3.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network Connection Control	0809.01n2Organizational.1234 - 01.n	Network traffic is controlled in accordance with the organizations access control policy through firewall and other network-related restrictions for each network access point or external telecommunication service's managed interface.	Virtual machines should be connected to an approved virtual network	1.0.0
Network Connection Control	0810.01n2Organizational.5 - 01.n	Transmitted information is secured and, at a minimum, encrypted over open, public networks.	Subnets should be associated with a Network Security Group	3.0.0
Network Connection Control	0810.01n2Organizational.5 - 01.n	Transmitted information is secured and, at a minimum, encrypted over open, public networks.	Virtual machines should be connected to an approved virtual network	1.0.0
Network Connection Control	0811.01n2Organizational.6 - 01.n	Exceptions to the traffic flow policy are documented with a supporting mission/business need, duration of the exception, and reviewed at least annually; traffic flow policy exceptions are removed when no longer supported by an explicit mission/business need.	Subnets should be associated with a Network Security Group	3.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network Connection Control	0811.01n2Organizational.6 - 01.n	Exceptions to the traffic flow policy are documented with a supporting mission/business need, duration of the exception, and reviewed at least annually; traffic flow policy exceptions are removed when no longer supported by an explicit mission/business need.	Virtual machines should be connected to an approved virtual network	1.0.0
Network Connection Control	0812.01n2Organizational.8 - 01.n	Remote devices establishing a non-remote connection are not allowed to communicate with external (remote) resources.	Subnets should be associated with a Network Security Group	3.0.0
Network Connection Control	0812.01n2Organizational.8 - 01.n	Remote devices establishing a non-remote connection are not allowed to communicate with external (remote) resources.	Virtual machines should be connected to an approved virtual network	1.0.0
Network Connection Control	0814.01n1Organizational.12 - 01.n	The ability of users to connect to the internal network is restricted using a deny-by-default and allow-by-exception policy at managed interfaces according to the access control policy and the requirements of clinical and business applications.	Subnets should be associated with a Network Security Group	3.0.0
Network Connection Control	0814.01n1Organizational.12 - 01.n	The ability of users to connect to the internal network is restricted using a deny-by-default and allow-by-exception policy at managed interfaces according to the access control policy and the requirements of clinical and business applications.	Virtual machines should be connected to an approved virtual network	1.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network Controls	0860.09m1Organizational.9 - 09.m	The organization formally manages equipment on the network, including equipment in user areas.	Deploy Diagnostic Settings for Network Security Groups	1.0.0
Security of Network Services	0837.09.n2Organizational.2 - 09.n	Formal agreements with external information system providers include specific obligations for security and privacy.	Network Watcher should be enabled	1.1.0
Security of Network Services	0886.09n2Organizational.4 - 09.n	The organization employs and documents in a formal agreement or other document, either i) allow-all, deny-by-exception, or, ii) deny-all, permit-by-exception (preferred), policy for allowing specific information systems to connect to external information systems.	Network Watcher should be enabled	1.1.0
Security of Network Services	0888.09n2Organizational.6 - 09.n	The contract with the external/outsourced service provider includes the specification that the service provider is responsible for the protection of covered information shared.	Network Watcher should be enabled	1.1.0

New Zealand ISM

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - New Zealand Information Security Manual](#). For more information about this compliance standard, see [New Zealand Information Security Manual](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
Network security	NS-5	18.3.19 Content of a Denial of Service (DoS) response plan	Azure DDoS Protection Standard should be enabled	3.0.0

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY	POLICY VERSION
Network security	NS-7	18.4.10 Malicious code counter-measures	All Internet traffic should be routed via your deployed Azure Firewall	3.0.0-preview
Network security	NS-7	18.4.10 Malicious code counter-measures	Web Application Firewall (WAF) should be enabled for Application Gateway	1.0.1
Network security	NS-7	18.4.10 Malicious code counter-measures	Web Application Firewall (WAF) should be enabled for Azure Front Door Service service	1.0.1
Network security	NS-7	18.4.10 Malicious code counter-measures	Web Application Firewall (WAF) should use the specified mode for Application Gateway	1.0.0
Network security	NS-7	18.4.10 Malicious code counter-measures	Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service	1.0.0

NIST SP 800-171 R2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-171 R2](#). For more information about this compliance standard, see [NIST SP 800-171 R2](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
System and Information Integrity	3.14.6	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Network Watcher should be enabled	1.1.0

NIST SP 800-53 R4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 R4](#). For more information about this compliance standard, see [NIST SP 800-53 R4](#).

DOMAIN	CONTROL ID	CONTROL TITLE	POLICY (AZURE PORTAL)	POLICY VERSION (GITHUB)
System and Communications Protection	SC-5	Denial of Service Protection	Azure DDoS Protection Standard should be enabled	3.0.0

Next steps

- Learn more about [Azure Policy Regulatory Compliance](#).
- See the built-ins on the [Azure Policy GitHub repo](#).

Subnet extension

2/18/2021 • 2 minutes to read • [Edit Online](#)

Workload migration to the public cloud requires careful planning and coordination. One of the key considerations can be the ability to retain your IP addresses. Which can be important especially if your applications have IP address dependency or you have compliance requirements to use specific IP addresses. Azure Virtual Network solves this problem for you by allowing you to create VNet and Subnets using an IP address range of your choice.

Migrations can get a bit challenging when the above requirement is coupled with an additional requirement to keep some applications on-premises. In such a situation, you'll have to split the applications between Azure and on-premises, without renumbering the IP addresses on either side. Additionally, you'll have to allow the applications to communicate as if they are in the same network.

One solution to the above problem is subnet extension. Extending a network allows applications to talk over the same broadcast domain when they exist at different physical locations, removing the need to rearchitect your network topology.

While extending your network isn't a good practice in general, below use cases can make it necessary.

- **Phased Migration:** The most common scenario is that you want to phase your migration. You want to bring a few applications first and over time migrate rest of the applications to Azure.
- **Latency:** Low latency requirements can be another reason for you to keep some applications on-premises to ensure that they're as close as possible to your datacenter.
- **Compliance:** Another use case is that you might have compliance requirements to keep some of your applications on-premises.

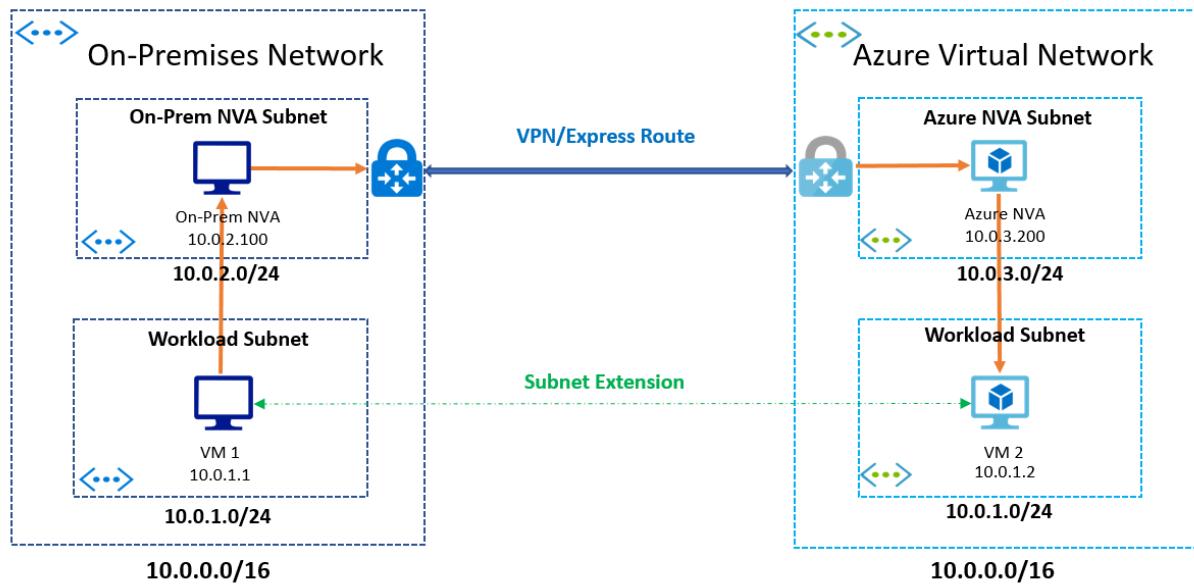
NOTE

You should not extend your subnets unless it is necessary. In the cases where you do extend your subnets, you should try to make it an intermediate step. With time, you should try re-number applications in your on-premises network and migrate them to Azure.

In the next section, we'll discuss how you can extend your subnets into Azure.

Extend your subnet to Azure

You can extend your on-premises subnets to Azure using a layer-3 overlay network based solution. Most solutions use an overlay technology such as VXLAN to extend the layer-2 network using an layer-3 overlay network. The diagram below shows a generalized solution. In this solution, the same subnet exists on both sides that is, Azure and on-premises.



The IP addresses from the subnet are assigned to VMs on Azure and on-premises. Both Azure and on-premises have an NVA inserted in their networks. When a VM in Azure tries to talk to a VM in on-premises network, the Azure NVA captures the packet, encapsulates it, and sends it over VPN/Express Route to the on-premises network. The on-premises NVA receives the packet, decapsulates it and forwards it to the intended recipient in its network. The return traffic uses a similar path and logic.

In the above example, the Azure NVA and the on-premises NVA communicate and learn about IP addresses behind each other. More complex networks can also have a mapping service, which maintains the mapping between the NVAs and the IP addresses behind them. When an NVA receives a packet, it queries the mapping service to find out the address of the NVA that has the destination IP address behind it.

In the next section, you'll find details on subnet extension solutions we've tested on Azure.

Next steps

[Extend your subnet to Azure using vendor solutions.](#)

What is subnet delegation?

2/18/2021 • 2 minutes to read • [Edit Online](#)

Subnet delegation enables you to designate a specific subnet for an Azure PaaS service of your choice that needs to be injected into your virtual network. Subnet delegation provides full control to the customer on managing the integration of Azure services into their virtual networks.

When you delegate a subnet to an Azure service, you allow that service to establish some basic network configuration rules for that subnet, which help the Azure service operate their instances in a stable manner. As a result, the Azure service may establish some pre or post deployment conditions, such as:

- deploy the service in a shared versus dedicated subnet.
- add to the service a set of Network Intent Policies post deployment that is required for the service to work properly.

Advantages of subnet delegation

Delegating a subnet to specific services provides the following advantages:

- helps to designate a subnet for one or more Azure services and manage the instances in the subnet as per requirements. For example, the virtual network owner can define the following for a delegated subnet to better manage resources and access as follows:
 - network filtering traffic policies with network security groups.
 - routing policies with user-defined routes.
 - services integration with service endpoints configurations.
- helps injected services to better integrate with the virtual network by defining their pre-conditions of deployments in the form of Network Intent Policies. This ensures any actions that can affect functioning of the injected service can be blocked at PUT.

Who can delegate?

Subnet delegation is an exercise that the virtual network owners need to perform to designate one of the subnets for a specific Azure Service. Azure Service in turn deploys the instances into this subnet for consumption by the customer workloads.

Impact of subnet delegation on your subnet

Each Azure service defines their own deployment model, where they can define what properties they do or do not support in a delegated subnet for injection purposes, such as follows:

- shared subnet with other Azure Services or VM / virtual machine scale set in the same subnet, or it only supports a dedicated subnet with only instances of this service in it.
- supports NSG association with the delegated subnet.
- supports NSG associated with the delegated subnet can be also associated with any other subnet.
- allows route table association with the delegated subnet.
- allows the route table associated with the delegated subnet to be associated with any other subnet.
- dictates the minimum number of IP Addresses in the delegated subnet.
- dictates the IP Address space in the delegated subnet to be from Private IP Address space (10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12).

- dictates that the custom DNS configuration has an Azure DNS entry.
- requires delegation to be removed before the subnet or virtual network can be deleted.
- cannot be used with a private endpoint if the subnet is delegated.

Injected services can also add their own policies as follows:

- **Security policies:** Collection of security rules required for a given service to work.
- **Route policies:** Collection of routes required for a given service to work.

What subnet delegation does not do

The Azure services being injected into a delegated subnet still have the basic set of properties that are available for non-delegated subnets, such as:

- Azure services can inject instances into customer subnets, but cannot impact the existing workloads.
- The policies or routes that these services apply are flexible and can be overridden by the customer.

Next steps

- [Delegate a subnet](#)

Plan virtual networks

2/18/2021 • 10 minutes to read • [Edit Online](#)

Creating a virtual network to experiment with is easy enough, but chances are, you will deploy multiple virtual networks over time to support the production needs of your organization. With some planning, you will be able to deploy virtual networks and connect the resources you need more effectively. The information in this article is most helpful if you're already familiar with virtual networks and have some experience working with them. If you are not familiar with virtual networks, it's recommended that you read [Virtual network overview](#).

Naming

All Azure resources have a name. The name must be unique within a scope, that may vary for each resource type. For example, the name of a virtual network must be unique within a [resource group](#), but can be duplicated within a [subscription](#) or Azure [region](#). Defining a naming convention that you can use consistently when naming resources is helpful when managing several network resources over time. For suggestions, see [Naming conventions](#).

Regions

All Azure resources are created in an Azure region and subscription. A resource can only be created in a virtual network that exists in the same region and subscription as the resource. You can however, connect virtual networks that exist in different subscriptions and regions. For more information, see [connectivity](#). When deciding which region(s) to deploy resources in, consider where consumers of the resources are physically located:

- Consumers of resources typically want the lowest network latency to their resources. To determine relative latencies between a specified location and Azure regions, see [View relative latencies](#).
- Do you have data residency, sovereignty, compliance, or resiliency requirements? If so, choosing the region that aligns to the requirements is critical. For more information, see [Azure geographies](#).
- Do you require resiliency across Azure Availability Zones within the same Azure region for the resources you deploy? You can deploy resources, such as virtual machines (VM) to different availability zones within the same virtual network. Not all Azure regions support availability zones however. To learn more about availability zones and the regions that support them, see [Availability zones](#).

Subscriptions

You can deploy as many virtual networks as required within each subscription, up to the [limit](#). Some organizations have different subscriptions for different departments, for example. For more information and considerations around subscriptions, see [Subscription governance](#).

Segmentation

You can create multiple virtual networks per subscription and per region. You can create multiple subnets within each virtual network. The considerations that follow help you determine how many virtual networks and subnets you require:

Virtual networks

A virtual network is a virtual, isolated portion of the Azure public network. Each virtual network is dedicated to your subscription. Things to consider when deciding whether to create one virtual network, or multiple virtual networks in a subscription:

- Do any organizational security requirements exist for isolating traffic into separate virtual networks? You can choose to connect virtual networks or not. If you connect virtual networks, you can implement a network virtual appliance, such as a firewall, to control the flow of traffic between the virtual networks. For more information, see [security](#) and [connectivity](#).
- Do any organizational requirements exist for isolating virtual networks into separate [subscriptions](#) or [regions](#)?
- A [network interface](#) enables a VM to communicate with other resources. Each network interface has one or more private IP addresses assigned to it. How many network interfaces and [private IP addresses](#) do you require in a virtual network? There are [limits](#) to the number of network interfaces and private IP addresses that you can have within a virtual network.
- Do you want to connect the virtual network to another virtual network or on-premises network? You may choose to connect some virtual networks to each other or on-premises networks, but not others. For more information, see [connectivity](#). Each virtual network that you connect to another virtual network, or on-premises network, must have a unique address space. Each virtual network has one or more public or private address ranges assigned to its address space. An address range is specified in classless internet domain routing (CIDR) format, such as 10.0.0.0/16. Learn more about [address ranges](#) for virtual networks.
- Do you have any organizational administration requirements for resources in different virtual networks? If so, you might separate resources into separate virtual network to simplify [permission assignment](#) to individuals in your organization or to assign different policies to different virtual networks.
- When you deploy some Azure service resources into a virtual network, they create their own virtual network. To determine whether an Azure service creates its own virtual network, see information for each [Azure service that can be deployed into a virtual network](#).

Subnets

A virtual network can be segmented into one or more subnets up to the [limits](#). Things to consider when deciding whether to create one subnet, or multiple virtual networks in a subscription:

- Each subnet must have a unique address range, specified in CIDR format, within the address space of the virtual network. The address range cannot overlap with other subnets in the virtual network.
- If you plan to deploy some Azure service resources into a virtual network, they may require, or create, their own subnet, so there must be enough unallocated space for them to do so. To determine whether an Azure service creates its own subnet, see information for each [Azure service that can be deployed into a virtual network](#). For example, if you connect a virtual network to an on-premises network using an Azure VPN Gateway, the virtual network must have a dedicated subnet for the gateway. Learn more about [gateway subnets](#).
- Azure routes network traffic between all subnets in a virtual network, by default. You can override Azure's default routing to prevent Azure routing between subnets, or to route traffic between subnets through a network virtual appliance, for example. If you require that traffic between resources in the same virtual network flow through a network virtual appliance (NVA), deploy the resources to different subnets. Learn more in [security](#).
- You can limit access to Azure resources such as an Azure storage account or Azure SQL Database, to specific subnets with a virtual network service endpoint. Further, you can deny access to the resources from the internet. You may create multiple subnets, and enable a service endpoint for some subnets, but not others. Learn more about [service endpoints](#), and the Azure resources you can enable them for.
- You can associate zero or one network security group to each subnet in a virtual network. You can associate the same, or a different, network security group to each subnet. Each network security group contains rules, which allow or deny traffic to and from sources and destinations. Learn more about [network security groups](#).

Security

You can filter network traffic to and from resources in a virtual network using network security groups and network virtual appliances. You can control how Azure routes traffic from subnets. You can also limit who in

your organization can work with resources in virtual networks.

Traffic filtering

- You can filter network traffic between resources in a virtual network using a network security group, an NVA that filters network traffic, or both. To deploy an NVA, such as a firewall, to filter network traffic, see the [Azure Marketplace](#). When using an NVA, you also create custom routes to route traffic from subnets to the NVA. Learn more about [traffic routing](#).
- A network security group contains several default security rules that allow or deny traffic to or from resources. A network security group can be associated to a network interface, the subnet the network interface is in, or both. To simplify management of security rules, it's recommended that you associate a network security group to individual subnets, rather than individual network interfaces within the subnet, whenever possible.
- If different VMs within a subnet need different security rules applied to them, you can associate the network interface in the VM to one or more application security groups. A security rule can specify an application security group in its source, destination, or both. That rule then only applies to the network interfaces that are members of the application security group. Learn more about [network security groups](#) and [application security groups](#).
- Azure creates several default security rules within each network security group. One default rule allows all traffic to flow between all resources in a virtual network. To override this behavior, use network security groups, custom routing to route traffic to an NVA, or both. It's recommended that you familiarize yourself with all of Azure's [default security rules](#) and understand how network security group rules are applied to a resource.

You can view sample designs for implementing a perimeter network (also known as a DMZ) between Azure and the internet using an [NVA](#).

Traffic routing

Azure creates several default routes for outbound traffic from a subnet. You can override Azure's default routing by creating a route table and associating it to a subnet. Common reasons for overriding Azure's default routing are:

- Because you want traffic between subnets to flow through an NVA. To learn more about how to [configure route tables to force traffic through an NVA](#).
- Because you want to force all internet-bound traffic through an NVA, or on-premises, through an Azure VPN gateway. Forcing internet traffic on-premises for inspection and logging is often referred to as forced tunneling. Learn more about how to configure [forced tunneling](#).

If you need to implement custom routing, it's recommended that you familiarize yourself with [routing in Azure](#).

Connectivity

You can connect a virtual network to other virtual networks using virtual network peering, or to your on-premises network, using an Azure VPN gateway.

Peering

When using [virtual network peering](#), the virtual networks can be in the same, or different, supported Azure regions. The virtual networks can be in the same or different Azure subscriptions (even subscriptions belonging to different Azure Active Directory tenants). Before creating a peering, it's recommended that you familiarize yourself with all of the peering [requirements and constraints](#). Bandwidth between resources in virtual networks peered in the same region is the same as if the resources were in the same virtual network.

VPN gateway

You can use an Azure [VPN Gateway](#) to connect a virtual network to your on-premises network using a [site-to-site VPN](#), or using a dedicated connection with Azure [ExpressRoute](#).

You can combine peering and a VPN gateway to create [hub and spoke networks](#), where spoke virtual networks connect to a hub virtual network, and the hub connects to an on-premises network, for example.

Name resolution

Resources in one virtual network cannot resolve the names of resources in a peered virtual network using Azure's [built-in DNS](#). To resolve names in a peered virtual network, [deploy your own DNS server](#), or use Azure DNS [private domains](#). Resolving names between resources in a virtual network and on-premises networks also requires you to deploy your own DNS server.

Permissions

Azure utilizes [Azure role-based access control \(Azure RBAC\)](#) to resources. Permissions are assigned to a [scope](#) in the following hierarchy: management group, subscription, resource group, and individual resource. To learn more about the hierarchy, see [Organize your resources](#). To work with Azure virtual networks and all of their related capabilities such as peering, network security groups, service endpoints, and route tables, you can assign members of your organization to the built-in [Owner](#), [Contributor](#), or [Network contributor](#) roles, and then assign the role to the appropriate scope. If you want to assign specific permissions for a subset of virtual network capabilities, create a [custom role](#) and assign the specific permissions required for [virtual networks](#), [subnets](#) and [service endpoints](#), [network interfaces](#), [peering](#), [network and application security groups](#), or [route tables](#) to the role.

Policy

Azure Policy enables you to create, assign, and manage policy definitions. Policy definitions enforce different rules over your resources, so the resources stay compliant with your organizational standards and service level agreements. Azure Policy runs an evaluation of your resources, scanning for resources that are not compliant with the policy definitions you have. For example, you can define and apply a policy that allows creation of virtual networks in only a specific resource group or region. Another policy can require that every subnet has a network security group associated to it. The policies are then evaluated when creating and updating resources.

Policies are applied to the following hierarchy: management group, subscription, and resource group. Learn more about [Azure Policy](#) or deploy some virtual network [Azure Policy definitions](#).

Next steps

Learn about all tasks, settings, and options for a [virtual network](#), [subnet](#) and [service endpoint](#), [network interface](#), [peering](#), [network and application security group](#), or [route table](#).

Name resolution for resources in Azure virtual networks

2/18/2021 • 15 minutes to read • [Edit Online](#)

Depending on how you use Azure to host IaaS, PaaS, and hybrid solutions, you might need to allow the virtual machines (VMs), and other resources deployed in a virtual network to communicate with each other. Although you can enable communication by using IP addresses, it is much simpler to use names that can be easily remembered, and do not change.

When resources deployed in virtual networks need to resolve domain names to internal IP addresses, they can use one of three methods:

- [Azure DNS private zones](#)
- [Azure-provided name resolution](#)
- [Name resolution that uses your own DNS server](#) (which might forward queries to the Azure-provided DNS servers)

The type of name resolution you use depends on how your resources need to communicate with each other. The following table illustrates scenarios and corresponding name resolution solutions:

NOTE

Azure DNS private zones is the preferred solution and gives you flexibility in managing your DNS zones and records. For more information, see [Using Azure DNS for private domains](#).

NOTE

If you use Azure Provided DNS then appropriate DNS suffix will be automatically applied to your virtual machines. For all other options you must either use Fully Qualified Domain Names (FQDN) or manually apply appropriate DNS suffix to your virtual machines.

SCENARIO	SOLUTION	DNS SUFFIX
Name resolution between VMs located in the same virtual network, or Azure Cloud Services role instances in the same cloud service.	Azure DNS private zones or Azure-provided name resolution	Hostname or FQDN
Name resolution between VMs in different virtual networks or role instances in different cloud services.	Azure DNS private zones or, Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server .	FQDN only
Name resolution from an Azure App Service (Web App, Function, or Bot) using virtual network integration to role instances or VMs in the same virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server .	FQDN only

SCENARIO	SOLUTION	DNS SUFFIX
Name resolution from App Service Web Apps to VMs in the same virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server .	FQDN only
Name resolution from App Service Web Apps in one virtual network to VMs in a different virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server .	FQDN only
Resolution of on-premises computer and service names from VMs or role instances in Azure.	Customer-managed DNS servers (on-premises domain controller, local read-only domain controller, or a DNS secondary synced using zone transfers, for example). See Name resolution using your own DNS server .	FQDN only
Resolution of Azure hostnames from on-premises computers.	Forward queries to a customer-managed DNS proxy server in the corresponding virtual network, the proxy server forwards queries to Azure for resolution. See Name resolution using your own DNS server .	FQDN only
Reverse DNS for internal IPs.	Azure DNS private zones or Azure-provided name resolution or Name resolution using your own DNS server .	Not applicable
Name resolution between VMs or role instances located in different cloud services, not in a virtual network.	Not applicable. Connectivity between VMs and role instances in different cloud services is not supported outside a virtual network.	Not applicable

Azure-provided name resolution

Azure provided name resolution provides only basic authoritative DNS capabilities. If you use this option the DNS zone names and records will be automatically managed by Azure and you will not be able to control the DNS zone names or the life cycle of DNS records. If you need a fully featured DNS solution for your virtual networks you must use [Azure DNS private zones](#) or [Customer-managed DNS servers](#).

Along with resolution of public DNS names, Azure provides internal name resolution for VMs and role instances that reside within the same virtual network or cloud service. VMs and instances in a cloud service share the same DNS suffix, so the host name alone is sufficient. But in virtual networks deployed using the classic deployment model, different cloud services have different DNS suffixes. In this situation, you need the FQDN to resolve names between different cloud services. In virtual networks deployed using the Azure Resource Manager deployment model, the DNS suffix is consistent across all virtual machines within a virtual network, so the FQDN is not needed. DNS names can be assigned to both VMs and network interfaces. Although Azure-provided name resolution does not require any configuration, it is not the appropriate choice for all deployment scenarios, as detailed in the previous table.

NOTE

When using cloud services web and worker roles, you can also access the internal IP addresses of role instances using the Azure Service Management REST API. For more information, see the [Service Management REST API Reference](#). The address is based on the role name and instance number.

Features

Azure-provided name resolution includes the following features:

- Ease of use. No configuration is required.
- High availability. You don't need to create and manage clusters of your own DNS servers.
- You can use the service in conjunction with your own DNS servers, to resolve both on-premises and Azure host names.
- You can use name resolution between VMs and role instances within the same cloud service, without the need for an FQDN.
- You can use name resolution between VMs in virtual networks that use the Azure Resource Manager deployment model, without need for an FQDN. Virtual networks in the classic deployment model require an FQDN when you are resolving names in different cloud services.
- You can use host names that best describe your deployments, rather than working with auto-generated names.

Considerations

Points to consider when you are using Azure-provided name resolution:

- The Azure-created DNS suffix cannot be modified.
- DNS lookup is scoped to a virtual network. DNS names created for one virtual networks can't be resolved from other virtual networks.
- You cannot manually register your own records.
- WINS and NetBIOS are not supported. You cannot see your VMs in Windows Explorer.
- Host names must be DNS-compatible. Names must use only 0-9, a-z, and '-', and cannot start or end with a '-'.
- DNS query traffic is throttled for each VM. Throttling shouldn't impact most applications. If request throttling is observed, ensure that client-side caching is enabled. For more information, see [DNS client configuration](#).
- Only VMs in the first 180 cloud services are registered for each virtual network in a classic deployment model. This limit does not apply to virtual networks in Azure Resource Manager.
- The Azure DNS IP address is 168.63.129.16. This is a static IP address and will not change.

Reverse DNS Considerations

Reverse DNS is supported in all ARM based virtual networks. You can issue reverse DNS queries (PTR queries) to map IP addresses of virtual machines to FQDNs of virtual machines.

- All PTR queries for IP addresses of virtual machines will return FQDNs of form [vmname].internal.cloudapp.net
- Forward lookup on FQDNs of form [vmname].internal.cloudapp.net will resolve to IP address assigned to the virtual machine.
- If the virtual network is linked to an [Azure DNS private zones](#) as a registration virtual network, the reverse DNS queries will return two records. One record will be of the form [vmname].[privatednszonename] and the other will be of the form [vmname].internal.cloudapp.net
- Reverse DNS lookup is scoped to a given virtual network even if it is peered to other virtual networks. Reverse DNS queries (PTR queries) for IP addresses of virtual machines located in peered virtual networks will return NXDOMAIN.

- If you want to turn off reverse DNS function in a virtual network you can do so by creating a reverse lookup zone using [Azure DNS private zones](#) and link this zone to your virtual network. For example if the IP address space of your virtual network is 10.20.0.0/16 then you can create a empty private DNS zone 20.10.in-addr.arpa and link it to the virtual network. While linking the zone to your virtual network you should disable auto registration on the link. This zone will override the default reverse lookup zones for the virtual network and since this zone is empty you will get NXDOMAIN for your reverse DNS queries. See our [Quickstart guide](#) for details on how to create a private DNS zone and link it to a virtual network.

NOTE

If you want reverse DNS lookup to span across virtual network you can create a reverse lookup zone (in-addr.arpa) [Azure DNS private zones](#) and links it to multiple virtual networks. You'll however have to manually manage the reverse DNS records for the virtual machines.

DNS client configuration

This section covers client-side caching and client-side retries.

Client-side caching

Not every DNS query needs to be sent across the network. Client-side caching helps reduce latency and improve resilience to network blips, by resolving recurring DNS queries from a local cache. DNS records contain a time-to-live (TTL) mechanism, which allows the cache to store the record for as long as possible without impacting record freshness. Thus, client-side caching is suitable for most situations.

The default Windows DNS client has a DNS cache built-in. Some Linux distributions do not include caching by default. If you find that there isn't a local cache already, add a DNS cache to each Linux VM.

There are a number of different DNS caching packages available (such as dnsmasq). Here's how to install dnsmasq on the most common distributions:

- Ubuntu (uses resolvconf):**
 - Install the dnsmasq package with `sudo apt-get install dnsmasq`.
- SUSE (uses netconf):**
 - Install the dnsmasq package with `sudo zypper install dnsmasq`.
 - Enable the dnsmasq service with `systemctl enable dnsmasq.service`.
 - Start the dnsmasq service with `systemctl start dnsmasq.service`.
 - Edit `/etc/sysconfig/network/config`, and change `NETCONFIG_DNS_FORWARDER=""` to `dnsmasq`.
 - Update `resolv.conf` with `netconfig update`, to set the cache as the local DNS resolver.
- CentOS (uses NetworkManager):**
 - Install the dnsmasq package with `sudo yum install dnsmasq`.
 - Enable the dnsmasq service with `systemctl enable dnsmasq.service`.
 - Start the dnsmasq service with `systemctl start dnsmasq.service`.
 - Add `prepend domain-name-servers 127.0.0.1;` to `/etc/dhclient-eth0.conf`.
 - Restart the network service with `service network restart`, to set the cache as the local DNS resolver.

NOTE

The dnsmasq package is only one of many DNS caches available for Linux. Before using it, check its suitability for your particular needs, and check that no other cache is installed.

Client-side retries

DNS is primarily a UDP protocol. Because the UDP protocol doesn't guarantee message delivery, retry logic is handled in the DNS protocol itself. Each DNS client (operating system) can exhibit different retry logic, depending on the creator's preference:

- Windows operating systems retry after one second, and then again after another two seconds, four seconds, and another four seconds.
- The default Linux setup retries after five seconds. We recommend changing the retry specifications to five times, at one-second intervals.

Check the current settings on a Linux VM with `cat /etc/resolv.conf`. Look at the *options* line, for example:

```
options timeout:1 attempts:5
```

The resolv.conf file is usually auto-generated, and should not be edited. The specific steps for adding the *options* line vary by distribution:

- **Ubuntu** (uses resolvconf):
 1. Add the *options* line to `/etc/resolvconf/resolv.conf.d/tail`.
 2. Run `resolvconf -u` to update.
- **SUSE** (uses netconf):
 1. Add *timeout:1 attempts:5* to the `NETCONFIG_DNS_RESOLVER_OPTIONS=""` parameter in `/etc/sysconfig/network/config`.
 2. Run `netconfig update` to update.
- **CentOS** (uses NetworkManager):
 1. Add `echo "options timeout:1 attempts:5"` to `/etc/NetworkManager/dispatcher.d/11-dhclient`.
 2. Update with `service network restart`.

Name resolution that uses your own DNS server

This section covers VMs, role instances, and web apps.

VMs and role instances

Your name resolution needs might go beyond the features provided by Azure. For example, you might need to use Microsoft Windows Server Active Directory domains, resolve DNS names between virtual networks. To cover these scenarios, Azure provides the ability for you to use your own DNS servers.

DNS servers within a virtual network can forward DNS queries to the recursive resolvers in Azure. This enables you to resolve host names within that virtual network. For example, a domain controller (DC) running in Azure can respond to DNS queries for its domains, and forward all other queries to Azure. Forwarding queries allows VMs to see both your on-premises resources (via the DC) and Azure-provided host names (via the forwarder). Access to the recursive resolvers in Azure is provided via the virtual IP 168.63.129.16.

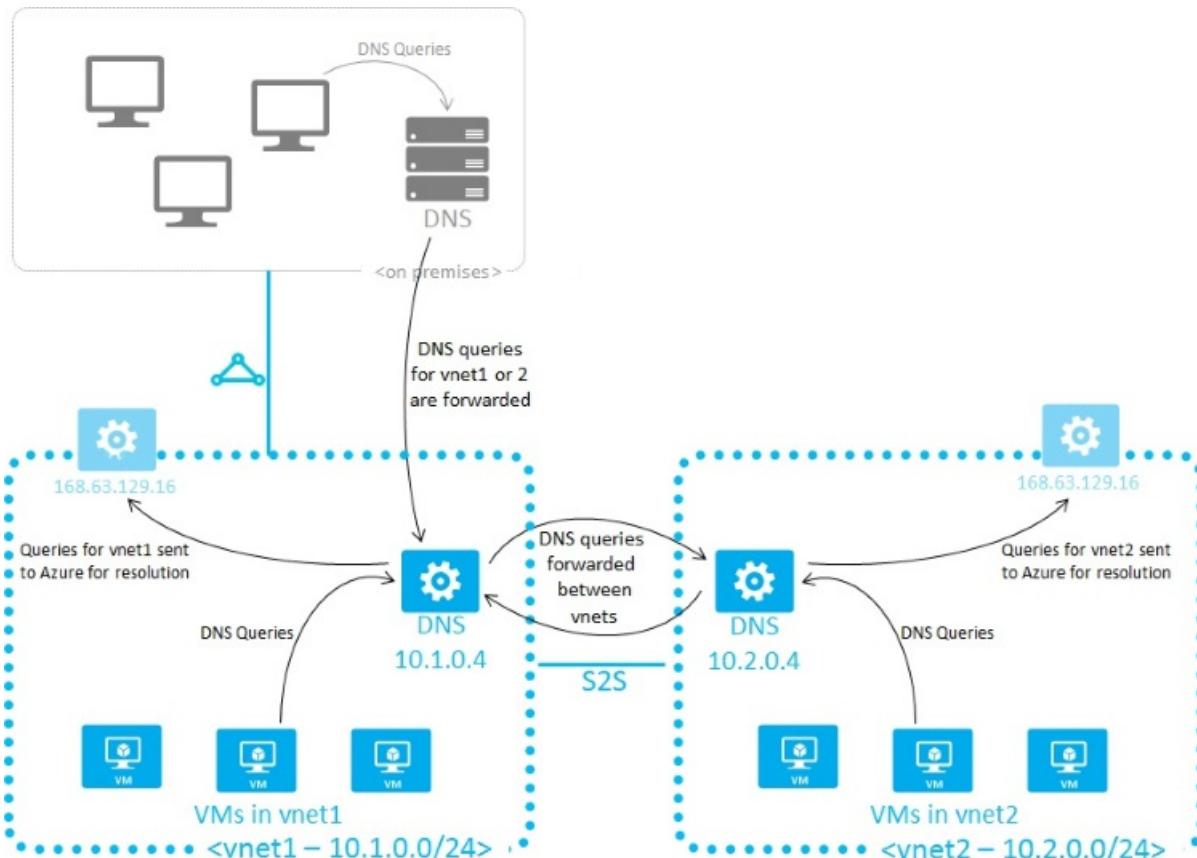
DNS forwarding also enables DNS resolution between virtual networks, and allows your on-premises machines to resolve Azure-provided host names. In order to resolve a VM's host name, the DNS server VM must reside in the same virtual network, and be configured to forward host name queries to Azure. Because the DNS suffix is different in each virtual network, you can use conditional forwarding rules to send DNS queries to the correct virtual network for resolution. The following image shows two virtual networks and an on-premises network doing DNS resolution between virtual networks, by using this method. An example DNS forwarder is available in the [Azure Quickstart Templates gallery](#) and [GitHub](#).

NOTE

A role instance can perform name resolution of VMs within the same virtual network. It does so by using the FQDN, which consists of the VM's host name and **internal.cloudapp.net** DNS suffix. However, in this case, name resolution is only successful if the role instance has the VM name defined in the [Role Schema \(.cscfg file\)](#).

```
<Role name=<role-name> vmName=<vm-name>>
```

Role instances that need to perform name resolution of VMs in another virtual network (FQDN by using the **internal.cloudapp.net** suffix) have to do so by using the method described in this section (custom DNS servers forwarding between the two virtual networks).



When you are using Azure-provided name resolution, Azure Dynamic Host Configuration Protocol (DHCP) provides an internal DNS suffix (**.internal.cloudapp.net**) to each VM. This suffix enables host name resolution because the host name records are in the **internal.cloudapp.net** zone. When you are using your own name resolution solution, this suffix is not supplied to VMs because it interferes with other DNS architectures (like domain-joined scenarios). Instead, Azure provides a non-functioning placeholder (*reddog.microsoft.com*).

If necessary, you can determine the internal DNS suffix by using PowerShell or the API:

- For virtual networks in Azure Resource Manager deployment models, the suffix is available via the [network interface REST API](#), the [Get-AzNetworkInterface](#) PowerShell cmdlet, and the [az network nic show](#) Azure CLI command.
- In classic deployment models, the suffix is available via the [Get Deployment API](#) call or the [Get-AzureVM - Debug](#) cmdlet.

If forwarding queries to Azure doesn't suit your needs, you should provide your own DNS solution. Your DNS solution needs to:

- Provide appropriate host name resolution, via [DDNS](#), for example. If you are using DDNS, you might need to disable DNS record scavenging. Azure DHCP leases are long, and scavenging might remove DNS records prematurely.

- Provide appropriate recursive resolution to allow resolution of external domain names.
- Be accessible (TCP and UDP on port 53) from the clients it serves, and be able to access the internet.
- Be secured against access from the internet, to mitigate threats posed by external agents.

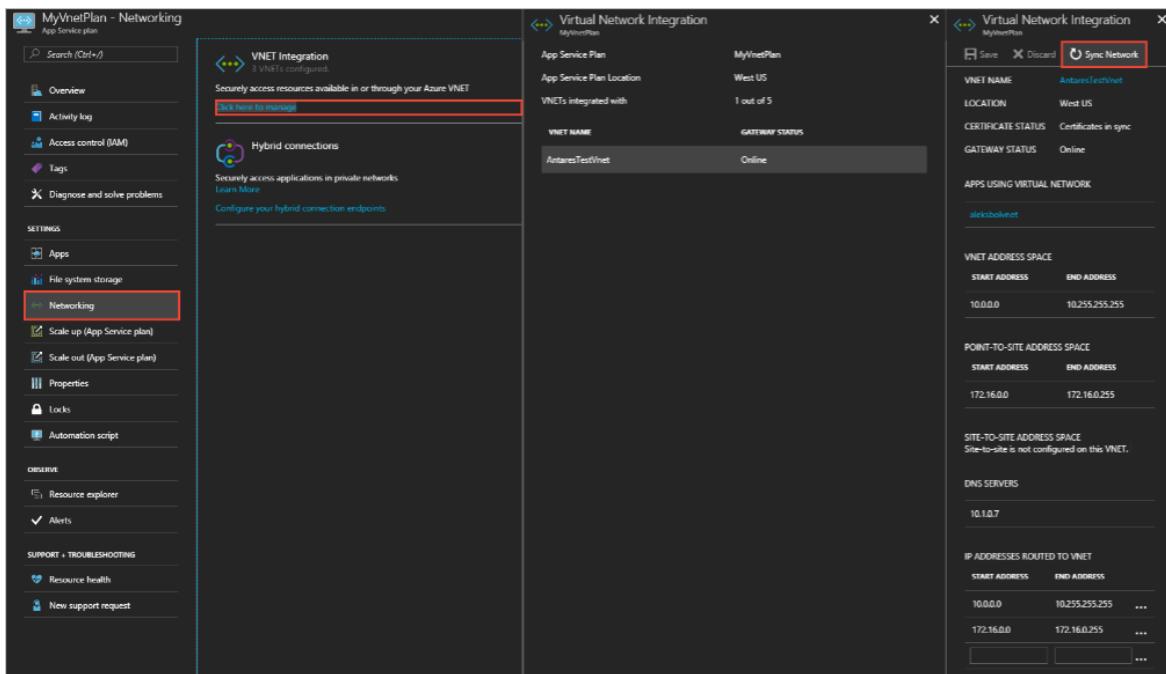
NOTE

For best performance, when you are using Azure VMs as DNS servers, IPv6 should be disabled.

Web apps

Suppose you need to perform name resolution from your web app built by using App Service, linked to a virtual network, to VMs in the same virtual network. In addition to setting up a custom DNS server that has a DNS forwarder that forwards queries to Azure (virtual IP 168.63.129.16), perform the following steps:

1. Enable virtual network integration for your web app, if not done already, as described in [Integrate your app with a virtual network](#).
2. In the Azure portal, for the App Service plan hosting the web app, select **Sync Network** under **Networking, Virtual Network Integration**.



If you need to perform name resolution from your web app built by using App Service, linked to a virtual network, to VMs in a different virtual network, you have to use custom DNS servers on both virtual networks, as follows:

- Set up a DNS server in your target virtual network, on a VM that can also forward queries to the recursive resolver in Azure (virtual IP 168.63.129.16). An example DNS forwarder is available in the [Azure Quickstart Templates gallery](#) and [GitHub](#).
- Set up a DNS forwarder in the source virtual network on a VM. Configure this DNS forwarder to forward queries to the DNS server in your target virtual network.
- Configure your source DNS server in your source virtual network's settings.
- Enable virtual network integration for your web app to link to the source virtual network, following the instructions in [Integrate your app with a virtual network](#).
- In the Azure portal, for the App Service plan hosting the web app, select **Sync Network** under **Networking, Virtual Network Integration**.

Specify DNS servers

When you are using your own DNS servers, Azure provides the ability to specify multiple DNS servers per virtual network. You can also specify multiple DNS servers per network interface (for Azure Resource Manager), or per cloud service (for the classic deployment model). DNS servers specified for a network interface or cloud service get precedence over DNS servers specified for the virtual network.

NOTE

Network connection properties, such as DNS server IPs, should not be edited directly within VMs. This is because they might get erased during service heal when the virtual network adaptor gets replaced. This applies to both Windows and Linux VMs.

When you are using the Azure Resource Manager deployment model, you can specify DNS servers for a virtual network and a network interface. For details, see [Manage a virtual network](#) and [Manage a network interface](#).

NOTE

If you opt for custom DNS server for your virtual network, you must specify at least one DNS server IP address; otherwise, virtual network will ignore the configuration and use Azure-provided DNS instead.

When you are using the classic deployment model, you can specify DNS servers for the virtual network in the Azure portal or the [Network Configuration file](#). For cloud services, you can specify DNS servers via the [Service Configuration file](#) or by using PowerShell, with [New-AzureVM](#).

NOTE

If you change the DNS settings for a virtual network or virtual machine that is already deployed, for the new DNS settings to take effect, you must perform a DHCP lease renewal on all affected VMs in the virtual network. For VMs running the Windows OS, you can do this by typing `ipconfig /renew` directly in the VM. The steps vary depending on the OS. See the relevant documentation for your OS type.

Next steps

Azure Resource Manager deployment model:

- [Manage a virtual network](#)
- [Manage a network interface](#)

Classic deployment model:

- [Azure Service Configuration Schema](#)
- [Virtual Network Configuration Schema](#)
- [Configure a Virtual Network by using a network configuration file](#)

Use dynamic DNS to register hostnames in your own DNS server

2/18/2021 • 3 minutes to read • [Edit Online](#)

Azure provides name resolution for virtual machines (VM) and role instances. When your name resolution needs exceed the capabilities provided by Azure's default DNS, you can provide your own DNS servers. Using your own DNS servers gives you the ability to tailor your DNS solution to suit your own specific needs. For example, you may need to access on-premises resources via your Active Directory domain controller.

When your custom DNS servers are hosted as Azure VMs, you can forward hostname queries for the same virtual network to Azure to resolve hostnames. If you do not wish to use this option, you can register your VM hostnames in your DNS server using dynamic DNS (DDNS). Azure doesn't have the credentials to directly create records in your DNS servers, so alternative arrangements are often needed. Some common scenarios, with alternatives follow:

Windows clients

Non-domain-joined Windows clients attempt unsecured DDNS updates when they boot, or when their IP address changes. The DNS name is the hostname plus the primary DNS suffix. Azure leaves the primary DNS suffix blank, but you can set the suffix in the VM, via the [user interface](#) or [PowerShell](#).

Domain-joined Windows clients register their IP addresses with the domain controller by using secure DDNS. The domain-join process sets the primary DNS suffix on the client and creates and maintains the trust relationship.

Linux clients

Linux clients generally don't register themselves with the DNS server on startup, they assume the DHCP server does it. Azure's DHCP servers do not have the credentials to register records in your DNS server. You can use a tool called `nsupdate`, which is included in the Bind package, to send DDNS updates. Because the DDNS protocol is standardized, you can use `nsupdate` even when you're not using Bind on the DNS server.

You can use the hooks that are provided by the DHCP client to create and maintain the hostname entry in the DNS server. During the DHCP cycle, the client executes the scripts in `/etc/dhcp/dhclient-exit-hooks.d/`. You can use the hooks to register the new IP address using `nsupdate`. For example:

```

#!/bin/sh
requireddomain=mydomain.local

# only execute on the primary nic
if [ "$interface" != "eth0" ]
then
    return
fi

# When you have a new IP, perform nsupdate
if [ "$reason" = BOUND ] || [ "$reason" = RENEW ] ||
[ "$reason" = REBIND ] || [ "$reason" = REBOOT ]
then
    host=`hostname`
    nsupdatecmds=/var/tmp/nsupdatecmds
    echo "update delete $host.$requireddomain a" > $nsupdatecmds
    echo "update add $host.$requireddomain 3600 a $new_ip_address" >> $nsupdatecmds
    echo "send" >> $nsupdatecmds

    nsupdate $nsupdatecmds
fi

```

You can also use the `nsupdate` command to perform secure DDNS updates. For example, when you're using a Bind DNS server, a public-private key pair is generated (<http://linux.yyz.us/nsupdate/>). The DNS server is configured (<http://linux.yyz.us/dns/ddns-server.html>) with the public part of the key, so that it can verify the signature on the request. To provide the key-pair to `nsupdate`, use the `-k` option, for the DDNS update request to be signed.

When you're using a Windows DNS server, you can use Kerberos authentication with the `-g` parameter in `nsupdate`, but it's not available in the Windows version of `nsupdate`. To use Kerberos, use `kinit` to load the credentials. For example, you can load credentials from a [keytab file](#), then `nsupdate -g` picks up the credentials, from the cache.

If needed, you can add a DNS search suffix to your VMs. The DNS suffix is specified in the `/etc/resolv.conf` file. Most Linux distros automatically manage the content of this file, so usually you can't edit it. However, you can override the suffix by using the DHCP client's `supersede` command. To override the suffix, add the following line to the `/etc/dhcp/dhclient.conf` file:

```
supersede domain-name <required-dns-suffix>;
```

Optimize network throughput for Azure virtual machines

2/18/2021 • 3 minutes to read • [Edit Online](#)

Azure virtual machines (VM) have default network settings that can be further optimized for network throughput. This article describes how to optimize network throughput for Microsoft Azure Windows and Linux VMs, including major distributions such as Ubuntu, CentOS, and Red Hat.

Windows VM

If your Windows VM supports [Accelerated Networking](#), enabling that feature would be the optimal configuration for throughput. For all other Windows VMs, using Receive Side Scaling (RSS) can reach higher maximal throughput than a VM without RSS. RSS may be disabled by default in a Windows VM. To determine whether RSS is enabled, and enable it if it's currently disabled, complete the following steps:

1. See if RSS is enabled for a network adapter with the `Get-NetAdapterRss` PowerShell command. In the following example output returned from the `Get-NetAdapterRss`, RSS is not enabled.

```
Name          : Ethernet
InterfaceDescription : Microsoft Hyper-V Network Adapter
Enabled       : False
```

2. To enable RSS, enter the following command:

```
Get-NetAdapter | % {Enable-NetAdapterRss -Name $_.Name}
```

The previous command does not have an output. The command changed NIC settings, causing temporary connectivity loss for about one minute. A Reconnecting dialog box appears during the connectivity loss. Connectivity is typically restored after the third attempt.

3. Confirm that RSS is enabled in the VM by entering the `Get-NetAdapterRss` command again. If successful, the following example output is returned:

```
Name      : Ethernet
InterfaceDescription : Microsoft Hyper-V Network Adapter
Enabled    : True
```

Linux VM

RSS is always enabled by default in an Azure Linux VM. Linux kernels released since October 2017 include new network optimizations options that enable a Linux VM to achieve higher network throughput.

Ubuntu for new deployments

The Ubuntu Azure kernel is the most optimized for network performance on Azure. To get the latest optimizations, first install the latest supported version of 18.04-LTS, as follows:

```
"Publisher": "Canonical",
"Offer": "UbuntuServer",
"Sku": "18.04-LTS",
"Version": "latest"
```

After the creation is complete, enter the following commands to get the latest updates. These steps also work for VMs currently running the Ubuntu Azure kernel.

```
#run as root or preface with sudo
apt-get -y update
apt-get -y upgrade
apt-get -y dist-upgrade
```

The following optional command set may be helpful for existing Ubuntu deployments that already have the Azure kernel but that have failed to further updates with errors.

```
#optional steps may be helpful in existing deployments with the Azure kernel
#run as root or preface with sudo
apt-get -f install
apt-get --fix-missing install
apt-get clean
apt-get -y update
apt-get -y upgrade
apt-get -y dist-upgrade
```

Ubuntu Azure kernel upgrade for existing VMs

Significant throughput performance can be achieved by upgrading to the Azure Linux kernel. To verify whether you have this kernel, check your kernel version. It should be the same or later than the example.

```
#Azure kernel name ends with "-azure"
uname -r

#sample output on Azure kernel:
#4.13.0-1007-azure
```

If your VM does not have the Azure kernel, the version number usually begins with "4.4." If the VM does not have the Azure kernel, run the following commands as root:

```
#run as root or preface with sudo
apt-get update
apt-get upgrade -y
apt-get dist-upgrade -y
apt-get install "linux-azure"
reboot
```

CentOS

In order to get the latest optimizations, it is best to create a VM with the latest supported version by specifying the following parameters:

```
"Publisher": "OpenLogic",
"Offer": "CentOS",
"Sku": "7.7",
"Version": "latest"
```

New and existing VMs can benefit from installing the latest Linux Integration Services (LIS). The throughput

optimization is in LIS, starting from 4.2.2-2, although later versions contain further improvements. Enter the following commands to install the latest LIS:

```
sudo yum update  
sudo reboot  
sudo yum install microsoft-hyper-v
```

Red Hat

In order to get the optimizations, it is best to create a VM with the latest supported version by specifying the following parameters:

```
"Publisher": "RedHat"  
"Offer": "RHEL"  
"Sku": "7-RAW"  
"Version": "latest"
```

New and existing VMs can benefit from installing the latest Linux Integration Services (LIS). The throughput optimization is in LIS, starting from 4.2. Enter the following commands to download and install LIS:

```
wget https://aka.ms/lis  
tar xvf lis  
cd LISISO  
sudo ./install.sh #or upgrade.sh if prior LIS was previously installed
```

Learn more about Linux Integration Services Version 4.2 for Hyper-V by viewing the [download page](#).

Next steps

- Deploy VMs close to each other for low latency with [Proximity Placement Group](#)
- See the optimized result with [Bandwidth/Throughput testing Azure VM](#) for your scenario.
- Read about how [bandwidth is allocated to virtual machines](#)
- Learn more with [Azure Virtual Network frequently asked questions \(FAQ\)](#)

Viewing and modifying hostnames

2/18/2021 • 2 minutes to read • [Edit Online](#)

To allow your role instances to be referenced by host name, you must set the value for the host name in the service configuration file for each role. You do that by adding the desired host name to the **vmName** attribute of the **Role** element. The value of the **vmName** attribute is used as a base for the host name of each role instance. For example, if **vmName** is *webrole* and there are three instances of that role, the host names of the instances will be *webrole0*, *webrole1*, and *webrole2*. You do not need to specify a host name for virtual machines in the configuration file, because the host name for a virtual machine is populated based on the virtual machine name. For more information about configuring a Microsoft Azure service, see [Azure Service Configuration Schema \(.cscfg File\)](#)

Viewing hostnames

You can view the host names of virtual machines and role instances in a cloud service by using any of the tools below.

Service configuration file

You can download the service configuration file for a deployed service from the **Configure** blade of the service in the Azure portal. You can then look for the **vmName** attribute for the **Role name** element to see the host name. Keep in mind that this host name is used as a base for the host name of each role instance. For example, if **vmName** is *webrole* and there are three instances of that role, the host names of the instances will be *webrole0*, *webrole1*, and *webrole2*.

Remote Desktop

After you enable Remote Desktop (Windows), Windows PowerShell remoting (Windows), or SSH (Linux and Windows) connections to your virtual machines or role instances, you can view the host name from an active Remote Desktop connection in various ways:

- Type hostname at the command prompt or SSH terminal.
- Type ipconfig /all at the command prompt (Windows only).
- View the computer name in the system settings (Windows only).

Azure Service Management REST API

From a REST client, follow these instructions:

1. Ensure that you have a client certificate to connect to the Azure portal. To obtain a client certificate, follow the steps presented in [How to: Download and Import Publish Settings and Subscription Information](#).
2. Set a header entry named x-ms-version with a value of 2013-11-01.
3. Send a request in the following format: `https://management.core.windows.net/<subscription-id>/services/hostedservices/<service-name>?embed-detail=true`
4. Look for the **HostName** element for each **RoleInstance** element.

WARNING

You can also view the internal domain suffix for your cloud service from the REST call response by checking the **InternalDnsSuffix** element, or by running ipconfig /all from a command prompt in a Remote Desktop session (Windows), or by running cat /etc/resolv.conf from an SSH terminal (Linux).

Modifying a hostname

You can modify the host name for any virtual machine or role instance by uploading a modified service configuration file, or by renaming the computer from a Remote Desktop session.

Next steps

[Name Resolution \(DNS\)](#)

[Azure Service Configuration Schema \(.cscfg\)](#)

[Azure Virtual Network Configuration Schema](#)

[Specify DNS settings using network configuration files](#)

Resource logging for a network security group

2/18/2021 • 7 minutes to read • [Edit Online](#)

A network security group (NSG) includes rules that allow or deny traffic to a virtual network subnet, network interface, or both.

When you enable logging for an NSG, you can gather the following types of resource log information:

- **Event:** Entries are logged for which NSG rules are applied to VMs, based on MAC address.
- **Rule counter:** Contains entries for how many times each NSG rule is applied to deny or allow traffic. The status for these rules is collected every 300 seconds.

Resource logs are only available for NSGs deployed through the Azure Resource Manager deployment model. You cannot enable resource logging for NSGs deployed through the classic deployment model. For a better understanding of the two models, see [Understanding Azure deployment models](#).

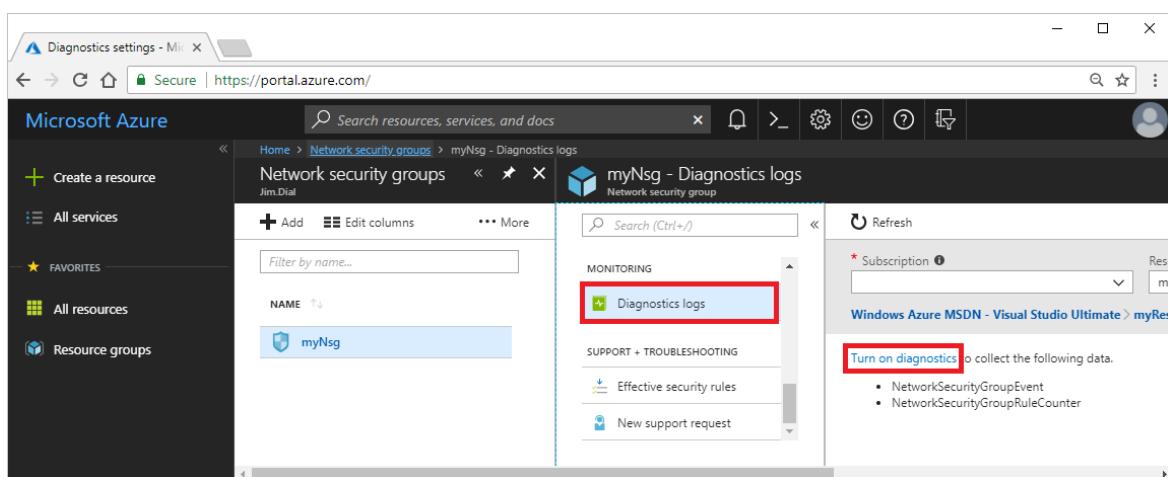
Resource logging is enabled separately for *each* NSG you want to collect diagnostic data for. If you're interested in activity (operational) logs instead, see [Azure activity logging](#).

Enable logging

You can use the [Azure Portal](#), [PowerShell](#), or the [Azure CLI](#) to enable resource logging.

Azure Portal

1. Sign in to the [portal](#).
2. Select **All services**, then type *network security groups*. When **Network security groups** appear in the search results, select it.
3. Select the NSG you want to enable logging for.
4. Under **MONITORING**, select **Diagnostics logs**, and then select **Turn on diagnostics**, as shown in the following picture:



5. Under **Diagnostics settings**, enter, or select the following information, and then select **Save**:

SETTING	VALUE

SETTING	VALUE
Name	A name of your choosing. For example: <i>myNsgDiagnostics</i>
Archive to a storage account, Stream to an event hub, and Send to Log Analytics	You can select as many destinations as you choose. To learn more about each, see Log destinations .
LOG	Select either, or both log categories. To learn more about the data logged for each category, see Log categories .

6. View and analyze logs. For more information, see [View and analyze logs](#).

PowerShell

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

You can run the commands that follow in the [Azure Cloud Shell](#), or by running PowerShell from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use with your account. If you run PowerShell from your computer, you need the Azure PowerShell module, version 1.0.0 or later. Run `Get-Module -ListAvailable Az` on your computer, to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to sign in to Azure with an account that has the [necessary permissions](#).

To enable resource logging, you need the Id of an existing NSG. If you don't have an existing NSG, you can create one with [New-AzNetworkSecurityGroup](#).

Retrieve the network security group that you want to enable resource logging for with [Get-AzNetworkSecurityGroup](#). For example, to retrieve an NSG named *myNsg* that exists in a resource group named *myResourceGroup*, enter the following command:

```
$Nsg=Get-AzNetworkSecurityGroup ` 
-Name myNsg ` 
-ResourceGroupName myResourceGroup
```

You can write resource logs to three destination types. For more information, see [Log destinations](#). In this article, logs are sent to the *Log Analytics* destination, as an example. Retrieve an existing Log Analytics workspace with [Get-AzOperationalInsightsWorkspace](#). For example, to retrieve an existing workspace named *myWorkspace* in a resource group named *myWorkspaces*, enter the following command:

```
$oms=Get-AzOperationalInsightsWorkspace ` 
-ResourceGroupName myWorkspaces ` 
-Name myWorkspace
```

If you don't have an existing workspace, you can create one with [New-AzOperationalInsightsWorkspace](#).

There are two categories of logging you can enable logs for. For more information, see [Log categories](#). Enable resource logging for the NSG with [Set-AzDiagnosticSetting](#). The following example logs both event and counter category data to the workspace for an NSG, using the IDs for the NSG and workspace you retrieved previously:

```
Set-AzDiagnosticSetting ` 
-ResourceId $Nsg.Id ` 
-WorkspaceId $Oms.ResourceId ` 
-Enabled $true
```

If you only want to log data for one category or the other, rather than both, add the `-Categories` option to the previous command, followed by *NetworkSecurityGroupEvent* or *NetworkSecurityGroupRuleCounter*. If you want to log to a different [destination](#) than a Log Analytics workspace, use the appropriate parameters for an Azure [Storage account](#) or [Event Hub](#).

View and analyze logs. For more information, see [View and analyze logs](#).

Azure CLI

You can run the commands that follow in the [Azure Cloud Shell](#), or by running the Azure CLI from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use with your account. If you run the CLI from your computer, you need version 2.0.38 or later. Run `az --version` on your computer, to find the installed version. If you need to upgrade, see [Install Azure CLI](#). If you are running the CLI locally, you also need to run `az login` to sign in to Azure with an account that has the [necessary permissions](#).

To enable resource logging, you need the Id of an existing NSG. If you don't have an existing NSG, you can create one with [az network nsg create](#).

Retrieve the network security group that you want to enable resource logging for with [az network nsg show](#). For example, to retrieve an NSG named *myNsg* that exists in a resource group named *myResourceGroup*, enter the following command:

```
nsgId=$(az network nsg show \
--name myNsg \
--resource-group myResourceGroup \
--query id \
--output tsv)
```

You can write resource logs to three destination types. For more information, see [Log destinations](#). In this article, logs are sent to the *Log Analytics* destination, as an example. For more information, see [Log categories](#).

Enable resource logging for the NSG with [az monitor diagnostic-settings create](#). The following example logs both event and counter category data to an existing workspace named *myWorkspace*, which exists in a resource group named *myWorkspaces*, and the ID of the NSG you retrieved previously:

```
az monitor diagnostic-settings create \
--name myNsgDiagnostics \
--resource $nsgId \
--logs '[ { "category": "NetworkSecurityGroupEvent", "enabled": true, "retentionPolicy": { "days": 30, "enabled": true } }, { "category": "NetworkSecurityGroupRuleCounter", "enabled": true, "retentionPolicy": { "days": 30, "enabled": true } } ]' \
--workspace myWorkspace \
--resource-group myWorkspaces
```

If you don't have an existing workspace, you can create one using the [Azure portal](#) or [PowerShell](#). There are two categories of logging you can enable logs for.

If you only want to log data for one category or the other, remove the category you don't want to log data for in the previous command. If you want to log to a different [destination](#) than a Log Analytics workspace, use the appropriate parameters for an Azure [Storage account](#) or [Event Hub](#).

View and analyze logs. For more information, see [View and analyze logs](#).

Log destinations

Diagnostics data can be:

- [Written to an Azure Storage account](#), for auditing or manual inspection. You can specify the retention time (in days) using resource diagnostic settings.
- [Streamed to an Event hub](#) for ingestion by a third-party service, or custom analytics solution, such as PowerBI.
- [Written to Azure Monitor logs](#).

Log categories

JSON-formatted data is written for the following log categories:

Event

The event log contains information about which NSG rules are applied to VMs, based on MAC address. The following data is logged for each event. In the following example, the data is logged for a virtual machine with the IP address 192.168.1.4 and a MAC address of 00-0D-3A-92-6A-7C:

```
{  
  "time": "[DATE-TIME]",  
  "systemId": "[ID]",  
  "category": "NetworkSecurityGroupEvent",  
  "resourceId": "/SUBSCRIPTIONS/[SUBSCRIPTION-ID]/RESOURCEGROUPS/[RESOURCE-GROUP-  
NAME]/PROVIDERS/MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/[NSG-NAME]",  
  "operationName": "NetworkSecurityGroupEvents",  
  "properties": {  
    "vnetResourceGuid": "[ID]",  
    "subnetPrefix": "192.168.1.0/24",  
    "macAddress": "00-0D-3A-92-6A-7C",  
    "primaryIPv4Address": "192.168.1.4",  
    "ruleName": "[SECURITY-RULE-NAME]",  
    "direction": "[DIRECTION-SPECIFIED-IN-RULE]",  
    "priority": "[PRIORITY-SPECIFIED-IN-RULE]",  
    "type": "[ALLOW-OR-DENY-AS-SPECIFIED-IN-RULE]",  
    "conditions": {  
      "protocols": "[PROTOCOLS-SPECIFIED-IN-RULE]",  
      "destinationPortRange": "[PORT-RANGE-SPECIFIED-IN-RULE]",  
      "sourcePortRange": "[PORT-RANGE-SPECIFIED-IN-RULE]",  
      "sourceIP": "[SOURCE-IP-OR-RANGE-SPECIFIED-IN-RULE]",  
      "destinationIP": "[DESTINATION-IP-OR-RANGE-SPECIFIED-IN-RULE]"  
    }  
  }  
}
```

Rule counter

The rule counter log contains information about each rule applied to resources. The following example data is logged each time a rule is applied. In the following example, the data is logged for a virtual machine with the IP address 192.168.1.4 and a MAC address of 00-0D-3A-92-6A-7C:

```
{
  "time": "[DATE-TIME]",
  "systemId": "[ID]",
  "category": "NetworkSecurityGroupRuleCounter",
  "resourceId": "/SUBSCRIPTIONS/[SUBSCRIPTION ID]/RESOURCEGROUPS/[RESOURCE-GROUP-NAME]/PROVIDERS/MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/[NSG-NAME]",
  "operationName": "NetworkSecurityGroupCounters",
  "properties": {
    "vnetResourceGuid": "[ID]",
    "subnetPrefix": "192.168.1.0/24",
    "macAddress": "00-0D-3A-92-6A-7C",
    "primaryIPv4Address": "192.168.1.4",
    "ruleName": "[SECURITY-RULE-NAME]",
    "direction": "[DIRECTION-SPECIFIED-IN-RULE]",
    "type": "[ALLOW-OR-DENY-AS-SPECIFIED-IN-RULE]",
    "matchedConnections": 125
  }
}
```

NOTE

The source IP address for the communication is not logged. You can enable [NSG flow logging](#) for an NSG however, which logs all of the rule counter information, as well as the source IP address that initiated the communication. NSG flow log data is written to an Azure Storage account. You can analyze the data with the [traffic analytics](#) capability of Azure Network Watcher.

View and analyze logs

To learn how to view resource log data, see [Azure platform logs overview](#). If you send diagnostics data to:

- **Azure Monitor logs:** You can use the [network security group analytics](#) solution for enhanced insights. The solution provides visualizations for NSG rules that allow or deny traffic, per MAC address, of the network interface in a virtual machine.
- **Azure Storage account:** Data is written to a PT1H.json file. You can find the:
 - Event log in the following path:
`insights-logs-networksecuritygroupevent/resourceId=/SUBSCRIPTIONS/[ID]/RESOURCEGROUPS/[RESOURCE-GROUP-NAME-FOR-NSG]/PROVIDERS/MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/[NSG NAME]/y=[YEAR]/m=[MONTH/d=[DAY]/h=[HOUR]/m=[MINUTE]`
 - Rule counter log in the following path:
`insights-logs-networksecuritygrouprulecounter/resourceId=/SUBSCRIPTIONS/[ID]/RESOURCEGROUPS/[RESOURCE-GROUP-NAME-FOR-NSG]/PROVIDERS/MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/[NSG NAME]/y=[YEAR]/m=[MONTH/d=[DAY]/h=[HOUR]/m=[MINUTE]`

Next steps

- Learn more about [Activity logging](#). Activity logging is enabled by default for NSGs created through either Azure deployment model. To determine which operations were completed on NSGs in the activity log, look for entries that contain the following resource types:
 - Microsoft.ClassicNetwork/networkSecurityGroups
 - Microsoft.ClassicNetwork/networkSecurityGroups/securityRules
 - Microsoft.Network/networkSecurityGroups
 - Microsoft.Network/networkSecurityGroups/securityRules
- To learn how to log diagnostic information, to include the source IP address for each flow, see [NSG flow logging](#).

Tutorial: Route network traffic with a route table using the Azure portal

2/18/2021 • 9 minutes to read • [Edit Online](#)

Azure routes traffic between all subnets within a virtual network, by default. You can create your own routes to override Azure's default routing. Custom routes are helpful when, for example, you want to route traffic between subnets through a network virtual appliance (NVA). In this tutorial, you learn how to:

- Create an NVA that routes traffic
- Create a route table
- Create a route
- Associate a route table to a subnet
- Deploy virtual machines (VM) into different subnets
- Route traffic from one subnet to another through an NVA

This tutorial uses the [Azure portal](#). You can also use [Azure CLI](#) or [Azure PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Create an NVA

Network virtual appliances (NVAs) are virtual machines that help with network functions, such as routing and firewall optimization. This tutorial assumes you're using **Windows Server 2016 Datacenter**. You can select a different operating system if you want.

1. On the [Azure portal](#) menu or from the **Home** page, select **Create a resource**.
2. Choose **Security > Windows Server 2016 Datacenter**.

The screenshot shows the Microsoft Azure Marketplace interface. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+/)". Below the search bar, the word "New" is displayed. On the left, there's a sidebar with categories like "Azure Marketplace" and "See all". The main area is titled "Popular" and lists various Azure services with their icons and names. A red box highlights the "Windows Server 2016 Datacenter" entry.

3. In the **Create a virtual machine** page, under **Basics**, enter or select this information:

SECTION	SETTING	ACTION
Project details	Subscription	Choose your subscription.
	Resource group	Select Create new , enter <i>myResourceGroup</i> , and select OK .
Instance details	Virtual machine name	Enter <i>myVmName</i> .
	Region	Choose (US) East US .
	Availability options	Choose No infrastructure redundancy required .
	Image	Choose Windows Server 2016 Datacenter .
	Size	Keep the default, Standard DS1 v2 .
Administrator account	Username	Enter a user name of your choosing.

SECTION	SETTING	ACTION
	Password	Enter a password of your choosing, which must be at least 12 characters long and meet the defined complexity requirements .
	Confirm Password	Enter the password again.
Inbound port rules	Public inbound ports	Pick None .
Save money	Already have a Windows Server license?	Pick No .

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Basics' tab is selected. The 'Project details' and 'Instance details' sections are highlighted with a red box. The 'Inbound port rules' section is also highlighted with a red box.

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Contoso
Resource group *: myResourceGroup (selected)

Instance details
Virtual machine name *: myVmNva
Region *: (US) East US
Availability options: No infrastructure redundancy required
Image *: Windows Server 2016 Datacenter

Azure Spot instance: No

Size *: Standard DS1 v2
1 vcpu, 3.5 GiB memory (\$???.??/month)
[Change size](#)

Administrator account
Username *: adminuser
Password *:
Confirm password *:

Inbound port rules
Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *: None (selected)

Select inbound ports: Select one or more ports
All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Review + create < Previous **Next : Disks >**

Then select **Next : Disks >**.

4. Under Disks, select the settings that are right for your needs, and then select **Next : Networking >**.

5. Under **Networking**:

a. For **Virtual network**, select **Create new**.

b. In the **Create virtual network** dialog box, under **Name**, enter *myVirtualNetwork*.

c. In **Address space**, replace the existing address range with *10.0.0.0/16*.

d. In **Subnets**, select the **Delete** icon to delete the existing subnet, and then enter the following combinations of **Subnet name** and **Address range**. Once a valid name and range is entered, a new empty row appears below it.

SUBNET NAME	ADDRESS RANGE
<i>Public</i>	<i>10.0.0.0/24</i>
<i>Private</i>	<i>10.0.1.0/24</i>
<i>DMZ</i>	<i>10.0.2.0/24</i>

e. Select **OK** to exit the dialog box.

f. In **Subnet**, choose **DMZ (10.0.2.0/24)**.

g. In **Public IP**, choose **None**, since this VM won't connect over the internet.

h. Select **Next : Management >**.

6. Under **Management**:

a. In **Diagnostics storage account**, select **Create New**.

b. In the **Create storage account** dialog box, enter or select this information:

SETTING	VALUE
Name	<i>mynvastorageaccount</i>
Account kind	Storage (general purpose v1)
Performance	Standard
Replication	Locally-redundant storage (LRS)

c. Select **OK** to exit the dialog box.

d. Select **Review + create**. You're taken to the **Review + create** page, and Azure validates your configuration.

7. When you see the **Validation passed** message, select **Create**.

The VM takes a few minutes to create. Wait until Azure finishes creating the VM. The **Your deployment is underway** page shows you deployment details.

8. When your VM is ready, select **Go to resource**.

Create a route table

1. On the [Azure portal](#) menu or from the **Home** page, select **Create a resource**.
2. In the search box, enter *Route table*. When **Route table** appears in the search results, select it.
3. In the **Route table** page, select **Create**.
4. In **Create route table**, enter or select this information:

SETTING	VALUE
Name	<i>myRouteTablePublic</i>
Subscription	Your subscription
Resource group	myResourceGroup
Location	(US) East US
Virtual network gateway route propagation	Enabled

The screenshot shows the 'Create route table' dialog box. At the top, there's a breadcrumb navigation: Home > New > Route table > Create route table. The main title is 'Create route table'. Below it, a note says 'You can add routes to this table after it's created.' The form fields are as follows:

- Name ***: myRouteTablePublic
- Subscription ***: Contoso
- Resource group ***: myResourceGroup
- Location ***: (US) East US
- Virtual network gateway route propagation**: Enabled (radio button selected)

At the bottom of the dialog are two buttons: 'Create' (highlighted in blue) and 'Automation options'.

5. Select **Create**.

Create a route

1. Go to the [Azure portal](#) to manage your route table. Search for and select **Route tables**.
2. Pick the name of your route table (**myRouteTablePublic**).
3. Choose **Routes > Add**.

4. In Add route, enter or select this information:

SETTING	VALUE
Route name	<i>ToPrivateSubnet</i>
Address prefix	<i>10.0.1.0/24</i> (the address range of the <i>Private</i> subnet created earlier)
Next hop type	Virtual appliance
Next hop address	<i>10.0.2.4</i> (an address within the address range of the <i>DMZ</i> subnet)

5. Select OK.

Associate a route table to a subnet

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Virtual networks**.
2. Pick the name of your virtual network (**myVirtualNetwork**).

3. In the virtual network's menu bar, choose **Subnets**.
4. In the virtual network's subnet list, choose **Public**.
5. In **Route table**, choose the route table you created (**myRouteTablePublic**), and then select **Save** to associate your route table to the *Public* subnet.

The screenshot shows the Azure portal interface for managing a subnet. At the top, the navigation path is Home > Virtual networks > myVirtualNetwork | Subnets > Public. The main section is titled "Public" and shows the following configuration:

- Address range (CIDR block)**: 10.0.0.0/24 (10.0.0.0 - 10.0.0.255 (256 addresses))
- Available addresses**: 250
- NAT gateway**: None
- Network security group**: None
- Route table**: myRouteTablePublic (highlighted with a red box)
- Users**: Manage users
- Service endpoints**: Services (0 selected)
- Subnet delegation**: Delegate subnet to a service (None)

Turn on IP forwarding

Next, turn on IP forwarding for your new NVA virtual machine, *myVmNva*. When Azure sends network traffic to *myVmNva*, if the traffic is destined for a different IP address, IP forwarding sends the traffic to the correct location.

1. Go to the [Azure portal](#) to manage your VM. Search for and select **Virtual machines**.
2. Pick the name of your VM (**myVmNva**).
3. In your NVA virtual machine's menu bar, select **Networking**.
4. Select **myvmnva123**. That's the network interface Azure created for your VM. Azure adds numbers to ensure a unique name.

myVmNva - Networking

Virtual machine

Search (Ctrl+ /) Attach network interface Detach network interface

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Networking

Connect Disks Size Security Extensions Continuous delivery (Preview) Availability + scaling Configuration Identity Properties Locks Export template

Operations

Bastion

Network Interface: myvmnva123 Effective security rule

Virtual network/subnet: myVirtualNetwork/DMZ NIC Public IP: Accelerated networking: **Disabled**

Inbound port rules Outbound port rules Application security rules

Network security group myVmNva-nsg (attached to network interface myvmnva123) Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port
65000	AllowVnetInBound	Any
65001	AllowAzureLoadBalanc...	Any
65500	DenyAllInBound	Any

5. In the network interface menu bar, select **IP configurations**.
6. In the **IP configurations** page, set **IP forwarding** to **Enabled**, and select **Save**.

The screenshot shows the Azure portal interface for managing IP configurations. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Settings, IP configurations (which is selected and highlighted in grey), DNS servers, Network security group, Properties, Locks, Export template, Support + troubleshooting, Effective security rules, Effective routes, and New support request. The main content area has tabs for IP forwarding settings, IP forwarding (with a toggle switch from Disabled to Enabled, also highlighted with a red box), Virtual network (set to myVirtualNetwork), and IP configurations. Under IP configurations, it shows a Subnet * field set to DMZ (10.0.2.0/24). Below this is a table with columns Name, IP Version, and Type, containing one row for ipconfig1 (IPv4, Primary). A search bar for IP configurations is also present.

Create public and private virtual machines

Create a public VM and a private VM in the virtual network. Later, you'll use them to see that Azure routes the *Public* subnet traffic to the *Private* subnet through the NVA.

To create the public VM and the private VM, follow the steps of [Create an NVA](#) earlier. You don't need to wait for deployment to finish or go to the VM resource. You'll use most of the same settings, except as described below.

Before you select **Create** to create the public or private VM, go to the following two subsections ([Public VM](#) and [Private VM](#)), which show the values that have to be different. You may continue to the next section ([Route traffic through an NVA](#)) after Azure finishes deploying both VMs.

Public VM

TAB	SETTING	VALUE
Basics	Resource group	myResourceGroup
	Virtual machine name	myVmPublic
	Public inbound ports	Allow selected ports
	Select inbound ports	RDP
Networking	Virtual network	myVirtualNetwork
	Subnet	Public (10.0.0.0/24)
	Public IP address	The default
Management	Diagnostics storage account	mynvastorageaccount

Private VM

TAB	SETTING	VALUE
Basics	Resource group	myResourceGroup
	Virtual machine name	myVmPrivate
	Public inbound ports	Allow selected ports
	Select inbound ports	RDP
Networking	Virtual network	myVirtualNetwork
	Subnet	Private (10.0.1.0/24)
	Public IP address	The default
Management	Diagnostics storage account	mynvastorageaccount

Route traffic through an NVA

Sign in to myVmPrivate over remote desktop

1. Go to the [Azure portal](#) to manage your private VM. Search for and select **Virtual machines**.
2. Pick the name of your private VM (**myVmPrivate**).
3. In the VM menu bar, select **Connect** to create a remote desktop connection to the private VM.
4. In the **Connect with RDP** page, select **Download RDP File**. Azure creates a Remote Desktop Protocol (.rdp) file and downloads it to your computer.
5. Open the downloaded .rdp file. If prompted, select **Connect**. Select **More choices > Use a different account**, and then enter the user name and password you specified when creating the private VM.
6. Select **OK**.
7. If you receive a certificate warning during the sign-in process, select **Yes** to connect to the VM.

Enable ICMP through the Windows firewall

In a later step, you'll use the trace route tool to test routing. Trace route uses the Internet Control Message Protocol (ICMP), which the Windows Firewall denies by default. Enable ICMP through the Windows firewall.

1. In the Remote Desktop of *myVmPrivate*, open PowerShell.
2. Enter this command:

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

You'll be using trace route to test routing in this tutorial. For production environments, we don't recommend allowing ICMP through the Windows Firewall.

Turn on IP forwarding within myVmNva

You [turned on IP forwarding](#) for the VM's network interface using Azure. The VM's operating system also has to forward network traffic. Turn on IP forwarding for *myVmNva* VM's operating system with these commands.

- From a command prompt on the *myVmPrivate* VM, open a remote desktop to the *myVmNva* VM:

```
mstsc /v:myVmNva
```

- From PowerShell on the *myVmNva* VM, enter this command to turn on IP forwarding:

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRouter -Value 1
```

- Restart the *myVmNva* VM: From the taskbar, select **Start > Power, Other (Planned) > Continue**.

This also disconnects the remote desktop session.

- After the *myVmNva* VM restarts, create a remote desktop session to the *myVmPublic* VM. While still connected to the *myVmPrivate* VM, open a command prompt and run this command:

```
mstsc /v:myVmPublic
```

- In the remote desktop of *myVmPublic*, open PowerShell.

- Enable ICMP through the Windows firewall by entering this command:

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

Test the routing of network traffic

First, let's test routing of network traffic from the *myVmPublic* VM to the *myVmPrivate* VM.

- From PowerShell on the *myVmPublic* VM, enter this command:

```
tracert myVmPrivate
```

The response is similar to this example:

```
Tracing route to myVmPrivate.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.1.4]
over a maximum of 30 hops:

 1  <1 ms      *      1 ms  10.0.2.4
 2  1 ms      1 ms      1 ms  10.0.1.4

Trace complete.
```

You can see the first hop is to 10.0.2.4, which is NVA's private IP address. The second hop is to the private IP address of the *myVmPrivate* VM: 10.0.1.4. Earlier, you added the route to the *myRouteTablePublic* route table and associated it to the *Public* subnet. As a result, Azure sent the traffic through the NVA and not directly to the *Private* subnet.

- Close the remote desktop session to the *myVmPublic* VM, which leaves you still connected to the *myVmPrivate* VM.
- From a command prompt on the *myVmPrivate* VM, enter this command:

```
tracert myVmPublic
```

This command tests the routing of network traffic from the *myVmPrivate* VM to the *myVmPublic* VM. The response is similar to this example:

```
Tracing route to myVmPublic.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.0.4]
over a maximum of 30 hops:

1      1 ms      1 ms      1 ms  10.0.0.4

Trace complete.
```

You can see that Azure routes traffic directly from the *myVmPrivate* VM to the *myVmPublic* VM. By default, Azure routes traffic directly between subnets.

4. Close the remote desktop session to the *myVmPrivate* VM.

Clean up resources

When the resource group is no longer needed, delete *myResourceGroup* and all resources it has:

1. Go to the [Azure portal](#) to manage your resource group. Search for and select **Resource groups**.
2. Pick the name of your resource group (*myResourceGroup*).
3. Select **Delete resource group**.
4. In the confirmation dialog box, enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME**, and then select **Delete**. Azure deletes the *myResourceGroup* and all resources tied to that resource group, including your route tables, storage accounts, virtual networks, VMs, network interfaces, and public IP addresses.

Next steps

In this tutorial, you created a route table and associated it to a subnet. You created a simple NVA that routed traffic from a public subnet to a private subnet. Now you can deploy different preconfigured NVAs from the [Azure Marketplace](#), which provide many useful network functions. To learn more about routing, see [Routing overview](#) and [Manage a route table](#).

While you can deploy many Azure resources within a virtual network, Azure can't deploy resources for some PaaS services into a virtual network. It's possible to restrict access to the resources of some Azure PaaS services, though the restriction must only be traffic from a virtual network subnet. To learn how to restrict network access to Azure PaaS resources, see the next tutorial.

[Restrict network access to PaaS resources](#)

NOTE

Azure services cost money. Azure Cost Management helps you set budgets and configure alerts to keep spending under control. Analyze, manage, and optimize your Azure costs with Cost Management. To learn more, see the [quickstart on analyzing your costs](#).

Route network traffic with a route table using PowerShell

2/18/2021 • 9 minutes to read • [Edit Online](#)

Azure automatically routes traffic between all subnets within a virtual network, by default. You can create your own routes to override Azure's default routing. The ability to create custom routes is helpful if, for example, you want to route traffic between subnets through a network virtual appliance (NVA). In this article, you learn how to:

- Create a route table
- Create a route
- Create a virtual network with multiple subnets
- Associate a route table to a subnet
- Create an NVA that routes traffic
- Deploy virtual machines (VM) into different subnets
- Route traffic from one subnet to another through an NVA

If you don't have an Azure subscription, create a [free account](#) before you begin.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

Create a route table

Before you can create a route table, create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup* for all resources created in this article.

```
New-AzResourceGroup -ResourceGroupName myResourceGroup -Location EastUS
```

Create a route table with [New-AzRouteTable](#). The following example creates a route table named *myRouteTablePublic*.

```
$routeTablePublic = New-AzRouteTable `  
    -Name 'myRouteTablePublic' `  
    -ResourceGroupName myResourceGroup `  
    -location EastUS
```

Create a route

Create a route by retrieving the route table object with [Get-AzRouteTable](#), create a route with [Add-AzRouteConfig](#), then write the route configuration to the route table with [Set-AzRouteTable](#).

```
Get-AzRouteTable `  
    -ResourceGroupName "myResourceGroup" `  
    -Name "myRouteTablePublic" `  
    | Add-AzRouteConfig `  
    -Name "ToPrivateSubnet" `  
    -AddressPrefix 10.0.1.0/24 `  
    -NextHopType "VirtualAppliance" `  
    -NextHopIpAddress 10.0.2.4 `  
    | Set-AzRouteTable
```

Associate a route table to a subnet

Before you can associate a route table to a subnet, you have to create a virtual network and subnet. Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVirtualNetwork* with the address prefix *10.0.0.0/16*.

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

Create three subnets by creating three subnet configurations with [New-AzVirtualNetworkSubnetConfig](#). The following example creates three subnet configurations for *Public*, *Private*, and *DMZ* subnets:

```
$subnetConfigPublic = Add-AzVirtualNetworkSubnetConfig ` 
    -Name Public ` 
    -AddressPrefix 10.0.0.0/24 ` 
    -VirtualNetwork $virtualNetwork

$subnetConfigPrivate = Add-AzVirtualNetworkSubnetConfig ` 
    -Name Private ` 
    -AddressPrefix 10.0.1.0/24 ` 
    -VirtualNetwork $virtualNetwork

$subnetConfigDmz = Add-AzVirtualNetworkSubnetConfig ` 
    -Name DMZ ` 
    -AddressPrefix 10.0.2.0/24 ` 
    -VirtualNetwork $virtualNetwork
```

Write the subnet configurations to the virtual network with [Set-AzVirtualNetwork](#), which creates the subnets in the virtual network:

```
$virtualNetwork | Set-AzVirtualNetwork
```

Associate the *myRouteTablePublic* route table to the *Public* subnet with [Set-AzVirtualNetworkSubnetConfig](#) and then write the subnet configuration to the virtual network with [Set-AzVirtualNetwork](#).

```
Set-AzVirtualNetworkSubnetConfig ` 
    -VirtualNetwork $virtualNetwork ` 
    -Name 'Public' ` 
    -AddressPrefix 10.0.0.0/24 ` 
    -RouteTable $myRouteTablePublic | ` 
Set-AzVirtualNetwork
```

Create an NVA

An NVA is a VM that performs a network function, such as routing, firewalling, or WAN optimization.

Before creating a VM, create a network interface.

Create a network interface

Before creating a network interface, you have to retrieve the virtual network Id with [Get-AzVirtualNetwork](#), then the subnet Id with [Get-AzVirtualNetworkSubnetConfig](#). Create a network interface with [New-AzNetworkInterface](#) in the *DMZ* subnet with IP forwarding enabled:

```
# Retrieve the virtual network object into a variable.
$virtualNetwork=Get-AzVirtualNetwork ` 
    -Name myVirtualNetwork ` 
    -ResourceGroupName myResourceGroup

# Retrieve the subnet configuration into a variable.
$subnetConfigDmz = Get-AzVirtualNetworkSubnetConfig ` 
    -Name DMZ ` 
    -VirtualNetwork $virtualNetwork

# Create the network interface.
$nic = New-AzNetworkInterface ` 
    -ResourceGroupName myResourceGroup ` 
    -Location EastUS ` 
    -Name 'myVmNva' ` 
    -SubnetId $subnetConfigDmz.Id ` 
    -EnableIPForwarding
```

Create a VM

To create a VM and attach an existing network interface to it, you must first create a VM configuration with [New-AzVMConfig](#). The configuration includes the network interface created in the previous step. When prompted for a username and password, select the user name and password you want to log into the VM with.

```
# Create a credential object.  
$cred = Get-Credential -Message "Enter a username and password for the VM."  
  
# Create a VM configuration.  
$vmConfig = New-AzVMConfig `  
    -VMName 'myVmNva' `  
    -VMSize Standard_DS2 | `  
    Set-AzVMOperatingSystem -Windows `  
        -ComputerName 'myVmNva' `  
        -Credential $cred | `  
    Set-AzVMSourceImage `  
        -PublisherName MicrosoftWindowsServer `  
        -Offer WindowsServer `  
        -Skus 2016-Datacenter `  
        -Version latest | `  
    Add-AzVMNetworkInterface -Id $nic.Id
```

Create the VM using the VM configuration with [New-AzVM](#). The following example creates a VM named *myVmNva*.

```
$vmNva = New-AzVM `  
    -ResourceGroupName myResourceGroup `  
    -Location EastUS `  
    -VM $vmConfig `  
    -AsJob
```

The `-AsJob` option creates the VM in the background, so you can continue to the next step.

Create virtual machines

Create two VMs in the virtual network so you can validate that traffic from the *Public* subnet is routed to the *Private* subnet through the network virtual appliance in a later step.

Create a VM in the *Public* subnet with [New-AzVM](#). The following example creates a VM named *myVmPublic* in the *Public* subnet of the *myVirtualNetwork* virtual network.

```
New-AzVm `  
    -ResourceGroupName "myResourceGroup" `  
    -Location "East US" `  
    -VirtualNetworkName "myVirtualNetwork" `  
    -SubnetName "Public" `  
    -ImageName "Win2016Datacenter" `  
    -Name "myVmPublic" `  
    -AsJob
```

Create a VM in the *Private* subnet.

```
New-AzVm ` 
-ResourceGroupName "myResourceGroup" ` 
-Location "East US" ` 
-VirtualNetworkName "myVirtualNetwork" ` 
-SubnetName "Private" ` 
-ImageName "Win2016Datacenter" ` 
-Name "myVmPrivate"
```

The VM takes a few minutes to create. Don't continue with the next step until the VM is created and Azure returns output to PowerShell.

Route traffic through an NVA

Use [Get-AzPublicIpAddress](#) to return the public IP address of the *myVmPrivate* VM. The following example returns the public IP address of the *myVmPrivate* VM:

```
Get-AzPublicIpAddress ` 
-Name myVmPrivate ` 
-ResourceGroupName myResourceGroup ` 
| SelectIpAddress
```

Use the following command to create a remote desktop session with the *myVmPrivate* VM from your local computer. Replace `<publicIpAddress>` with the IP address returned from the previous command.

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

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), then select **OK**. You may receive a certificate warning during the sign-in process. Select **Yes** to proceed with the connection.

In a later step, the `tracert.exe` command is used to test routing. Tracert uses the Internet Control Message Protocol (ICMP), which is denied through the Windows Firewall. Enable ICMP through the Windows firewall by entering the following command from PowerShell on the *myVmPrivate* VM:

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

Though trace route is used to test routing in this article, allowing ICMP through the Windows Firewall for production deployments is not recommended.

You enabled IP forwarding within Azure for the VM's network interface in Enable IP forwarding. Within the VM, the operating system, or an application running within the VM, must also be able to forward network traffic. Enable IP forwarding within the operating system of the *myVmNva*.

From a command prompt on the *myVmPrivate* VM, remote desktop to the *myVmNva*:

```
mstsc /v:myvmnva
```

To enable IP forwarding within the operating system, enter the following command in PowerShell from the *myVmNva* VM:

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRouter -Value 1
```

Restart the *myVmNva* VM, which also disconnects the remote desktop session.

While still connected to the *myVmPrivate* VM, create a remote desktop session to the *myVmPublic* VM, after the *myVmNva* VM restarts:

```
mstsc /v:myVmPublic
```

Enable ICMP through the Windows firewall by entering the following command from PowerShell on the *myVmPublic* VM:

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

To test routing of network traffic to the *myVmPrivate* VM from the *myVmPublic* VM, enter the following command from PowerShell on the *myVmPublic* VM:

```
tracert myVmPrivate
```

The response is similar to the following example:

```
Tracing route to myVmPrivate.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.1.4]
over a maximum of 30 hops:

1 <1 ms * 1 ms 10.0.2.4
2 1 ms 1 ms 1 ms 10.0.1.4

Trace complete.
```

You can see that the first hop is 10.0.2.4, which is the NVA's private IP address. The second hop is 10.0.1.4, the private IP address of the *myVmPrivate* VM. The route added to the *myRouteTablePublic* route table and associated to the *Public* subnet caused Azure to route the traffic through the NVA, rather than directly to the *Private* subnet.

Close the remote desktop session to the *myVmPublic* VM, which leaves you still connected to the *myVmPrivate* VM.

To test routing of network traffic to the *myVmPublic* VM from the *myVmPrivate* VM, enter the following command from a command prompt on the *myVmPrivate* VM:

```
tracert myVmPublic
```

The response is similar to the following example:

```
Tracing route to myVmPublic.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.0.4]
over a maximum of 30 hops:

1 1 ms 1 ms 1 ms 10.0.0.4

Trace complete.
```

You can see that traffic is routed directly from the *myVmPrivate* VM to the *myVmPublic* VM. By default, Azure

routes traffic directly between subnets.

Close the remote desktop session to the *myVmPrivate* VM.

Clean up resources

When no longer needed, use [Remove-AzResourcegroup](#) to remove the resource group and all of the resources it contains.

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

Next steps

In this article, you created a route table and associated it to a subnet. You created a simple network virtual appliance that routed traffic from a public subnet to a private subnet. Deploy a variety of pre-configured network virtual appliances that perform network functions such as firewall and WAN optimization from the [Azure Marketplace](#). To learn more about routing, see [Routing overview](#) and [Manage a route table](#).

While you can deploy many Azure resources within a virtual network, resources for some Azure PaaS services cannot be deployed into a virtual network. You can still restrict access to the resources of some Azure PaaS services to traffic only from a virtual network subnet though. To learn how, see [Restrict network access to PaaS resources](#).

Route network traffic with a route table using the Azure CLI

2/18/2021 • 7 minutes to read • [Edit Online](#)

Azure automatically routes traffic between all subnets within a virtual network, by default. You can create your own routes to override Azure's default routing. The ability to create custom routes is helpful if, for example, you want to route traffic between subnets through a network virtual appliance (NVA). In this article, you learn how to:

- Create a route table
- Create a route
- Create a virtual network with multiple subnets
- Associate a route table to a subnet
- Create an NVA that routes traffic
- Deploy virtual machines (VM) into different subnets
- Route traffic from one subnet to another through an NVA

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
 - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
 - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
 - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a route table

Before you can create a route table, create a resource group with `az group create` for all resources created in this article.

```
# Create a resource group.
az group create \
    --name myResourceGroup \
    --location eastus
```

Create a route table with `az network route-table create`. The following example creates a route table named

myRouteTablePublic.

```
# Create a route table
az network route-table create \
--resource-group myResourceGroup \
--name myRouteTablePublic
```

Create a route

Create a route in the route table with [az network route-table route create](#).

```
az network route-table route create \
--name ToPrivateSubnet \
--resource-group myResourceGroup \
--route-table-name myRouteTablePublic \
--address-prefix 10.0.1.0/24 \
--next-hop-type VirtualAppliance \
--next-hop-ip-address 10.0.2.4
```

Associate a route table to a subnet

Before you can associate a route table to a subnet, you have to create a virtual network and subnet. Create a virtual network with one subnet with [az network vnet create](#).

```
az network vnet create \
--name myVirtualNetwork \
--resource-group myResourceGroup \
--address-prefix 10.0.0.0/16 \
--subnet-name Public \
--subnet-prefix 10.0.0.0/24
```

Create two additional subnets with [az network vnet subnet create](#).

```
# Create a private subnet.
az network vnet subnet create \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name Private \
--address-prefix 10.0.1.0/24

# Create a DMZ subnet.
az network vnet subnet create \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name DMZ \
--address-prefix 10.0.2.0/24
```

Associate the *myRouteTablePublic* route table to the *Public* subnet with [az network vnet subnet update](#).

```
az network vnet subnet update \
--vnet-name myVirtualNetwork \
--name Public \
--resource-group myResourceGroup \
--route-table myRouteTablePublic
```

Create an NVA

An NVA is a VM that performs a network function, such as routing, firewalling, or WAN optimization.

Create an NVA in the *DMZ* subnet with [az vm create](#). When you create a VM, Azure creates and assigns a public IP address to the VM, by default. The `--public-ip-address ""` parameter instructs Azure not to create and assign a public IP address to the VM, since the VM doesn't need to be connected to from the internet. If SSH keys do not already exist in a default key location, the command creates them. To use a specific set of keys, use the `--ssh-key-value` option.

```
az vm create \
--resource-group myResourceGroup \
--name myVmNva \
--image UbuntuLTS \
--public-ip-address "" \
--subnet DMZ \
--vnet-name myVirtualNetwork \
--generate-ssh-keys
```

The VM takes a few minutes to create. Do not continue to the next step until Azure finishes creating the VM and returns output about the VM.

For a network interface to be able to forward network traffic sent to it, that is not destined for its own IP address, IP forwarding must be enabled for the network interface. Enable IP forwarding for the network interface with [az network nic update](#).

```
az network nic update \
--name myVmNvaVMNic \
--resource-group myResourceGroup \
--ip-forwarding true
```

Within the VM, the operating system, or an application running within the VM, must also be able to forward network traffic. Enable IP forwarding within the VM's operating system with [az vm extension set](#):

```
az vm extension set \
--resource-group myResourceGroup \
--vm-name myVmNva \
--name customScript \
--publisher Microsoft.Azure.Extensions \
--settings '{"commandToExecute":"sudo sysctl -w net.ipv4.ip_forward=1"}'
```

The command may take up to a minute to execute.

Create virtual machines

Create two VMs in the virtual network so you can validate that traffic from the *Public* subnet is routed to the *Private* subnet through the NVA in a later step.

Create a VM in the *Public* subnet with [az vm create](#). The `--no-wait` parameter enables Azure to execute the command in the background so you can continue to the next command. To streamline this article, a password is used. Keys are typically used in production deployments. If you use keys, you must also configure SSH agent forwarding. For more information, see the documentation for your SSH client. Replace `<replace-with-your-password>` in the following command with a password of your choosing.

```
adminPassword=<replace-with-your-password>

az vm create \
--resource-group myResourceGroup \
--name myVmPublic \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet Public \
--admin-username azureuser \
--admin-password $adminPassword \
--no-wait
```

Create a VM in the *Private* subnet.

```
az vm create \
--resource-group myResourceGroup \
--name myVmPrivate \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet Private \
--admin-username azureuser \
--admin-password $adminPassword
```

The VM takes a few minutes to create. After the VM is created, the Azure CLI shows information similar to the following example:

```
{
  "fqdns": "",
  "id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVmPrivate",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.1.4",
  "publicIpAddress": "13.90.242.231",
  "resourceGroup": "myResourceGroup"
}
```

Take note of the **publicIpAddress**. This address is used to access the VM from the internet in a later step.

Route traffic through an NVA

Use the following command to create an SSH session with the *myVmPrivate* VM. Replace *<publicIpAddress>* with the public IP address of your VM. In the example above, the IP address is *13.90.242.231*.

```
ssh azureuser@<publicIpAddress>
```

When prompted for a password, enter the password you selected in [Create virtual machines](#).

Use the following command to install trace route on the *myVmPrivate* VM:

```
sudo apt-get install traceroute
```

Use the following command to test routing for network traffic to the *myVmPublic* VM from the *myVmPrivate* VM.

```
traceroute myVmPublic
```

The response is similar to the following example:

```
traceroute to myVmPublic (10.0.0.4), 30 hops max, 60 byte packets
1 10.0.0.4 (10.0.0.4) 1.404 ms 1.403 ms 1.398 ms
```

You can see that traffic is routed directly from the *myVmPrivate* VM to the *myVmPublic* VM. Azure's default routes, route traffic directly between subnets.

Use the following command to SSH to the *myVmPublic* VM from the *myVmPrivate* VM:

```
ssh azureuser@myVmPublic
```

Use the following command to install trace route on the *myVmPublic* VM:

```
sudo apt-get install traceroute
```

Use the following command to test routing for network traffic to the *myVmPrivate* VM from the *myVmPublic* VM.

```
traceroute myVmPrivate
```

The response is similar to the following example:

```
traceroute to myVmPrivate (10.0.1.4), 30 hops max, 60 byte packets
1 10.0.2.4 (10.0.2.4) 0.781 ms 0.780 ms 0.775 ms
2 10.0.1.4 (10.0.0.4) 1.404 ms 1.403 ms 1.398 ms
```

You can see that the first hop is 10.0.2.4, which is the NVA's private IP address. The second hop is 10.0.1.4, the private IP address of the *myVmPrivate* VM. The route added to the *myRouteTablePublic* route table and associated to the *Public* subnet caused Azure to route the traffic through the NVA, rather than directly to the *Private* subnet.

Close the SSH sessions to both the *myVmPublic* and *myVmPrivate* VMs.

Clean up resources

When no longer needed, use [az group delete](#) to remove the resource group and all of the resources it contains.

```
az group delete --name myResourceGroup --yes
```

Next steps

In this article, you created a route table and associated it to a subnet. You created a simple NVA that routed traffic from a public subnet to a private subnet. Deploy a variety of pre-configured NVAs that perform network functions such as firewall and WAN optimization from the [Azure Marketplace](#). To learn more about routing, see [Routing overview](#) and [Manage a route table](#).

While you can deploy many Azure resources within a virtual network, resources for some Azure PaaS services cannot be deployed into a virtual network. You can still restrict access to the resources of some Azure PaaS

services to traffic only from a virtual network subnet though. To learn how, see [Restrict network access to PaaS resources](#).

Create, change, or delete a route table

2/18/2021 • 11 minutes to read • [Edit Online](#)

Azure automatically routes traffic between Azure subnets, virtual networks, and on-premises networks. If you want to change any of Azure's default routing, you do so by creating a route table. If you're new to routing in virtual networks, you can learn more about it in [virtual network traffic routing](#) or by completing a [tutorial](#).

Before you begin

If you don't have one, set up an Azure account with an active subscription. [Create an account for free](#). Then complete one of these tasks before starting steps in any section of this article:

- **Portal users:** Sign in to the [Azure portal](#) with your Azure account.
- **PowerShell users:** Either run the commands in the [Azure Cloud Shell](#), or run PowerShell from your computer. 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. In the Azure Cloud Shell browser tab, find the **Select environment** dropdown list, then choose **PowerShell** if it isn't already selected.

If you're running PowerShell locally, use Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az.Network` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). Also run `Connect-AzAccount` to create a connection with Azure.

- **Azure Command-line interface (CLI) users:** Either run the commands in the [Azure Cloud Shell](#), or run the CLI from your computer. Use Azure CLI version 2.0.31 or later if you're running the Azure CLI locally. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). Also run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with must be assigned to the [Network contributor role](#) or to a [Custom role](#) that's assigned the appropriate actions listed in [Permissions](#).

Create a route table

There's a limit to how many route tables you can create per Azure location and subscription. For details, see [Networking limits - Azure Resource Manager](#).

1. On the [Azure portal](#) menu or from the [Home](#) page, select **Create a resource**.
2. In the search box, enter *Route table*. When **Route table** appears in the search results, select it.
3. In the **Route table** page, select **Create**.
4. In the **Create route table** dialog box:
 - a. Enter a **Name** for the route table.
 - b. Choose your **Subscription**.
 - c. Choose an existing **Resource group** or select **Create new** to create a new resource group.
 - d. Choose a **Location**.
 - e. If you plan to associate the route table to a subnet in a virtual network that's connected to your on-premises network through a VPN gateway, and you don't want to propagate your on-premises routes to the network interfaces in the subnet, set **Virtual network gateway route propagation** to

Disabled.

5. Select **Create** to create your new route table.

Create route table - commands

TOOL	COMMAND
Azure CLI	az network route-table create
PowerShell	New-AzRouteTable

View route tables

Go to the [Azure portal](#) to manage your virtual network. Search for and select **Route tables**. The route tables that exist in your subscription are listed.

View route table - commands

TOOL	COMMAND
Azure CLI	az network route-table list
PowerShell	Get-AzRouteTable

View details of a route table

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Route tables**.
2. In the route table list, choose the route table that you want to view details for.
3. In the route table page, under **Settings**, view the **Routes** in the route table or the **Subnets** the route table is associated to.

To learn more about common Azure settings, see the following information:

- [Activity log](#)
- [Access control \(IAM\)](#)
- [Tags](#)
- [Locks](#)
- [Automation script](#)

View details of route table - commands

TOOL	COMMAND
Azure CLI	az network route-table show
PowerShell	Get-AzRouteTable

Change a route table

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Route tables**.
2. In the route table list, choose the route table that you want to change.

The most common changes are to [add](#) routes, [remove](#) routes, [associate](#) route tables to subnets, or [dissociate](#)

route tables from subnets.

Change a route table - commands

TOOL	COMMAND
Azure CLI	az network route-table update
PowerShell	Set-AzRouteTable

Associate a route table to a subnet

You can optionally associate a route table to a subnet. A route table can be associated to zero or more subnets. Because route tables aren't associated to virtual networks, you must associate a route table to each subnet you want the route table associated to. Azure routes all traffic leaving the subnet based on routes you've created within route tables, [default routes](#), and routes propagated from an on-premises network, if the virtual network is connected to an Azure virtual network gateway (ExpressRoute or VPN). You can only associate a route table to subnets in virtual networks that exist in the same Azure location and subscription as the route table.

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Virtual networks**.
2. In the virtual network list, choose the virtual network that contains the subnet you want to associate a route table to.
3. In the virtual network menu bar, choose **Subnets**.
4. Select the subnet you want to associate the route table to.
5. In **Route table**, choose the route table you want to associate to the subnet.
6. Select **Save**.

If your virtual network is connected to an Azure VPN gateway, don't associate a route table to the [gateway subnet](#) that includes a route with a destination of *0.0.0.0/0*. Doing so can prevent the gateway from functioning properly. For more information about using *0.0.0.0/0* in a route, see [Virtual network traffic routing](#).

Associate a route table - commands

TOOL	COMMAND
Azure CLI	az network vnet subnet update
PowerShell	Set-AzVirtualNetworkSubnetConfig

Dissociate a route table from a subnet

When you dissociate a route table from a subnet, Azure routes traffic based on its [default routes](#).

1. Go to the [Azure portal](#) to manage your virtual network. Search for and select **Virtual networks**.
2. In the virtual network list, choose the virtual network that contains the subnet you want to dissociate a route table from.
3. In the virtual network menu bar, choose **Subnets**.
4. Select the subnet you want to dissociate the route table from.
5. In **Route table**, choose **None**.

6. Select **Save**.

Dissociate a route table - commands

TOOL	COMMAND
Azure CLI	az network vnet subnet update
PowerShell	Set-AzVirtualNetworkSubnetConfig

Delete a route table

You can't delete a route table that's associated to any subnets. [Dissociate](#) a route table from all subnets before attempting to delete it.

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table you want to delete.
3. Select **Delete**, and then select **Yes** in the confirmation dialog box.

Delete a route table - commands

TOOL	COMMAND
Azure CLI	az network route-table delete
PowerShell	Remove-AzRouteTable

Create a route

There's a limit to how many routes per route table can create per Azure location and subscription. For details, see [Networking limits - Azure Resource Manager](#).

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table you want to add a route to.
3. From the route table menu bar, choose **Routes > Add**.
4. Enter a unique **Route name** for the route within the route table.
5. Enter the **Address prefix**, in Classless Inter-Domain Routing (CIDR) notation, that you want to route traffic to. The prefix can't be duplicated in more than one route within the route table, though the prefix can be within another prefix. For example, if you defined `10.0.0.0/16` as a prefix in one route, you can still define another route with the `10.0.0.0/22` address prefix. Azure selects a route for traffic based on longest prefix match. To learn more, see [How Azure selects a route](#).
6. Choose a **Next hop type**. To learn more about next hop types, see [Virtual network traffic routing](#).
7. If you chose a **Next hop type** of **Virtual appliance**, enter an IP address for **Next hop address**.
8. Select **OK**.

Create a route - commands

TOOL	COMMAND
Azure CLI	az network route-table route create
PowerShell	New-AzRouteConfig

View routes

A route table contains zero or more routes. To learn more about the information listed when viewing routes, see [Virtual network traffic routing](#).

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table you want to view routes for.
3. In the route table menu bar, choose **Routes** to see the list of routes.

View routes - commands

TOOL	COMMAND
Azure CLI	az network route-table route list
PowerShell	Get-AzRouteConfig

View details of a route

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table containing the route you want to view details for.
3. In the route table menu bar, choose **Routes** to see the list of routes.
4. Select the route you want to view details of.

View details of a route - commands

TOOL	COMMAND
Azure CLI	az network route-table route show
PowerShell	Get-AzRouteConfig

Change a route

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table containing the route you want to change.
3. In the route table menu bar, choose **Routes** to see the list of routes.
4. Choose the route you want to change.
5. Change existing settings to their new settings, then select **Save**.

Change a route - commands

TOOL	COMMAND
Azure CLI	az network route-table route update
PowerShell	Set-AzRouteConfig

Delete a route

1. Go to the [Azure portal](#) to manage your route tables. Search for and select **Route tables**.
2. In the route table list, choose the route table containing the route you want to delete.
3. In the route table menu bar, choose **Routes** to see the list of routes.
4. Choose the route you want to delete.
5. Select **Delete**, then select **Yes** in the confirmation dialog box.

Delete a route - commands

TOOL	COMMAND
Azure CLI	az network route-table route delete
PowerShell	Remove-AzRouteConfig

View effective routes

The effective routes for each VM-attached network interface are a combination of route tables that you've created, Azure's default routes, and any routes propagated from on-premises networks via the Border Gateway Protocol (BGP) through an Azure virtual network gateway. Understanding the effective routes for a network interface is helpful when troubleshooting routing problems. You can view the effective routes for any network interface that's attached to a running VM.

1. Go to the [Azure portal](#) to manage your VMs. Search for and select **Virtual machines**.
2. In the virtual machine list, choose the VM you want to view effective routes for.
3. In the VM menu bar, choose **Networking**.
4. Select the name of a network interface.
5. In the network interface menu bar, select **Effective routes**.
6. Review the list of effective routes to see whether the correct route exists for where you want to route traffic to. Learn more about next hop types that you see in this list in [Virtual network traffic routing](#).

View effective routes - commands

TOOL	COMMAND
Azure CLI	az network nic show-effective-route-table
PowerShell	Get-AzEffectiveRouteTable

Validate routing between two endpoints

You can determine the next hop type between a virtual machine and the IP address of another Azure resource, an on-premises resource, or a resource on the Internet. Determining Azure's routing is helpful when troubleshooting routing problems. To complete this task, you must have an existing network watcher. If you don't have an existing network watcher, create one by completing the steps in [Create a Network Watcher instance](#).

1. Go to the [Azure portal](#) to manage your network watchers. Search for and select **Network Watcher**.
2. In the network watcher menu bar, choose **Next hop**.
3. In the **Network Watcher | Next hop** page:
 - a. Choose your **Subscription** and the **Resource group** of the source VM you want to validate routing from.
 - b. Choose the **Virtual machine** and the **Network interface** that's attached to the VM.
 - c. Enter a **Source IP address** assigned to the network interface that you want to validate routing from.
 - d. Enter a **Destination IP address** that you want to validate routing to.
4. Select **Next hop**.

After a short wait, Azure tells you the next hop type and the ID of the route that routed the traffic. Learn more about next hop types that you see returned in [Virtual network traffic routing](#).

Validate routing between two endpoints - commands

TOOL	COMMAND
Azure CLI	az network watcher show-next-hop
PowerShell	Get-AzNetworkWatcherNextHop

Permissions

To do tasks on route tables and routes, your account must be assigned to the [Network contributor role](#) or to a [Custom role](#) that's assigned the appropriate actions listed in the following table:

ACTION	NAME
Microsoft.Network/routeTables/read	Read a route table
Microsoft.Network/routeTables/write	Create or update a route table
Microsoft.Network/routeTables/delete	Delete a route table
Microsoft.Network/routeTables/join/action	Associate a route table to a subnet
Microsoft.Network/routeTables/routes/read	Read a route
Microsoft.Network/routeTables/routes/write	Create or update a route
Microsoft.Network/routeTables/routes/delete	Delete a route

ACTION	NAME
Microsoft.Network/networkInterfaces/effectiveRouteTable/action	Get the effective route table for a network interface
Microsoft.Network/networkWatchers/nextHop/action	Gets the next hop from a VM

Next steps

- Create a route table using [PowerShell](#) or [Azure CLI](#) sample scripts, or Azure [Resource Manager](#) templates
- Create and assign [Azure Policy definitions](#) for virtual networks

Create, change, or delete a network interface

2/18/2021 • 21 minutes to read • [Edit Online](#)

Learn how to create, change settings for, and delete a network interface. A network interface enables an Azure Virtual Machine to communicate with internet, Azure, and on-premises resources. When creating a virtual machine using the Azure portal, the portal creates one network interface with default settings for you. You may instead choose to create network interfaces with custom settings and add one or more network interfaces to a virtual machine when you create it. You may also want to change default network interface settings for an existing network interface. This article explains how to create a network interface with custom settings, change existing settings, such as network filter (network security group) assignment, subnet assignment, DNS server settings, and IP forwarding, and delete a network interface.

If you need to add, change, or remove IP addresses for a network interface, see [Manage IP addresses](#). If you need to add network interfaces to, or remove network interfaces from virtual machines, see [Add or remove network interfaces](#).

Before you begin

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with your Azure account.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.28 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Permissions](#).

Create a network interface

When creating a virtual machine using the Azure portal, the portal creates a network interface with default settings for you. If you'd rather specify all your network interface settings, you can create a network interface with custom settings and attach the network interface to a virtual machine when creating the virtual machine

(using PowerShell or the Azure CLI). You can also create a network interface and add it to an existing virtual machine (using PowerShell or the Azure CLI). To learn how to create a virtual machine with an existing network interface or to add to, or remove network interfaces from existing virtual machines, see [Add or remove network interfaces](#). Before creating a network interface, you must have an existing [virtual network](#) in the same location and subscription you create a network interface in.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*.

When **network interfaces** appear in the search results, select it.

2. Select + Add under **Network interfaces**.

3. Enter, or select values for the following settings, then select **Create**:

SETTING	REQUIRED?	DETAILS
Name	Yes	The name must be unique within the resource group you select. Over time, you'll likely have several network interfaces in your Azure subscription. For suggestions when creating a naming convention to make managing several network interfaces easier, see Naming conventions . The name cannot be changed after the network interface is created.
Virtual network	Yes	Select the virtual network for the network interface. You can only assign a network interface to a virtual network that exists in the same subscription and location as the network interface. Once a network interface is created, you cannot change the virtual network it is assigned to. The virtual machine you add the network interface to must also exist in the same location and subscription as the network interface.
Subnet	Yes	Select a subnet within the virtual network you selected. You can change the subnet the network interface is assigned to after it's created.

SETTING	REQUIRED?	DETAILS
Private IP address assignment	Yes	In this setting, you're choosing the assignment method for the IPv4 address. Choose from the following assignment methods: Dynamic : When selecting this option, Azure automatically assigns the next available address from the address space of the subnet you selected. Static : When selecting this option, you must manually assign an available IP address from within the address space of the subnet you selected. Static and dynamic addresses do not change until you change them or the network interface is deleted. You can change the assignment method after the network interface is created. The Azure DHCP server assigns this address to the network interface within the operating system of the virtual machine.
Network security group	No	Leave set to None , select an existing network security group , or create a network security group . Network security groups enable you to filter network traffic in and out of a network interface. You can apply zero or one network security group to a network interface. Zero or one network security group can also be applied to the subnet the network interface is assigned to. When a network security group is applied to a network interface and the subnet the network interface is assigned to, sometimes unexpected results occur. To troubleshoot network security groups applied to network interfaces and subnets, see Troubleshoot network security groups .
Subscription	Yes	Select one of your Azure subscriptions . The virtual machine you attach a network interface to and the virtual network you connect it to must exist in the same subscription.

SETTING	REQUIRED?	DETAILS
Private IP address (IPv6)	No	If you select this checkbox, an IPv6 address is assigned to the network interface, in addition to the IPv4 address assigned to the network interface. See the IPv6 section of this article for important information about use of IPv6 with network interfaces. You cannot select an assignment method for the IPv6 address. If you choose to assign an IPv6 address, it is assigned with the dynamic method.
IPv6 name (only appears when the Private IP address (IPv6) checkbox is checked)	Yes, if the Private IP address (IPv6) checkbox is checked.	This name is assigned to a secondary IP configuration for the network interface. To learn more about IP configurations, see View network interface settings .
Resource group	Yes	Select an existing resource group or create one. A network interface can exist in the same, or different resource group, than the virtual machine you attach it to, or the virtual network you connect it to.
Location	Yes	The virtual machine you attach a network interface to and the virtual network you connect it to must exist in the same location , also referred to as a region.

The portal doesn't provide the option to assign a public IP address to the network interface when you create it, though the portal does create a public IP address and assign it to a network interface when you create a virtual machine using the portal. To learn how to add a public IP address to the network interface after creating it, see [Manage IP addresses](#). If you want to create a network interface with a public IP address, you must use the CLI or PowerShell to create the network interface.

The portal doesn't provide the option to assign the network interface to application security groups when creating a network interface, but the Azure CLI and PowerShell do. You can assign an existing network interface to an application security group using the portal however, as long as the network interface is attached to a virtual machine. To learn how to assign a network interface to an application security group, see [Add to or remove from application security groups](#).

NOTE

Azure assigns a MAC address to the network interface only after the network interface is attached to a virtual machine and the virtual machine is started the first time. You cannot specify the MAC address that Azure assigns to the network interface. The MAC address remains assigned to the network interface until the network interface is deleted or the private IP address assigned to the primary IP configuration of the primary network interface is changed. To learn more about IP addresses and IP configurations, see [Manage IP addresses](#)

Commands

TOOL	COMMAND
CLI	az network nic create
PowerShell	New-AzNetworkInterface

View network interface settings

You can view and change most settings for a network interface after it's created. The portal does not display the DNS suffix or application security group membership for the network interface. You can use the PowerShell or Azure CLI [commands](#) to view the DNS suffix and application security group membership.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface you want to view or change settings for from the list.
3. The following items are listed for the network interface you selected:
 - **Overview:** Provides information about the network interface, such as the IP addresses assigned to it, the virtual network/subnet the network interface is assigned to, and the virtual machine the network interface is attached to (if it's attached to one). The following picture shows the overview settings for a network interface named **mywebserver256**:

Setting	Value
Resource group (change)	Private IP address MyRG 10.0.0.4
Location	Virtual network/subnet West US MyVNet/Front-end
Subscription name (change)	Public IP address <Subscription name> 13.64.66.250 (MyWebServer-ip)
Subscription ID	Network security group <Subscription ID> MyWebServer-nsg
	Attached to MyWebServer

You can move a network interface to a different resource group or subscription by selecting **(change)** next to the **Resource group** or **Subscription name**. If you move the network interface to a new subscription, you must move all resources related to the network interface with it. If the network interface is attached to a virtual machine, for example, you must also move the virtual machine, and other virtual machine-related resources. To move a network interface, see [Move resource to a new resource group or subscription](#). The article lists prerequisites, and how to move resources using the Azure portal, PowerShell, and the Azure CLI.

- **IP configurations:** Public and private IPv4 and IPv6 addresses assigned to IP configurations are listed here. If an IPv6 address is assigned to an IP configuration, the address is not displayed. To learn more about IP configurations and how to add and remove IP addresses, see [Configure IP addresses for an Azure network interface](#). IP forwarding and subnet assignment are also configured in this section. To learn more about these settings, see [Enable or disable IP forwarding](#) and [Change subnet assignment](#).
- **DNS servers:** You can specify which DNS server a network interface is assigned by the Azure DHCP servers. The network interface can inherit the setting from the virtual network the network interface is assigned to, or have a custom setting that overrides the setting for the virtual network it's assigned to. To modify what's displayed, see [Change DNS servers](#).
- **Network security group (NSG):** Displays which NSG is associated to the network interface (if

any). An NSG contains inbound and outbound rules to filter network traffic for the network interface. If an NSG is associated to the network interface, the name of the associated NSG is displayed. To modify what's displayed, see [Associate or dissociate a network security group](#).

- **Properties:** Displays key settings about the network interface, including its MAC address (blank if the network interface isn't attached to a virtual machine), and the subscription it exists in.
- **Effective security rules:** Security rules are listed if the network interface is attached to a running virtual machine, and an NSG is associated to the network interface, the subnet it's assigned to, or both. To learn more about what's displayed, see [View effective security rules](#). To learn more about NSGs, see [Network security groups](#).
- **Effective routes:** Routes are listed if the network interface is attached to a running virtual machine. The routes are a combination of the Azure default routes, any user-defined routes, and any BGP routes that may exist for the subnet the network interface is assigned to. To learn more about what's displayed, see [View effective routes](#). To learn more about Azure default routes and user-defined routes, see [Routing overview](#). Common Azure Resource Manager settings: To learn more about common Azure Resource Manager settings, see [Activity log](#), [Access control \(IAM\)](#), [Tags](#), [Locks](#), and [Automation script](#).

Commands

If an IPv6 address is assigned to a network interface, the PowerShell output returns the fact that the address is assigned, but it doesn't return the assigned address. Similarly, the CLI returns the fact that the address is assigned, but returns *null* in its output for the address.

TOOL	COMMAND
CLI	<code>az network nic list</code> to view network interfaces in the subscription; <code>az network nic show</code> to view settings for a network interface
PowerShell	<code>Get-AzNetworkInterface</code> to view network interfaces in the subscription or view settings for a network interface

Change DNS servers

The DNS server is assigned by the Azure DHCP server to the network interface within the virtual machine operating system. The DNS server assigned is whatever the DNS server setting is for a network interface. To learn more about name resolution settings for a network interface, see [Name resolution for virtual machines](#). The network interface can inherit the settings from the virtual network, or use its own unique settings that override the setting for the virtual network.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface that you want to change a DNS server for from the list.
3. Select **DNS servers** under **SETTINGS**.
4. Select either:
 - **Inherit from virtual network:** Choose this option to inherit the DNS server setting defined for the virtual network the network interface is assigned to. At the virtual network level, either a custom DNS server or the Azure-provided DNS server is defined. The Azure-provided DNS server can resolve hostnames for resources assigned to the same virtual network. FQDN must be used to resolve for resources assigned to different virtual networks.
 - **Custom:** You can configure your own DNS server to resolve names across multiple virtual networks. Enter the IP address of the server you want to use as a DNS server. The DNS server address you

specify is assigned only to this network interface and overrides any DNS setting for the virtual network the network interface is assigned to.

NOTE

If the VM uses a NIC that's part of an availability set, all the DNS servers that are specified for each of the VMs from all NICs that are part of the availability set will be inherited.

5. Select **Save**.

Commands

TOOL	COMMAND
CLI	az network nic update
PowerShell	Set-AzNetworkInterface

Enable or disable IP forwarding

IP forwarding enables the virtual machine a network interface is attached to:

- Receive network traffic not destined for one of the IP addresses assigned to any of the IP configurations assigned to the network interface.
- Send network traffic with a different source IP address than the one assigned to one of a network interface's IP configurations.

The setting must be enabled for every network interface that is attached to the virtual machine that receives traffic that the virtual machine needs to forward. A virtual machine can forward traffic whether it has multiple network interfaces or a single network interface attached to it. While IP forwarding is an Azure setting, the virtual machine must also run an application able to forward the traffic, such as firewall, WAN optimization, and load balancing applications. When a virtual machine is running network applications, the virtual machine is often referred to as a network virtual appliance. You can view a list of ready to deploy network virtual appliances in the [Azure Marketplace](#). IP forwarding is typically used with user-defined routes. To learn more about user-defined routes, see [User-defined routes](#).

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface that you want to enable or disable IP forwarding for.
3. Select **IP configurations** in the **SETTINGS** section.
4. Select **Enabled** or **Disabled** (default setting) to change the setting.
5. Select **Save**.

Commands

TOOL	COMMAND
CLI	az network nic update
PowerShell	Set-AzNetworkInterface

Change subnet assignment

You can change the subnet, but not the virtual network, that a network interface is assigned to.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface that you want to change subnet assignment for.
3. Select **IP configurations** under **SETTINGS**. If any private IP addresses for any IP configurations listed have **(Static)** next to them, you must change the IP address assignment method to dynamic by completing the steps that follow. All private IP addresses must be assigned with the dynamic assignment method to change the subnet assignment for the network interface. If the addresses are assigned with the dynamic method, continue to step five. If any IPv4 addresses are assigned with the static assignment method, complete the following steps to change the assignment method to dynamic:
 - Select the IP configuration you want to change the IPv4 address assignment method for from the list of IP configurations.
 - Select **Dynamic** for the private IP address **Assignment** method. You cannot assign an IPv6 address with the static assignment method.
 - Select **Save**.
4. Select the subnet you want to move the network interface to from the **Subnet** drop-down list.
5. Select **Save**. New dynamic addresses are assigned from the subnet address range for the new subnet. After assigning the network interface to a new subnet, you can assign a static IPv4 address from the new subnet address range if you choose. To learn more about adding, changing, and removing IP addresses for a network interface, see [Manage IP addresses](#).

Commands

TOOL	COMMAND
CLI	<code>az network nic ip-config update</code>
PowerShell	<code>Set-AzNetworkInterfaceIpConfig</code>

Add to or remove from application security groups

You can only add a network interface to, or remove a network interface from an application security group using the portal if the network interface is attached to a virtual machine. You can use PowerShell or the Azure CLI to add a network interface to, or remove a network interface from an application security group, whether the network interface is attached to a virtual machine or not. Learn more about [Application security groups](#) and how to [create an application security group](#).

1. In the *Search resources, services, and docs* box at the top of the portal, begin typing the name of a virtual machine that has a network interface that you want to add to, or remove from, an application security group. When the name of your VM appears in the search results, select it.
2. Under **SETTINGS**, select **Networking**. Select **Application Security Groups** then **Configure the application security group**select the application security groups that you want to add the network interface to, or unselect the application security groups that you want to remove the network interface from, and then select **Save**. Only network interfaces that exist in the same virtual network can be added to the same application security group. The application security group must exist in the same location as the network interface.

Commands

TOOL	COMMAND
CLI	az network nic update
PowerShell	Set-AzNetworkInterface

Associate or dissociate a network security group

1. In the search box at the top of the portal, enter *network interfaces* in the search box. When **network interfaces** appear in the search results, select it.
2. Select the network interface in the list that you want to associate a network security group to, or dissociate a network security group from.
3. Select **Network security group** under **SETTINGS**.
4. Select **Edit**.
5. Select **Network security group** and then select the network security group you want to associate to the network interface, or select **None**, to dissociate a network security group.
6. Select **Save**.

Commands

- Azure CLI: [az network nic update](#)
- PowerShell: [Set-AzNetworkInterface](#)

Delete a network interface

You can delete a network interface as long as it's not attached to a virtual machine. If a network interface is attached to a virtual machine, you must first place the virtual machine in the stopped (deallocated) state, then detach the network interface from the virtual machine. To detach a network interface from a virtual machine, complete the steps in [Detach a network interface from a virtual machine](#). You cannot detach a network interface from a virtual machine if it's the only network interface attached to the virtual machine however. A virtual machine must always have at least one network interface attached to it. Deleting a virtual machine detaches all network interfaces attached to it, but does not delete the network interfaces.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface in the list that you want to delete.
3. Under **Overview** Select **Delete**.
4. Select **Yes** to confirm deletion of the network interface.

When you delete a network interface, any MAC or IP addresses assigned to it are released.

Commands

TOOL	COMMAND
CLI	az network nic delete
PowerShell	Remove-AzNetworkInterface

Resolve connectivity issues

If you are unable to communicate to or from a virtual machine, network security group security rules or routes effective for a network interface, may be causing the problem. You have the following options to help resolve the

issue:

View effective security rules

The effective security rules for each network interface attached to a virtual machine are a combination of the rules you've created in a network security group and [default security rules](#). Understanding the effective security rules for a network interface may help you determine why you're unable to communicate to or from a virtual machine. You can view the effective rules for any network interface that is attached to a running virtual machine.

1. In the search box at the top of the portal, enter the name of a virtual machine you want to view effective security rules for. If you don't know the name of a virtual machine, enter *virtual machines* in the search box. When **Virtual machines** appear in the search results, select it, and then select a virtual machine from the list.
2. Select **Networking** under **SETTINGS**.
3. Select the name of a network interface.
4. Select **Effective security rules** under **SUPPORT + TROUBLESHOOTING**.
5. Review the list of effective security rules to determine if the correct rules exist for your required inbound and outbound communication. Learn more about what you see in the list in [Network security group overview](#).

The IP flow verify feature of Azure Network Watcher can also help you determine if security rules are preventing communication between a virtual machine and an endpoint. To learn more, see [IP flow verify](#).

Commands

- Azure CLI: [az network nic list-effective-nsg](#)
- PowerShell: [Get-AzEffectiveNetworkSecurityGroup](#)

View effective routes

The effective routes for the network interfaces attached to a virtual machine are a combination of default routes, any routes you've created, and any routes propagated from on-premises networks via BGP through an Azure virtual network gateway. Understanding the effective routes for a network interface may help you determine why you're unable to communicate to or from a virtual machine. You can view the effective routes for any network interface that is attached to a running virtual machine.

1. In the search box at the top of the portal, enter the name of a virtual machine you want to view effective security rules for. If you don't know the name of a virtual machine, enter *virtual machines* in the search box. When **Virtual machines** appear in the search results, select it, and then select a virtual machine from the list.
2. Select **Networking** under **SETTINGS**.
3. Select the name of a network interface.
4. Select **Effective routes** under **SUPPORT + TROUBLESHOOTING**.
5. Review the list of effective routes to determine if the correct routes exist for your required inbound and outbound communication. Learn more about what you see in the list in [Routing overview](#).

The next hop feature of Azure Network Watcher can also help you determine if routes are preventing communication between a virtual machine and an endpoint. To learn more, see [Next hop](#).

Commands

- Azure CLI: [az network nic show-effective-route-table](#)
- PowerShell: [Get-AzEffectiveRouteTable](#)

Permissions

To perform tasks on network interfaces, your account must be assigned to the [network contributor](#) role or to a [custom](#) role that is assigned the appropriate permissions listed in the following table:

ACTION	NAME
Microsoft.Network/networkInterfaces/read	Get network interface
Microsoft.Network/networkInterfaces/write	Create or update network interface
Microsoft.Network/networkInterfaces/join/action	Attach a network interface to a virtual machine
Microsoft.Network/networkInterfaces/delete	Delete network interface
Microsoft.Network/networkInterfaces/joinViaPrivateIp/action	Join a resource to a network interface via a servi...
Microsoft.Network/networkInterfaces/effectiveRouteTable/act ion	Get network interface effective route table
Microsoft.Network/networkInterfaces/effectiveNetworkSecuri tyGroups/action	Get network interface effective security groups
Microsoft.Network/networkInterfaces/loadBalancers/read	Get network interface load balancers
Microsoft.Network/networkInterfaces/serviceAssociations/re a d	Get service association
Microsoft.Network/networkInterfaces/serviceAssociations/wri te	Create or update a service association
Microsoft.Network/networkInterfaces/serviceAssociations/del ete	Delete service association
Microsoft.Network/networkInterfaces/serviceAssociations/val idate/action	Validate service association
Microsoft.Network/networkInterfaces/ipconfigurations/read	Get network interface IP configuration

Next steps

- Create a VM with multiple NICs using the [Azure CLI](#) or [PowerShell](#)
- Create a single NIC VM with multiple IPv4 addresses using the [Azure CLI](#) or [PowerShell](#)
- Create a single NIC VM with a private IPv6 address (behind an Azure Load Balancer) using the [Azure CLI](#), [PowerShell](#), or [Azure Resource Manager template](#)
- Create a network interface using [PowerShell](#) or [Azure CLI](#) sample scripts, or using Azure [Resource Manager template](#)
- Create and assign [Azure Policy definitions](#) for virtual networks

Create, change, or delete a virtual network

2/18/2021 • 13 minutes to read • [Edit Online](#)

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Learn how to create and delete a virtual network and change settings, like DNS servers and IP address spaces, for an existing virtual network. If you're new to virtual networks, you can learn more about them in the [Virtual network overview](#) or by completing a [tutorial](#). A virtual network contains subnets. To learn how to create, change, and delete subnets, see [Manage subnets](#).

Before you begin

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with your Azure account.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.31 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.
- The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Permissions](#).

Create a virtual network

1. Select + **Create a resource > Networking > Virtual network**.

2. Enter or select values for the following settings, then select **Create**:

- **Name:** The name must be unique in the [resource group](#) that you select to create the virtual network in. You cannot change the name after the virtual network is created. You can create multiple virtual networks over time. For naming suggestions, see [Naming conventions](#). Following a naming convention can help make it easier to manage multiple virtual networks.
- **Address space:** The address space for a virtual network is composed of one or more non-overlapping address ranges that are specified in CIDR notation. The address range you define can be public or private (RFC 1918). Whether you define the address range as public or private, the address range is reachable only from within the virtual network, from interconnected virtual

networks, and from any on-premises networks that you have connected to the virtual network. You cannot add the following address ranges:

- 224.0.0.0/4 (Multicast)
- 255.255.255.255/32 (Broadcast)
- 127.0.0.0/8 (Loopback)
- 169.254.0.0/16 (Link-local)
- 168.63.129.16/32 (Internal DNS, DHCP, and Azure Load Balancer [health probe](#))

Although you can define only one address range when you create the virtual network in the portal, you can add more address ranges to the address space after the virtual network is created. To learn how to add an address range to an existing virtual network, see [Add or remove an address range](#).

WARNING

If a virtual network has address ranges that overlap with another virtual network or on-premises network, the two networks can't be connected. Before you define an address range, consider whether you might want to connect the virtual network to other virtual networks or on-premises networks in the future. Microsoft recommends configuring virtual network address ranges with private address space or public address space owned by your organization.

- **Subnet name:** The subnet name must be unique within the virtual network. You cannot change the subnet name after the subnet is created. The portal requires that you define one subnet when you create a virtual network, even though a virtual network isn't required to have any subnets. In the portal, you can define only one subnet when you create a virtual network. You can add more subnets to the virtual network later, after the virtual network is created. To add a subnet to a virtual network, see [Manage subnets](#). You can create a virtual network that has multiple subnets by using Azure CLI or PowerShell.

TIP

Sometimes, administrators create different subnets to filter or control traffic routing between the subnets. Before you define subnets, consider how you might want to filter and route traffic between your subnets. To learn more about filtering traffic between subnets, see [Network security groups](#). Azure automatically routes traffic between subnets, but you can override Azure default routes. To learn more about Azures default subnet traffic routing, see [Routing overview](#).

- **Subnet address range:** The range must be within the address space you entered for the virtual network. The smallest range you can specify is /29, which provides eight IP addresses for the subnet. Azure reserves the first and last address in each subnet for protocol conformance. Three additional addresses are reserved for Azure service usage. As a result, a virtual network with a subnet address range of /29 has only three usable IP addresses. If you plan to connect a virtual network to a VPN gateway, you must create a gateway subnet. Learn more about [specific address range considerations for gateway subnets](#). You can change the address range after the subnet is created, under specific conditions. To learn how to change a subnet address range, see [Manage subnets](#).
- **Subscription:** Select a [subscription](#). You cannot use the same virtual network in more than one Azure subscription. However, you can connect a virtual network in one subscription to virtual networks in other subscriptions with [virtual network peering](#). Any Azure resource that you connect to the virtual network must be in the same subscription as the virtual network.

- **Resource group:** Select an existing [resource group](#) or create a new one. An Azure resource that you connect to the virtual network can be in the same resource group as the virtual network or in a different resource group.
- **Location:** Select an Azure [location](#), also known as a region. A virtual network can be in only one Azure location. However, you can connect a virtual network in one location to a virtual network in another location by using a VPN gateway. Any Azure resource that you connect to the virtual network must be in the same location as the virtual network.

Commands

- Azure CLI: [az network vnet create](#)
- PowerShell: [New-AzVirtualNetwork](#)

View virtual networks and settings

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it.
2. From the list of virtual networks, select the virtual network that you want to view settings for.
3. The following settings are listed for the virtual network you selected:
 - **Overview:** Provides information about the virtual network, including address space and DNS servers. The following screenshot shows the overview settings for a virtual network named **MyVNet**:

DEVICE	TYPE	IP ADDRESS	SUBNET
myvm2951	Network interface	10.0.0.4	MySubnet1
myvm1607	Network interface	10.0.0.5	MySubnet1
MyLB1	Load balancer	10.0.0.6	MySubnet1
myvm3276	Network interface	10.0.1.4	MySubnet2

You can move a virtual network to a different subscription or resource group by selecting **Change** next to **Resource group** or **Subscription name**. To learn how to move a virtual network, see [Move resources to a different resource group or subscription](#). The article lists prerequisites, and how to move resources by using the Azure portal, PowerShell, and Azure CLI. All resources that are connected to the virtual network must move with the virtual network.

- **Address space:** The address spaces that are assigned to the virtual network are listed. To learn

how to add and remove an address range to the address space, complete the steps in [Add or remove an address range](#).

- **Connected devices:** Any resources that are connected to the virtual network are listed. In the preceding screenshot, three network interfaces and one load balancer are connected to the virtual network. Any new resources that you create and connect to the virtual network are listed. If you delete a resource that was connected to the virtual network, it no longer appear in the list.
- **Subnets:** A list of subnets that exist within the virtual network is shown. To learn how to add and remove a subnet, see [Manage subnets](#).
- **DNS servers:** You can specify whether the Azure internal DNS server or a custom DNS server provides name resolution for devices that are connected to the virtual network. When you create a virtual network by using the Azure portal, Azure's DNS servers are used for name resolution within a virtual network, by default. To modify the DNS servers, complete the steps in [Change DNS servers](#) in this article.
- **Peerings:** If there are existing peerings in the subscription, they are listed here. You can view settings for existing peerings, or create, change, or delete peerings. To learn more about peerings, see [Virtual network peering](#).
- **Properties:** Displays settings about the virtual network, including the virtual network's resource ID and the Azure subscription it is in.
- **Diagram:** The diagram provides a visual representation of all devices that are connected to the virtual network. The diagram has some key information about the devices. To manage a device in this view, in the diagram, select the device.
- **Common Azure settings:** To learn more about common Azure settings, see the following information:
 - [Activity log](#)
 - [Access control \(IAM\)](#)
 - [Tags](#)
 - [Locks](#)
 - [Automation script](#)

Commands

- Azure CLI: `az network vnet show`
- PowerShell: `Get-AzVirtualNetwork`

Add or remove an address range

You can add and remove address ranges for a virtual network. An address range must be specified in CIDR notation, and cannot overlap with other address ranges within the same virtual network. The address ranges you define can be public or private (RFC 1918). Whether you define the address range as public or private, the address range is reachable only from within the virtual network, from interconnected virtual networks, and from any on-premises networks that you have connected to the virtual network.

You can decrease the address range for a virtual network as long as it still includes the ranges of any associated subnets. Additionally, you can extend the address range, for example, changing a /16 to /8.

You cannot add the following address ranges:

- 224.0.0.0/4 (Multicast)
- 255.255.255.255/32 (Broadcast)

- 127.0.0.0/8 (Loopback)
- 169.254.0.0/16 (Link-local)
- 168.63.129.16/32 (Internal DNS, DHCP, and Azure Load Balancer [health probe](#))

To add or remove an address range:

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it.
2. From the list of virtual networks, select the virtual network for which you want to add or remove an address range.
3. Select **Address space**, under **SETTINGS**.
4. Complete one of the following options:
 - **Add an address range:** Enter the new address range. The address range cannot overlap with an existing address range that is defined for the virtual network.
 - **Remove an address range:** On the right of the address range you want to remove, select ..., then select **Remove**. If a subnet exists in the address range, you cannot remove the address range. To remove an address range, you must first delete any subnets (and any resources in the subnets) that exist in the address range.
5. Select **Save**.

Commands

- Azure CLI: [az network vnet update](#)
- PowerShell: [Set-AzVirtualNetwork](#)

Change DNS servers

All VMs that are connected to the virtual network register with the DNS servers that you specify for the virtual network. They also use the specified DNS server for name resolution. Each network interface (NIC) in a VM can have its own DNS server settings. If a NIC has its own DNS server settings, they override the DNS server settings for the virtual network. To learn more about NIC DNS settings, see [Network interface tasks and settings](#). To learn more about name resolution for VMs and role instances in Azure Cloud Services, see [Name resolution for VMs and role instances](#). To add, change, or remove a DNS server:

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it.
2. From the list of virtual networks, select the virtual network for which you want to change DNS servers for.
3. Select **DNS servers**, under **SETTINGS**.
4. Select one of the following options:
 - **Default (Azure-provided):** All resource names and private IP addresses are automatically registered to the Azure DNS servers. You can resolve names between any resources that are connected to the same virtual network. You cannot use this option to resolve names across virtual networks. To resolve names across virtual networks, you must use a custom DNS server.
 - **Custom:** You can add one or more servers, up to the Azure limit for a virtual network. To learn more about DNS server limits, see [Azure limits](#). You have the following options:
 - **Add an address:** Adds the server to your virtual network DNS servers list. This option also registers the DNS server with Azure. If you've already registered a DNS server with Azure, you can select that DNS server in the list.
 - **Remove an address:** Next to the server that you want to remove, select ..., then **Remove**. Deleting the server removes the server only from this virtual network list. The DNS server remains registered in Azure for your other virtual networks to use.
 - **Reorder DNS server addresses:** It's important to verify that you list your DNS servers in the

correct order for your environment. DNS server lists are used in the order that they are specified. They do not work as a round-robin setup. If the first DNS server in the list can be reached, the client uses that DNS server, regardless of whether the DNS server is functioning properly. Remove all the DNS servers that are listed, and then add them back in the order that you want.

- **Change an address:** Highlight the DNS server in the list, and then enter the new address.

5. Select **Save**.

6. Restart the VMs that are connected to the virtual network, so they are assigned the new DNS server settings. VMs continue to use their current DNS settings until they are restarted.

Commands

- Azure CLI: [az network vnet update](#)
- PowerShell: [Set-AzVirtualNetwork](#)

Delete a virtual network

You can delete a virtual network only if there are no resources connected to it. If there are resources connected to any subnet within the virtual network, you must first delete the resources that are connected to all subnets within the virtual network. The steps you take to delete a resource vary depending on the resource. To learn how to delete resources that are connected to subnets, read the documentation for each resource type you want to delete. To delete a virtual network:

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it.
2. From the list of virtual networks, select the virtual network you want to delete.
3. Confirm that there are no devices connected to the virtual network by selecting **Connected devices**, under **SETTINGS**. If there are connected devices, you must delete them before you can delete the virtual network. If there are no connected devices, select **Overview**.
4. Select **Delete**.
5. To confirm the deletion of the virtual network, select **Yes**.

Commands

- Azure CLI: [azure network vnet delete](#)
- PowerShell: [Remove-AzVirtualNetwork](#)

Permissions

To perform tasks on virtual networks, your account must be assigned to the [network contributor](#) role or to a [custom](#) role that is assigned the appropriate actions listed in the following table:

ACTION	NAME
Microsoft.Network/virtualNetworks/read	Read a virtual Network
Microsoft.Network/virtualNetworks/write	Create or update a virtual network
Microsoft.Network/virtualNetworks/delete	Delete a virtual network

Next steps

- Create a virtual network using [PowerShell](#) or [Azure CLI](#) sample scripts, or using Azure [Resource Manager templates](#)

- Create and assign [Azure Policy definitions](#) for virtual networks

Add, change, or delete a virtual network subnet

2/18/2021 • 8 minutes to read • [Edit Online](#)

Learn how to add, change, or delete a virtual network subnet. All Azure resources deployed into a virtual network are deployed into a subnet within a virtual network. If you're new to virtual networks, you can learn more about them in the [Virtual network overview](#) or by completing a [quickstart](#). To learn more about managing a virtual network, see [Create, change, or delete a virtual network](#).

Before you begin

If you don't have one, set up an Azure account with an active subscription. [Create an account for free](#). Then complete one of these tasks before starting steps in any section of this article:

- **Portal users:** Sign in to the [Azure portal](#) with your Azure account.
- **PowerShell users:** Either run the commands in the [Azure Cloud Shell](#), or run PowerShell from your computer. 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. In the Azure Cloud Shell browser tab, find the **Select environment** dropdown list, then choose **PowerShell** if it isn't already selected.

If you're running PowerShell locally, use Azure PowerShell module version 1.0.0 or later. Run

```
Get-Module -ListAvailable Az.Network
```

 to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). Also run

```
Connect-AzAccount
```

 to create a connection with Azure.

- **Azure Command-line interface (CLI) users:** Either run the commands in the [Azure Cloud Shell](#), or run the CLI from your computer. Use Azure CLI version 2.0.31 or later if you're running the Azure CLI locally. Run

```
az --version
```

 to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). Also run

```
az login
```

 to create a connection with Azure.

The account you sign in to, or connect to Azure with, must be assigned to the [Network contributor role](#) role or to a [Custom role](#) that's assigned the appropriate actions listed in [Permissions](#).

Add a subnet

1. Go to the [Azure portal](#) to view your virtual networks. Search for and select **Virtual networks**.
2. Select the name of the virtual network you want to add a subnet to.
3. From **Settings**, select **Subnets > Subnet**.
4. In the **Add subnet** dialog box, enter values for the following settings:

SETTING	DESCRIPTION
Name	The name must be unique within the virtual network. For maximum compatibility with other Azure services, we recommend using a letter as the first character of the name. For example, Azure Application Gateway won't deploy into a subnet that has a name that starts with a number.

SETTING	DESCRIPTION
Address range	<p>The range must be unique within the address space for the virtual network. The range can't overlap with other subnet address ranges within the virtual network. The address space must be specified by using Classless Inter-Domain Routing (CIDR) notation.</p> <p>For example, in a virtual network with address space <code>10.0.0.0/16</code>, you might define a subnet address space of <code>10.0.0.0/22</code>. The smallest range you can specify is <code>/29</code>, which provides eight IP addresses for the subnet. Azure reserves the first and last address in each subnet for protocol conformance. Three additional addresses are reserved for Azure service usage. As a result, defining a subnet with a <code>/29</code> address range results in three usable IP addresses in the subnet.</p> <p>If you plan to connect a virtual network to a VPN gateway, you must create a gateway subnet. Learn more about specific address range considerations for gateway subnets. You can change the address range after the subnet is added, under specific conditions. To learn how to change a subnet address range, see Change subnet settings.</p>
Network security group	To filter inbound and outbound network traffic for the subnet, you may associate an existing network security group to a subnet. The network security group must exist in the same subscription and location as the virtual network. Learn more about network security groups and how to create a network security group .
Route table	To control network traffic routing to other networks, you may optionally associate an existing route table to a subnet. The route table must exist in the same subscription and location as the virtual network. Learn more about Azure routing and how to create a route table .

SETTING	DESCRIPTION
Service endpoints	<p>A subnet may optionally have one or more service endpoints enabled for it. To enable a service endpoint for a service, select the service or services that you want to enable service endpoints for from the Services list. Azure configures the location automatically for an endpoint. By default, Azure configures the service endpoints for the virtual network's region. To support regional failover scenarios, Azure automatically configures endpoints to Azure paired regions for Azure Storage.</p> <p>To remove a service endpoint, unselect the service you want to remove the service endpoint for. To learn more about service endpoints, and the services they can be enabled for, see Virtual network service endpoints. Once you enable a service endpoint for a service, you must also enable network access for the subnet for a resource created with the service. For example, if you enable the service endpoint for Microsoft.Storage, you must also enable network access to all Azure Storage accounts you want to grant network access to. To enable network access to subnets that a service endpoint is enabled for, see the documentation for the individual service you enabled the service endpoint for.</p> <p>To validate that a service endpoint is enabled for a subnet, view the effective routes for any network interface in the subnet. When you configure an endpoint, you see a <i>default</i> route with the address prefixes of the service, and a next hop type of VirtualNetworkServiceEndpoint. To learn more about routing, see Virtual network traffic routing.</p>
Subnet delegation	<p>A subnet may optionally have one or more delegations enabled for it. Subnet delegation gives explicit permissions to the service to create service-specific resources in the subnet using a unique identifier during service deployment. To delegate for a service, select the service you want to delegate to from the Services list.</p>

5. To add the subnet to the virtual network that you selected, select **OK**.

Commands

TOOL	COMMAND
Azure CLI	az network vnet subnet create
PowerShell	Add-AzVirtualNetworkSubnetConfig

Change subnet settings

1. Go to the [Azure portal](#) to view your virtual networks. Search for and select **Virtual networks**.
2. Select the name of the virtual network containing the subnet you want to change.
3. From **Settings**, select **Subnets**.

4. In the list of subnets, select the subnet you want to change settings for.

5. In the subnet page, change any of the following settings:

SETTING	DESCRIPTION
Address range	If no resources are deployed within the subnet, you can change the address range. If any resources exist in the subnet, you must either move the resources to another subnet, or delete them from the subnet first. The steps you take to move or delete a resource vary depending on the resource. To learn how to move or delete resources that are in subnets, read the documentation for each of those resource types. See the constraints for Address range in step 4 of Add a subnet .
Users	You can control access to the subnet by using built-in roles or your own custom roles. To learn more about assigning roles and users to access the subnet, see Assign Azure roles .
Network security group and Route table	See step 4 of Add a subnet .
Service endpoints	<p>See service endpoints in step 4 of Add a subnet. When enabling a service endpoint for an existing subnet, ensure that no critical tasks are running on any resource in the subnet. Service endpoints switch routes on every network interface in the subnet. The service endpoints go from using the default route with the <i>0.0.0.0/0</i> address prefix and next hop type of <i>Internet</i>, to using a new route with the address prefixes of the service and a next hop type of <i>VirtualNetworkServiceEndpoint</i>.</p> <p>During the switch, any open TCP connections may be terminated. The service endpoint isn't enabled until traffic flows to the service for all network interfaces are updated with the new route. To learn more about routing, see Virtual network traffic routing.</p>
Subnet delegation	See service endpoints in step 4 of Add a subnet . Subnet delegation can be modified to zero or multiple delegations enabled for it. If a resource for a service is already deployed in the subnet, subnet delegation can't be added or removed until all the resources for the service are removed. To delegate for a different service, select the service you want to delegate to from the Services list.

6. Select **Save**.

Commands

TOOL	COMMAND
Azure CLI	<code>az network vnet subnet update</code>
PowerShell	<code>Set-AzVirtualNetworkSubnetConfig</code>

Delete a subnet

You can delete a subnet only if there are no resources in the subnet. If resources are in the subnet, you must delete those resources before you can delete the subnet. The steps you take to delete a resource vary depending on the resource. To learn how to delete resources that are in subnets, read the documentation for each of those resource types.

1. Go to the [Azure portal](#) to view your virtual networks. Search for and select **Virtual networks**.
2. Select the name of the virtual network containing the subnet you want to delete.
3. From **Settings**, select **Subnets**.
4. In the list of subnets, select the subnet you want to delete.
5. Select **Delete**, and then select **Yes** in the confirmation dialog box.

Commands

TOOL	COMMAND
Azure CLI	az network vnet subnet delete
PowerShell	Remove-AzVirtualNetworkSubnetConfig

Permissions

To do tasks on subnets, your account must be assigned to the [Network contributor role](#) or to a [Custom role](#) that's assigned the appropriate actions in the following table:

ACTION	NAME
Microsoft.Network/virtualNetworks/subnets/read	Read a virtual network subnet
Microsoft.Network/virtualNetworks/subnets/write	Create or update a virtual network subnet
Microsoft.Network/virtualNetworks/subnets/delete	Delete a virtual network subnet
Microsoft.Network/virtualNetworks/subnets/join/action	Join a virtual network
Microsoft.Network/virtualNetworks/subnets/joinViaServiceEndpoint/action	Enable a service endpoint for a subnet
Microsoft.Network/virtualNetworks/subnets/virtualMachines/read	Get the virtual machines in a subnet

Next steps

- Create a virtual network and subnets using [PowerShell](#) or [Azure CLI](#) sample scripts, or using Azure [Resource Manager templates](#)
- Create and assign [Azure Policy definitions](#) for virtual networks

Add or remove a subnet delegation

2/18/2021 • 5 minutes to read • [Edit Online](#)

Subnet delegation gives explicit permissions to the service to create service-specific resources in the subnet using a unique identifier when deploying the service. This article describes how to add or remove a delegated subnet for an Azure service.

Portal

Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

Create the virtual network

In this section, you create a virtual network and the subnet that you'll later delegate to an Azure service.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network**.
2. In **Create virtual network**, enter or select this information:

SETTING	VALUE
Name	Enter <i>MyVirtualNetwork</i> .
Address space	Enter <i>10.0.0.0/16</i> .
Subscription	Select your subscription.
Resource group	Select Create new , enter <i>myResourceGroup</i> , then select OK .
Location	Select EastUS .
Subnet - Name	Enter <i>mySubnet</i> .
Subnet - Address range	Enter <i>10.0.0.0/24</i> .

3. Leave the rest as default, and then select **Create**.

Permissions

If you didn't create the subnet you would like to delegate to an Azure service, you need the following permission: `Microsoft.Network/virtualNetworks/subnets/write`.

The built-in **Network Contributor** role also contains the necessary permissions.

Delegate a subnet to an Azure service

In this section, you delegate the subnet that you created in the preceding section to an Azure service.

1. In the portal's search bar, enter *myVirtualNetwork*. When **myVirtualNetwork** appears in the search results, select it.
2. In the search results, select *myVirtualNetwork*.

3. Select **Subnets**, under **SETTINGS**, and then select **mySubnet**.
4. On the *mySubnet* page, for the **Subnet delegation** list, select from the services listed under **Delegate subnet to a service** (for example, **Microsoft.DBforPostgreSQL/serversv2**).

Remove subnet delegation from an Azure service

1. In the portal's search bar, enter *myVirtualNetwork*. When **myVirtualNetwork** appears in the search results, select it.
2. In the search results, select *myVirtualNetwork*.
3. Select **Subnets**, under **SETTINGS**, and then select **mySubnet**.
4. In *mySubnet* page, for the **Subnet delegation** list, select **None** from the services listed under **Delegate subnet to a service**.

Azure CLI

Prepare your environment for the Azure CLI.

- Use the Bash environment in [Azure Cloud Shell](#).
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
 - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
 - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
 - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a resource group

Create a resource group with [az group create](#). An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named **myResourceGroup** in the **eastus** location:

```
az group create \
--name myResourceGroup \
--location eastus
```

Create a virtual network

Create a virtual network named **myVnet** with a subnet named **mySubnet** in the **myResourceGroup** using [az network vnet create](#).

```
az network vnet create \
--resource-group myResourceGroup \
--location eastus \
--name myVnet \
--address-prefix 10.0.0.0/16 \
--subnet-name mySubnet \
--subnet-prefix 10.0.0.0/24
```

Permissions

If you didn't create the subnet you would like to delegate to an Azure service, you need the following permission: `Microsoft.Network/virtualNetworks/subnets/write`.

The built-in [Network Contributor](#) role also contains the necessary permissions.

Delegate a subnet to an Azure service

In this section, you delegate the subnet that you created in the preceding section to an Azure service.

Use [az network vnet subnet update](#) to update the subnet named `mySubnet` with a delegation to an Azure service. In this example `Microsoft.DBforPostgreSQL/serversv2` is used for the example delegation:

```
az network vnet subnet update \
--resource-group myResourceGroup \
--name mySubnet \
--vnet-name myVnet \
--delegations Microsoft.DBforPostgreSQL/serversv2
```

To verify the delegation was applied, use [az network vnet subnet show](#). Verify the service is delegated to the subnet under the property `serviceName`:

```
az network vnet subnet show \
--resource-group myResourceGroup \
--name mySubnet \
--vnet-name myVnet \
--query delegations
```

```
[  
 {  
   "actions": [  
     "Microsoft.Network/virtualNetworks/subnets/join/action"  
   ],  
   "etag": "W/\\"8a8bf16a-38cf-409f-9434-fe3b5ab9ae54\\\"",  
   "id": "/subscriptions/3bf09329-ca61-4fee-88cb-  
7e30b9ee305b/resourceGroups/myResourceGroup/providers/Microsoft.Network/virtualNetworks/myVnet/subnets/mySub-  
net/delegations/0",  
   "name": "0",  
   "provisioningState": "Succeeded",  
   "resourceGroup": "myResourceGroup",  
   "serviceName": "Microsoft.DBforPostgreSQL/serversv2",  
   "type": "Microsoft.Network/virtualNetworks/subnets/delegations"  
 }  
]
```

Remove subnet delegation from an Azure service

Use [az network vnet subnet update](#) to remove the delegation from the subnet named `mySubnet`:

```
az network vnet subnet update \
--resource-group myResourceGroup \
--name mySubnet \
--vnet-name myVnet \
--remove delegations
```

To verify the delegation was removed, use [az network vnet subnet show](#). Verify the service is removed from the subnet under the property **serviceName**:

```
az network vnet subnet show \
--resource-group myResourceGroup \
--name mySubnet \
--vnet-name myVnet \
--query delegations
```

Output from command is a null bracket:

```
[]
```

Azure PowerShell

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Connect to Azure

```
Connect-AzAccount
```

Create a resource group

Create a resource group with [New-AzResourceGroup](#). An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named *myResourceGroup* in the *eastus* location:

```
New-AzResourceGroup -Name myResourceGroup -Location eastus
```

Create virtual network

Create a virtual network named **myVnet** with a subnet named **mySubnet** using [New-AzVirtualNetworkSubnetConfig](#) in the **myResourceGroup** using [New-AzVirtualNetwork](#). The IP address space for the virtual network is **10.0.0.0/16**. The subnet within the virtual network is **10.0.0.0/24**.

```
$subnet = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix "10.0.0.0/24"

New-AzVirtualNetwork -Name myVnet -ResourceGroupName myResourceGroup -Location eastus -AddressPrefix
"10.0.0.0/16" -Subnet $subnet
```

Permissions

If you didn't create the subnet you would like to delegate to an Azure service, you need the following

permission: `Microsoft.Network/virtualNetworks/subnets/write`.

The built-in [Network Contributor](#) role also contains the necessary permissions.

Delegate a subnet to an Azure service

In this section, you delegate the subnet that you created in the preceding section to an Azure service.

Use [Add-AzDelegation](#) to update the subnet named `mySubnet` with a delegation named `myDelegation` to an Azure service. In this example `Microsoft.DBforPostgreSQL/serversv2` is used for the example delegation:

```
$vnet = Get-AzVirtualNetwork -Name "myVNet" -ResourceGroupName "myResourceGroup"
$subnet = Get-AzVirtualNetworkSubnetConfig -Name "mySubnet" -VirtualNetwork $vnet
$subnet = Add-AzDelegation -Name "myDelegation" -ServiceName "Microsoft.DBforPostgreSQL/serversv2" -Subnet
$subnet
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

Use [Get-AzDelegation](#) to verify the delegation:

```
$subnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName "myResourceGroup" | Get-
AzVirtualNetworkSubnetConfig -Name "mySubnet"
Get-AzDelegation -Name "myDelegation" -Subnet $subnet

ProvisioningState : Succeeded
ServiceName      : Microsoft.DBforPostgreSQL/serversv2
Actions          : {Microsoft.Network/virtualNetworks/subnets/join/action}
Name             : myDelegation
Etag             : W/"9cba4b0e-2ceb-444b-b553-454f8da07d8a"
Id               : /subscriptions/3bf09329-ca61-4fee-88cb-
7e30b9ee305b/resourceGroups/myResourceGroup/providers/Microsoft.Network/virtualNetworks/myVnet/subnets/mySub-
net/delegations/myDelegation
```

Remove subnet delegation from an Azure service

Use [Remove-AzDelegation](#) to remove the delegation from the subnet named `mySubnet`:

```
$vnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName "myResourceGroup"
$subnet = Get-AzVirtualNetworkSubnetConfig -Name "mySubnet" -VirtualNetwork $vnet
$subnet = Remove-AzDelegation -Name "myDelegation" -Subnet $subnet
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

Use [Get-AzDelegation](#) to verify the delegation was removed:

```
$subnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName "myResourceGroup" | Get-
AzVirtualNetworkSubnetConfig -Name "mySubnet"
Get-AzDelegation -Name "myDelegation" -Subnet $subnet

Get-AzDelegation: Sequence contains no matching element
```

Next steps

- Learn how to [manage subnets in Azure](#).

Connect virtual networks with virtual network peering using PowerShell

2/18/2021 • 6 minutes to read • [Edit Online](#)

You can connect virtual networks to each other with virtual network peering. Once virtual networks are peered, resources in both virtual networks are able to communicate with each other, with the same latency and bandwidth as if the resources were in the same virtual network. In this article, you learn how to:

- Create two virtual networks
- Connect two virtual networks with a virtual network peering
- Deploy a virtual machine (VM) into each virtual network
- Communicate between VMs

If you don't have an Azure subscription, create a [free account](#) before you begin.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the Copy button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select Enter to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

Create virtual networks

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup* in the *eastus* location.

```
New-AzResourceGroup -ResourceGroupName myResourceGroup -Location EastUS
```

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVirtualNetwork1* with the address prefix *10.0.0.0/16*.

```
$virtualNetwork1 = New-AzVirtualNetwork `  
    -ResourceGroupName myResourceGroup `  
    -Location EastUS `  
    -Name myVirtualNetwork1 `  
    -AddressPrefix 10.0.0.0/16
```

Create a subnet configuration with [New-AzVirtualNetworkSubnetConfig](#). The following example creates a subnet configuration with a *10.0.0.0/24* address prefix:

```
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `  
    -Name Subnet1 `  
    -AddressPrefix 10.0.0.0/24 `  
    -VirtualNetwork $virtualNetwork1
```

Write the subnet configuration to the virtual network with [Set-AzVirtualNetwork](#), which creates the subnet:

```
$virtualNetwork1 | Set-AzVirtualNetwork
```

Create a virtual network with a *10.1.0.0/16* address prefix and one subnet:

```
# Create the virtual network.  
$virtualNetwork2 = New-AzVirtualNetwork `  
    -ResourceGroupName myResourceGroup `  
    -Location EastUS `  
    -Name myVirtualNetwork2 `  
    -AddressPrefix 10.1.0.0/16  
  
# Create the subnet configuration.  
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `  
    -Name Subnet1 `  
    -AddressPrefix 10.1.0.0/24 `  
    -VirtualNetwork $virtualNetwork2  
  
# Write the subnet configuration to the virtual network.  
$virtualNetwork2 | Set-AzVirtualNetwork
```

Peer virtual networks

Create a peering with [Add-AzVirtualNetworkPeering](#). The following example peers *myVirtualNetwork1* to *myVirtualNetwork2*.

```
Add-AzVirtualNetworkPeering ` 
-Name myVirtualNetwork1-myVirtualNetwork2 ` 
-VirtualNetwork $virtualNetwork1 ` 
-RemoteVirtualNetworkId $virtualNetwork2.Id
```

In the output returned after the previous command executes, you see that the **PeeringState** is *Initiated*. The peering remains in the *Initiated* state until you create the peering from *myVirtualNetwork2* to *myVirtualNetwork1*. Create a peering from *myVirtualNetwork2* to *myVirtualNetwork1*.

```
Add-AzVirtualNetworkPeering ` 
-Name myVirtualNetwork2-myVirtualNetwork1 ` 
-VirtualNetwork $virtualNetwork2 ` 
-RemoteVirtualNetworkId $virtualNetwork1.Id
```

In the output returned after the previous command executes, you see that the **PeeringState** is *Connected*. Azure also changed the peering state of the *myVirtualNetwork1-myVirtualNetwork2* peering to *Connected*. Confirm that the peering state for the *myVirtualNetwork1-myVirtualNetwork2* peering changed to *Connected* with [Get-AzVirtualNetworkPeering](#).

```
Get-AzVirtualNetworkPeering ` 
-ResourceGroupName myResourceGroup ` 
-VirtualNetworkName myVirtualNetwork1 ` 
| Select PeeringState
```

Resources in one virtual network cannot communicate with resources in the other virtual network until the **PeeringState** for the peerings in both virtual networks is *Connected*.

Create virtual machines

Create a VM in each virtual network so that you can communicate between them in a later step.

Create the first VM

Create a VM with [New-AzVm](#). The following example creates a VM named *myVm1* in the *myVirtualNetwork1* virtual network. The `-AsJob` option creates the VM in the background, so you can continue to the next step. When prompted, enter the user name and password you want to log in to the VM with.

```
New-AzVm ` 
-ResourceGroupName "myResourceGroup" ` 
-Location "East US" ` 
-VirtualNetworkName "myVirtualNetwork1" ` 
-SubnetName "Subnet1" ` 
-ImageName "Win2016Datacenter" ` 
-Name "myVm1" ` 
-AsJob
```

Create the second VM

```
New-AzVm ` 
-ResourceGroupName "myResourceGroup" ` 
-Location "East US" ` 
-VirtualNetworkName "myVirtualNetwork2" ` 
-SubnetName "Subnet1" ` 
-ImageName "Win2016Datacenter" ` 
-Name "myVm2"
```

The VM takes a few minutes to create. Do not continue with later steps until Azure creates the VM and returns output to PowerShell.

Communicate between VMs

You can connect to a VM's public IP address from the internet. Use [Get-AzPublicIpAddress](#) to return the public IP address of a VM. The following example returns the public IP address of the *myVm1* VM:

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

Use the following command to create a remote desktop session with the *myVm1* VM from your local computer. Replace `<publicIpAddress>` with the IP address returned from the previous command.

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

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

On the *myVm1* VM, enable the Internet Control Message Protocol (ICMP) through the Windows firewall so you can ping this VM from *myVm2* in a later step, using PowerShell:

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

Though ping is used to communicate between VMs in this article, allowing ICMP through the Windows Firewall for production deployments is not recommended.

To connect to the *myVm2* VM, enter the following command from a command prompt on the *myVm1* VM:

```
mstsc /v:10.1.0.4
```

Since you enabled ping on *myVm1*, you can now ping it by IP address from a command prompt on the *myVm2* VM:

```
ping 10.0.0.4
```

You receive four replies. Disconnect your RDP sessions to both *myVm1* and *myVm2*.

Clean up resources

When no longer needed, use [Remove-AzResourcegroup](#) to remove the resource group and all of the resources it contains.

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

Next steps

In this article, you learned how to connect two networks in the same Azure region, with virtual network peering.

You can also peer virtual networks in different [supported regions](#) and in [different Azure subscriptions](#), as well as create [hub and spoke network designs](#) with peering. To learn more about virtual network peering, see [Virtual network peering overview](#) and [Manage virtual network peerings](#).

You can [connect your own computer to a virtual network](#) through a VPN, and interact with resources in a virtual network, or in peered virtual networks. For reusable scripts to complete many of the tasks covered in the virtual network articles, see [script samples](#).

Connect virtual networks with virtual network peering using the Azure CLI

2/18/2021 • 5 minutes to read • [Edit Online](#)

You can connect virtual networks to each other with virtual network peering. Once virtual networks are peered, resources in both virtual networks are able to communicate with each other, with the same latency and bandwidth as if the resources were in the same virtual network. In this article, you learn how to:

- Create two virtual networks
- Connect two virtual networks with a virtual network peering
- Deploy a virtual machine (VM) into each virtual network
- Communicate between VMs

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
 - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
 - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
 - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create virtual networks

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with [az group create](#). The following example creates a resource group named `myResourceGroup` in the `eastus` location.

```
az group create --name myResourceGroup --location eastus
```

Create a virtual network with [az network vnet create](#). The following example creates a virtual network named `myVirtualNetwork1` with the address prefix `10.0.0.0/16`.

```
az network vnet create \
--name myVirtualNetwork1 \
--resource-group myResourceGroup \
--address-prefixes 10.0.0.0/16 \
--subnet-name Subnet1 \
--subnet-prefix 10.0.0.0/24
```

Create a virtual network named *myVirtualNetwork2* with the address prefix *10.1.0.0/16*:

```
az network vnet create \
--name myVirtualNetwork2 \
--resource-group myResourceGroup \
--address-prefixes 10.1.0.0/16 \
--subnet-name Subnet1 \
--subnet-prefix 10.1.0.0/24
```

Peer virtual networks

Peerings are established between virtual network IDs, so you must first get the ID of each virtual network with [az network vnet show](#) and store the ID in a variable.

```
# Get the id for myVirtualNetwork1.
vNet1Id=$(az network vnet show \
--resource-group myResourceGroup \
--name myVirtualNetwork1 \
--query id --out tsv)

# Get the id for myVirtualNetwork2.
vNet2Id=$(az network vnet show \
--resource-group myResourceGroup \
--name myVirtualNetwork2 \
--query id \
--out tsv)
```

Create a peering from *myVirtualNetwork1* to *myVirtualNetwork2* with [az network vnet peering create](#). If the `--allow-vnet-access` parameter is not specified, a peering is established, but no communication can flow through it.

```
az network vnet peering create \
--name myVirtualNetwork1-myVirtualNetwork2 \
--resource-group myResourceGroup \
--vnet-name myVirtualNetwork1 \
--remote-vnet $vNet2Id \
--allow-vnet-access
```

In the output returned after the previous command executes, you see that the **peeringState** is *Initiated*. The peering remains in the *Initiated* state until you create the peering from *myVirtualNetwork2* to *myVirtualNetwork1*. Create a peering from *myVirtualNetwork2* to *myVirtualNetwork1*.

```
az network vnet peering create \
--name myVirtualNetwork2-myVirtualNetwork1 \
--resource-group myResourceGroup \
--vnet-name myVirtualNetwork2 \
--remote-vnet $vNet1Id \
--allow-vnet-access
```

In the output returned after the previous command executes, you see that the **peeringState** is *Connected*.

Azure also changed the peering state of the *myVirtualNetwork1-myVirtualNetwork2* peering to *Connected*. Confirm that the peering state for the *myVirtualNetwork1-myVirtualNetwork2* peering changed to *Connected* with [az network vnet peering show](#).

```
az network vnet peering show \
--name myVirtualNetwork1-myVirtualNetwork2 \
--resource-group myResourceGroup \
--vnet-name myVirtualNetwork1 \
--query peeringState
```

Resources in one virtual network cannot communicate with resources in the other virtual network until the **peeringState** for the peerings in both virtual networks is *Connected*.

Create virtual machines

Create a VM in each virtual network so that you can communicate between them in a later step.

Create the first VM

Create a VM with [az vm create](#). The following example creates a VM named *myVm1* in the *myVirtualNetwork1* virtual network. If SSH keys do not already exist in a default key location, the command creates them. To use a specific set of keys, use the `--ssh-key-value` option. The `--no-wait` option creates the VM in the background, so you can continue to the next step.

```
az vm create \
--resource-group myResourceGroup \
--name myVm1 \
--image UbuntuLTS \
--vnet-name myVirtualNetwork1 \
--subnet Subnet1 \
--generate-ssh-keys \
--no-wait
```

Create the second VM

Create a VM in the *myVirtualNetwork2* virtual network.

```
az vm create \
--resource-group myResourceGroup \
--name myVm2 \
--image UbuntuLTS \
--vnet-name myVirtualNetwork2 \
--subnet Subnet1 \
--generate-ssh-keys
```

The VM takes a few minutes to create. After the VM is created, the Azure CLI shows information similar to the following example:

```
{  
    "fqdns": "",  
    "id": "/subscriptions/00000000-0000-0000-0000-  
0000000000/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVm2",  
    "location": "eastus",  
    "macAddress": "00-0D-3A-23-9A-49",  
    "powerState": "VM running",  
    "privateIpAddress": "10.1.0.4",  
    "publicIpAddress": "13.90.242.231",  
    "resourceGroup": "myResourceGroup"  
}
```

Take note of the **publicIpAddress**. This address is used to access the VM from the internet in a later step.

Communicate between VMs

Use the following command to create an SSH session with the *myVm2* VM. Replace <publicIpAddress> with the public IP address of your VM. In the previous example, the public IP address is *13.90.242.231*.

```
ssh <publicIpAddress>
```

Ping the VM in *myVirtualNetwork1*.

```
ping 10.0.0.4 -c 4
```

You receive four replies.

Close the SSH session to the *myVm2* VM.

Clean up resources

When no longer needed, use [az group delete](#) to remove the resource group and all of the resources it contains.

```
az group delete --name myResourceGroup --yes
```

Next steps

In this article, you learned how to connect two networks in the same Azure region, with virtual network peering. You can also peer virtual networks in different [supported regions](#) and in [different Azure subscriptions](#), as well as create [hub and spoke network designs](#) with peering. To learn more about virtual network peering, see [Virtual network peering overview](#) and [Manage virtual network peerings](#).

You can [connect your own computer to a virtual network](#) through a VPN, and interact with resources in a virtual network, or in peered virtual networks. For reusable scripts to complete many of the tasks covered in the virtual network articles, see [script samples](#).

Create a virtual network peering - Resource Manager, different subscriptions and Azure Active Directory tenants

2/18/2021 • 16 minutes to read • [Edit Online](#)

In this tutorial, you learn to create a virtual network peering between virtual networks created through Resource Manager. The virtual networks exist in different subscriptions that may belong to different Azure Active Directory (Azure AD) tenants. Peering two virtual networks enables resources in different virtual networks to communicate with each other with the same bandwidth and latency as though the resources were in the same virtual network. Learn more about [Virtual network peering](#).

The steps to create a virtual network peering are different, depending on whether the virtual networks are in the same, or different, subscriptions, and which [Azure deployment model](#) the virtual networks are created through. Learn how to create a virtual network peering in other scenarios by selecting the scenario from the following table:

AZURE DEPLOYMENT MODEL	AZURE SUBSCRIPTION
Both Resource Manager	Same
One Resource Manager, one classic	Same
One Resource Manager, one classic	Different

A virtual network peering cannot be created between two virtual networks deployed through the classic deployment model. If you need to connect virtual networks that were both created through the classic deployment model, you can use an Azure [VPN Gateway](#) to connect the virtual networks.

This tutorial peers virtual networks in the same region. You can also peer virtual networks in different [supported regions](#). It's recommended that you familiarize yourself with the [peering requirements and constraints](#) before peering virtual networks.

You can use the [Azure portal](#), the Azure [command-line interface](#) (CLI), Azure [PowerShell](#), or an [Azure Resource Manager template](#) to create a virtual network peering. Select any of the previous tool links to go directly to the steps for creating a virtual network peering using your tool of choice.

If the virtual networks are in different subscriptions, and the subscriptions are associated with different Azure Active Directory tenants, complete the following steps before continuing:

1. Add the user from each Active Directory tenant as a [guest user](#) in the opposite Azure Active Directory tenant.
2. Each user must accept the guest user invitation from the opposite Azure Active Directory tenant.

Create peering - Azure portal

The following steps use different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of the portal, and skip the steps for assigning another user permissions to the virtual networks.

1. Log in to the [Azure portal](#) as *UserA*. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).

2. Select + **Create a resource**, select **Networking**, and then select **Virtual network**.
3. Select or enter the following example values for the following settings, then select **Create**:
 - **Name:** *myVnetA*
 - **Address space:** *10.0.0.0/16*
 - **Subnet name:** *default*
 - **Subnet address range:** *10.0.0.0/24*
 - **Subscription:** Select subscription A.
 - **Resource group:** Select **Create new** and enter *myResourceGroupA*
 - **Location:** *East US*
4. In the **Search resources** box at the top of the portal, type *myVnetA*. Select **myVnetA** when it appears in the search results.
5. Select **Access control (IAM)** from the vertical list of options on the left side.
6. Under **myVnetA - Access control (IAM)**, select + **Add role assignment**.
7. Select **Network contributor** in the **Role** box.
8. In the **Select** box, select *UserB*, or type UserB's email address to search for it.
9. Select **Save**.
10. Under **myVnetA - Access control (IAM)**, select **Properties** from the vertical list of options on the left side. Copy the RESOURCE ID, which is used in a later step. The resource ID is similar to the following example:

```
/subscriptions/<Subscription Id>/resourceGroups/myResourceGroupA/providers/Microsoft.Network/virtualNetworks/myVnetA
```
11. Log out of the portal as UserA, then log in as UserB.
12. Complete steps 2-3, entering or selecting the following values in step 3:
 - **Name:** *myVnetB*
 - **Address space:** *10.1.0.0/16*
 - **Subnet name:** *default*
 - **Subnet address range:** *10.1.0.0/24*
 - **Subscription:** Select subscription B.
 - **Resource group:** Select **Create new** and enter *myResourceGroupB*
 - **Location:** *East US*
13. In the **Search resources** box at the top of the portal, type *myVnetB*. Select **myVnetB** when it appears in the search results.
14. Under **myVnetB**, select **Properties** from the vertical list of options on the left side. Copy the RESOURCE ID, which is used in a later step. The resource ID is similar to the following example:

```
/subscriptions/<Subscription Id>/resourceGroups/myResourceGroupB/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnetB
```
15. Select **Access control (IAM)** under **myVnetB**, and then complete steps 5-10 for myVnetB, entering **UserA** in step 8.
16. Log out of the portal as UserB and log in as UserA.
17. In the **Search resources** box at the top of the portal, type *myVnetA*. Select **myVnetA** when it appears in the search results.

18. Select **myVnetA**.
19. Under **SETTINGS**, select **Peerings**.
20. Under **myVnetA - Peerings**, select **+ Add**
21. Under **Add peering**, enter, or select, the following options, then select **OK**:
 - **Name:** *myVnetAToMyVnetB*
 - **Virtual network deployment model:** Select **Resource Manager**.
 - **I know my resource ID:** Check this box.
 - **Resource ID:** Enter the resource ID from step 14.
 - **Allow virtual network access:** Ensure that **Enabled** is selected. No other settings are used in this tutorial. To learn about all peering settings, read [Manage virtual network peerings](#).
22. The peering you created appears a short wait after selecting **OK** in the previous step. **Initiated** is listed in the **PEERING STATUS** column for the **myVnetAToMyVnetB** peering you created. You've peered **myVnetA** to **myVnetB**, but now you must peer **myVnetB** to **myVnetA**. The peering must be created in both directions to enable resources in the virtual networks to communicate with each other.
23. Log out of the portal as UserA and log in as UserB.
24. Complete steps 17-21 again for **myVnetB**. In step 21, name the peering *myVnetBToMyVnetA*, select **myVnetA** for **Virtual network**, and enter the ID from step 10 in the **Resource ID** box.
25. A few seconds after selecting **OK** to create the peering for **myVnetB**, the **myVnetBToMyVnetA** peering you just created is listed with **Connected** in the **PEERING STATUS** column.
26. Log out of the portal as UserB and log in as UserA.
27. Complete steps 17-19 again. The **PEERING STATUS** for the **myVnetAToVNetB** peering is now also **Connected**. The peering is successfully established after you see **Connected** in the **PEERING STATUS** column for both virtual networks in the peering. Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).
28. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
29. **Optional:** To delete the resources that you create in this tutorial, complete the steps in the [Delete resources](#) section of this article.

Create peering - Azure CLI

This tutorial uses different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of Azure, and remove the lines of script that create user role assignments. Replace `UserA@azure.com` and `UserB@azure.com` in all of the following scripts with the usernames you're using for UserA and UserB.

The following scripts:

- Requires the Azure CLI version 2.0.4 or later. To find the version, run `az --version`. If you need to upgrade, see [Install Azure CLI](#).
- Works in a Bash shell. For options on running Azure CLI scripts on Windows client, see [Install the Azure CLI on Windows](#).

Instead of installing the CLI and its dependencies, you can use the Azure Cloud Shell. The Azure Cloud Shell is a free Bash shell that you can run directly within the Azure portal. It has the Azure CLI preinstalled and configured to use with your account. Select the Try it button in the script that follows, which invokes a Cloud Shell that you can log in to your Azure account with.

1. Open a CLI session and log in to Azure as UserA using the `azure login` command. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
2. Copy the following script to a text editor on your PC, replace `<SubscriptionA-Id>` with the ID of SubscriptionA, then copy the modified script, paste it in your CLI session, and press `Enter`. If you don't know your subscription Id, enter the `az account show` command. The value for `id` in the output is your subscription Id.

```
# Create a resource group.  
az group create \  
  --name myResourceGroupA \  
  --location eastus  
  
# Create virtual network A.  
az network vnet create \  
  --name myVnetA \  
  --resource-group myResourceGroupA \  
  --location eastus \  
  --address-prefix 10.0.0.0/16  
  
# Assign UserB permissions to virtual network A.  
az role assignment create \  
  --assignee UserB@azure.com \  
  --role "Network Contributor" \  
  --scope /subscriptions/<SubscriptionA-  
Id>/resourceGroups/myResourceGroupA/providers/Microsoft.Network/VirtualNetworks/myVnetA
```

3. Log out of Azure as UserA using the `az logout` command, then log in to Azure as UserB. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
4. Create myVnetB. Copy the script contents in step 2 to a text editor on your PC. Replace `<SubscriptionA-Id>` with the ID of SubscriptionB. Change 10.0.0.0/16 to 10.1.0.0/16, change all As to B, and all Bs to A. Copy the modified script, paste it in to your CLI session, and press `Enter`.
5. Log out of Azure as UserB and log in to Azure as UserA.
6. Create a virtual network peering from myVnetA to myVnetB. Copy the following script contents to a text editor on your PC. Replace `<SubscriptionB-Id>` with the ID of SubscriptionB. To execute the script, copy the modified script, paste it into your CLI session, and press Enter.

```

# Get the id for myVnetA.
vnetAId=$(az network vnet show \
--resource-group myResourceGroupA \
--name myVnetA \
--query id --out tsv)

# Peer myVNetA to myVnetB.
az network vnet peering create \
--name myVnetAToMyVnetB \
--resource-group myResourceGroupA \
--vnet-name myVnetA \
--remote-vnet /subscriptions/<SubscriptionB-  

Id>/resourceGroups/myResourceGroupB/providers/Microsoft.Network/VirtualNetworks/myVnetB \
--allow-vnet-access

```

7. View the peering state of myVnetA.

```

az network vnet peering list \
--resource-group myResourceGroupA \
--vnet-name myVnetA \
--output table

```

The state is **Initiated**. It changes to **Connected** once you create the peering to myVnetA from myVnetB.

8. Log out UserA from Azure and log in to Azure as UserB.

9. Create the peering from myVnetB to myVnetA. Copy the script contents in step 6 to a text editor on your PC. Replace `<SubscriptionB-Id>` with the ID for SubscriptionA and change all As to B and all Bs to A. Once you've made the changes, copy the modified script, paste it into your CLI session, and press `Enter`.
10. View the peering state of myVnetB. Copy the script contents in step 7 to a text editor on your PC. Change A to B for the resource group and virtual network names, copy the script, paste the modified script in to your CLI session, and then press `Enter`. The peering state is **Connected**. The peering state of myVnetA changes to **Connected** after you've created the peering from myVnetB to myVnetA. You can log UserA back in to Azure and complete step 7 again to verify the peering state of myVnetA.

NOTE

The peering is not established until the peering state is **Connected** for both virtual networks.

11. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
12. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

Create peering - PowerShell

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

This tutorial uses different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of Azure, and remove the lines of script that create user role assignments. Replace UserA@azure.com and UserB@azure.com in all of the following scripts with the usernames you're using for UserA and UserB.

1. Confirm that you have Azure PowerShell version 1.0.0 or higher. You can do this by running the `Get-Module -Name Az` command. We recommend installing the latest version of the PowerShell [Az module](#). If you're new to Azure PowerShell, see [Azure PowerShell overview](#).
2. Start a PowerShell session.
3. In PowerShell, log in to Azure as UserA by entering the `Connect-AzAccount` command. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
4. Create a resource group and virtual network A. Copy the following script to a text editor on your PC. Replace `<SubscriptionA-Id>` with the ID of SubscriptionA. If you don't know your subscription Id, enter the `Get-AzSubscription` command to view it. The value for `Id` in the returned output is your subscription ID. To execute the script, copy the modified script, paste it in to PowerShell, and then press `Enter`.

```
# Create a resource group.  
New-AzResourceGroup `  
    -Name MyResourceGroupA `  
    -Location eastus  
  
# Create virtual network A.  
$vNetA = New-AzVirtualNetwork `  
    -ResourceGroupName MyResourceGroupA `  
    -Name 'myVnetA' `  
    -AddressPrefix '10.0.0.0/16' `  
    -Location eastus  
  
# Assign UserB permissions to myVnetA.  
New-AzRoleAssignment `  
    -SignInName UserB@azure.com `  
    -RoleDefinitionName "Network Contributor" `  
    -Scope /subscriptions/<SubscriptionA-  
Id>/resourceGroups/myResourceGroupA/providers/Microsoft.Network/VirtualNetworks/myVnetA
```

5. Log out UserA from Azure and log in UserB. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
6. Copy the script contents in step 4 to a text editor on your PC. Replace `<SubscriptionA-Id>` with the ID for subscription B. Change 10.0.0.0/16 to 10.1.0.0/16. Change all As to B and all Bs to A. To execute the script, copy the modified script, paste into PowerShell, and then press `Enter`.
7. Log out UserB from Azure and log in UserA.
8. Create the peering from myVnetA to myVnetB. Copy the following script to a text editor on your PC. Replace `<SubscriptionB-Id>` with the ID of subscription B. To execute the script, copy the modified script, paste in to PowerShell, and then press `Enter`.

```
# Peer myVnetA to myVnetB.  
$vNetA=Get-AzVirtualNetwork -Name myVnetA -ResourceGroupName myResourceGroupA  
Add-AzVirtualNetworkPeering `  
-Name 'myVnetAToMyVnetB' `  
-VirtualNetwork $vNetA `  
-RemoteVirtualNetworkId "/subscriptions/<SubscriptionB-  
Id>/resourceGroups/myResourceGroupB/providers/Microsoft.Network/virtualNetworks/myVnetB"
```

9. View the peering state of myVnetA.

```
Get-AzVirtualNetworkPeering `  
-ResourceGroupName myResourceGroupA `  
-VirtualNetworkName myVnetA `  
| Format-Table VirtualNetworkName, PeeringState
```

The state is **Initiated**. It changes to **Connected** once you set up the peering to myVnetA from myVnetB.

10. Log out UserA from Azure and log in UserB.

11. Create the peering from myVnetB to myVnetA. Copy the script contents in step 8 to a text editor on your PC. Replace `<SubscriptionB-Id>` with the ID of subscription A and change all As to B and all Bs to A. To execute the script, copy the modified script, paste it in to PowerShell, and then press `Enter`.

12. View the peering state of myVnetB. Copy the script contents in step 9 to a text editor on your PC. Change A to B for the resource group and virtual network names. To execute the script, paste the modified script into PowerShell, and then press `Enter`. The state is **Connected**. The peering state of myVnetA changes to **Connected** after you've created the peering from myVnetB to myVnetA. You can log UserA back in to Azure and complete step 9 again to verify the peering state of myVnetA.

NOTE

The peering is not established until the peering state is **Connected** for both virtual networks.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

13. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
14. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

Create peering - Resource Manager template

1. To create a virtual network and assign the appropriate [permissions](#), complete the steps in the [Portal](#), [Azure CLI](#), or [PowerShell](#) sections of this article.
2. Save the text that follows to a file on your local computer. Replace `<subscription ID>` with UserA's subscription ID. You might save the file as vnetpeeringA.json, for example.

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [
        {
            "apiVersion": "2016-06-01",
            "type": "Microsoft.Network/virtualNetworks/virtualNetworkPeerings",
            "name": "myVnetA/myVnetAToMyVnetB",
            "location": "[resourceGroup().location]",
            "properties": {
                "allowVirtualNetworkAccess": true,
                "allowForwardedTraffic": false,
                "allowGatewayTransit": false,
                "useRemoteGateways": false,
                "remoteVirtualNetwork": {
                    "id": "/subscriptions/<subscription
ID>/resourceGroups/PeeringTest/providers/Microsoft.Network/virtualNetworks/myVnetB"
                }
            }
        }
    ]
}
```

3. Log in to Azure as UserA and deploy the template using the [portal](#), [PowerShell](#), or the [Azure CLI](#). Specify the file name you saved the example json text in step 2 to.
4. Copy the example json from step 2 to a file on your computer and make changes to the lines that begin with:
 - **name:** Change *myVnetA/myVnetAToMyVnetB* to *myVnetB/myVnetBToMyVnetA*.
 - **id:** Replace `<subscription ID>` with UserB's subscription ID and change *myVnetB* to *myVnetA*.
5. Complete step 3 again, logged in to Azure as UserB.
6. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
7. **Optional:** To delete the resources that you create in this tutorial, complete the steps in the [Delete resources](#) section of this article, using either the Azure portal, PowerShell, or the Azure CLI.

Delete resources

When you've finished this tutorial, you might want to delete the resources you created in the tutorial, so you don't incur usage charges. Deleting a resource group also deletes all resources that are in the resource group.

Azure portal

1. Log in to the Azure portal as UserA.
2. In the portal search box, enter **myResourceGroupA**. In the search results, select **myResourceGroupA**.
3. Select **Delete**.
4. To confirm the deletion, in the **TYPE THE RESOURCE GROUP NAME** box, enter **myResourceGroupA**, and then select **Delete**.
5. Log out of the portal as UserA and log in as UserB.
6. Complete steps 2-4 for **myResourceGroupB**.

Azure CLI

1. Log in to Azure as UserA and execute the following command:

```
az group delete --name myResourceGroupA --yes
```

2. Log out of Azure as UserA and log in as UserB.

3. Execute the following command:

```
az group delete --name myResourceGroupB --yes
```

PowerShell

1. Log in to Azure as UserA and execute the following command:

```
Remove-AzResourceGroup -Name myResourceGroupA -force
```

2. Log out of Azure as UserA and log in as UserB.

3. Execute the following command:

```
Remove-AzResourceGroup -Name myResourceGroupB -force
```

Next steps

- Thoroughly familiarize yourself with important [virtual network peering constraints and behaviors](#) before creating a virtual network peering for production use.
- Learn about all [virtual network peering settings](#).
- Learn how to [create a hub and spoke network topology](#) with virtual network peering.

Create a virtual network peering - different deployment models, same subscription

2/18/2021 • 10 minutes to read • [Edit Online](#)

In this tutorial, you learn to create a virtual network peering between virtual networks created through different deployment models. Both virtual networks exist in the same subscription. Peering two virtual networks enables resources in different virtual networks to communicate with each other with the same bandwidth and latency as though the resources were in the same virtual network. Learn more about [Virtual network peering](#).

The steps to create a virtual network peering are different, depending on whether the virtual networks are in the same, or different, subscriptions, and which [Azure deployment model](#) the virtual networks are created through. Learn how to create a virtual network peering in other scenarios by clicking the scenario from the following table:

AZURE DEPLOYMENT MODEL	AZURE SUBSCRIPTION
Both Resource Manager	Same
Both Resource Manager	Different
One Resource Manager, one classic	Different

A virtual network peering can't be created between two virtual networks deployed through the classic deployment model. If you need to connect virtual networks that were both created through the classic deployment model, you can use an Azure [VPN Gateway](#) to connect the virtual networks.

This tutorial peers virtual networks in the same region. You can also peer virtual networks in different [supported regions](#). It's recommended that you familiarize yourself with the [peering requirements and constraints](#) before peering virtual networks.

You can use the Azure portal, the Azure [command-line interface](#) (CLI), Azure [PowerShell](#), or an Azure Resource Manager template to create a virtual network peering. Click any of the previous tool links to go directly to the steps for creating a virtual network peering using your tool of choice.

Create peering - Azure portal

1. Sign in to the [Azure portal](#). The account you sign in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
2. Click **+ New**, click **Networking**, then click **Virtual network**.
3. In the **Create virtual network** blade, enter, or select values for the following settings, then click **Create**:
 - **Name:** *myVnet1*
 - **Address space:** *10.0.0.0/16*
 - **Subnet name:** *default*
 - **Subnet address range:** *10.0.0.0/24*
 - **Subscription:** Select your subscription
 - **Resource group:** Select **Create new** and enter *myResourceGroup*
 - **Location:** *East US*

4. Click + New. In the Search the Marketplace box, type *Virtual network*. Click **Virtual network** when it appears in the search results.
5. In the **Virtual network** blade, select **Classic** in the Select a deployment model box, and then click **Create**.
6. In the **Create virtual network** blade, enter, or select values for the following settings, then click **Create**:
 - **Name:** *myVnet2*
 - **Address space:** *10.1.0.0/16*
 - **Subnet name:** *default*
 - **Subnet address range:** *10.1.0.0/24*
 - **Subscription:** Select your subscription
 - **Resource group:** Select **Use existing** and select *myResourceGroup*
 - **Location:** *East US*
7. In the **Search resources** box at the top of the portal, type *myResourceGroup*. Click **myResourceGroup** when it appears in the search results. A blade appears for the **myresourcegroup** resource group. The resource group holds the two virtual networks created in previous steps.
8. Click **myVNet1**.
9. In the **myVnet1** blade that appears, click **Peerings** from the vertical list of options on the left side of the blade.
10. In the **myVnet1 - Peerings** blade that appeared, click + **Add**
11. In the **Add peering** blade that appears, enter, or select the following options, then click **OK**:
 - **Name:** *myVnet1ToMyVnet2*
 - **Virtual network deployment model:** Select **Classic**.
 - **Subscription:** Select your subscription
 - **Virtual network:** Click **Choose a virtual network**, then click **myVnet2**.
 - **Allow virtual network access:** Ensure that **Enabled** is selected. No other settings are used in this tutorial. To learn about all peering settings, read [Manage virtual network peerings](#).
12. After clicking **OK** in the previous step, the **Add peering** blade closes and you see the **myVnet1 - Peerings** blade again. After a few seconds, the peering you created appears in the blade. **Connected** is listed in the **PEERING STATUS** column for the **myVnet1ToMyVnet2** peering you created.

The peering is now established. Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks aren't able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).
13. **Optional:** Though creating virtual machines isn't covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
14. **Optional:** To delete the resources that you create in this tutorial, complete the steps in the [Delete resources](#) section of this article.

Create peering - Azure CLI

Complete the following steps using the Azure classic CLI and the Azure CLI. You can complete the steps from the Azure Cloud Shell, by just selecting the Try it button in any of the following steps, or by installing the [classic CLI](#)

and [CLI](#) and running the commands on your local computer.

1. If using the Cloud Shell, skip to step 2, because the Cloud Shell automatically signs you in to Azure. Open a command session and sign in to Azure using the `azure login` command.
2. Run the CLI in Service Management mode by entering the `azure config mode asm` command.
3. Enter the following command to create the virtual network (classic):

```
azure network vnet create --vnet myVnet2 --address-space 10.1.0.0 --cidr 16 --location "East US"
```

4. Execute the following bash CLI script using the CLI, not the classic CLI. For options on running bash CLI scripts on Windows computer, see [Install the Azure CLI on Windows](#).

```
#!/bin/bash

# Create a resource group.
az group create \
    --name myResourceGroup \
    --location eastus

# Create the virtual network (Resource Manager).
az network vnet create \
    --name myVnet1 \
    --resource-group myResourceGroup \
    --location eastus \
    --address-prefix 10.0.0.0/16
```

5. Create a virtual network peering between the two virtual networks created through the different deployment models using the CLI. Copy the following script to a text editor on your PC. Replace `<subscription id>` with your subscription ID. If you don't know your subscription ID, enter the `az account show` command. The value for `id` in the output is your subscription ID. Paste the modified script in to your CLI session, and then press `Enter`.

```
# Get the ID for VNet1.
vnet1Id=$(az network vnet show \
    --resource-group myResourceGroup \
    --name myVnet1 \
    --query id --out tsv)

# Peer VNet1 to VNet2.
az network vnet peering create \
    --name myVnet1ToMyVnet2 \
    --resource-group myResourceGroup \
    --vnet-name myVnet1 \
    --remote-vnet-id /subscriptions/<subscription id>/resourceGroups/Default-
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnet2 \
    --allow-vnet-access
```

6. After the script executes, review the peering for the virtual network (Resource Manager). Copy the following command, paste it in your CLI session, and then press `Enter`:

```
az network vnet peering list \
    --resource-group myResourceGroup \
    --vnet-name myVnet1 \
    --output table
```

The output shows **Connected** in the **PeeringState** column.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks aren't able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

7. **Optional:** Though creating virtual machines isn't covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
8. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

Create peering - PowerShell

1. Install the latest version of the PowerShell [Azure](#) and [Az](#) modules. If you're new to Azure PowerShell, see [Azure PowerShell overview](#).
2. Start a PowerShell session.
3. In PowerShell, sign in to Azure by entering the `Add-AzureAccount` command. The account you sign in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
4. To create a virtual network (classic) with PowerShell, you must create a new, or modify an existing, network configuration file. Learn how to [export, update, and import network configuration files](#). The file should include the following **VirtualNetworkSite** element for the virtual network used in this tutorial:

```
<VirtualNetworkSite name="myVnet2" Location="East US">
  <AddressSpace>
    <AddressPrefix>10.1.0.0/16</AddressPrefix>
  </AddressSpace>
  <Subnets>
    <Subnet name="default">
      <AddressPrefix>10.1.0.0/24</AddressPrefix>
    </Subnet>
  </Subnets>
</VirtualNetworkSite>
```

WARNING

Importing a changed network configuration file can cause changes to existing virtual networks (classic) in your subscription. Ensure you only add the previous virtual network and that you don't change or remove any existing virtual networks from your subscription.

5. Sign in to Azure to create the virtual network (Resource Manager) by entering the `Connect-AzAccount` command. The account you sign in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
6. Create a resource group and a virtual network (Resource Manager). Copy the script, paste it into PowerShell, and then press `Enter`.

```

# Create a resource group.
New-AzResourceGroup -Name myResourceGroup -Location eastus

# Create the virtual network (Resource Manager).
$vnet1 = New-AzVirtualNetwork ` 
-ResourceGroupName myResourceGroup ` 
-Name 'myVnet1' ` 
-AddressPrefix '10.0.0.0/16' ` 
-Location eastus

```

7. Create a virtual network peering between the two virtual networks created through the different deployment models. Copy the following script to a text editor on your PC. Replace <subscription id> with your subscription ID. If you don't know your subscription ID, enter the `Get-AzSubscription` command to view it. The value for `Id` in the returned output is your subscription ID. To execute the script, copy the modified script from your text editor, then right-click in your PowerShell session, and then press `Enter`.

```

# Peer VNet1 to VNet2.
Add-AzVirtualNetworkPeering ` 
-Name myVnet1ToMyVnet2 ` 
-VirtualNetwork $vnet1 ` 
-RemoteVirtualNetworkId /subscriptions/<subscription Id>/resourceGroups/Default-` 
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnet2

```

8. After the script executes, review the peering for the virtual network (Resource Manager). Copy the following command, paste it in your PowerShell session, and then press `Enter`:

```

Get-AzVirtualNetworkPeering ` 
-ResourceGroupName myResourceGroup ` 
-VirtualNetworkName myVnet1 ` 
| Format-Table VirtualNetworkName, PeeringState

```

The output shows **Connected** in the **PeeringState** column.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks aren't able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

9. **Optional:** Though creating virtual machines isn't covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
10. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

Delete resources

When you've finished this tutorial, you might want to delete the resources you created in the tutorial, so you don't incur usage charges. Deleting a resource group also deletes all resources that are in the resource group.

Azure portal

1. In the portal search box, enter **myResourceGroup**. In the search results, click **myResourceGroup**.
2. On the **myResourceGroup** blade, click the **Delete** icon.
3. To confirm the deletion, in the **TYPE THE RESOURCE GROUP NAME** box, enter **myResourceGroup**, and

then click **Delete**.

Azure CLI

1. Use the Azure CLI to delete the virtual network (Resource Manager) with the following command:

```
az group delete --name myResourceGroup --yes
```

2. Use the classic CLI to delete the virtual network (classic) with the following commands:

```
azure config mode asm  
azure network vnet delete --vnet myVnet2 --quiet
```

PowerShell

1. Enter the following command to delete the virtual network (Resource Manager):

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

2. To delete the virtual network (classic) with PowerShell, you must modify an existing network configuration file. Learn how to [export, update, and import network configuration files](#). Remove the following VirtualNetworkSite element for the virtual network used in this tutorial:

```
<VirtualNetworkSite name="myVnet2" Location="East US">  
  <AddressSpace>  
    <AddressPrefix>10.1.0.0/16</AddressPrefix>  
  </AddressSpace>  
  <Subnets>  
    <Subnet name="default">  
      <AddressPrefix>10.1.0.0/24</AddressPrefix>  
    </Subnet>  
  </Subnets>  
</VirtualNetworkSite>
```

WARNING

Importing a changed network configuration file can cause changes to existing virtual networks (classic) in your subscription. Ensure you only remove the previous virtual network and that you don't change or remove any other existing virtual networks from your subscription.

Next steps

- Thoroughly familiarize yourself with important [virtual network peering constraints and behaviors](#) before creating a virtual network peering for production use.
- Learn about all [virtual network peering settings](#).
- Learn how to [create a hub and spoke network topology](#) with virtual network peering.

Create a virtual network peering - different deployment models and subscriptions

2/18/2021 • 14 minutes to read • [Edit Online](#)

In this tutorial, you learn to create a virtual network peering between virtual networks created through different deployment models. The virtual networks exist in different subscriptions. Peering two virtual networks enables resources in different virtual networks to communicate with each other with the same bandwidth and latency as though the resources were in the same virtual network. Learn more about [Virtual network peering](#).

The steps to create a virtual network peering are different, depending on whether the virtual networks are in the same, or different, subscriptions, and which [Azure deployment model](#) the virtual networks are created through. Learn how to create a virtual network peering in other scenarios by clicking the scenario from the following table:

AZURE DEPLOYMENT MODEL	AZURE SUBSCRIPTION
Both Resource Manager	Same
Both Resource Manager	Different
One Resource Manager, one classic	Same

A virtual network peering cannot be created between two virtual networks deployed through the classic deployment model. This tutorial uses virtual networks that exist in the same region. This tutorial peers virtual networks in the same region. You can also peer virtual networks in different [supported regions](#). It's recommended that you familiarize yourself with the [peering requirements and constraints](#) before peering virtual networks.

When creating a virtual network peering between virtual networks that exist in different subscriptions, the subscriptions can be associated to the same Azure Active Directory tenant. If you don't already have an Azure Active Directory tenant, you can quickly [create one](#).

You can use the [Azure portal](#), the Azure [command-line interface](#) (CLI), or Azure [PowerShell](#) to create a virtual network peering. Click any of the previous tool links to go directly to the steps for creating a virtual network peering using your tool of choice.

Create peering - Azure portal

This tutorial uses different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of the portal, and skip the steps for assigning another user permissions to the virtual networks.

1. Log in to the [Azure portal](#) as UserA. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
2. Click + New, click **Networking**, then click **Virtual network**.
3. In the **Create virtual network** blade, enter, or select values for the following settings, then click **Create**:
 - **Name:** *myVnetA*
 - **Address space:** *10.0.0.0/16*

- **Subnet name:** *default*
 - **Subnet address range:** *10.0.0.0/24*
 - **Subscription:** Select subscription A.
 - **Resource group:** Select **Create new** and enter *myResourceGroupA*
 - **Location:** *East US*
4. In the **Search resources** box at the top of the portal, type *myVnetA*. Click **myVnetA** when it appears in the search results. A blade appears for the **myVnetA** virtual network.
5. In the **myVnetA** blade that appears, click **Access control (IAM)** from the vertical list of options on the left side of the blade.
6. In the **myVnetA - Access control (IAM)** blade that appears, click **+ Add role assignment**.
7. In the **Add role assignment** blade that appears, select **Network contributor** in the **Role** box.
8. In the **Select** box, select UserB, or type UserB's email address to search for it. The list of users shown is from the same Azure Active Directory tenant as the virtual network you're setting up the peering for. Click UserB when it appears in the list.
9. Click **Save**.
10. Log out of the portal as UserA, then log in as UserB.
11. Click **+ New**, type *Virtual network* in the **Search the Marketplace** box, then click **Virtual network** in the search results.
12. In the **Virtual Network** blade that appears, select **Classic** in the **Select a deployment model** box, then click **Create**.
13. In the **Create virtual network (classic)** box that appears, enter the following values:
- **Name:** *myVnetB*
 - **Address space:** *10.1.0.0/16*
 - **Subnet name:** *default*
 - **Subnet address range:** *10.1.0.0/24*
 - **Subscription:** Select subscription B.
 - **Resource group:** Select **Create new** and enter *myResourceGroupB*
 - **Location:** *East US*
14. In the **Search resources** box at the top of the portal, type *myVnetB*. Click **myVnetB** when it appears in the search results. A blade appears for the **myVnetB** virtual network.
15. In the **myVnetB** blade that appears, click **Properties** from the vertical list of options on the left side of the blade. Copy the **RESOURCE ID**, which is used in a later step. The resource ID is similar to the following example:
- ```
/subscriptions/<Subscription ID>/resourceGroups/myResourceGroupB/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnetB
```
16. Complete steps 5-9 for myVnetB, entering **UserA** in step 8.
17. Log out of the portal as UserB and log in as UserA.
18. In the **Search resources** box at the top of the portal, type *myVnetA*. Click **myVnetA** when it appears in the search results. A blade appears for the **myVnet** virtual network.
19. Click **myVnetA**.
20. In the **myVnetA** blade that appears, click **Peerings** from the vertical list of options on the left side of the blade.

21. In the **myVnetA - Peerings** blade that appeared, click **+ Add**
22. In the **Add peering** blade that appears, enter, or select the following options, then click **OK**:
  - **Name:** *myVnetAToMyVnetB*
  - **Virtual network deployment model:** Select **Classic**.
  - **I know my resource ID:** Check this box.
  - **Resource ID:** Enter the resource ID of myVnetB from step 15.
  - **Allow virtual network access:** Ensure that **Enabled** is selected. No other settings are used in this tutorial. To learn about all peering settings, read [Manage virtual network peerings](#).
23. After clicking **OK** in the previous step, the **Add peering** blade closes and you see the **myVnetA - Peerings** blade again. After a few seconds, the peering you created appears in the blade. **Connected** is listed in the **PEERING STATUS** column for the **myVnetAToMyVnetB** peering you created. The peering is now established. There is no need to peer the virtual network (classic) to the virtual network (Resource Manager).

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).
24. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
25. **Optional:** To delete the resources that you create in this tutorial, complete the steps in the [Delete resources](#) section of this article.

## Create peering - Azure CLI

This tutorial uses different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of Azure, and remove the lines of script that create user role assignments. Replace UserA@azure.com and UserB@azure.com in all of the following scripts with the usernames you're using for UserA and UserB. Complete the following steps using the Azure classic CLI and the Azure CLI. You can complete the steps from the Azure Cloud Shell, by just selecting the **Try it** button in any of the following steps, or by installing the [classic CLI](#) and [CLI](#) and running the commands on your local computer.

1. If using the Cloud Shell, skip to step 2, because the Cloud Shell automatically signs you in to Azure. Open a command session and sign in to Azure using the `azure login` command.
2. Run the classic CLI in Service Management mode by entering the `azure config mode asm` command.
3. Enter the following classic CLI command to create the virtual network (classic):

```
azure network vnet create --vnet myVnetB --address-space 10.1.0.0 --cidr 16 --location "East US"
```

4. The remaining steps must be completed using a bash shell with the Azure CLI (not the classic CLI).
5. Copy the following script to a text editor on your PC. Replace `<SubscriptionB-Id>` with your subscription ID. If you don't know your subscription Id, enter the `az account show` command. The value for `id` in the output is your subscription Id. Copy the modified script, paste it in to your CLI session, and then press `Enter`.

```
az role assignment create \
--assignee UserA@azure.com \
--role "Classic Network Contributor" \
--scope /subscriptions/<SubscriptionB-Id>/resourceGroups/Default-
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnetB
```

When you created the virtual network (classic) in step 4, Azure created the virtual network in the *Default-Networking* resource group.

6. Log UserB out of Azure and log in as UserA in the CLI.
7. Create a resource group and a virtual network (Resource Manager). Copy the following script, paste it in to your CLI session, and then press **Enter**.

```
#!/bin/bash

Variables for common values used throughout the script.
rgName="myResourceGroupA"
location="eastus"

Create a resource group.
az group create \
--name $rgName \
--location $location

Create virtual network A (Resource Manager).
az network vnet create \
--name myVnetA \
--resource-group $rgName \
--location $location \
--address-prefix 10.0.0.0/16

Get the id for myVnetA.
vNetAId=$(az network vnet show \
--resource-group $rgName \
--name myVnetA \
--query id --out tsv)

Assign UserB permissions to myVnetA.
az role assignment create \
--assignee UserB@azure.com \
--role "Network Contributor" \
--scope $vNetAId
```

8. Create a virtual network peering between the two virtual networks created through the different deployment models. Copy the following script to a text editor on your PC. Replace `<SubscriptionB-id>` with your subscription Id. If you don't know your subscription Id, enter the `az account show` command. The value for `id` in the output is your subscription Id. Azure created the virtual network (classic) you created in step 4 in a resource group named *Default-Networking*. Paste the modified script in your CLI session, and then press **Enter**.

```
Peer VNet1 to VNet2.
az network vnet peering create \
--name myVnetAToMyVnetB \
--resource-group $rgName \
--vnet-name myVnetA \
--remote-vnet-id /subscriptions/<SubscriptionB-id>/resourceGroups/Default-
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnetB \
--allow-vnet-access
```

9. After the script executes, review the peering for the virtual network (Resource Manager). Copy the following script, and then paste it in your CLI session:

```
az network vnet peering list \
--resource-group $rgName \
--vnet-name myVnetA \
--output table
```

The output shows **Connected** in the **PeeringState** column.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

10. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
11. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

## Create peering - PowerShell

This tutorial uses different accounts for each subscription. If you're using an account that has permissions to both subscriptions, you can use the same account for all steps, skip the steps for logging out of Azure, and remove the lines of script that create user role assignments. Replace UserA@azure.com and UserB@azure.com in all of the following scripts with the usernames you're using for UserA and UserB.

1. Install the latest version of the PowerShell [Azure](#) and [Az](#) modules. If you're new to Azure PowerShell, see [Azure PowerShell overview](#).
2. Start a PowerShell session.
3. In PowerShell, log in to UserB's subscription as UserB by entering the `Add-AzureAccount` command. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
4. To create a virtual network (classic) with PowerShell, you must create a new, or modify an existing, network configuration file. Learn how to [export, update, and import network configuration files](#). The file should include the following **VirtualNetworkSite** element for the virtual network used in this tutorial:

```
<VirtualNetworkSite name="myVnetB" Location="East US">
 <AddressSpace>
 <AddressPrefix>10.1.0.0/16</AddressPrefix>
 </AddressSpace>
 <Subnets>
 <Subnet name="default">
 <AddressPrefix>10.1.0.0/24</AddressPrefix>
 </Subnet>
 </Subnets>
</VirtualNetworkSite>
```

### WARNING

Importing a changed network configuration file can cause changes to existing virtual networks (classic) in your subscription. Ensure you only add the previous virtual network and that you don't change or remove any existing virtual networks from your subscription.

5. Log in to UserB's subscription as UserB to use Resource Manager commands by entering the `Connect-AzAccount` command.
6. Assign UserA permissions to virtual network B. Copy the following script to a text editor on your PC and replace `<SubscriptionB-id>` with the ID of subscription B. If you don't know the subscription Id, enter the `Get-AzSubscription` command to view it. The value for `Id` in the returned output is your subscription ID. Azure created the virtual network (classic) you created in step 4 in a resource group named `Default-Networking`. To execute the script, copy the modified script, paste it in to PowerShell, and then press `Enter`.

```
New-AzRoleAssignment `
 -SignInName UserA@azure.com `
 -RoleDefinitionName "Classic Network Contributor" `
 -Scope /subscriptions/<SubscriptionB-id>/resourceGroups/Default-
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnet
```

7. Log out of Azure as UserB and log in to UserA's subscription as UserA by entering the `Connect-AzAccount` command. The account you log in with must have the necessary permissions to create a virtual network peering. For a list of permissions, see [Virtual network peering permissions](#).
8. Create the virtual network (Resource Manager) by copying the following script, pasting it in to PowerShell, and then pressing `Enter`:

```
Variables for common values
$rgName='MyResourceGroupA'
$location='eastus'

Create a resource group.
New-AzResourceGroup `
 -Name $rgName `
 -Location $location

Create virtual network A.
$vnetA = New-AzVirtualNetwork `
 -ResourceGroupName $rgName `
 -Name 'myVnetA' `
 -AddressPrefix '10.0.0.0/16' `
 -Location $location
```

9. Assign UserB permissions to myVnetA. Copy the following script to a text editor on your PC and replace `<SubscriptionA-Id>` with the ID of subscription A. If you don't know the subscription Id, enter the `Get-AzSubscription` command to view it. The value for `Id` in the returned output is your subscription ID. Paste the modified version of the script in PowerShell, and then press `Enter` to execute it.

```
New-AzRoleAssignment `
 -SignInName UserB@azure.com `
 -RoleDefinitionName "Network Contributor" `
 -Scope /subscriptions/<SubscriptionA-
Id>/resourceGroups/myResourceGroupA/providers/Microsoft.Network/VirtualNetworks/myVnetA
```

10. Copy the following script to a text editor on your PC, and replace <SubscriptionB-id> with the ID of subscription B. To peer myVnetA to myVNetB, copy the modified script, paste it in to PowerShell, and then press **Enter**.

```
Add-AzVirtualNetworkPeering `
-Name 'myVnetAToMyVnetB' `
-VirtualNetwork $vnetA `
-RemoteVirtualNetworkId /subscriptions/<SubscriptionB-id>/resourceGroups/Default-
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/myVnetB
```

11. View the peering state of myVnetA by copying the following script, pasting it into PowerShell, and pressing **Enter**.

```
Get-AzVirtualNetworkPeering `
-ResourceGroupName $rgName `
-VirtualNetworkName myVnetA `
| Format-Table VirtualNetworkName, PeeringState
```

The state is **Connected**. It changes to **Connected** once you set up the peering to myVnetA from myVnetB.

Any Azure resources you create in either virtual network are now able to communicate with each other through their IP addresses. If you're using default Azure name resolution for the virtual networks, the resources in the virtual networks are not able to resolve names across the virtual networks. If you want to resolve names across virtual networks in a peering, you must create your own DNS server. Learn how to set up [Name resolution using your own DNS server](#).

12. **Optional:** Though creating virtual machines is not covered in this tutorial, you can create a virtual machine in each virtual network and connect from one virtual machine to the other, to validate connectivity.
13. **Optional:** To delete the resources that you create in this tutorial, complete the steps in [Delete resources](#) in this article.

## Delete resources

When you've finished this tutorial, you might want to delete the resources you created in the tutorial, so you don't incur usage charges. Deleting a resource group also deletes all resources that are in the resource group.

### Azure portal

1. In the portal search box, enter **myResourceGroupA**. In the search results, click **myResourceGroupA**.
2. On the **myResourceGroupA** blade, click the **Delete** icon.
3. To confirm the deletion, in the **TYPE THE RESOURCE GROUP NAME** box, enter **myResourceGroupA**, and then click **Delete**.
4. In the **Search resources** box at the top of the portal, type **myVnetB**. Click **myVnetB** when it appears in the search results. A blade appears for the **myVnetB** virtual network.
5. In the **myVnetB** blade, click **Delete**.
6. To confirm the deletion, click **Yes** in the **Delete virtual network** box.

### Azure CLI

1. Log in to Azure using the CLI to delete the virtual network (Resource Manager) with the following command:

```
az group delete --name myResourceGroupA --yes
```

2. Sign in to Azure using the classic CLI to delete the virtual network (classic) with the following commands:

```
azure config mode asm
azure network vnet delete --vnet myVnetB --quiet
```

## PowerShell

1. At the PowerShell command prompt, enter the following command to delete the virtual network (Resource Manager):

```
Remove-AzResourceGroup -Name myResourceGroupA -Force
```

2. To delete the virtual network (classic) with PowerShell, you must modify an existing network configuration file. Learn how to [export, update, and import network configuration files](#). Remove the following VirtualNetworkSite element for the virtual network used in this tutorial:

```
<VirtualNetworkSite name="myVnetB" Location="East US">
 <AddressSpace>
 <AddressPrefix>10.1.0.0/16</AddressPrefix>
 </AddressSpace>
 <Subnets>
 <Subnet name="default">
 <AddressPrefix>10.1.0.0/24</AddressPrefix>
 </Subnet>
 </Subnets>
</VirtualNetworkSite>
```

### WARNING

Importing a changed network configuration file can cause changes to existing virtual networks (classic) in your subscription. Ensure you only remove the previous virtual network and that you don't change or remove any other existing virtual networks from your subscription.

## Next steps

- Thoroughly familiarize yourself with important [virtual network peering constraints and behaviors](#) before creating a virtual network peering for production use.
- Learn about all [virtual network peering settings](#).
- Learn how to [create a hub and spoke network topology](#) with virtual network peering.

# Create, change, or delete a virtual network peering

2/18/2021 • 17 minutes to read • [Edit Online](#)

Learn how to create, change, or delete a virtual network peering. Virtual network peering enables you to connect virtual networks in the same region and across regions (also known as Global VNet Peering) through the Azure backbone network. Once peered, the virtual networks are still managed as separate resources. If you're new to virtual network peering, you can learn more about it in the [virtual network peering overview](#) or by completing a [tutorial](#).

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with an account that has the [necessary permissions](#) to work with peerings.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` with an account that has the [necessary permissions](#) to work with peering, to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.31 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` with an account that has the [necessary permissions](#) to work with peering, to create a connection with Azure.

The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Permissions](#).

## Create a peering

Before creating a peering, familiarize yourself with the requirements and constraints and [necessary permissions](#).

1. In the search box at the top of the Azure portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it. Do not select **Virtual networks (classic)** if it appears in the list, as you cannot create a peering from a virtual network deployed through the classic deployment model.
2. Select the virtual network in the list that you want to create a peering for.
3. Under **SETTINGS**, select **Peerings**.

4. Select + Add.
5. Enter or select values for the following settings:
  - **Name:** The name for the peering must be unique within the virtual network.
  - **Virtual network deployment model:** Select which deployment model the virtual network you want to peer with was deployed through.
  - **I know my resource ID:** If you have read access to the virtual network you want to peer with, leave this checkbox unchecked. If you don't have read access to the virtual network or subscription you want to peer with, check this box. Enter the full resource ID of the virtual network you want to peer with in the **Resource ID** box that appeared when you checked the box. The resource ID you enter must be for a virtual network that exists in the same, or [supported different Azure region](#) as this virtual network. The full resource ID looks similar to

```
/subscriptions/<Id>/resourceGroups/<resource-group-name>/providers/Microsoft.Network/virtualNetworks/<virtual-network-name>
```

. You can get the resource ID for a virtual network by viewing the properties for a virtual network. To learn how to view the properties for a virtual network, see [Manage virtual networks](#). If the subscription is associated to a different Azure Active Directory tenant than the subscription with the virtual network you're creating the peering from, first add a user from each tenant as a [guest user](#) in the opposite tenant.
  - **Subscription:** Select the [subscription](#) of the virtual network you want to peer with. One or more subscriptions are listed, depending on how many subscriptions your account has read access to. If you checked the **Resource ID** checkbox, this setting isn't available.
  - **Virtual network:** Select the virtual network you want to peer with. You can select a virtual network created through either Azure deployment model. If you want to select a virtual network in a different region, you must select a virtual network in a [supported region](#). You must have read access to the virtual network for it to be visible in the list. If a virtual network is listed, but grayed out, it may be because the address space for the virtual network overlaps with the address space for this virtual network. If virtual network address spaces overlap, they cannot be peered. If you checked the **Resource ID** checkbox, this setting isn't available.
  - **Allow virtual network access:** Select **Enabled** (default) if you want to enable communication between the two virtual networks through the default `VirtualNetwork` flow. Enabling communication between virtual networks allows resources connected to either virtual network to communicate with each other with the same bandwidth and latency as if they were connected to the same virtual network. All communication between resources in the two virtual networks is over the Azure private network. The `VirtualNetwork` service tag for network security groups encompasses the virtual network and peered virtual network when this setting is **Enabled**. (To learn more about network security group service tags, see [Network security groups overview](#).) Select **Disabled** if you don't want traffic to flow to the peered virtual network by default. You might select **Disabled** if you've peered a virtual network with another virtual network, but occasionally want to disable default traffic flow between the two virtual networks. You may find enabling/disabling is more convenient than deleting and re-creating peerings. When this setting is disabled, traffic doesn't flow between the peered virtual networks by default; however, traffic may still flow if explicitly allowed through a [network security group](#) rule that includes the appropriate IP addresses or application security groups.

**WARNING**

Disabling the **Allow virtual network access** setting only changes the definition of the **VirtualNetwork** service tag. It *doesn't* fully prevent traffic flow across the peer connection, as explained in this setting description.

- **Allow forwarded traffic:** Check this box to allow traffic *forwarded* by a network virtual appliance in a virtual network (that didn't originate from the virtual network) to flow to this virtual network through a peering. For example, consider three virtual networks named Spoke1, Spoke2, and Hub. A peering exists between each spoke virtual network and the Hub virtual network, but peerings don't exist between the spoke virtual networks. A network virtual appliance is deployed in the Hub virtual network, and user-defined routes are applied to each spoke virtual network that route traffic between the subnets through the network virtual appliance. If this checkbox is not checked for the peering between each spoke virtual network and the hub virtual network, traffic doesn't flow between the spoke virtual networks because the hub is not forwarding the traffic between the virtual networks. While enabling this capability allows the forwarded traffic through the peering, it does not create any user-defined routes or network virtual appliances. User-defined routes and network virtual appliances are created separately. Learn about [user-defined routes](#). You don't need to check this setting if traffic is forwarded between virtual networks through an Azure VPN Gateway.
- **Allow gateway transit:** Check this box if you have a virtual network gateway attached to this virtual network and want to allow traffic from the peered virtual network to flow through the gateway. For example, this virtual network may be attached to an on-premises network through a virtual network gateway. The gateway can be an ExpressRoute or VPN gateway. Checking this box allows traffic from the peered virtual network to flow through the gateway attached to this virtual network to the on-premises network. If you check this box, the peered virtual network cannot have a gateway configured. The peered virtual network must have the **Use remote gateways** checkbox checked when setting up the peering from the other virtual network to this virtual network. If you leave this box unchecked (default), traffic from the peered virtual network still flows to this virtual network, but cannot flow through a virtual network gateway attached to this virtual network. If the peering is between a virtual network (Resource Manager) and a virtual network (classic), the gateway must be in the virtual network (Resource Manager).

In addition to forwarding traffic to an on-premises network, a VPN gateway can forward network traffic between virtual networks that are peered with the virtual network the gateway is in, without the virtual networks needing to be peered with each other. Using a VPN gateway to forward traffic is useful when you want to use a VPN gateway in a hub (see the hub and spoke example described for **Allow forwarded traffic**) virtual network to route traffic between spoke virtual networks that aren't peered with each other. To learn more about allowing use of a gateway for transit, see [Configure a VPN gateway for transit in a virtual network peering](#). This scenario requires implementing user-defined routes that specify the virtual network gateway as the next hop type. Learn about [user-defined routes](#). You can only specify a VPN gateway as a next hop type in a user-defined route, you cannot specify an ExpressRoute gateway as the next hop type in a user-defined route.

- **Use remote gateways:** Check this box to allow traffic from this virtual network to flow through a virtual network gateway attached to the virtual network you're peering with. For example, the virtual network you're peering with has a VPN gateway attached that enables communication to an on-premises network. Checking this box allows traffic from this virtual network to flow through the VPN gateway attached to the peered virtual network. If you check this box, the peered virtual network must have a virtual network gateway attached to it and must have the **Allow gateway transit** checkbox checked. If you leave this box unchecked (default), traffic from the peered virtual

network can still flow to this virtual network, but cannot flow through a virtual network gateway attached to this virtual network.

Only one peering for this virtual network can have this setting enabled.

You can't use remote gateways if you already have a gateway configured in your virtual network. To learn more about using a gateway for transit, see [Configure a VPN gateway for transit in a virtual network peering](#)

#### NOTE

If you use a Virtual Network Gateway to send on-premises traffic transitively to a peered VNet, the peered VNet IP range for the on-premises VPN device must be set to 'interesting' traffic. Otherwise, your on-premises resources won't be able to communicate with resources in the peered VNet.

6. Select **OK** to add the peering to the virtual network you selected.

For step-by-step instructions for implementing peering between virtual networks in different subscriptions and deployment models, see [next steps](#).

#### Commands

- **Azure CLI:** `az network vnet peering create`
- **PowerShell:** `Add-AzVirtualNetworkPeering`

## View or change peering settings

Before changing a peering, familiarize yourself with the requirements and constraints and [necessary permissions](#).

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it. Do not select **Virtual networks (classic)** if it appears in the list, as you cannot create a peering from a virtual network deployed through the classic deployment model.
2. Select the virtual network in the list that you want to change peering settings for.
3. Under **SETTINGS**, select **Peerings**.
4. Select the peering you want to view or change settings for.
5. Change the appropriate setting. Read about the options for each setting in [step 5](#) of Create a peering.
6. Select **Save**.

#### Commands

- **Azure CLI:** `az network vnet peering list` to list peerings for a virtual network, `az network vnet peering show` to show settings for a specific peering, and `az network vnet peering update` to change peering settings.|
- **PowerShell:** `Get-AzVirtualNetworkPeering` to retrieve view peering settings and `Set-AzVirtualNetworkPeering` to change settings.

## Delete a peering

Before deleting a peering, ensure your account has the [necessary permissions](#).

When a peering is deleted, traffic from a virtual network no longer flows to the peered virtual network. When virtual networks deployed through Resource Manager are peered, each virtual network has a peering to the other virtual network. Though deleting the peering from one virtual network disables the communication between the virtual networks, it does not delete the peering from the other virtual network. The peering status for the peering that exists in the other virtual network is **Disconnected**. You cannot recreate the peering until you re-create the peering in the first virtual network and the peering status for both virtual networks changes to

*Connected.*

If you want virtual networks to communicate sometimes, but not always, rather than deleting a peering, you can set the **Allow virtual network access** setting to **Disabled** instead. To learn how, read step 6 of the Create a peering section of this article. You may find disabling and enabling network access easier than deleting and recreating peerings.

1. In the search box at the top of the portal, enter *virtual networks* in the search box. When **Virtual networks** appear in the search results, select it. Do not select **Virtual networks (classic)** if it appears in the list, as you cannot create a peering from a virtual network deployed through the classic deployment model.
2. Select the virtual network in the list that you want to delete a peering for.
3. Under **SETTINGS**, select **Peerings**.
4. On the right side of the peering you want to delete, select ..., select **Delete**, then select **Yes** to delete the peering from the first virtual network.
5. Complete the previous steps to delete the peering from the other virtual network in the peering.

## Commands

- Azure CLI: [az network vnet peering delete](#)
- PowerShell: [Remove-AzVirtualNetworkPeering](#)

## Requirements and constraints

- You can peer virtual networks in the same region, or different regions. Peering virtual networks in different regions is also referred to as *Global VNet Peering*.
- When creating a global peering, the peered virtual networks can exist in any Azure public cloud region or China cloud regions or Government cloud regions. You cannot peer across clouds. For example, a VNet in Azure public cloud cannot be peered to a VNet in Azure China cloud.
- Resources in one virtual network cannot communicate with the front-end IP address of a Basic internal load balancer in a globally peered virtual network. Support for Basic Load Balancer only exists within the same region. Support for Standard Load Balancer exists for both, VNet Peering and Global VNet Peering. Services that use a Basic load balancer which will not work over Global VNet Peering are documented [here](#).
- You can use remote gateways or allow gateway transit in globally peered virtual networks and locally peered virtual networks.
- The virtual networks can be in the same, or different subscriptions. When you peer virtual networks in different subscriptions, both subscriptions can be associated to the same or different Azure Active Directory tenant. If you don't already have an AD tenant, you can [create one](#).
- The virtual networks you peer must have non-overlapping IP address spaces.
- You can't add address ranges to, or delete address ranges from a virtual network's address space once a virtual network is peered with another virtual network. To add or remove address ranges, delete the peering, add or remove the address ranges, then re-create the peering. To add address ranges to, or remove address ranges from virtual networks, see [Manage virtual networks](#).
- You can peer two virtual networks deployed through Resource Manager or a virtual network deployed through Resource Manager with a virtual network deployed through the classic deployment model. You cannot peer two virtual networks created through the classic deployment model. If you're not familiar with Azure deployment models, read the [Understand Azure deployment models](#) article. You can use a [VPN Gateway](#) to connect two virtual networks created through the classic deployment model.
- When peering two virtual networks created through Resource Manager, a peering must be configured for

each virtual network in the peering. You see one of the following types for peering status:

- *Initiated*: When you create the peering to the second virtual network from the first virtual network, the peering status is *Initiated*.
- *Connected*: When you create the peering from the second virtual network to the first virtual network, its peering status is *Connected*. If you view the peering status for the first virtual network, you see its status changed from *Initiated* to *Connected*. The peering is not successfully established until the peering status for both virtual network peerings is *Connected*.
- When peering a virtual network created through Resource Manager with a virtual network created through the classic deployment model, you only configure a peering for the virtual network deployed through Resource Manager. You cannot configure peering for a virtual network (classic), or between two virtual networks deployed through the classic deployment model. When you create the peering from the virtual network (Resource Manager) to the virtual network (Classic), the peering status is *Updating*, then shortly changes to *Connected*.
- A peering is established between two virtual networks. Peerings by itself are not transitive. If you create peerings between:
  - VirtualNetwork1 & VirtualNetwork2 - VirtualNetwork1 & VirtualNetwork2
  - VirtualNetwork2 & VirtualNetwork3 - VirtualNetwork2 & VirtualNetwork3

There is no peering between VirtualNetwork1 and VirtualNetwork3 through VirtualNetwork2. If you want to create a virtual network peering between VirtualNetwork1 and VirtualNetwork3, you have to create a peering between VirtualNetwork1 and VirtualNetwork3. There is no peering between VirtualNetwork1 and VirtualNetwork3 through VirtualNetwork2. If you want VirtualNetwork1 and VirtualNetwork3 to directly communicate, you have to create an explicit peering between VirtualNetwork1 and VirtualNetwork3 or go through an NVA in the Hub network.

- You can't resolve names in peered virtual networks using default Azure name resolution. To resolve names in other virtual networks, you must use [Azure DNS for private domains](#) or a custom DNS server. To learn how to set up your own DNS server, see [Name resolution using your own DNS server](#).
- Resources in peered virtual networks in the same region can communicate with each other with the same bandwidth and latency as if they were in the same virtual network. Each virtual machine size has its own maximum network bandwidth however. To learn more about maximum network bandwidth for different virtual machine sizes, see [Windows](#) or [Linux](#) virtual machine sizes.
- A virtual network can be peered to another virtual network, and also be connected to another virtual network with an Azure virtual network gateway. When virtual networks are connected through both peering and a gateway, traffic between the virtual networks flows through the peering configuration, rather than the gateway.
- Point-to-Site VPN clients must be downloaded again after virtual network peering has been successfully configured to ensure the new routes are downloaded to the client.
- There is a nominal charge for ingress and egress traffic that utilizes a virtual network peering. For more information, see the [pricing page](#).

## Permissions

The accounts you use to work with virtual network peering must be assigned to the following roles:

- [Network Contributor](#): For a virtual network deployed through Resource Manager.
- [Classic Network Contributor](#): For a virtual network deployed through the classic deployment model.

If your account is not assigned to one of the previous roles, it must be assigned to a [custom role](#) that is assigned the necessary actions from the following table:

ACTION	NAME
Microsoft.Network/virtualNetworks/virtualNetworkPeerings/write	Required to create a peering from virtual network A to virtual network B. Virtual network A must be a virtual network (Resource Manager)
Microsoft.Network/virtualNetworks/peer/action	Required to create a peering from virtual network B (Resource Manager) to virtual network A
Microsoft.ClassicNetwork/virtualNetworks/peer/action	Required to create a peering from virtual network B (classic) to virtual network A
Microsoft.Network/virtualNetworks/virtualNetworkPeerings/read	Read a virtual network peering
Microsoft.Network/virtualNetworks/virtualNetworkPeerings/delete	Delete a virtual network peering

## Next steps

- A virtual network peering is created between virtual networks created through the same, or different deployment models that exist in the same, or different subscriptions. Complete a tutorial for one of the following scenarios:

AZURE DEPLOYMENT MODEL	SUBSCRIPTION
Both Resource Manager	Same
	Different
One Resource Manager, one classic	Same
	Different

- Learn how to create a [hub and spoke network topology](#)
- Create a virtual network peering using [PowerShell](#) or [Azure CLI](#) sample scripts, or using Azure [Resource Manager templates](#)
- Create and assign [Azure Policy definitions](#) for virtual networks

# Configure a VNet-to-VNet VPN gateway connection using PowerShell

2/18/2021 • 18 minutes to read • [Edit Online](#)

This article helps you connect virtual networks by using the VNet-to-VNet connection type. The virtual networks can be in the same or different regions, and from the same or different subscriptions. When connecting VNets from different subscriptions, the subscriptions do not need to be associated with the same Active Directory tenant.

The steps in this article apply to the Resource Manager deployment model and use PowerShell. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:

## About connecting VNets

There are multiple ways to connect VNets. The sections below describe different ways to connect virtual networks.

### VNet-to-VNet

Configuring a VNet-to-VNet connection is a good way to easily connect VNets. Connecting a virtual network to another virtual network using the VNet-to-VNet connection type (VNet2VNet) is similar to creating a Site-to-Site IPsec connection to an on-premises location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE, and both function the same way when communicating. The difference between the connection types is the way the local network gateway is configured. When you create a VNet-to-VNet connection, you do not see the local network gateway address space. It is automatically created and populated. If you update the address space for one VNet, the other VNet automatically knows to route to the updated address space. Creating a VNet-to-VNet connection is typically faster and easier than creating a Site-to-Site connection between VNets.

### Site-to-Site (IPsec)

If you are working with a complicated network configuration, you may prefer to connect your VNets using the [Site-to-Site](#) steps, instead the VNet-to-VNet steps. When you use the Site-to-Site steps, you create and configure the local network gateways manually. The local network gateway for each VNet treats the other VNet as a local site. This lets you specify additional address space for the local network gateway in order to route traffic. If the address space for a VNet changes, you need to update the corresponding local network gateway to reflect the change. It does not automatically update.

### VNet peering

You may want to consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway and has different constraints. Additionally, [VNet peering pricing](#) is calculated differently than [VNet-to-VNet VPN Gateway pricing](#). For more information, see [VNet peering](#).

## Why create a VNet-to-VNet connection?

You may want to connect virtual networks using a VNet-to-VNet connection for the following reasons:

- **Cross region geo-redundancy and geo-presence**
  - You can set up your own geo-replication or synchronization with secure connectivity without going over Internet-facing endpoints.

- With Azure Traffic Manager and Load Balancer, you can set up highly available workload with geo-redundancy across multiple Azure regions. One important example is to set up SQL Always On with Availability Groups spreading across multiple Azure regions.
- Regional multi-tier applications with isolation or administrative boundary**
  - Within the same region, you can set up multi-tier applications with multiple virtual networks connected together due to isolation or administrative requirements.

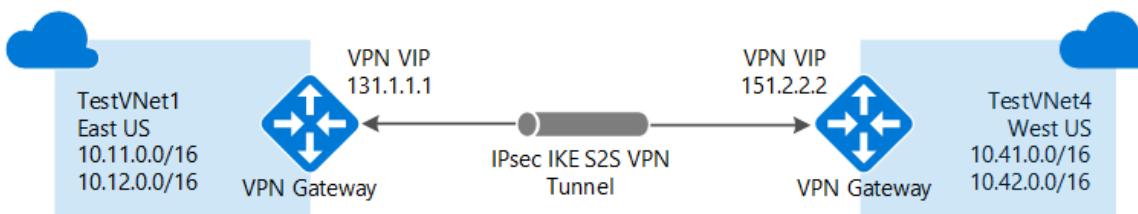
VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

## Which VNet-to-VNet steps should I use?

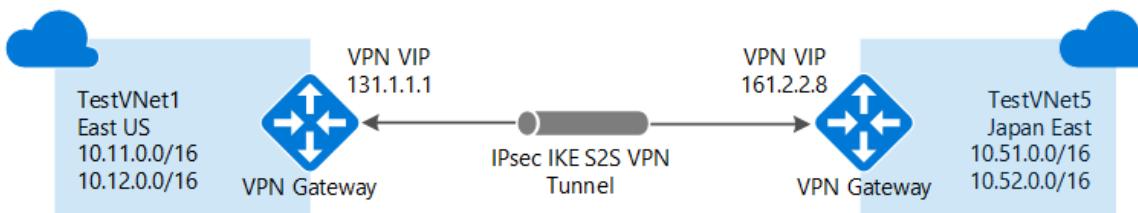
In this article, you see two different sets of steps. One set of steps for [VNets that reside in the same subscription](#) and one for [VNets that reside in different subscriptions](#). The key difference between the sets is that you must use separate PowerShell sessions when configuring the connections for VNets that reside in different subscriptions.

For this exercise, you can combine configurations, or just choose the one that you want to work with. All of the configurations use the VNet-to-VNet connection type. Network traffic flows between the VNets that are directly connected to each other. In this exercise, traffic from TestVNet4 does not route to TestVNet5.

- [VNets that reside in the same subscription](#): The steps for this configuration use TestVNet1 and TestVNet4.



- [VNets that reside in different subscriptions](#): The steps for this configuration use TestVNet1 and TestVNet5.



## How to connect VNets that are in the same subscription

### Before you begin

#### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

- Because it takes up to 45 minutes to create a gateway, Azure Cloud Shell will timeout periodically during this exercise. You can restart Cloud Shell by clicking in the upper left of the terminal. Be sure to redeclare any variables when you restart the terminal.
- If you would rather install latest version of the Azure PowerShell module locally, see [How to install and configure Azure PowerShell](#).

### Step 1 - Plan your IP address ranges

In the following steps, you create two virtual networks along with their respective gateway subnets and configurations. You then create a VPN connection between the two VNets. It's important to plan the IP address ranges for your network configuration. Keep in mind that you must make sure that none of your VNet ranges or local network ranges overlap in any way. In these examples, we do not include a DNS server. If you want name resolution for your virtual networks, see [Name resolution](#).

We use the following values in the examples:

#### Values for TestVNet1:

- VNet Name: TestVNet1
- Resource Group: TestRG1
- Location: East US
- TestVNet1: 10.11.0.0/16 & 10.12.0.0/16
- FrontEnd: 10.11.0.0/24
- BackEnd: 10.12.0.0/24
- GatewaySubnet: 10.12.255.0/27
- GatewayName: VNet1GW
- Public IP: VNet1GWIP
- VPNTYPE: RouteBased
- Connection(1to4): VNet1toVNet4
- Connection(1to5): VNet1toVNet5 (For VNets in different subscriptions)
- ConnectionType: VNet2VNet

#### Values for TestVNet4:

- VNet Name: TestVNet4
- TestVNet2: 10.41.0.0/16 & 10.42.0.0/16
- FrontEnd: 10.41.0.0/24
- BackEnd: 10.42.0.0/24
- GatewaySubnet: 10.42.255.0/27
- Resource Group: TestRG4
- Location: West US
- GatewayName: VNet4GW
- Public IP: VNet4GWIP
- VPNTYPE: RouteBased
- Connection: VNet4toVNet1
- ConnectionType: VNet2VNet

#### Step 2 - Create and configure TestVNet1

1. Verify your subscription settings.

Connect to your account if you are running PowerShell locally on your computer. If you are using Azure Cloud Shell, you are connected automatically.

```
Connect-AzAccount
```

Check the subscriptions for the account.

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName nameofsubscription
```

2. Declare your variables. This example declares the variables using the values for this exercise. In most cases, you should replace the values with your own. However, you can use these variables if you are running through the steps to become familiar with this type of configuration. Modify the variables if needed, then copy and paste them into your PowerShell console.

```
$RG1 = "TestRG1"
$Location1 = "East US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$VNetPrefix11 = "10.11.0.0/16"
$VNetPrefix12 = "10.12.0.0/16"
$FESubPrefix1 = "10.11.0.0/24"
$BESubPrefix1 = "10.12.0.0/24"
$GWSUBPREFIX1 = "10.12.255.0/27"
$GWName1 = "VNet1GW"
$GWIPName1 = "VNet1GWIP"
$GWIPconfName1 = "gwipconf1"
$Connection14 = "VNet1toVNet4"
$Connection15 = "VNet1toVNet5"
```

3. Create a resource group.

```
New-AzResourceGroup -Name $RG1 -Location $Location1
```

4. Create the subnet configurations for TestVNet1. This example creates a virtual network named TestVNet1 and three subnets, one called GatewaySubnet, one called FrontEnd, and one called Backend. When substituting values, it's important that you always name your gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation fails. For this reason, it is not assigned via variable below.

The following example uses the variables that you set earlier. In this example, the gateway subnet is using a /27. While it is possible to create a gateway subnet as small as /29, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. This will allow for enough addresses to accommodate possible additional configurations that you may want in the future.

```
$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix $GWSUBPREFIX1
```

5. Create TestVNet1.

```
New-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 `
-Location $Location1 -AddressPrefix $VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1
```

6. Request a public IP address to be allocated to the gateway you will create for your VNet. Notice that the AllocationMethod is Dynamic. You cannot specify the IP address that you want to use. It's dynamically allocated to your gateway.

```
$gwpip1 = New-AzPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 `
-Location $Location1 -AllocationMethod Dynamic
```

7. Create the gateway configuration. The gateway configuration defines the subnet and the public IP address to use. Use the example to create your gateway configuration.

```
$vnet1 = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 `
-Subnet $subnet1 -PublicIpAddress $gwpip1
```

8. Create the gateway for TestVNet1. In this step, you create the virtual network gateway for your TestVNet1. VNet-to-VNet configurations require a RouteBased VpnType. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

```
New-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 `
-Location $Location1 -IpConfigurations $gwipconf1 -GatewayType Vpn `
-VpnType RouteBased -GatewaySku VpnGw1
```

After you finish the commands, it will take up to 45 minutes to create this gateway. If you are using Azure Cloud Shell, you can restart your Cloud Shell session by clicking in the upper left of the Cloud Shell terminal, then configure TestVNet4. You don't need to wait until the TestVNet1 gateway completes.

### Step 3 - Create and configure TestVNet4

Once you've configured TestVNet1, create TestVNet4. Follow the steps below, replacing the values with your own when needed.

1. Connect and declare your variables. Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG4 = "TestRG4"
$Location4 = "West US"
$VnetName4 = "TestVNet4"
$FESubName4 = "FrontEnd"
$BESubName4 = "Backend"
$VnetPrefix41 = "10.41.0.0/16"
$VnetPrefix42 = "10.42.0.0/16"
$FESubPrefix4 = "10.41.0.0/24"
$BESubPrefix4 = "10.42.0.0/24"
$GWSubPrefix4 = "10.42.255.0/27"
$GWName4 = "VNet4GW"
$GWIPName4 = "VNet4GWIP"
$GWIPconfName4 = "gwipconf4"
$Connection41 = "VNet4toVNet1"
```

2. Create a resource group.

```
New-AzResourceGroup -Name $RG4 -Location $Location4
```

3. Create the subnet configurations for TestVNet4.

```
$fesub4 = New-AzVirtualNetworkSubnetConfig -Name $FESubName4 -AddressPrefix $FESubPrefix4
$besub4 = New-AzVirtualNetworkSubnetConfig -Name $BESubName4 -AddressPrefix $BESubPrefix4
$gwsub4 = New-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix $GWSubPrefix4
```

4. Create TestVNet4.

```
New-AzVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4 `
-Location $Location4 -AddressPrefix $VnetPrefix41,$VnetPrefix42 -Subnet $fesub4,$besub4,$gwsb4
```

5. Request a public IP address.

```
$gwpip4 = New-AzPublicIpAddress -Name $GWIPName4 -ResourceGroupName $RG4 `
-Location $Location4 -AllocationMethod Dynamic
```

6. Create the gateway configuration.

```
$vnet4 = Get-AzVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4
$subnet4 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet4
$gwipconf4 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName4 -Subnet $subnet4 -
PublicIpAddress $gwpip4
```

7. Create the TestVNet4 gateway. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

```
New-AzVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4 `
-Location $Location4 -IpConfigurations $gwipconf4 -GatewayType Vpn `
-VpnType RouteBased -GatewaySku VpnGw1
```

#### Step 4 - Create the connections

Wait until both gateways are completed. Restart your Azure Cloud Shell session and copy and paste the variables from the beginning of Step 2 and Step 3 into the console to redeclare values.

1. Get both virtual network gateways.

```
$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet4gw = Get-AzVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4
```

2. Create the TestVNet1 to TestVNet4 connection. In this step, you create the connection from TestVNet1 to TestVNet4. You'll see a shared key referenced in the examples. You can use your own values for the shared key. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

```
New-AzVirtualNetworkGatewayConnection -Name $Connection14 -ResourceGroupName $RG1 `
-VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet4gw -Location $Location1 `
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

3. Create the TestVNet4 to TestVNet1 connection. This step is similar to the one above, except you are creating the connection from TestVNet4 to TestVNet1. Make sure the shared keys match. The connection will be established after a few minutes.

```
New-AzVirtualNetworkGatewayConnection -Name $Connection41 -ResourceGroupName $RG4 `
-VirtualNetworkGateway1 $vnet4gw -VirtualNetworkGateway2 $vnet1gw -Location $Location4 `
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

4. Verify your connection. See the section [How to verify your connection](#).

## How to connect VNets that are in different subscriptions

In this scenario, you connect TestVNet1 and TestVNet5. TestVNet1 and TestVNet5 reside in different subscriptions. The subscriptions do not need to be associated with the same Active Directory tenant.

The difference between these steps and the previous set is that some of the configuration steps need to be performed in a separate PowerShell session in the context of the second subscription. Especially when the two subscriptions belong to different organizations.

Due to changing subscription context in this exercise, you may find it easier to use PowerShell locally on your computer, rather than using the Azure Cloud Shell, when you get to Step 8.

### Step 5 - Create and configure TestVNet1

You must complete [Step 1](#) and [Step 2](#) from the previous section to create and configure TestVNet1 and the VPN Gateway for TestVNet1. For this configuration, you are not required to create TestVNet4 from the previous section, although if you do create it, it will not conflict with these steps. Once you complete Step 1 and Step 2, continue with Step 6 to create TestVNet5.

### Step 6 - Verify the IP address ranges

It is important to make sure that the IP address space of the new virtual network, TestVNet5, does not overlap with any of your VNet ranges or local network gateway ranges. In this example, the virtual networks may belong to different organizations. For this exercise, you can use the following values for the TestVNet5:

**Values for TestVNet5:**

- VNet Name: TestVNet5
- Resource Group: TestRG5
- Location: Japan East
- TestVNet5: 10.51.0.0/16 & 10.52.0.0/16
- FrontEnd: 10.51.0.0/24
- BackEnd: 10.52.0.0/24
- GatewaySubnet: 10.52.255.0.0/27
- GatewayName: VNet5GW
- Public IP: VNet5GWIP
- VPNTYPE: RouteBased
- Connection: VNet5toVNet1
- ConnectionType: VNet2VNet

### Step 7 - Create and configure TestVNet5

This step must be done in the context of the new subscription. This part may be performed by the administrator in a different organization that owns the subscription.

1. Declare your variables. Be sure to replace the values with the ones that you want to use for your configuration.

```
$Sub5 = "Replace_With_the_New_Subscription_Name"
$RG5 = "TestRG5"
$Location5 = "Japan East"
$VnetName5 = "TestVNet5"
$FESubName5 = "FrontEnd"
$BESubName5 = "Backend"
$GWSubName5 = "GatewaySubnet"
$VnetPrefix51 = "10.51.0.0/16"
$VnetPrefix52 = "10.52.0.0/16"
$FESubPrefix5 = "10.51.0.0/24"
$BESubPrefix5 = "10.52.0.0/24"
$GWSubPrefix5 = "10.52.255.0/27"
$GWName5 = "VNet5GW"
$GWIPName5 = "VNet5GWIP"
$GWIPconfName5 = "gwipconf5"
$Connection51 = "VNet5toVNet1"
```

2. Connect to subscription 5. Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzAccount
```

Check the subscriptions for the account.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName $Sub5
```

3. Create a new resource group.

```
New-AzResourceGroup -Name $RG5 -Location $Location5
```

4. Create the subnet configurations for TestVNet5.

```
$fesub5 = New-AzVirtualNetworkSubnetConfig -Name $FESubName5 -AddressPrefix $FESubPrefix5
$besub5 = New-AzVirtualNetworkSubnetConfig -Name $BESubName5 -AddressPrefix $BESubPrefix5
$gwsb5 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName5 -AddressPrefix $GWSubPrefix5
```

5. Create TestVNet5.

```
New-AzVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5 -Location $Location5 `
-AddressPrefix $VnetPrefix51,$VnetPrefix52 -Subnet $fesub5,$besub5,$gwsb5
```

6. Request a public IP address.

```
$gwip5 = New-AzPublicIpAddress -Name $GWIPName5 -ResourceGroupName $RG5 `
-Location $Location5 -AllocationMethod Dynamic
```

7. Create the gateway configuration.

```
$vnet5 = Get-AzVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5
$subnet5 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet5
$gwpipconf5 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName5 -Subnet $subnet5 -
PublicIpAddress $gwpip5
```

## 8. Create the TestVNet5 gateway.

```
New-AzVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5 -Location $Location5 ` -IpConfigurations $gwpipconf5 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1
```

### Step 8 - Create the connections

In this example, because the gateways are in the different subscriptions, we've split this step into two PowerShell sessions marked as [Subscription 1] and [Subscription 5].

1. [Subscription 1] Get the virtual network gateway for Subscription 1. Sign in and connect to Subscription 1 before running the following example:

```
$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
```

Copy the output of the following elements and send these to the administrator of Subscription 5 via email or another method.

```
$vnet1gw.Name
$vnet1gw.Id
```

These two elements will have values similar to the following example output:

```
PS D:\> $vnet1gw.Name
VNet1GW
PS D:\> $vnet1gw.Id
/subscriptions/b636ca99-6f88-4df4-a7c3-
2f8dc4545509/resourceGroupsTestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW
```

2. [Subscription 5] Get the virtual network gateway for Subscription 5. Sign in and connect to Subscription 5 before running the following example:

```
$vnet5gw = Get-AzVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5
```

Copy the output of the following elements and send these to the administrator of Subscription 1 via email or another method.

```
$vnet5gw.Name
$vnet5gw.Id
```

These two elements will have values similar to the following example output:

```
PS C:\> $vnet5gw.Name
VNet5GW
PS C:\> $vnet5gw.Id
/subscriptions/66c8e4f1-ecd6-47ed-9de7-
7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW
```

3. [Subscription 1] Create the TestVNet1 to TestVNet5 connection. In this step, you create the connection from TestVNet1 to TestVNet5. The difference here is that \$vnet5gw cannot be obtained directly because it is in a different subscription. You will need to create a new PowerShell object with the values communicated from Subscription 1 in the steps above. Use the example below. Replace the Name, ID, and shared key with your own values. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

Connect to Subscription 1 before running the following example:

```
$vnet5gw = New-Object -TypeName Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet5gw.Name = "VNet5GW"
$vnet5gw.Id = "/subscriptions/66c8e4f1-ecd6-47ed-9de7-7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW"
$Connection15 = "VNet1toVNet5"
New-AzVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -
VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet5gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

4. [Subscription 5] Create the TestVNet5 to TestVNet1 connection. This step is similar to the one above, except you are creating the connection from TestVNet5 to TestVNet1. The same process of creating a PowerShell object based on the values obtained from Subscription 1 applies here as well. In this step, be sure that the shared keys match.

Connect to Subscription 5 before running the following example:

```
$vnet1gw = New-Object -TypeName Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet1gw.Name = "VNet1GW"
$vnet1gw.Id = "/subscriptions/b636ca99-6f88-4df4-a7c3-2f8dc4545509/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW"
$Connection51 = "VNet5toVNet1"
New-AzVirtualNetworkGatewayConnection -Name $Connection51 -ResourceGroupName $RG5 -
VirtualNetworkGateway1 $vnet5gw -VirtualNetworkGateway2 $vnet1gw -Location $Location5 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

## How to verify a connection

### IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

You can verify that your connection succeeded by using the 'Get-AzVirtualNetworkGatewayConnection' cmdlet, with or without '-Debug'.

1. Use the following cmdlet example, configuring the values to match your own. If prompted, select 'A' in order to run 'All'. In the example, '-Name' refers to the name of the connection that you want to test.

```
Get-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. After the cmdlet has finished, view the values. In the example below, the connection status shows as 'Connected' and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

## VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN gateway connections. For information about VNet peering, see [Virtual network peering](#).

### **Does Azure charge for traffic between VNets?**

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

### **Does VNet-to-VNet traffic travel across the internet?**

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

### **Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?**

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

### **Is VNet-to-VNet traffic secure?**

Yes, it's protected by IPsec/IKE encryption.

### **Do I need a VPN device to connect VNets together?**

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

### **Do my VNets need to be in the same region?**

No. The virtual networks can be in the same or different Azure regions (locations).

### **If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?**

No.

### **Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?**

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

### **Can I use VNet-to-VNet along with multi-site connections?**

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

### **How many on-premises sites and virtual networks can one virtual network connect to?**

See the [Gateway requirements](#) table.

### **Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?**

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

### **Can a cloud service or a load-balancing endpoint span VNets?**

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

### **Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?**

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

**Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?**

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

**Do VPN tunnels share bandwidth?**

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

**Are redundant tunnels supported?**

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

**Can I have overlapping address spaces for VNet-to-VNet configurations?**

No. You can't have overlapping IP address ranges.

**Can there be overlapping address spaces among connected virtual networks and on-premises local sites?**

No. You can't have overlapping IP address ranges.

## Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. See the [Virtual Machines documentation](#) for more information.
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).

# Connect virtual networks from different deployment models using the portal

2/18/2021 • 21 minutes to read • [Edit Online](#)

This article shows you how to connect classic VNets to Resource Manager VNets to allow the resources located in the separate deployment models to communicate with each other. The steps in this article primarily use the Azure portal, but you can also create this configuration using the PowerShell by selecting the article from this list.

Connecting a classic VNet to a Resource Manager VNet is similar to connecting a VNet to an on-premises site location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE. You can create a connection between VNets that are in different subscriptions and in different regions. You can also connect VNets that already have connections to on-premises networks, as long as the gateway that they have been configured with is dynamic or route-based. For more information about VNet-to-VNet connections, see the [VNet-to-VNet FAQ](#) at the end of this article.

If you do not already have a virtual network gateway and do not want to create one, you may want to instead consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway. For more information, see [VNet peering](#).

## Before you begin

- These steps assume that both VNets have already been created. If you are using this article as an exercise and don't have VNets, there are links in the steps to help you create them.
- Verify that the address ranges for the VNets do not overlap with each other, or overlap with any of the ranges for other connections that the gateways may be connected to.
- Install the latest PowerShell cmdlets for both Resource Manager and Service Management (classic). In this article, we use both the Azure portal and PowerShell. PowerShell is required to create the connection from the classic VNet to the Resource Manager VNet. For more information, see [How to install and configure Azure PowerShell](#).

## Example settings

You can use these values to create a test environment, or refer to them to better understand the examples in this article.

### Classic VNet

VNet name = ClassicVNet

Address space = 10.0.0.0/24

Subnet name = Subnet-1

Subnet address range = 10.0.0.0/27

Subscription = the subscription you want to use

Resource Group = ClassicRG

Location = West US

GatewaySubnet = 10.0.0.32/28

Local site = RMVNetLocal

### Resource Manager VNet

VNet name = RMVNet

Address space = 192.168.0.0/16

Resource Group = RG1  
 Location = East US  
 Subnet name = Subnet-1  
 Address range = 192.168.1.0/24  
 GatewaySubnet = 192.168.0.0/26  
 Virtual network gateway name = RMGateway  
 Gateway type = VPN  
 VPN type = Route-based  
 SKU = VpnGw1  
 Location = East US  
 Virtual network = RMVNet  
 (associate the VPN gateway to this VNet) First IP configuration = rmgwpip  
 (gateway public IP address) Local network gateway = ClassicVNetLocal  
 Connection name = RMtoClassic

### Connection overview

For this configuration, you create a VPN gateway connection over an IPsec/IKE VPN tunnel between the virtual networks. Make sure that none of your VNet ranges overlap with each other, or with any of the local networks that they connect to.

The following table shows an example of how the example VNets and local sites are defined:

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE
ClassicVNet	(10.0.0.0/24)	West US	RMVNetLocal (192.168.0.0/16)
RMVNet	(192.168.0.0/16)	East US	ClassicVNetLocal (10.0.0.0/24)

## Section 1 - Configure the classic VNet settings

In this section, you create the classic VNet, the local network (local site), and the virtual network gateway. Screenshots are provided as examples. Be sure to replace the values with your own, or use the [Example](#) values.

### 1. Create a classic VNet

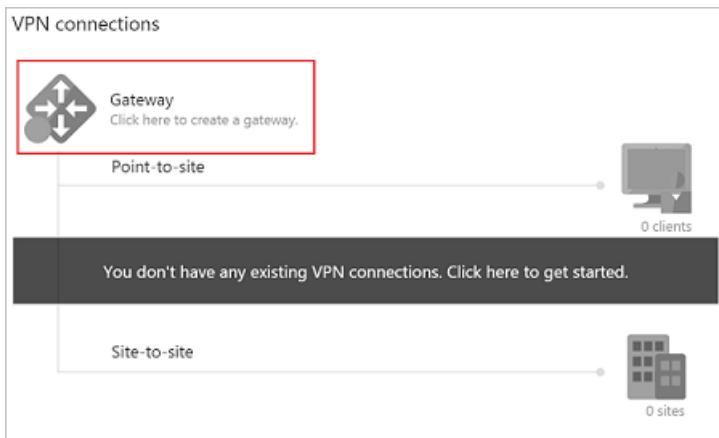
If you don't have a classic VNet and are running these steps as an exercise, you can create a VNet by using [this article](#) and the [Example](#) settings values from above.

If you already have a VNet with a VPN gateway, verify that the gateway is Dynamic. If it's Static, you must first delete the VPN gateway before you proceed to [Configure the local site](#).

1. Open the [Azure portal](#) and sign in with your Azure account.
2. Click **+ Create a resource** to open the 'New' page.
3. In the 'Search the marketplace' field, type 'Virtual Network'. If you instead, select Networking -> Virtual Network, you will not get the option to create a classic VNet.
4. Locate 'Virtual Network' from the returned list and click it to open the Virtual Network page.
5. On the virtual network page, select 'Classic' to create a classic VNet. If you take the default here, you will wind up with a Resource Manager VNet instead.

### 2. Configure the local site

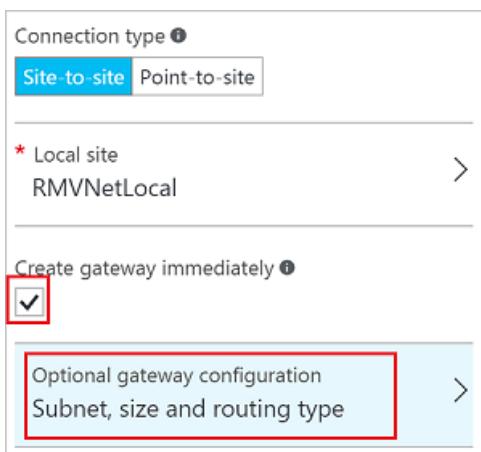
1. Navigate to **All resources** and locate the **ClassicVNet** in the list.
2. Click **Gateway** in the **Settings** section of the menu, and then click on the banner to create a gateway.



3. On the **New VPN Connection** page, for **Connection type**, select **Site-to-site**.
4. For **Local site**, click **Configure required settings**. This opens the **Local site** page.
5. On the **Local site** page, create a name to refer to the Resource Manager VNet. For example, 'RMVNetLocal'.
6. If the VPN gateway for the Resource Manager VNet already has a Public IP address, use the value for the **VPN gateway IP address** field. If you are doing these steps as an exercise, or don't yet have a virtual network gateway for your Resource Manager VNet, you can make up a placeholder IP address. Make sure that the placeholder IP address uses a valid format. Later, you replace the placeholder IP address with the Public IP address of the Resource Manager virtual network gateway.
7. For **Client Address Space**, use the **values** for the virtual network IP address spaces for the Resource Manager VNet. This setting is used to specify the address spaces to route to the Resource Manager virtual network. In the example, we use 192.168.0.0/16, the address range for the RMVNet.
8. Click **OK** to save the values and return to the **New VPN Connection** page.

### 3. Create the virtual network gateway

1. On the **New VPN Connection** page, select the **Create gateway immediately** checkbox.
2. Click **Optional gateway configuration** to open the **Gateway configuration** page.



3. Click **Subnet - Configure required settings** to open the **Add subnet** page. The **Name** is already configured with the required value: **GatewaySubnet**.
4. The **Address range** refers to the range for the gateway subnet. Although you can create a gateway subnet with a /29 address range (3 addresses), we recommend creating a gateway subnet that contains more IP addresses. This will accommodate future configurations that may require more available IP addresses. If possible, use /27 or /28. If you are using these steps as an exercise, you can refer to the **Example values**. For this example, we use '10.0.0.32/28'. Click **OK** to create the gateway subnet.
5. On the **Gateway configuration** page, **Size** refers to the gateway SKU. Select the gateway SKU for your VPN gateway.
6. Verify the **Routing Type** is **Dynamic**, then click **OK** to return to the **New VPN Connection** page.

7. On the **New VPN Connection** page, click **OK** to begin creating your VPN gateway. Creating a VPN gateway can take up to 45 minutes to complete.

#### 4. Copy the virtual network gateway Public IP address

After the virtual network gateway has been created, you can view the gateway IP address.

1. Navigate to your classic VNet, and click **Overview**.
2. Click **VPN connections** to open the VPN connections page. On the VPN connections page, you can view the Public IP address. This is the Public IP address assigned to your virtual network gateway. Make a note of the IP address. You use it in later steps when you work with your Resource Manager local network gateway configuration settings.
3. You can view the status of your gateway connections. Notice the local network site you created is listed as 'Connecting'. The status will change after you have created your connections. You can close this page when you are finished viewing the status.

## Section 2 - Configure the Resource Manager VNet settings

In this section, you create the virtual network gateway and the local network gateway for your Resource Manager VNet. Screenshots are provided as examples. Be sure to replace the values with your own, or use the [Example](#) values.

### 1. Create a virtual network

Example values:

- VNet name = RMVNet
- Address space = 192.168.0.0/16
- Resource Group = RG1
- Location = East US
- Subnet name = Subnet-1
- Address range = 192.168.1.0/24

If you don't have a Resource Manager VNet and are running these steps as an exercise, create a virtual network with the steps in [Create a virtual network](#), using the example values.

### 2. Create a virtual network gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

Example values:

- Virtual network gateway name = RMGateway
- Gateway type = VPN
- VPN type = Route-based
- SKU = VpnGw1
- Location = East US
- Virtual network = RMVNet
- GatewaySubnet = 192.168.0.0/26
- First IP configuration = rmgwpip

1. From the [Azure portal](#), in **Search resources, services, and docs (G+)** type **virtual network gateway**. Locate **Virtual network gateway** in the search results and select it.

2. On the **Virtual network gateway** page, select **+ Add**. This opens the **Create virtual network gateway** page.

3. On the **Basics** tab, fill in the values for your virtual network gateway.

[Home](#) > [New](#) > [Virtual network gateway](#) >

## Create virtual network gateway

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group ⓘ

(derived from virtual network's resource group)

### Instance details

Name \*

 ✓

Region \*

 ✓

Gateway type \* ⓘ

 VPN  ExpressRoute

VPN type \* ⓘ

 Route-based  Policy-based

SKU \* ⓘ

 ✓

Generation ⓘ

 ✓

Virtual network \* ⓘ

 ✓[Create virtual network](#)

i Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range \* ⓘ

 ✓

### Public IP address

Public IP address \* ⓘ

 Create new  Use existing

Public IP address name \*

 ✓

Public IP address SKU

Basic

Assignment

 Dynamic  Static

Enable active-active mode \* ⓘ

 Enabled  Disabled

Configure BGP \* ⓘ

 Enabled  Disabled

Azure recommends using a validated VPN device with your virtual network gateway. To view a list of validated devices and instructions for configuration, refer to Azure's [documentation](#) regarding validated VPN devices.

[Review + create](#)[Previous](#)[Next : Tags >](#)[Download a template for automation](#)

### Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

### Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.

- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).
- **Generation:** For information about VPN Gateway Generation, see [Gateway SKUs](#).
- **Virtual network:** From the dropdown, select the virtual network to which you want to add this gateway.
- **Gateway subnet address range:** This field only appears if your VNet doesn't have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.). We don't recommend creating a range any smaller than /28. If you already have a gateway subnet, you can view **GatewaySubnet** details by navigating to your virtual network. Click **Subnets** to view the range. If you want to change the range, you can delete and recreate the **GatewaySubnet**.

#### Public IP address

This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only **Dynamic**.
- **Enable active-active mode:** Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting **Disabled**.
- Leave **Configure BGP** as **Disabled**, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

4. Select **Review + create** to run validation.

5. Once validation passes, select **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

#### IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

### 3. Create a local network gateway

Example values: Local network gateway = ClassicVNetLocal

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE	GATEWAY PUBLIC IP ADDRESS

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE	GATEWAY PUBLIC IP ADDRESS
ClassicVNet	(10.0.0.0/24)	West US	RMVNetLocal (192.168.0.0/16)	The Public IP address that is assigned to the ClassicVNet gateway
RMVNet	(192.168.0.0/16)	East US	ClassicVNetLocal (10.0.0.0/24)	The Public IP address that is assigned to the RMVNet gateway.

The local network gateway specifies the address range and the Public IP address associated with your classic VNet and its virtual network gateway. If you are doing these steps as an exercise, refer to the Example values.

1. From the [Azure portal](#), in **Search resources, services, and docs (G+)** type **local network gateway**. Locate **local network gateway** under **Marketplace** in the search results and select it. This opens the **Create local network gateway** page.
2. On the **Create local network gateway** page, specify the values for your local network gateway.

## Create local network gateway

Name \*

Endpoint ⓘ

IP address     FQDN

IP address \* ⓘ

Address space ⓘ

...	
...	
Add additional address range	...

Configure BGP settings

Subscription \*

Resource group \* ⓘ

[Create new](#)

Location \*

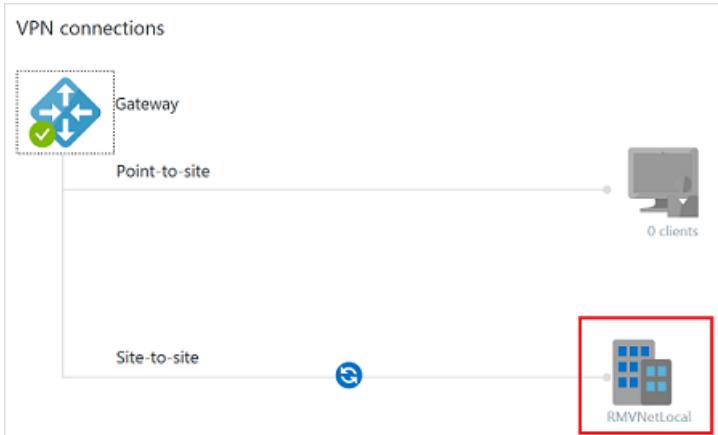
- **Name:** Specify a name for your local network gateway object.
- **Endpoint:** Select the endpoint type for the on-premises VPN device - **IP address or FQDN (Fully Qualified Domain Name)**.
  - **IP address:** If you have a static public IP address allocated from your Internet service provider for your VPN device, select the IP address option and fill in the IP address as shown in the example. This is the public IP address of the VPN device that you want Azure VPN gateway to connect to. If you don't have the IP address right now, you can use the values shown in the example, but you'll need to go back and replace your placeholder IP address with the public IP address of your VPN device. Otherwise, Azure will not be able to connect.
- **Address Space** refers to the address ranges for the network that this local network represents. You can add multiple address space ranges. Make sure that the ranges you specify here do not overlap with ranges of other networks that you want to connect to. Azure will route the address range that you specify to the on-premises VPN device IP address. *Use your own values here if you want to connect to your on-premises site, not the values shown in the example.*
- **Configure BGP settings:** Use only when configuring BGP. Otherwise, don't select this.
- **Subscription:** Verify that the correct subscription is showing.
- **Resource Group:** Select the resource group that you want to use. You can either create a new resource group, or select one that you have already created.
- **Location:** The location is the same as **Region** in other settings. Select the location that this object will be created in. You may want to select the same location that your VNet resides in, but you are not required to do so.

3. When you have finished specifying the values, select the **Create** button at the bottom of the page to create the local network gateway.

## Section 3 - Modify the classic VNet local site settings

In this section, you replace the placeholder IP address that you used when specifying the local site settings, with the Resource Manager VPN gateway IP address. This section uses the classic (SM) PowerShell cmdlets.

1. In the Azure portal, navigate to the classic virtual network.
2. On the page for your virtual network, click **Overview**.
3. In the **VPN connections** section, click the name of your local site in the graphic.



4. On the **Site-to-site VPN connections** page, click the name of the site.

Site-to-site VPN connections			
ClassicVNet1			
NAME		STATUS	
RMVNetLocal		Connecting	1/9/2017 4:18:08 PM 0 B / 0 B

5. On the connection page for your local site, click the name of the local site to open the Local site page.

RMVNetLocal

Site-to-site VPN connections

Connect Disconnect Delete

Local site RMVNetLocal

STATUS  
Connecting

LAST EVENT TIME STAMP  
1/9/2017 4:18:08 PM

LAST EVENT ID  
24401

LAST EVENT MESSAGE  
The connectivity state for the local network site 'D8E4CA69\_RMVNetLocal' changed from Initializing to Connecting.

DATA IN / OUT  
0 B / 0 B

6. On the Local site page, replace the **VPN gateway IP address** with the IP address of the Resource Manager gateway.

Local site

\* Name  
RMVNetLocal

\* VPN gateway IP address

CLIENT ADDRESS SPACE

7. Click OK to update the IP address.

## Section 4 - Create Resource Manager to classic connection

In these steps, you configure the connection from the Resource Manager VNet to the classic VNet using the Azure portal.

- In **All resources**, locate the local network gateway. In our example, the local network gateway is **ClassicVNetLocal**.
- Click **Configuration** and verify that the IP address value is the VPN gateway for the classic VNet. Update, if needed, then click **Save**. Close the page.
- In **All resources**, click the local network gateway.
- Click **Connections** to open the Connections page.
- On the **Connections** page, click **+** to add a connection.

6. On the **Add connection** page, name the connection. For example, 'RMtoClassic'.
7. **Site-to-Site** is already selected on this page.
8. Select the virtual network gateway that you want to associate with this site.
9. Create a **shared key**. This key is also used in the connection that you create from the classic VNet to the Resource Manager VNet. You can generate the key or make one up. In our example, we use 'abc123', but you can (and should) use something more complex.
10. Click **OK** to create the connection.

## Section 5 - Create classic to Resource Manager connection

In these steps, you configure the connection from the classic VNet to the Resource Manager VNet. These steps require PowerShell. You can't create this connection in the portal. Make sure you have downloaded and installed both the classic (SM) and Resource Manager (RM) PowerShell cmdlets.

### 1. Connect to your Azure account

Open the PowerShell console with elevated rights and log in to your Azure account. After logging in, your account settings are downloaded so that they are available to Azure PowerShell. The following cmdlet prompts you for the login credentials for your Azure Account for the Resource Manager deployment model:

```
Connect-AzAccount
```

Get a list of your Azure subscriptions.

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Name of subscription"
```

Next, log in to use the classic PowerShell cmdlets (Service Management). Use the following command to add your Azure account for the classic deployment model:

```
Add-AzureAccount
```

Get a list of your subscriptions. This step may be necessary when adding the Service Management cmdlets, depending on your Azure module install.

```
Get-AzureSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionName "Name of subscription"
```

### 2. View the network configuration file values

When you create a VNet in the Azure portal, the full name that Azure uses is not visible in the Azure portal. For example, a VNet that appears to be named 'ClassicVNet' in the Azure portal may have a much longer name in the network configuration file. The name might look something like: 'Group ClassicRG ClassicVNet'. In these steps, you download the network configuration file and view the values.

Create a directory on your computer and then export the network configuration file to the directory. In this

example, the network configuration file is exported to C:\AzureNet.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

Open the file with a text editor and view the name for your classic VNet. Use the names in the network configuration file when running your PowerShell cmdlets.

- VNet names are listed as **VirtualNetworkSite name =**
- Site names are listed as **LocalNetworkSite name=**

### 3. Create the connection

Set the shared key and create the connection from the classic VNet to the Resource Manager VNet. You cannot set the shared key using the portal. Make sure you run these steps while logged in using the classic version of the PowerShell cmdlets. To do so, use **Add-AzureAccount**. Otherwise, you will not be able to set the '**-AzureVNetGatewayKey**'.

- In this example, **-VNetName** is the name of the classic VNet as found in your network configuration file.
- The **-LocalNetworkSiteName** is the name you specified for the local site, as found in your network configuration file.
- The **-SharedKey** is a value that you generate and specify. For this example, we used *abc123*, but you can generate something more complex. The important thing is that the value you specify here must be the same value that you specified when creating your Resource Manager to classic connection.

```
Set-AzureVNetGatewayKey -VNetName "Group ClassicRG ClassicVNet" `
-LocalNetworkSiteName "172B9E16_RMVNetLocal" -SharedKey abc123
```

## Section 6 - Verify your connections

You can verify your connections by using the Azure portal or PowerShell. When verifying, you may need to wait a minute or two as the connection is being created. When a connection is successful, the connectivity state changes from 'Connecting' to 'Connected'.

### To verify the connection from your classic VNet to your Resource Manager VNet

In the Azure portal, you can view the connection status for a classic VNet VPN Gateway by navigating to the connection. The following steps show one way to navigate to your connection and verify.

1. In the [Azure portal](#), click **All resources** and navigate to your classic virtual network (VNet).
2. On the virtual network page, select the type of connection that you want to see. For example, **Site-to-site connections**.

The screenshot shows the Azure portal interface for managing a classic virtual network named 'TestVNet1'. The user has selected the 'Site-to-site connections' tab under the 'Virtual Network (classic)' section. A single connection entry is displayed: 'Site/Net4' with a status of 'Not connected'. The connection was last updated on 10/15/2020 at 10:20:52 AM. The data in/out is shown as 0 B / 0 B. The left sidebar shows other navigation options like Overview, Activity log, Access control (IAM), Settings (Address space, Subnets, DNS servers, Diagram, Gateway), and Site-to-site connections (which is currently selected).

3. On the **Site-to-site connections** page, under **Name**, select the site connection you want to view.

**TestVNet4 | Site-to-site connections**

Virtual network (classic)

Search (Ctrl+ /) Add Refresh VPN Device Script

Overview Activity log Access control (IAM)

Name ↑↓

**SiteVNet1**

**Settings**

- Address space
- Subnets
- DNS servers
- Diagram
- Gateway
- Site-to-site connections**

- On the Properties page, view the information about the connection.

#### To verify the connection from your Resource Manager VNet to your classic VNet

In the Azure portal, you can view the connection status of a Resource Manager VPN Gateway by navigating to the connection. The following steps show one way to navigate to your connection and verify.

- In the [Azure portal](#), click **All resources** and navigate to your virtual network gateway.
- On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.

**VNet4GW | Connections**

Name	Status	Connection type	Peer
VNet4-VNet1	Connected	VNet-to-VNet	VNet1GW
VNet1-VNet4	Connected	VNet-to-VNet	VNet1GW

- Click the name of the connection that you want to verify. In **Essentials**, you can view more information about your connection. The **Status** values are 'Succeeded' and 'Connected' when you have made a successful connection.

**VNet1-VNet4**

Connection

Resource group (change)	TestRG1
Status	Connected
Location	East US
Subscription (change)	Content Development
Subscription ID	:
Tags (Change)	: Click here to add tags

Data in : 0 B  
Data out : 0 B  
Virtual network : VNet1, VNet4  
Virtual network gateway 1 : VNet1GW  
Virtual network gateway 2 : VNet4GW

## VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN gateway connections. For information about VNet peering, see [Virtual network peering](#).

**Does Azure charge for traffic between VNets?**

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

**Does VNet-to-VNet traffic travel across the internet?**

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

**Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?**

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

**Is VNet-to-VNet traffic secure?**

Yes, it's protected by IPsec/IKE encryption.

**Do I need a VPN device to connect VNets together?**

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

**Do my VNets need to be in the same region?**

No. The virtual networks can be in the same or different Azure regions (locations).

**If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?**

No.

**Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?**

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

**Can I use VNet-to-VNet along with multi-site connections?**

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

**How many on-premises sites and virtual networks can one virtual network connect to?**

See the [Gateway requirements](#) table.

**Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?**

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

**Can a cloud service or a load-balancing endpoint span VNets?**

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

**Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?**

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

**Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?**

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

**Do VPN tunnels share bandwidth?**

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

**Are redundant tunnels supported?**

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

**Can I have overlapping address spaces for VNet-to-VNet configurations?**

No. You can't have overlapping IP address ranges.

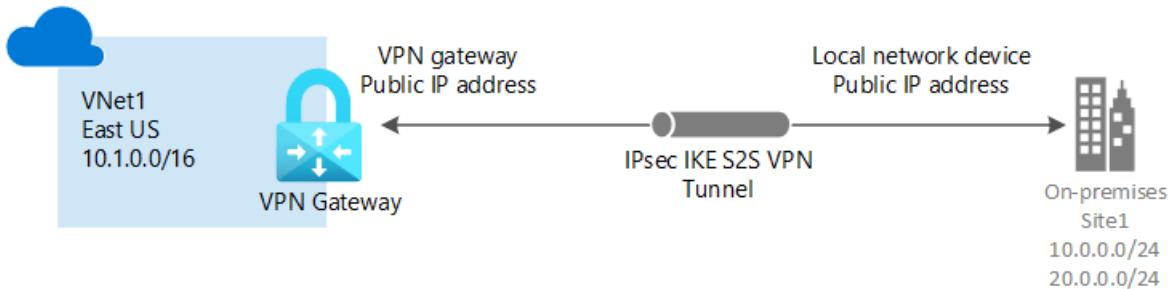
**Can there be overlapping address spaces among connected virtual networks and on-premises local sites?**

No. You can't have overlapping IP address ranges.

# Tutorial: Create a Site-to-Site connection in the Azure portal

2/18/2021 • 19 minutes to read • [Edit Online](#)

Azure VPN gateways provide cross-premises connectivity between customer premises and Azure. This tutorial shows you how to use the Azure portal to create a Site-to-Site VPN gateway connection from your on-premises network to the VNet. You can also create this configuration using [Azure PowerShell](#) or [Azure CLI](#).



In this tutorial, you learn how to:

- Create a virtual network
- Create a VPN gateway
- Create a local network gateway
- Create a VPN connection
- Verify the connection
- Connect to a virtual machine

## Prerequisites

- An Azure account with an active subscription. If you don't have one, [create one for free](#).
- Make sure you have a compatible VPN device and someone who is able to configure it. For more information about compatible VPN devices and device configuration, see [About VPN Devices](#).
- Verify that you have an externally facing public IPv4 address for your VPN device.
- If you are unfamiliar with the IP address ranges located in your on-premises network configuration, you need to coordinate with someone who can provide those details for you. When you create this configuration, you must specify the IP address range prefixes that Azure will route to your on-premises location. None of the subnets of your on-premises network can overlap with the virtual network subnets that you want to connect to.

## Create a virtual network

Create a virtual network (VNet) using the following values:

- **Resource group:** TestRG1
- **Name:** VNet1
- **Region:** (US) East US
- **IPv4 address space:** 10.1.0.0/16
- **Subnet name:** FrontEnd
- **Subnet address space:** 10.1.0.0/24

## NOTE

When using a virtual network as part of a cross-premises architecture, be sure to coordinate with your on-premises network administrator to carve out an IP address range that you can use specifically for this virtual network. If a duplicate address range exists on both sides of the VPN connection, traffic will route in an unexpected way. Additionally, if you want to connect this virtual network to another virtual network, the address space cannot overlap with the other virtual network. Plan your network configuration accordingly.

1. Sign in to the [Azure portal](#).
2. In **Search resources, service, and docs (G+/)**, type *virtual network*.

The screenshot shows the Microsoft Azure portal interface. At the top, there is a navigation bar with a search bar containing the placeholder text "Search resources, services, and docs (G+/-)". Below the search bar is a "Microsoft Azure" logo and several icons for different services: Create a resource, Resource groups, All resources, Virtual network gateways, Virtual machines, Virtual WANs, App Services, and Storage accounts. A large blue arrow points downwards to a "More services" link.

3. Select **Virtual Network** from the **Marketplace** results.

The screenshot shows the search results for "virtual network". On the left, there is a sidebar with sections for "Azure", "Create", "Recent", and "Name". The main search results area has two columns. The left column under "Azure" includes links for Virtual networks, Virtual network gateways, Virtual networks (classic), Private DNS zones, Virtual machines, Network interfaces, Network Watcher, Virtual clusters, and Virtual WANs. The right column under "Marketplace" includes a link for Virtual Network, which is highlighted with a red box. Other links in the Marketplace column include Virtual network gateway, Documentation, About Azure VPN Gateway | Microsoft Docs, Azure Virtual Network Documentation - Tutorials ..., Azure Virtual Network FAQ | Microsoft Docs, Create a virtual network - quickstart - Azure portal ..., and Resource Groups.

4. On the **Virtual Network** page, select **Create**.

The screenshot shows the "Virtual Network" page. The top navigation bar includes the Microsoft Azure logo, a search bar, and a breadcrumb trail: Home > Virtual Network. The main content area has a large blue header with the text "Virtual Network" and a "Create" button. Below the header, there is a "Save for later" link, the Microsoft logo, and a note about deployment options: "Deploy with Resource Manager (change to Classic)".

5. Once you select **Create**, the **Create virtual network** page opens.
6. On the **Basics** tab, configure **Project details** and **Instance details** VNet settings.

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

**Project details**

Subscription \* ⓘ Content Development

Resource group \* ⓘ (New) TestRG1 Create new

**Instance details**

Name \* VNet1

Region \* (US) East US

When you fill in the fields, you see a green check mark when the characters you enter in the field are validated. Some values are autofilled, which you can replace with your own values:

- **Subscription:** Verify that the subscription listed is the correct one. You can change subscriptions by using the drop-down.
- **Resource group:** Select an existing resource group, or click **Create new** to create a new one. For more information about resource groups, see [Azure Resource Manager overview](#).
- **Name:** Enter the name for your virtual network.
- **Region:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will live.

7. On the **IP Addresses** tab, configure the values. The values shown in the examples below are for demonstration purposes. Adjust these values according to the settings that you require.

**Create virtual network**

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

10.1.0.0/16

Add IPv6 address space ⓘ

Subnet name	Subnet address range
FrontEnd	10.1.0.0/24

+ Add subnet Remove subnet

- **IPv4 address space:** By default, an address space is automatically created. You can click the address space to adjust it to reflect your own values. You can also add additional address spaces.
- **Subnet:** If you use the default address space, a default subnet is created automatically. If you change the address space, you need to add a subnet. Select **+ Add subnet** to open the **Add subnet** window. Configure the following settings and then select **Add** to add the values:

- **Subnet name:** In this example, we named the subnet "FrontEnd".
  - **Subnet address range:** The address range for this subnet.
8. On the **Security** tab, at this time, leave the default values:
- **DDos protection:** Basic
  - **Firewall:** Disabled
9. Select **Review + create** to validate the virtual network settings.

10. After the settings have been validated, select **Create**.

## Create a VPN gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

### About the gateway subnet

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses a /27 or /28.

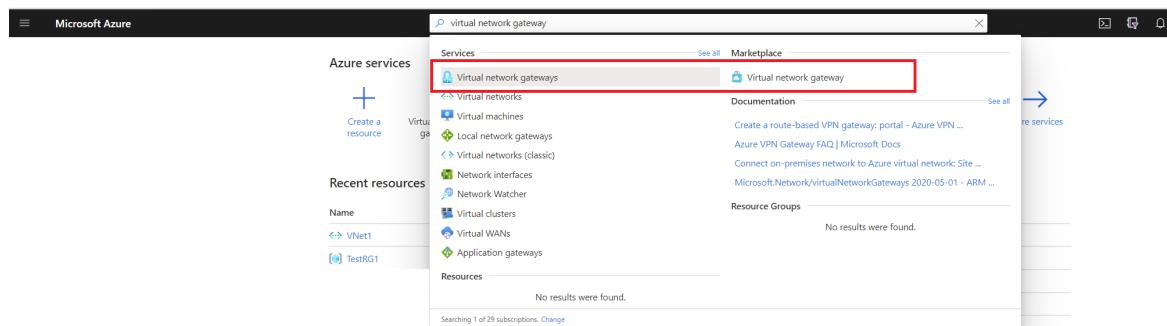
If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

### Create the gateway

Create a VPN gateway using the following values:

- **Name:** VNet1GW
- **Region:** East US
- **Gateway type:** VPN
- **VPN type:** Route-based
- **SKU:** VpnGw1
- **Generation:** Generation1
- **Virtual network:** VNet1
- **Gateway subnet address range:** 10.1.255.0/27
- **Public IP address:** Create new
- **Public IP address name:** VNet1GWpip
- **Enable active-active mode:** Disabled
- **Configure BGP:** Disabled

1. From the [Azure portal](#), in **Search resources, services, and docs (G+)** type **virtual network gateway**. Locate **Virtual network gateway** in the search results and select it.



2. On the **Virtual network gateway** page, select **+ Add**. This opens the **Create virtual network gateway** page.

[Home](#) >

## Virtual network gateways

Microsoft

**+ Add** [Edit columns](#) [Refresh](#) | [Assign](#)

**Subscriptions:** 1 of 29 selected – Don't see a subscription? [Change](#)

[C](#)

3. On the **Basics** tab, fill in the values for your virtual network gateway.

Microsoft Azure

Home > New > Virtual network gateway >

### Create virtual network gateway

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Content Development
Resource group ⓘ	TestRG1 (derived from virtual network's resource group)
<b>Instance details</b>	
Name *	VNet1GW
Region *	East US
Gateway type * ⓘ	<input checked="" type="radio"/> VPN <input type="radio"/> ExpressRoute
VPN type * ⓘ	<input checked="" type="radio"/> Route-based <input type="radio"/> Policy-based
SKU * ⓘ	VpnGw1
Generation ⓘ	Generation1

Virtual network \* ⓘ

VNet1

Create virtual network

ⓘ Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range \* ⓘ

10.1.255.0/27

10.1.255.0 - 10.1.255.31 (32 addresses)

**Public IP address**

Public IP address \* ⓘ

Create new  Use existing

Public IP address name \*

VNet1GWpip

Public IP address SKU

Basic

Assignment

Dynamic  Static

Enable active-active mode \* ⓘ

Enabled  Disabled

Configure BGP \* ⓘ

Enabled  Disabled

Azure recommends using a validated VPN device with your virtual network gateway. To view a list of validated devices and instructions for configuration, refer to Azure's [documentation](#) regarding validated VPN devices.

[Review + create](#)

[Previous](#)

[Next : Tags >](#)

[Download a template for automation](#)

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

#### Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).
- **Generation:** For information about VPN Gateway Generation, see [Gateway SKUs](#).
- **Virtual network:** From the dropdown, select the virtual network to which you want to add this gateway.
- **Gateway subnet address range:** This field only appears if your VNet doesn't have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.). We don't recommend creating a range any smaller than /28. If you already have a gateway subnet, you can view GatewaySubnet details by navigating to your virtual network. Click **Subnets** to view the range. If you want to change the range, you can delete and recreate the GatewaySubnet.

#### Public IP address

This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.

- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only Dynamic.
- **Enable active-active mode:** Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting **Disabled**.
- Leave **Configure BGP as Disabled**, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

4. Select **Review + create** to run validation.

5. Once validation passes, select **Create** to deploy the VPN gateway.

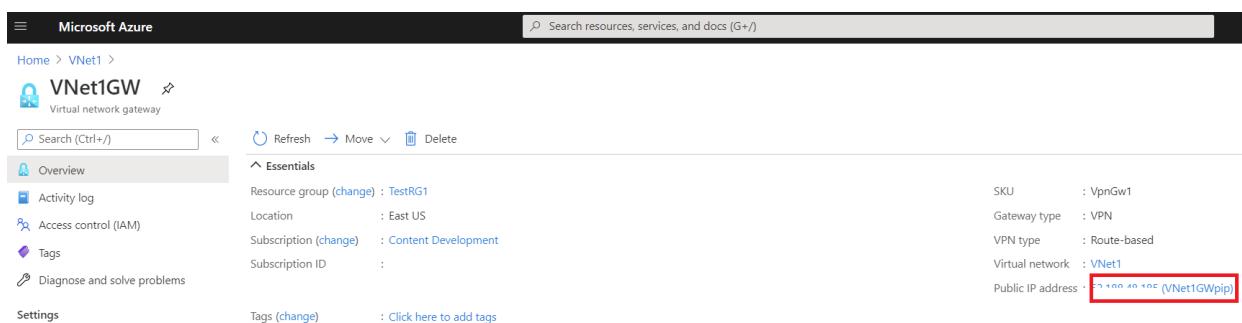
A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway. After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

#### **IMPORTANT**

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

## View the public IP address

You can view the gateway public IP address on the **Overview** page for your gateway.



The screenshot shows the Azure portal interface for a virtual network gateway named 'VNet1GW'. The 'Overview' tab is selected. Key details shown include:

- Resource group: TestRG1
- Location: East US
- Subscription: Content Development
- SKU: VpnGw1
- Gateway type: VPN
- VPN type: Route-based
- Virtual network: VNet1
- Public IP address: 52.100.40.105 (VNet1GWpip)

The 'Public IP address' field is highlighted with a red box.

To see additional information about the public IP address object, click the name/IP address link next to **Public IP address**.

## Create a local network gateway

The local network gateway is a specific object that represents your on-premises location (the site) for routing purposes. You give the site a name by which Azure can refer to it, then specify the IP address of the on-premises VPN device to which you will create a connection. You also specify the IP address prefixes that will be routed through the VPN gateway to the VPN device. The address prefixes you specify are the prefixes located on your on-premises network. If your on-premises network changes or you need to change the public IP address for the VPN device, you can easily update the values later.

Create a local network gateway using the following values:

- **Name:** Site1
- **Resource Group:** TestRG1
- **Location:** East US

1. From the [Azure portal](#), in **Search resources, services, and docs (G+)** type **local network gateway**. Locate **local network gateway** under **Marketplace** in the search results and select it. This opens the **Create local network gateway** page.

2. On the [Create local network gateway](#) page, specify the values for your local network gateway.

## Create local network gateway

Name \*

Site1

Endpoint ⓘ

IP address

FQDN

IP address \* ⓘ

4.3.2.1

Address space ⓘ

10.0.0.0/24

...

20.0.0.0/24

...

[Add additional address range](#)

...

Configure BGP settings

Subscription \*

Content Development

Resource group \* ⓘ

TestRG1

[Create new](#)

Location \*

East US

- **Name:** Specify a name for your local network gateway object.
- **Endpoint:** Select the endpoint type for the on-premises VPN device - **IP address** or **FQDN** (Fully Qualified Domain Name).
  - **IP address:** If you have a static public IP address allocated from your Internet service provider for your VPN device, select the IP address option and fill in the IP address as shown in the example. This is the public IP address of the VPN device that you want Azure VPN gateway to connect to. If you don't have the IP address right now, you can use the values shown in the example, but you'll need to go back and replace your placeholder IP address with the public IP address of your VPN device. Otherwise, Azure will not be able to connect.
  - **FQDN:** If you have a dynamic public IP address that could change after certain period of time, usually determined by your Internet service provider, you can use a constant DNS name with a Dynamic DNS service to point to your current public IP address of your VPN device. Your Azure VPN gateway will resolve the FQDN to determine the public IP address to connect to.
- **Address Space** refers to the address ranges for the network that this local network represents. You can add multiple address space ranges. Make sure that the ranges you specify here do not overlap with ranges of other networks that you want to connect to. Azure will route the address range that you specify to the on-premises VPN device IP address. *Use your own values here if you want to connect to your on-premises site, not the values shown in the example.*

- **Configure BGP settings:** Use only when configuring BGP. Otherwise, don't select this.
- **Subscription:** Verify that the correct subscription is showing.
- **Resource Group:** Select the resource group that you want to use. You can either create a new resource group, or select one that you have already created.
- **Location:** The location is the same as **Region** in other settings. Select the location that this object will be created in. You may want to select the same location that your VNet resides in, but you are not required to do so.

**NOTE**

- Azure VPN supports only one IPv4 address for each FQDN. If the domain name resolves to multiple IP addresses, Azure VPN Gateway will use the first IP address returned by the DNS servers. To eliminate the uncertainty, we recommend that your FQDN always resolve to a single IPv4 address. IPv6 is not supported.
- Azure VPN Gateway maintains a DNS cache refreshed every 5 minutes. The gateway tries to resolve the FQDNs for disconnected tunnels only. Resetting the gateway will also trigger FQDN resolution.

3. When you have finished specifying the values, select the **Create** button at the bottom of the page to create the local network gateway.

## Configure your VPN device

Site-to-Site connections to an on-premises network require a VPN device. In this step, you configure your VPN device. When configuring your VPN device, you need the following values:

- A shared key. This is the same shared key that you specify when creating your Site-to-Site VPN connection. In our examples, we use a basic shared key. We recommend that you generate a more complex key to use.
- The Public IP address of your virtual network gateway. You can view the public IP address by using the Azure portal, PowerShell, or CLI. To find the Public IP address of your VPN gateway using the Azure portal, navigate to **Virtual network gateways**, then select the name of your gateway.

### To download VPN device configuration scripts:

Depending on the VPN device that you have, you may be able to download a VPN device configuration script. For more information, see [Download VPN device configuration scripts](#).

### See the following links for additional configuration information:

- For information about compatible VPN devices, see [VPN Devices](#).
- Before configuring your VPN device, check for any [Known device compatibility issues](#) for the VPN device that you want to use.
- For links to device configuration settings, see [Validated VPN Devices](#). The device configuration links are provided on a best-effort basis. It's always best to check with your device manufacturer for the latest configuration information. The list shows the versions we have tested. If your OS is not on that list, it is still possible that the version is compatible. Check with your device manufacturer to verify that OS version for your VPN device is compatible.
- For an overview of VPN device configuration, see [Overview of 3rd party VPN device configurations](#).
- For information about editing device configuration samples, see [Editing samples](#).
- For cryptographic requirements, see [About cryptographic requirements and Azure VPN gateways](#).
- For information about IPsec/IKE parameters, see [About VPN devices and IPsec/IKE parameters for Site-to-Site VPN gateway connections](#). This link shows information about IKE version, Diffie-Hellman Group, Authentication method, encryption and hashing algorithms, SA lifetime, PFS, and DPD, in addition to

other parameter information that you need to complete your configuration.

- For IPsec/IKE policy configuration steps, see [Configure IPsec/IKE policy for S2S VPN or VNet-to-VNet connections](#).
- To connect multiple policy-based VPN devices, see [Connect Azure VPN gateways to multiple on-premises policy-based VPN devices using PowerShell](#).

## Create a VPN connection

Create the Site-to-Site VPN connection between your virtual network gateway and your on-premises VPN device.

Create a connection using the following values:

- **Local network gateway name:** Site1
  - **Connection name:** VNet1toSite1
  - **Shared key:** For this example, we use abc123. But, you can use whatever is compatible with your VPN hardware. The important thing is that the values match on both sides of the connection.
1. Open the page for your virtual network gateway. You can navigate to the gateway by going to **Name of your VNet** -> **Overview** -> **Connected devices** -> **Name of your gateway**, although there are multiple other ways to navigate as well.
  2. On the page for the gateway, select **Connections**. At the top of the Connections page, select **+Add** to open the **Add connection** page.

## Add connection



VNet1GW

Name \*

VNet1toSite1

Connection type

Site-to-site (IPsec)

\*Virtual network gateway



VNet1GW

\*Local network gateway



Site1

Shared key (PSK) \*

abc123

Use Azure Private IP Address

Enable BGP

IKE Protocol

IKEv1  IKEv2

Subscription

Content Development

Resource group

TestRG1

Create new

Location

East US

3. On the **Add connection** page, configure the values for your connection.

- **Name:** Name your connection.
- **Connection type:** Select **Site-to-site (IPSec)**.
- **Virtual network gateway:** The value is fixed because you are connecting from this gateway.
- **Local network gateway:** Select **Choose a local network gateway** and select the local network gateway that you want to use.
- **Shared Key:** the value here must match the value that you are using for your local on-premises VPN device. The example uses 'abc123', but you can (and should) use something more complex. The important thing is that the value you specify here must be the same value that you specify when configuring your VPN device.
- Leave **Use Azure Private IP Address** unchecked.
- Leave **Enable BGP** unchecked.

- Select **IKEv2**.
  - The remaining values for **Subscription**, **Resource Group**, and **Location** are fixed.
4. Select **OK** to create your connection. You'll see *Creating Connection* flash on the screen.
5. You can view the connection in the **Connections** page of the virtual network gateway. The Status will go from *Unknown* to *Connecting*, and then to *Succeeded*.

## Verify the VPN connection

In the Azure portal, you can view the connection status of a Resource Manager VPN Gateway by navigating to the connection. The following steps show one way to navigate to your connection and verify.

1. In the [Azure portal](#) menu, select **All resources** or search for and select **All resources** from any page.
2. Select to your virtual network gateway.
3. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
4. Click the name of the connection that you want to verify to open **Essentials**. In Essentials, you can view more information about your connection. The **Status** is 'Succeeded' and 'Connected' when you have made a successful connection.

→ Move     Delete     Refresh

---

Resource group ([change](#)) : [TestRG1](#)

Status	: Connected
Location	: East US
Subscription ( <a href="#">change</a> )	: <a href="#">Content Development</a>
Subscription ID	:
Tags ( <a href="#">change</a> )	: <a href="#">Click here to add tags</a>

## Connect to a virtual machine

You can connect to a VM that is deployed to your VNet by creating a Remote Desktop Connection to your VM. The best way to initially verify that you can connect to your VM is to connect by using its private IP address, rather than computer name. That way, you are testing to see if you can connect, not whether name resolution is configured properly.

1. Locate the private IP address. You can find the private IP address of a VM by either looking at the properties for the VM in the Azure portal, or by using PowerShell.
  - Azure portal - Locate your virtual machine in the Azure portal. View the properties for the VM. The private IP address is listed.
  - PowerShell - Use the example to view a list of VMs and private IP addresses from your resource groups. You don't need to modify this example before using it.

```

$VMs = Get-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
 $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
 $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
 $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
 Write-Output "$($VM.Name): $Prv,$Alloc"
}

```

2. Verify that you are connected to your VNet using the Point-to-Site VPN connection.
3. Open **Remote Desktop Connection** by typing "RDP" or "Remote Desktop Connection" in the search box on the taskbar, then select Remote Desktop Connection. You can also open Remote Desktop Connection using the 'mstsc' command in PowerShell.
4. In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

### Troubleshoot a connection

If you are having trouble connecting to a virtual machine over your VPN connection, check the following:

- Verify that your VPN connection is successful.
- Verify that you are connecting to the private IP address for the VM.
- If you can connect to the VM using the private IP address, but not the computer name, verify that you have configured DNS properly. For more information about how name resolution works for VMs, see [Name Resolution for VMs](#).
- For more information about RDP connections, see [Troubleshoot Remote Desktop connections to a VM](#).

## Optional steps

### Add additional connections to the gateway

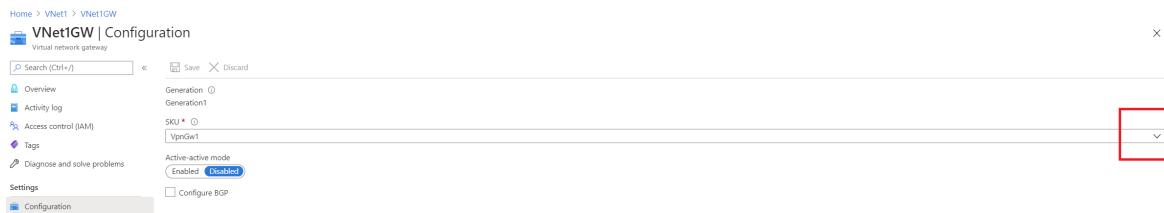
You can add additional connections, provided that none of the address spaces overlap between connections.

1. To add an additional connection, navigate to the VPN gateway, then select **Connections** to open the Connections page.
2. Select **+Add** to add your connection. Adjust the connection type to reflect either VNet-to-VNet (if connecting to another VNet gateway), or Site-to-site.
3. If you are connecting using Site-to-site and you have not already created a local network gateway for the site you want to connect to, you can create a new one.
4. Specify the shared key that you want to use, then select **OK** to create the connection.

### Resize a gateway SKU

There are specific rules regarding resizing vs. changing a gateway SKU. In this section, we will resize the SKU. For more information, see [Gateway settings - resizing and changing SKUs](#).

1. Go to the **Configuration** page for your virtual network gateway.
2. Select the arrows for the dropdown.



3. Select the SKU from the dropdown.



## Reset a gateway

Resetting an Azure VPN gateway is helpful if you lose cross-premises VPN connectivity on one or more Site-to-Site VPN tunnels. In this situation, your on-premises VPN devices are all working correctly, but are not able to establish IPsec tunnels with the Azure VPN gateways.

1. In the portal, navigate to the virtual network gateway that you want to reset.
2. On the page for the virtual network gateway, select **Reset**.

[Home > VNet1GW](#)

## VNet1GW | Reset

Virtual network gateway

 «

- █ Overview
- █ Activity log
- █ Access control (IAM)
- █ Tags
- █ Diagnose and solve problems

### Settings

- █ Configuration
- █ Connections
- █ Point-to-site configuration
- █ Properties
- █ Locks

### Monitoring

- █ Logs
- █ Alerts
- █ Metrics

### Automation

- █ Tasks (preview)
- █ Export template

### Support + troubleshooting

- █ Resource health
- █ **Reset**
- █ New support request

3. On the **Reset** page, click **Reset**. Once the command is issued, the current active instance of the Azure VPN gateway is rebooted immediately. Resetting the gateway will cause a gap in VPN connectivity, and may limit future root cause analysis of the issue.

The screenshot shows the Azure portal interface for a Virtual Network Gateway named 'VNet1GW'. At the top, there's a search bar and a breadcrumb trail: Home > VNet1GW. Below the title, there are several navigation links: Overview, Activity log, Access control (IAM), and a 'Reset' button which is highlighted with a red box. A descriptive text block explains that resetting the gateway helps if you lose cross-premises VPN connectivity, noting it will reboot the current instance and cause a temporary gap in connectivity.

## Additional configuration considerations

S2S configurations can be customized in a variety of ways. For more information, see the following articles:

- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).
- For information about forced tunneling, see [About forced tunneling](#).
- For information about Highly Available Active-Active connections, see [Highly Available cross-premises and VNet-to-VNet connectivity](#).
- For information about how to limit network traffic to resources in a virtual network, see [Network Security](#).
- For information about how Azure routes traffic between Azure, on-premises, and Internet resources, see [Virtual network traffic routing](#).

## Clean up resources

If you're not going to continue to use this application or go to the next tutorial, delete these resources using the following steps:

1. Enter the name of your resource group in the **Search** box at the top of the portal and select it from the search results.
2. Select **Delete resource group**.
3. Enter your resource group for **TYPE THE RESOURCE GROUP NAME** and select **Delete**.

## Next steps

Once you have configured a S2S connection, you can add a P2S connection to the same gateway.

[Point-to-Site VPN connections](#)

# Tutorial: Connect a virtual network to an ExpressRoute circuit using the portal

11/2/2020 • 5 minutes to read • [Edit Online](#)

This tutorial helps you create a connection to link a virtual network to an Azure ExpressRoute circuit using the Azure portal. The virtual networks that you connect to your Azure ExpressRoute circuit can either be in the same subscription or be part of another subscription.

In this tutorial, you learn how to:

- Connect a virtual networking to a circuit in the same subscription.
- Connect a virtual networking to a circuit in a different subscription.
- Delete the link between the virtual network and ExpressRoute circuit.

## Prerequisites

- Review the [prerequisites](#), [routing requirements](#), and [workflows](#) before you begin configuration.
- You must have an active ExpressRoute circuit.
  - Follow the instructions to [create an ExpressRoute circuit](#) and have the circuit enabled by your connectivity provider.
  - Ensure that you have Azure private peering configured for your circuit. See the [Create and modify peering for an ExpressRoute circuit](#) article for peering and routing instructions.
  - Ensure that Azure private peering gets configured and establishes BGP peering between your network and Microsoft for end-to-end connectivity.
  - Ensure that you have a virtual network and a virtual network gateway created and fully provisioned. Follow the instructions to [create a virtual network gateway for ExpressRoute](#). A virtual network gateway for ExpressRoute uses the GatewayType 'ExpressRoute', not VPN.
- You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit.
- A single VNet can be linked to up to four ExpressRoute circuits. Use the following process to create a new connection object for each ExpressRoute circuit you're connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.
- If you enable the ExpressRoute premium add-on, you can link virtual networks outside of the geopolitical region of the ExpressRoute circuit. The premium add-on will also allow you to connect more than 10 virtual networks to your ExpressRoute circuit depending on the bandwidth chosen. Check the [FAQ](#) for more details on the premium add-on.
- You can [view a video](#) before beginning to better understand the steps.

## Connect a VNet to a circuit - same subscription

### NOTE

BGP configuration information will not appear if the layer 3 provider configured your peerings. If your circuit is in a provisioned state, you should be able to create connections.

## To create a connection

1. Ensure that your ExpressRoute circuit and Azure private peering have been configured successfully.

Follow the instructions in [Create an ExpressRoute circuit](#) and [Create and modify peering for an ExpressRoute circuit](#). Your ExpressRoute circuit should look like the following image:

The screenshot shows the 'Essentials' section of the 'TestERCircuit' ExpressRoute circuit configuration page. It displays the following details:

Setting	Value
Resource group (change)	ExpressRouteResourceGroup
Circuit status	Enabled
Location	West US 2
Subscription (change)	Azure Subscription
Subscription ID	01234567-abcd-ef01-2345-6789abcdef01
Tags (change)	Click here to add tags
Type	Azure public
Status	Not provisioned
Primary subnet	-
Secondary subnet	-
Last modified by	-

Below this, the 'Peerings' section shows a table with one row:

Type	Status	Primary subnet	Secondary subnet	Last modified by
Azure private	Provisioned	> One subnet configured	One subnet configured	Customer

2. You can now start provisioning a connection to link your virtual network gateway to your ExpressRoute circuit. Select **Connection > Add** to open the **Add connection** page.

The screenshot shows the 'Connections' configuration page for the 'TestERCircuit' ExpressRoute circuit. The 'Connections' tab is selected. A red box highlights the '+ Add' button. The page displays the following interface:

- Search bar: Search (Ctrl+ /)
- Action buttons: + Add, Delete, Refresh
- Table headers: Name ↑↓, Status ↑↓, Connection type ↑↓, Peer ↑↓
- Message: No results.
- Left sidebar: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, **Connections**, Authorizations.

3. Enter a name for the connection and then select **Next: Settings >**.

## Create connection

Basics Settings Tags Review + create

Create a secure connection to your virtual network by using VPN Gateway or ExpressRoute. [Learn more about VPN Gateway](#)  [Learn more about ExpressRoute](#) 

### Project details

Subscription

Azure Subscription

Resource group

ExpressRouteResourceGroup

[Create new](#)

### Instance details

Connection type 

ExpressRoute

Name \*

ER-VNet-Connection

Region

(US) West US 2

[Review + create](#)

< Previous

Next : [Settings >](#)

[Download a template for automation](#)

4. Select the gateway that belongs to the virtual network that you want to link to the circuit and select **Review + create**. Then select **Create** after validation completes.

## Create connection

Basics **Settings** Tags Review + create

### Virtual network gateway

To use a virtual network with a connection, it must be associated to a virtual network gateway.

Virtual network gateway \* 

ERGW

ExpressRoute circuit 

TestERCircuit

Redeem authorization 



Routing weight \*

0

[Review + create](#)

< Previous

Next : [Tags >](#)

[Download a template for automation](#)

5. After your connection has been successfully configured, your connection object will show the information for the connection.

**TestERCircuit | Connections**

ExpressRoute circuit

Search (Ctrl+ /) Add Delete Refresh

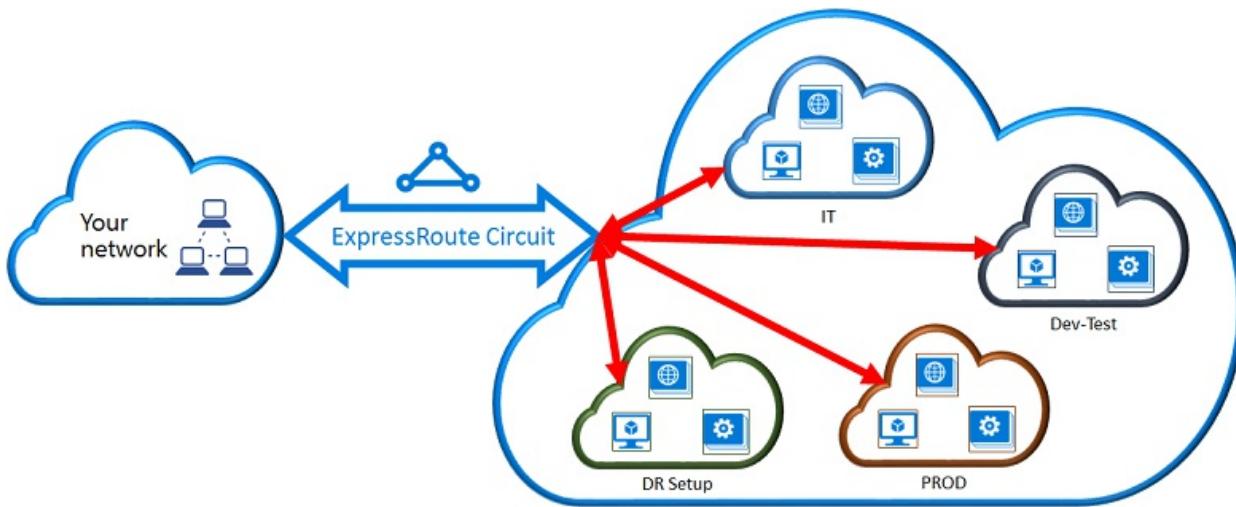
Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Configuration Connections Authorizations

Name ↑↓	Status ↑↓	Connection type ↑↓	Peer ↑↓
ER-VNet-Connection	Succeeded	ExpressRoute	ERGW

## Connect a VNet to a circuit - different subscription

You can share an ExpressRoute circuit across multiple subscriptions. The following figure shows a simple schematic of how sharing works for ExpressRoute circuits across multiple subscriptions.



Each of the smaller clouds within the large cloud is used to represent subscriptions that belong to different departments within an organization. Each of the departments within the organization uses their own subscription for deploying their services--but they can share a single ExpressRoute circuit to connect back to your on-premises network. A single department (in this example: IT) can own the ExpressRoute circuit. Other subscriptions within the organization may use the ExpressRoute circuit.

### NOTE

Connectivity and bandwidth charges for the dedicated circuit will be applied to the ExpressRoute circuit owner. All virtual networks share the same bandwidth.

## Administration - About circuit owners and circuit users

The 'circuit owner' is an authorized Power User of the ExpressRoute circuit resource. The circuit owner can create authorizations that can be redeemed by 'circuit users'. Circuit users are owners of virtual network gateways that aren't within the same subscription as the ExpressRoute circuit. Circuit users can redeem authorizations (one authorization per virtual network).

The circuit owner has the power to modify and revoke authorizations at any time. Revoking an authorization results in all link connections being deleted from the subscription whose access was revoked.

## Circuit owner operations

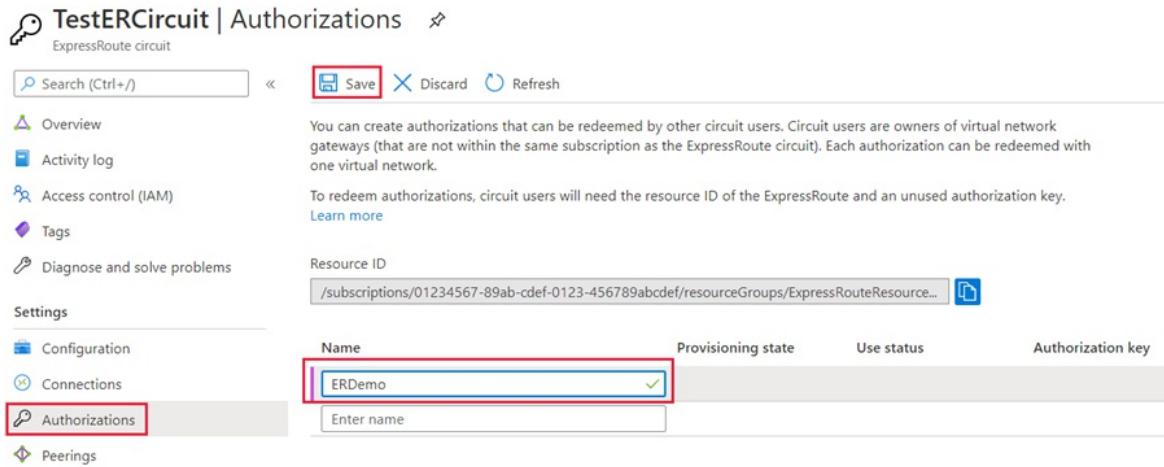
## To create a connection authorization

The circuit owner creates an authorization, which creates an authorization key to be used by a circuit user to connect their virtual network gateways to the ExpressRoute circuit. An authorization is valid for only one connection.

### NOTE

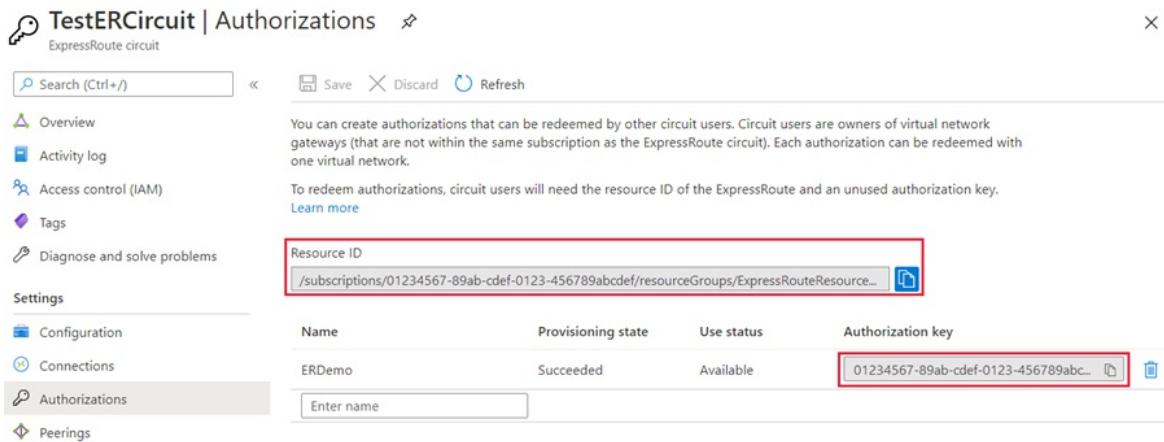
Each connection requires a separate authorization.

1. In the ExpressRoute page, select **Authorizations** and then type a **name** for the authorization and select **Save**.



The screenshot shows the 'Authorizations' blade for a 'TestERCircuit' ExpressRoute circuit. The 'Authorizations' tab is selected. A new authorization named 'ERDemo' is being created, with its name highlighted in a red box. The 'Save' button is visible at the top.

2. Once the configuration is saved, copy the **Resource ID** and the **Authorization Key**.



The screenshot shows the 'Authorizations' blade for a 'TestERCircuit' ExpressRoute circuit. The 'Authorizations' tab is selected. The newly created authorization 'ERDemo' is listed with its details. The 'Resource ID' and 'Authorization key' are highlighted in red boxes.

## To delete a connection authorization

You can delete a connection by selecting the **Delete** icon for the authorization key for your connection.

**TestERCircuit | Authorizations**

ExpressRoute circuit

Search (Ctrl+ /) Save Discard Refresh

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Configuration Connections Authorizations Peerings

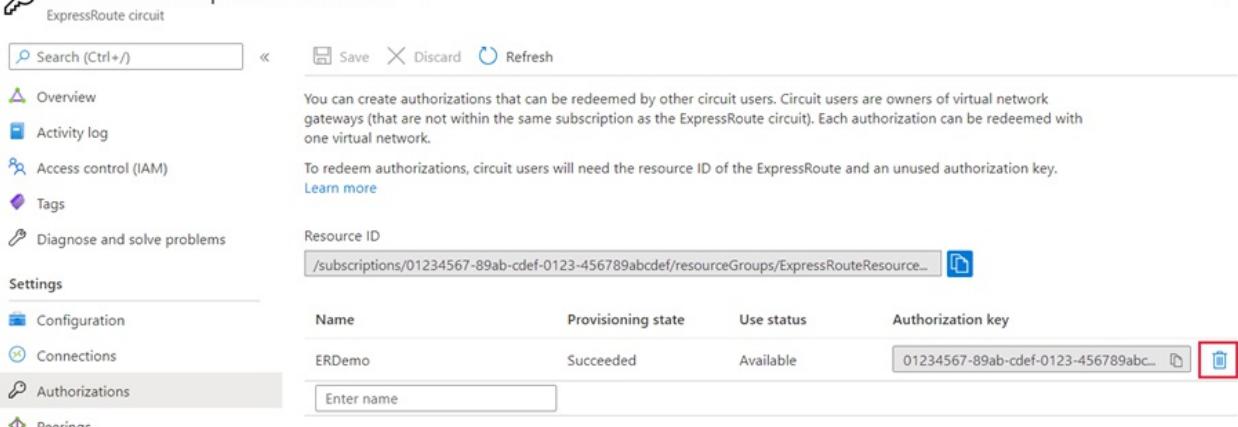
You can create authorizations that can be redeemed by other circuit users. Circuit users are owners of virtual network gateways (that are not within the same subscription as the ExpressRoute circuit). Each authorization can be redeemed with one virtual network.

To redeem authorizations, circuit users will need the resource ID of the ExpressRoute and an unused authorization key. [Learn more](#)

Resource ID: /subscriptions/01234567-89ab-cdef-0123-456789abcdef/resourceGroups/ExpressRouteResource...

Name	Provisioning state	Use status	Authorization key
ERDemo	Succeeded	Available	01234567-89ab-cdef-0123-456789abc... <a href="#">Copy</a> <a href="#">Delete</a>

Enter name:



If you want to delete the connection but retain the authorization key, you can delete the connection from the connection page of the circuit.

**TestERCircuit | Connections**

ExpressRoute circuit

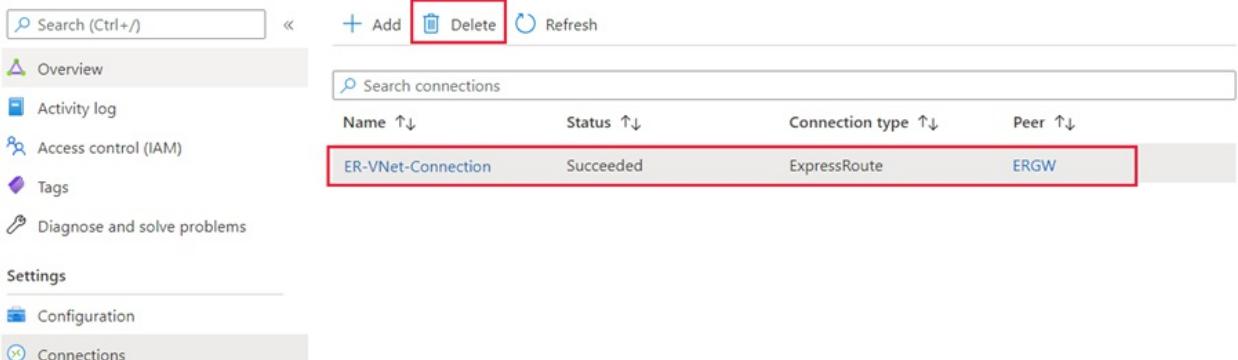
Search (Ctrl+ /) Add Delete Refresh

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Configuration Connections Authorizations

Search connections:

Name ↑↓	Status ↑↓	Connection type ↑↓	Peer ↑↓
ER-VNet-Connection	Succeeded	ExpressRoute	ERGW



## Circuit user operations

The circuit user needs the resource ID and an authorization key from the circuit owner.

### To redeem a connection authorization

1. Select the + Create a resource button. Search for Connection and select Create.

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/#create/Microsoft.Network/vpnConnection](#). The left sidebar is visible with various service icons. The main content area is titled 'Connection' and shows a summary of the connection type as 'ExpressRoute'. It includes sections for 'Overview', 'Plans', and 'Usage Information + Support'. A large blue 'Create' button is prominently displayed at the bottom of the main content area.

2. Make sure the *Connection type* is set to **ExpressRoute**. Select the *Resource group* and *Location*, then select **OK** in the Basics page.

#### NOTE

The location *must* match the virtual network gateway location you're creating the connection for.

## Basics

**Connection type \*** ⓘ

**Subscription \***

**Resource group \*** ⓘ

[Create new](#)

**Location \***

**OK**

3. In the **Settings** page, Select the *Virtual network gateway* and check the **Redeem authorization** check box. Enter the *Authorization key* and the *Peer circuit URI* and give the connection a name. Select **OK**.

**NOTE**

The *Peer Circuit URI* is the Resource ID of the ExpressRoute circuit (which you can find under the Properties Setting pane of the ExpressRoute Circuit).

## Settings

\*Virtual network gateway ⓘ >  
ERGW

Redeem authorization ⓘ

Authorization key \*

01234567-89ab-cdef-0123-456789abcdef ✓

Peer circuit URI \*

/subscriptions/01234567-89ab-cdef-012... ✓

Connection name \*

ER-VNet-Connection ✓

Routing weight \*

0

OK

4. Review the information in the **Summary** page and select **OK**.

## Summary

### Basics

Connection type	ExpressRoute
Subscription	Azure Subscription
Resource Group	ExpressRouteResourceGroup
Location	East US

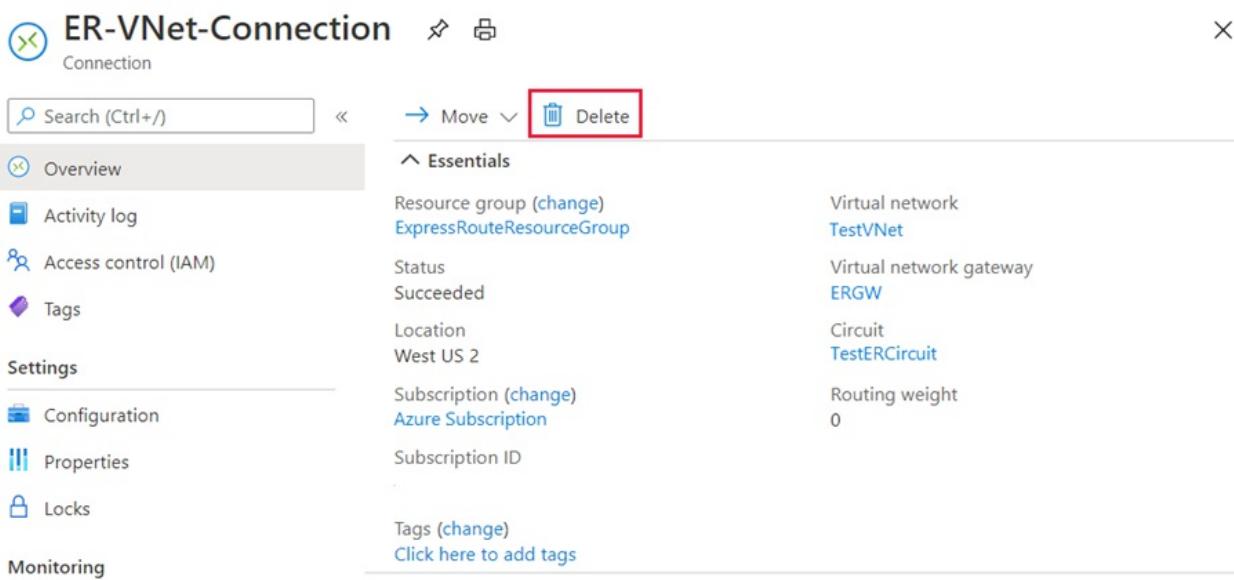
### Settings

Virtual network gateway	ERGW
Redeem ExpressRoute authorization	Yes
Authorization key	01234567-89ab-cdef-0123-456789abcdef
Peer circuit URI	/subscriptions/01234567-89ab-cdef-0123-45678...
Connection name	ER-VNet-Connection

OK

## Clean up resources

You can delete a connection and unlink your VNet to an ExpressRoute circuit by selecting the **Delete** icon on the page for your connection.



The screenshot shows the 'ER-VNet-Connection' blade in the Azure portal. The top navigation bar includes a 'Search (Ctrl+ /)' field, a 'Move' dropdown, and a 'Delete' button, which is highlighted with a red box. The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, Configuration, Properties, Locks, and Monitoring. The main content area is titled 'Essentials' and displays the following information:

Setting	Value
Resource group (change)	ExpressRouteResourceGroup
Status	Succeeded
Location	West US 2
Subscription (change)	Azure Subscription
Tags (change)	Click here to add tags
Virtual network	TestVNet
Virtual network gateway	ERGW
Circuit	TestERCircuit
Routing weight	0

## Next steps

In this tutorial, you learned how to connect a virtual network to a circuit in the same subscription and a different subscription. For more information about the ExpressRoute gateway, see:

[About ExpressRoute virtual network gateways](#)

# Work with a virtual network TAP using the Azure CLI

11/2/2020 • 3 minutes to read • [Edit Online](#)

Azure virtual network TAP (Terminal Access Point) allows you to continuously stream your virtual machine network traffic to a network packet collector or analytics tool. The collector or analytics tool is provided by a [network virtual appliance](#) partner. For a list of partner solutions that are validated to work with virtual network TAP, see [partner solutions](#).

## Create a virtual network TAP resource

Read [prerequisites](#) before you create a virtual network TAP resource. You can run the commands that follow in the [Azure Cloud Shell](#), or by running the Azure command-line interface (CLI) from your computer. The Azure Cloud Shell is a free interactive shell, that doesn't require installing the Azure CLI on your computer. You must sign in to Azure with an account that has the appropriate [permissions](#). This article requires the Azure CLI version 2.0.46 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI 2.0](#). Virtual network TAP is currently available as an extension. To install the extension you need to run `az extension add -n virtual-network-tap`. If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.

1. Retrieve the ID of your subscription into a variable that is used in a later step:

```
subscriptionId=$(az account show \
--query id \
--out tsv)
```

2. Set the subscription id that you will use to create a virtual network TAP resource.

```
az account set --subscription $subscriptionId
```

3. Re-register the subscription ID that you'll use to create a virtual network TAP resource. If you get a registration error when you create a TAP resource, run the following command:

```
az provider register --namespace Microsoft.Network --subscription $subscriptionId
```

4. If the destination for the virtual network TAP is the network interface on the network virtual appliance for collector or analytics tool -

- Retrieve the IP configuration of the network virtual appliance's network interface into a variable that is used in a later step. The ID is the end point that will aggregate the TAP traffic. The following example retrieves the ID of the `ipconfig1` IP configuration for a network interface named `myNetworkInterface`, in a resource group named `myResourceGroup`.

```
IpConfigId=$(az network nic ip-config show \
--name ipconfig1 \
--nic-name myNetworkInterface \
--resource-group myResourceGroup \
--query id \
--out tsv)
```

- Create the virtual network TAP in westcentralus azure region using the ID of the IP configuration as the destination and an optional port property. The port specifies the destination port on network interface IP configuration where the TAP traffic will be received :

```
az network vnet tap create \
--resource-group myResourceGroup \
--name myTap \
--destination $IpConfigId \
--port 4789 \
--location westcentralus
```

## 5. If the destination for the virtual network TAP is an azure internal load balancer:

- Retrieve the front end IP configuration of the Azure internal load balancer into a variable that is used in a later step. The ID is the end point that will aggregate the TAP traffic. The following example retrieves the ID of the *frontendipconfig1* front end IP configuration for a load balancer named *myInternalLoadBalancer*, in a resource group named *myResourceGroup*.

```
FrontendIpConfigId=$(az network lb frontend-ip show \
--name frontendipconfig1 \
--lb-name myInternalLoadBalancer \
--resource-group myResourceGroup \
--query id \
--out tsv)
```

- Create the virtual network TAP using the ID of the frontend IP configuration as the destination and an optional port property. The port specifies the destination port on front end IP configuration where the TAP traffic will be received :

```
az network vnet tap create \
--resource-group myResourceGroup \
--name myTap \
--destination $FrontendIpConfigId \
--port 4789 \
--location westcentralus
```

## 6. Confirm creation of the virtual network TAP:

```
az network vnet tap show \
--resource-group myResourceGroup \
--name myTap
```

## Add a TAP configuration to a network interface

1. Retrieve the ID of an existing virtual network TAP resource. The following example retrieves a virtual network TAP named *myTap* in a resource group named *myResourceGroup*.

```
tapId=$(az network vnet tap show \
--name myTap \
--resource-group myResourceGroup \
--query id \
--out tsv)
```

2. Create a TAP configuration on the network interface of the monitored virtual machine. The following example creates a TAP configuration for a network interface named *myNetworkInterface*.

```
az network nic vtap-config create \
--resource-group myResourceGroup \
--nic myNetworkInterface \
--vnet-tap $tapId \
--name mytapconfig \
--subscription subscriptionId
```

3. Confirm creation of the TAP configuration:

```
az network nic vtap-config show \
--resource-group myResourceGroup \
--nic-name myNetworkInterface \
--name mytapconfig \
--subscription subscriptionId
```

## Delete the TAP configuration on a network interface

```
az network nic vtap-config delete \
--resource-group myResourceGroup \
--nic myNetworkInterface \
--name myTapConfig \
--subscription subscriptionId
```

## List virtual network TAPs in a subscription

```
az network vnet tap list
```

## Delete a virtual network TAP in a resource group

```
az network vnet tap delete \
--resource-group myResourceGroup \
--name myTap
```

# Deploy the Azure Virtual Network container network interface plug-in

2/18/2021 • 5 minutes to read • [Edit Online](#)

The Azure Virtual Network container network interface (CNI) plug-in installs in an Azure virtual machine and brings virtual network capabilities to Kubernetes Pods and Docker containers. To learn more about the plug-in, see [Enable containers to use Azure Virtual Network capabilities](#). Additionally, the plug-in can be used with the Azure Kubernetes Service (AKS) by choosing the [Advanced Networking](#) option, which automatically places AKS containers in a virtual network.

## Deploy plug-in for ACS-Engine Kubernetes cluster

The ACS-Engine deploys a Kubernetes cluster with an Azure Resource Manager template. The cluster configuration is specified in a JSON file that is passed to the tool when generating the template. To learn more about the entire list of supported cluster settings and their descriptions, see [Microsoft Azure Container Service Engine - Cluster Definition](#). The plug-in is the default networking plug-in for clusters created using the ACS-Engine. The following network configuration settings are important when configuring the plug-in:

SETTING	DESCRIPTION
firstConsecutiveStaticIP	The IP address that is allocated to the Master node. This is a mandatory setting.
clusterSubnet under kubernetesConfig	CIDR of the virtual network subnet where the cluster is deployed, and from which IP addresses are allocated to Pods
vnetSubnetId under masterProfile	Specifies the Azure Resource Manager resource ID of the subnet where the cluster is to be deployed
vnetCidr	CIDR of the virtual network where the cluster is deployed
max_Pods under kubeletConfig	Maximum number of Pods on every agent virtual machine. For the plug-in, the default is 30. You can specify up to 250

### Example configuration

The json example that follows is for a cluster with the following properties:

- 1 Master node and 2 Agent nodes
- Is deployed in a subnet named *KubeClusterSubnet* (10.0.0.0/20), with both Master and Agent nodes residing in it.

```
{
 "apiVersion": "vlabs",
 "properties": {
 "orchestratorProfile": {
 "orchestratorType": "Kubernetes",
 "kubernetesConfig": {
 "clusterSubnet": "10.0.0.0/20" --> Subnet allocated for the cluster
 }
 },
 "masterProfile": {
 "count": 1,
 "dnsPrefix": "ACSKubeMaster",
 "vmSize": "Standard_A2",
 "vnetSubnetId": "/subscriptions/<subscription ID>/resourceGroups/<Resource Group Name>/providers/Microsoft.Network/virtualNetworks/<Vnet Name>/subnets/KubeClusterSubnet",
 "firstConsecutiveStaticIP": "10.0.1.50", --> IP address allocated to the Master node
 "vnetCidr": "10.0.0.0/16" --> Virtual network address space
 },
 "agentPoolProfiles": [
 {
 "name": "k8sagentpool1",
 "count": 2,
 "vmSize": "Standard_A2_v2",
 "vnetSubnetId": "/subscriptions/<subscription ID>/resourceGroups/<Resource Group Name>/providers/Microsoft.Network/virtualNetworks/<VNet Name>/subnets/KubeClusterSubnet",
 "availabilityProfile": "AvailabilitySet"
 }
],
 "linuxProfile": {
 "adminUsername": "KubeServerAdmin",
 "ssh": {
 "publicKeys": [
 {...}
]
 }
 },
 "servicePrincipalProfile": {
 "clientId": "dd438987-aa12-4754-b47d-375811889714",
 "secret": "azure123"
 }
 }
}
```

## Deploy plug-in for a Kubernetes cluster

Complete the following steps to install the plug-in on every Azure virtual machine in a Kubernetes cluster:

1. [Download and install the plug-in.](#)
2. Pre-allocate a virtual network IP address pool on every virtual machine from which IP addresses will be assigned to Pods. Every Azure virtual machine comes with a primary virtual network private IP address on each network interface. The pool of IP addresses for Pods is added as secondary addresses (*ipconfigs*) on the virtual machine network interface, using one of the following options:
  - **CLI:** [Assign multiple IP addresses using the Azure CLI](#)
  - **PowerShell:** [Assign multiple IP addresses using PowerShell](#)
  - **Portal:** [Assign multiple IP addresses using the Azure portal](#)
  - **Azure Resource Manager template:** [Assign multiple IP addresses using templates](#)

Ensure that you add enough IP addresses for all of the Pods that you expect to bring up on the virtual machine.

3. Select the plug-in for providing networking for your cluster by passing Kubelet the `-network-plugin=cni`

command-line option during cluster creation. Kubernetes, by default, looks for the plug-in and the configuration file in the directories where they are already installed.

4. If you want your Pods to access the internet, add the following *iptables* rule on your Linux virtual machines to source-NAT internet traffic. In the following example, the specified IP range is 10.0.0.0/8.

```
iptables -t nat -A POSTROUTING -m iprange ! --dst-range 168.63.129.16 -m
addrtype ! --dst-type local ! -d 10.0.0.0/8 -j MASQUERADE
```

The rules NAT traffic that is not destined to the specified IP ranges. The assumption is that all traffic outside the previous ranges is internet traffic. You can choose to specify the IP ranges of the virtual machine's virtual network, that of peered virtual networks, and on-premises networks.

Windows virtual machines automatically source NAT traffic that has a destination outside the subnet to which the virtual machine belongs. It is not possible to specify custom IP ranges.

After completing the previous steps, Pods brought up on the Kubernetes Agent virtual machines are automatically assigned private IP addresses from the virtual network.

## Deploy plug-in for Docker containers

1. [Download and install the plug-in](#).
2. Create Docker containers with the following command:

```
./docker-run.sh <container-name> <container-namespace> <image>
```

The containers automatically start receiving IP addresses from the allocated pool. If you want to load balance traffic to the Docker containers, they must be placed behind a software load balancer, and you must configure a load balancer probe, the same way you create a policy and probes for a virtual machine.

### CNI network configuration file

The CNI network configuration file is described in JSON format. It is, by default, present in `/etc/cni/net.d` for Linux and `c:\cni\netconf` for Windows. The file specifies the configuration of the plug-in and is different for Windows and Linux. The json that follows is a sample Linux configuration file, followed by an explanation for some of the key settings. You don't need to make any changes to the file:

```
{
 "cniVersion": "0.3.0",
 "name": "azure",
 "plugins": [
 {
 "type": "azure-vnet",
 "mode": "bridge",
 "bridge": "azure0",
 "ipam": {
 "type": "azure-vnet-ipam"
 }
 },
 {
 "type": "portmap",
 "capabilities": {
 "portMappings": true
 },
 "snat": true
 }
]
}
```

#### Settings explanation

- **cniVersion:** The Azure Virtual Network CNI plug-ins support versions 0.3.0 and 0.3.1 of the [CNI spec](#).
- **name:** Name of the network. This property can be set to any unique value.
- **type:** Name of the network plug-in. Set to *azure-vnet*.
- **mode:** Operational mode. This field is optional. The only mode supported is "bridge". For more information, see [operational modes](#).
- **bridge:** Name of the bridge that will be used to connect containers to a virtual network. This field is optional. If omitted, the plugin automatically picks a unique name, based on the master interface index.
- **ipam type:** Name of the IPAM plug-in. Always set to *azure-vnet-ipam*.

## Download and install the plug-in

Download the plug-in from [GitHub](#). Download the latest version for the platform that you're using:

- **Linux:** [azure-vnet-cni-linux-amd64-<version no.>.tgz](#)
- **Windows:** [azure-vnet-cni-windows-amd64-<version no.>.zip](#)

Copy the install script for [Linux](#) or [Windows](#) to your computer. Save the script to a `scripts` directory on your computer and name the file `install-cni-plugin.sh` for Linux, or `install-cni-plugin.ps1` for Windows. To install the plug-in, run the appropriate script for your platform, specifying the version of the plug-in you are using. For example, you might specify *v1.0.12-rc3*:

```
\$scripts/install-cni-plugin.sh [version]
```

```
scripts\install-cni-plugin.ps1 [version]
```

The script installs the plug-in under `/opt/cni/bin` for Linux and `c:\cni\bin` for Windows. The installed plug-in comes with a simple network configuration file that works after installation. It doesn't need to be updated. To learn more about the settings in the file, see [CNI network configuration file](#).

# Upgrade an IPv4 application to IPv6 in Azure virtual network - PowerShell

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article shows you how to add IPv6 connectivity to an existing IPv4 application in an Azure virtual network with a Standard Load Balancer and Public IP. The in-place upgrade includes:

- IPv6 address space for the virtual network and subnet
- a Standard Load Balancer with both IPv4 and IPv6 frontend configurations
- VMs with NICs that have both an IPv4 + IPv6 configuration
- IPv6 Public IP so the load balancer has Internet-facing IPv6 connectivity

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the <a href="#">Azure portal</a> .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the Copy button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 6.9.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Prerequisites

This article assumes that you deployed a Standard Load Balancer as described in [Quickstart: Create a Standard Load Balancer - Azure PowerShell](#).

## Retrieve the resource group

Before you can create your dual-stack virtual network, you must retrieve the resource group with [Get-AzResourceGroup](#).

```
$rg = Get-AzResourceGroup -ResourceGroupName "myResourceGroupSLB"
```

## Create an IPv6 IP addresses

Create a public IPv6 address with [New-AzPublicIpAddress](#) for your Standard Load Balancer. The following example creates an IPv6 public IP address named *PublicIP\_v6* in the *myResourceGroupSLB* resource group:

```
$PublicIP_v6 = New-AzPublicIpAddress `
 -Name "PublicIP_v6" `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -Sku Standard `
 -AllocationMethod Static `
 -IpAddressVersion IPv6
```

## Configure load balancer frontend

Retrieve the existing load balancer configuration and then add the new IPv6 IP address using [Add-AzLoadBalancerFrontendIpConfig](#) as follows:

```
Retrieve the load balancer configuration
$lb = Get-AzLoadBalancer -ResourceGroupName $rg.ResourceGroupName -Name "MyLoadBalancer"

Add IPv6 components to the local copy of the load balancer configuration
$lb | Add-AzLoadBalancerFrontendIpConfig `
 -Name "dsLbFrontEnd_v6" `
 -PublicIpAddress $PublicIP_v6

#Update the running load balancer with the new frontend
$lb | Set-AzLoadBalancer
```

## Configure load balancer backend pool

Create the backend pool on the local copy of the load balancer configuration and update the running load balancer with the new backend pool configuration as follows:

```
$lb | Add-AzLoadBalancerBackendAddressPoolConfig -Name "LbBackEndPool_v6"

Update the running load balancer with the new backend pool
$lb | Set-AzLoadBalancer
```

## Configure load balancer rules

Retrieve the existing load balancer frontend and backend pool configuration and then add new load-balancing rules using [Add-AzLoadBalancerRuleConfig](#).

```

Retrieve the updated (live) versions of the frontend and backend pool
$frontendIPv6 = Get-AzLoadBalancerFrontendIpConfig -Name "dsLbFrontEnd_v6" -LoadBalancer $lb
$backendPoolv6 = Get-AzLoadBalancerBackendAddressPoolConfig -Name "LbBackEndPool_v6" -LoadBalancer $lb

Create new LB rule with the frontend and backend
$lb | Add-AzLoadBalancerRuleConfig `
 -Name "dsLBrule_v6" `
 -FrontendIpConfiguration $frontendIPv6 `
 -BackendAddressPool $backendPoolv6 `
 -Protocol Tcp `
 -FrontendPort 80 `
 -BackendPort 80

#Finalize all the load balancer updates on the running load balancer
$lb | Set-AzLoadBalancer

```

## Add IPv6 address ranges

Add IPv6 address ranges to the virtual network and subnet hosting the VMs as follows:

```

#Add IPv6 ranges to the VNET and subnet
#Retreive the VNET object
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rg.ResourceGroupName -Name "myVnet"

#Add IPv6 prefix to the VNET
$vnet.addressspace.addressprefixes.add("fd00:db8:deca::/48")

#Update the running VNET
$vnet | Set-AzVirtualNetwork

#Retrieve the subnet object from the local copy of the VNET
$subnet= $vnet.subnets[0]

#Add IPv6 prefix to the Subnet (subnet of the VNET prefix, of course)
$subnet.addressprefix.add("fd00:db8:deca::/64")

#Update the running VNET with the new subnet configuration
$vnet | Set-AzVirtualNetwork

```

## Add IPv6 configuration to NIC

Configure all of the VM NICs with an IPv6 address using [Add-AzNetworkInterfaceIpConfig](#) as follows:

```

#Retrieve the NIC objects
$NIC_1 = Get-AzNetworkInterface -Name "myNic1" -ResourceGroupName $rg.ResourceGroupName
$NIC_2 = Get-AzNetworkInterface -Name "myNic2" -ResourceGroupName $rg.ResourceGroupName
$NIC_3 = Get-AzNetworkInterface -Name "myNic3" -ResourceGroupName $rg.ResourceGroupName

#Add an IPv6 IPconfig to NIC_1 and update the NIC on the running VM
$NIC_1 | Add-AzNetworkInterfaceIpConfig -Name MyIPv6Config -Subnet $vnet.Subnets[0] -
PrivateIpAddressVersion IPv6 -LoadBalancerBackendAddressPool $backendPoolv6
$NIC_1 | Set-AzNetworkInterface

#Add an IPv6 IPconfig to NIC_2 and update the NIC on the running VM
$NIC_2 | Add-AzNetworkInterfaceIpConfig -Name MyIPv6Config -Subnet $vnet.Subnets[0] -
PrivateIpAddressVersion IPv6 -LoadBalancerBackendAddressPool $backendPoolv6
$NIC_2 | Set-AzNetworkInterface

#Add an IPv6 IPconfig to NIC_3 and update the NIC on the running VM
$NIC_3 | Add-AzNetworkInterfaceIpConfig -Name MyIPv6Config -Subnet $vnet.Subnets[0] -
PrivateIpAddressVersion IPv6 -LoadBalancerBackendAddressPool $backendPoolv6
$NIC_3 | Set-AzNetworkInterface

```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *myVnet*.
2. When **myVnet** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *myVNet*. The dual stack virtual network shows the three NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *mySubnet*.

Device	Type	IP Address	Subnet
MyNic1	Network interface	10.0.2.4	mySubnet
MyNic2	Network interface	10.0.2.5	mySubnet
MyNic3	Network interface	10.0.2.6	mySubnet
MyNic1	Network interface	ace:cab:deca:4	mySubnet
MyNic2	Network interface	ace:cab:deca:5	mySubnet
MyNic3	Network interface	ace:cab:deca:6	mySubnet

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name MyAzureResourceGroupSLB
```

## Next steps

In this article, you updated an existing Standard Load Balancer with a IPv4 frontend IP configuration to a dual stack (IPv4 and IPv6) configuration. You also added IPv6 configurations to the NICs of the VMs in the backend pool and to the Virtual Network that hosts them. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Add IPv6 to an IPv4 application in Azure virtual network - Azure CLI

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article shows you how to add IPv6 addresses to an application that is using IPv4 public IP address in an Azure virtual network for a Standard Load Balancer using Azure CLI. The in-place upgrade includes a virtual network and subnet, a Standard Load Balancer with IPv4 + IPV6 frontend configurations, VMs with NICs that have a IPv4 + IPv6 configurations, network security group, and public IPs.

## Prerequisites

- This article assumes that you deployed a Standard Load Balancer as described in [Quickstart: Create a Standard Load Balancer - Azure CLI](#).
- Use the Bash environment in [Azure Cloud Shell](#).  
A blue rectangular button with a white triangle icon on the left and the text "Launch Cloud Shell" in white.
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create IPv6 addresses

Create public IPv6 address with with [az network public-ip create](#) for your Standard Load Balancer. The following example creates an IPv6 public IP address named *PublicIP\_v6* in the *myResourceGroupSLB* resource group:

```
az network public-ip create \
--name PublicIP_v6 \
--resource-group MyResourceGroupSLB \
--location EastUS \
--sku Standard \
--allocation-method static \
--version IPv6
```

## Configure IPv6 load balancer frontend

Configure the load balancer with the new IPv6 IP address using [az network lb frontend-ip create](#) as follows:

```
az network lb frontend-ip create \
--lb-name myLoadBalancer \
--name dsLbFrontEnd_v6 \
--resource-group MyResourceGroupSLB \
--public-ip-address PublicIP_v6
```

## Configure IPv6 load balancer backend pool

Create the backend pool for NICs with IPv6 addresses using [az network lb address-pool create](#) as follows:

```
az network lb address-pool create \
--lb-name myLoadBalancer \
--name dsLbBackEndPool_v6 \
--resource-group MyResourceGroupSLB
```

## Configure IPv6 load balancer rules

Create IPv6 load balancer rules with [az network lb rule create](#).

```
az network lb rule create \
--lb-name myLoadBalancer \
--name dsLBrule_v6 \
--resource-group MyResourceGroupSLB \
--frontend-ip-name dsLbFrontEnd_v6 \
--protocol Tcp \
--frontend-port 80 \
--backend-port 80 \
--backend-pool-name dsLbBackEndPool_v6
```

## Add IPv6 address ranges

Add IPv6 address ranges to the virtual network and subnet hosting the load balancer as follows:

```
az network vnet update \
--name myVnet \
--resource-group MyResourceGroupSLB \
--address-prefixes "10.0.0.0/16" "fd00:db8:deca::/48"

az network vnet subnet update \
--vnet-name myVnet \
--name mySubnet \
--resource-group MyResourceGroupSLB \
--address-prefixes "10.0.0.0/24" "fd00:db8:deca:deed::/64"
```

## Add IPv6 configuration to NICs

Configure the VM NICs with an IPv6 address using [az network nic ip-config create](#) as follows:

```

az network nic ip-config create \
--name dsIp6Config_NIC1 \
--nic-name myNicVM1 \
--resource-group MyResourceGroupSLB \
--vnet-name myVnet \
--subnet mySubnet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name dsLB

az network nic ip-config create \
--name dsIp6Config_NIC2 \
--nic-name myNicVM2 \
--resource-group MyResourceGroupSLB \
--vnet-name myVnet \
--subnet mySubnet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name myLoadBalancer

az network nic ip-config create \
--name dsIp6Config_NIC3 \
--nic-name myNicVM3 \
--resource-group MyResourceGroupSLB \
--vnet-name myVnet \
--subnet mySubnet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name myLoadBalancer

```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *myVnet*.
2. When **myVnet** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *myVNet*. The dual stack virtual network shows the three NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *mySubnet*.

Device	Type	IP Address	Subnet
MyNic1	Network interface	10.0.2.4	mySubnet
MyNic2	Network interface	10.0.2.5	mySubnet
MyNic3	Network interface	10.0.2.6	mySubnet
MyNic1	Network interface	ace:cab:deca::4	mySubnet
MyNic2	Network interface	ace:cab:deca::5	mySubnet
MyNic3	Network interface	ace:cab:deca::6	mySubnet

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
az group delete --name MyAzureResourceGroupSLB
```

## Next steps

In this article, you updated an existing Standard Load Balancer with a IPv4 frontend IP configuration to a dual stack (IPv4 and IPv6) configuration. You also added IPv6 configurations to the NICs of the VMs in the backend pool. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy an IPv6 dual stack application using Basic Load Balancer - PowerShell

2/18/2021 • 9 minutes to read • [Edit Online](#)

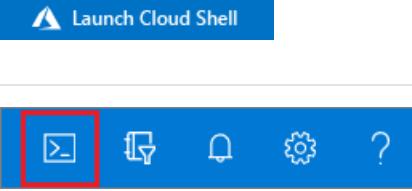
This article shows you how to deploy a dual stack (IPv4 + IPv6) application with Basic Load Balancer using Azure PowerShell that includes a dual stack virtual network and subnet, a Basic Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, network security group, and public IPs.

To deploy a dual stack (IPV4 + IPV6) application using Standard Load Balancer, see [Deploy an IPv6 dual stack application with Standard Load Balancer using Azure PowerShell](#).

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the Copy button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 6.9.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a resource group

Before you can create your dual-stack virtual network, you must create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myRGDualStack* in the *east us* location:

```
$rg = New-AzResourceGroup
-ResourceGroupName "dsRG1"
-Location "east us"
```

## Create IPv4 and IPv6 public IP addresses

To access your virtual machines from the Internet, you need IPv4 and IPv6 public IP addresses for the load balancer. Create public IP addresses with [New-AzPublicIpAddress](#). The following example creates IPv4 and IPv6 public IP address named *dsPublicIP\_v4* and *dsPublicIP\_v6* in the *dsRG1* resource group:

```
$PublicIP_v4 = New-AzPublicIpAddress
-Name "dsPublicIP_v4"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Dynamic
-IpAddressVersion IPv4

$PublicIP_v6 = New-AzPublicIpAddress
-Name "dsPublicIP_v6"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Dynamic
-IpAddressVersion IPv6
```

To access your virtual machines using a RDP connection, create a IPV4 public IP addresses for the virtual machines with [New-AzPublicIpAddress](#).

```
$RdpPublicIP_1 = New-AzPublicIpAddress
-Name "RdpPublicIP_1"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Dynamic
-IpAddressVersion IPv4

$RdpPublicIP_2 = New-AzPublicIpAddress
-Name "RdpPublicIP_2"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Dynamic
-IpAddressVersion IPv4
```

## Create Basic Load Balancer

In this section, you configure dual frontend IP (IPv4 and IPv6) and the back-end address pool for the load balancer and then create a Basic Load Balancer.

### Create front-end IP

Create a front-end IP with [New-AzLoadBalancerFrontendIpConfig](#). The following example creates IPv4 and IPv6 frontend IP configurations named *dsLbFrontEnd\_v4* and *dsLbFrontEnd\_v6*:

```
$frontendIPv4 = New-AzLoadBalancerFrontendIpConfig `
-Name "dsLbFrontEnd_v4" `
-PublicIpAddress $PublicIP_v4

$frontendIPv6 = New-AzLoadBalancerFrontendIpConfig `
-Name "dsLbFrontEnd_v6" `
-PublicIpAddress $PublicIP_v6
```

## Configure back-end address pool

Create a back-end address pool with [New-AzLoadBalancerBackendAddressPoolConfig](#). The VMs attach to this back-end pool in the remaining steps. The following example creates back-end address pools named *dsLbBackEndPool\_v4* and *dsLbBackEndPool\_v6* to include VMs with both IPv4 and IPv6 NIC configurations:

```
$backendPoolv4 = New-AzLoadBalancerBackendAddressPoolConfig `
-Name "dsLbBackEndPool_v4"

$backendPoolv6 = New-AzLoadBalancerBackendAddressPoolConfig `
-Name "dsLbBackEndPool_v6"
```

## Create a health probe

Use [Add-AzLoadBalancerProbeConfig](#) to create a health probe to monitor the health of the VMs.

```
$probe = New-AzLoadBalancerProbeConfig -Name MyProbe -Protocol tcp -Port 3389 -IntervalInSeconds 15 -
ProbeCount 2
```

## Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port. To make sure only healthy VMs receive traffic, you can optionally define a health probe. Basic load balancer uses an IPv4 probe to assess health for both IPv4 and IPv6 endpoints on the VMs. Standard load balancer includes support for explicitly IPv6 health probes.

Create a load balancer rule with [Add-AzLoadBalancerRuleConfig](#). The following example creates load balancer rules named *dsLBrule\_v4* and *dsLBrule\_v6* and balances traffic on *TCP* port *80* to the IPv4 and IPv6 frontend IP configurations:

```
$lbrule_v4 = New-AzLoadBalancerRuleConfig `
-Name "dsLBrule_v4" `
-FrontendIpConfiguration $frontendIPv4 `
-BackendAddressPool $backendPoolv4 `
-Protocol Tcp `
-FrontendPort 80 `
-BackendPort 80 `
-probe $probe

$lbrule_v6 = New-AzLoadBalancerRuleConfig `
-Name "dsLBrule_v6" `
-FrontendIpConfiguration $frontendIPv6 `
-BackendAddressPool $backendPoolv6 `
-Protocol Tcp `
-FrontendPort 80 `
-BackendPort 80 `
-probe $probe
```

## Create load balancer

Create the Basic Load Balancer with [New-AzLoadBalancer](#). The following example creates a public Basic Load Balancer named *myLoadBalancer* using the IPv4 and IPv6 frontend IP configurations, backend pools, and load-balancing rules that you created in the preceding steps:

```
$lb = New-AzLoadBalancer `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "MyLoadBalancer" `
-Sku "Basic" `
-FrontendIpConfiguration $frontendIPv4,$frontendIPv6 `
-BackendAddressPool $backendPoolv4,$backendPoolv6 `
-LoadBalancingRule $lbrule_v4,$lbrule_v6
```

## Create network resources

Before you deploy some VMs and can test your balancer, you must create supporting network resources - availability set, network security group, virtual network, and virtual NICs.

### Create an availability set

To improve the high availability of your app, place your VMs in an availability set.

Create an availability set with [New-AzAvailabilitySet](#). The following example creates an availability set named *myAvailabilitySet*.

```
$avset = New-AzAvailabilitySet `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsAVset" `
-PlatformFaultDomainCount 2 `
-PlatformUpdateDomainCount 2 `
-Sku aligned
```

### Create network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNET.

#### Create a network security group rule for port 3389

Create a network security group rule to allow RDP connections through port 3389 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule1 = New-AzNetworkSecurityRuleConfig `
-Name 'myNetworkSecurityGroupRuleRDP' `
-Description 'Allow RDP' `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 100 `
-SourceAddressPrefix * `
-SourcePortRange * `
-DestinationAddressPrefix * `
-DestinationPortRange 3389
```

#### Create a network security group rule for port 80

Create a network security group rule to allow internet connections through port 80 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule2 = New-AzNetworkSecurityRuleConfig `
-Name 'myNetworkSecurityGroupRuleHTTP' `
-Description 'Allow HTTP' `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 200 `
-SourceAddressPrefix * `
-SourcePortRange 80 `
-DestinationAddressPrefix * `
-DestinationPortRange 80
```

### Create a network security group

Create a network security group with [New-AzNetworkSecurityGroup](#).

```
$nsg = New-AzNetworkSecurityGroup `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsNSG1" `
-SecurityRules $rule1,$rule2
```

### Create a virtual network

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVnet* with *mySubnet*.

```
Create dual stack subnet
$subnet = New-AzVirtualNetworkSubnetConfig `
-Name "dsSubnet" `
-AddressPrefix "10.0.0.0/24","fd00:db8:deca:deed::/64"

Create the virtual network
$vnet = New-AzVirtualNetwork `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsVnet" `
-AddressPrefix "10.0.0.0/16","fd00:db8:deca::/48" `
-Subnet $subnet
```

### Create NICs

Create virtual NICs with [New-AzNetworkInterface](#). The following example creates two virtual NICs both with IPv4 and IPv6 configurations. (One virtual NIC for each VM you create for your app in the following steps).

```

$Ip4Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp4Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv4 `
-LoadBalancerBackendAddressPool $backendPoolv4 `
-PublicIpAddress $RdpPublicIP_1

$Ip6Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp6Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv6 `
-LoadBalancerBackendAddressPool $backendPoolv6

$NIC_1 = New-AzNetworkInterface `
-Name "dsNIC1" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-NetworkSecurityGroupId $nsg.Id `
-IpConfiguration $Ip4Config,$Ip6Config

$Ip4Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp4Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv4 `
-LoadBalancerBackendAddressPool $backendPoolv4 `
-PublicIpAddress $RdpPublicIP_2

$NIC_2 = New-AzNetworkInterface `
-Name "dsNIC2" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-NetworkSecurityGroupId $nsg.Id `
-IpConfiguration $Ip4Config,$Ip6Config

```

## Create virtual machines

Set an administrator username and password for the VMs with [Get-Credential](#):

```
$cred = get-credential -Message "DUAL STACK VNET SAMPLE: Please enter the Administrator credential to log into the VMs."
```

Now you can create the VMs with [New-AzVM](#). The following example creates two VMs and the required virtual network components if they do not already exist.

```

$vmsize = "Standard_A2"
$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "WindowsServer"
$imageSKU = "2019-Datacenter"

$vmName= "dsVM1"
$VMconfig1 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-
AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-
AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null
| Set-AzVMOSDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id
$NIC_1.Id 3> $null
$VM1 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig1

$vmName= "dsVM2"
$VMconfig2 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-
AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-
AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null
| Set-AzVMOSDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id
$NIC_2.Id 3> $null
$VM2 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig2

```

## Determine IP addresses of the IPv4 and IPv6 endpoints

Get all Network Interface Objects in the resource group to summarize the IP's used in this deployment with

```
get-AzNetworkInterface
```

. Also, get the Load Balancer's frontend addresses of the IPv4 and IPv6 endpoints with

```
get-AzpublicIpAddress
```

```

$rgName= "dsRG1"
$NICsInRG= get-AzNetworkInterface -resourceGroupName $rgName
write-host `nSummary of IPs in this Deployment:
write-host ****
foreach ($NIC in $NICsInRG) {

 $VMid= $NIC.virtualmachine.id
 $VMnamebits= $VMid.split("/")
 $VMname= $VMnamebits[($VMnamebits.count-1)]
 write-host `nPrivate IP addresses for $VMname
 $IPconfigsInNIC= $NIC.IPconfigurations
 foreach ($IPconfig in $IPconfigsInNIC) {

 $IPAddress= $IPconfig.privateipaddress
 write-host " "$IPAddress
 IF ($IPconfig.PublicIpAddress.ID) {

 $IDbits= ($IPconfig.PublicIpAddress.ID).split("/")
 $PipName= $IDbits[($IDbits.count-1)]
 $PipObject= get-azPublicIpAddress -name $PipName -resourceGroup $rgName
 write-host " RDP address: $PipObject.IpAddress
 }
 }
}

write-host `nPublic IP addresses on Load Balancer:
(get-AzpublicIpAddress -resourcegroupname $rgName | where { $_.name -notlike "RdpPublicIP*" }).IpAddress

```

The following figure shows a sample output that lists the private IPv4 and IPv6 addresses of the two VMs, and the frontend IPv4 and IPv6 IP addresses of the Load Balancer.

## Summary of IPs in this Deployment:

```

```

### Private IP addresses for DsVM0

```
10.0.0.4
RDP address: 40.118.190.180
ace:cab:deca:deed::4
```

### Private IP addresses for DsVM1

```
10.0.0.5
RDP address: 40.118.190.195
ace:cab:deca:deed::5
```

### Public IP addresses on Load Balancer:

```
40.118.190.251
```

```
2a01:111:f100:3000::a83e:19c3
```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *dsVnet*.
2. When **myVirtualNetwork** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *dsVnet*. The dual stack virtual network shows the two NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *dsSubnet*.

The screenshot shows the Azure portal interface for managing a virtual network. The left sidebar navigation includes Home, Resource groups, DsRG02, VNET, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Address space, Connected devices, Subnets, DDoS protection, Firewall, Security, DNS servers, Peerings, Service endpoints, and Properties. The main content area displays the 'Overview' tab for 'dsVnet'. It shows the resource group (DsRG02), location (West US), subscription (change), and subscription ID. There is a 'Tags (change)' section with a 'Click here to add tags' button. Below this, the 'Connected devices' section lists four entries: DsVM0 (Network interface, IP ADDRESS: 10.0.0.4, SUBNET: DualStackSubnet), DsVM0 (Network interface, IP ADDRESS: ace:cab:deca:deed::4, SUBNET: DualStackSubnet), DsVM1 (Network interface, IP ADDRESS: 10.0.0.5, SUBNET: DualStackSubnet), and DsVM1 (Network interface, IP ADDRESS: ace:cab:deca:deed::5, SUBNET: DualStackSubnet).

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name dsRG1
```

## Next steps

In this article, you created a Basic Load Balancer with a dual frontend IP configuration (IPv4 and IPv6). You also created a two virtual machines that included NICs with dual IP configurations (IPV4 + IPV6) that were added to the back-end pool of the load balancer. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy an IPv6 dual stack application using Basic Load Balancer - CLI

2/18/2021 • 8 minutes to read • [Edit Online](#)

This article shows you how to deploy a dual stack (IPv4 + IPv6) application with Basic Load Balancer using Azure CLI that includes a dual stack virtual network with a dual stack subnet, a Basic Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, dual network security group rules, and dual public IPs.

To deploy a dual stack (IPV4 + IPV6) application using Standard Load Balancer, see [Deploy an IPv6 dual stack application with Standard Load Balancer using Azure CLI](#).

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
A blue rectangular button with a white triangle icon on the left and the text "Launch Cloud Shell" in white on the right.
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.49 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

Before you can create your dual-stack virtual network, you must create a resource group with [az group create](#). The following example creates a resource group named *DsResourceGroup01* in the *eastus* location:

```
az group create \
--name DsResourceGroup01 \
--location eastus
```

## Create IPv4 and IPv6 public IP addresses for load balancer

To access your IPv4 and IPv6 endpoints on the Internet, you need IPv4 and IPv6 public IP addresses for the load balancer. Create a public IP address with [az network public-ip create](#). The following example creates IPv4 and IPv6 public IP address named *dsPublicIP\_v4* and *dsPublicIP\_v6* in the *DsResourceGroup01* resource group:

```
Create an IPV4 IP address
az network public-ip create \
--name dsPublicIP_v4 \
--resource-group DsResourceGroup01 \
--location eastus \
--sku BASIC \
--allocation-method dynamic \
--version IPv4

Create an IPV6 IP address
az network public-ip create \
--name dsPublicIP_v6 \
--resource-group DsResourceGroup01 \
--location eastus \
--sku BASIC \
--allocation-method dynamic \
--version IPv6
```

## Create public IP addresses for VMs

To remotely access your VMs on the internet, you need IPv4 public IP addresses for the VMs. Create a public IP address with [az network public-ip create](#).

```
az network public-ip create \
--name dsVM0_remote_access \
--resource-group DsResourceGroup01 \
--location eastus \
--sku BASIC \
--allocation-method dynamic \
--version IPv4

az network public-ip create \
--name dsVM1_remote_access \
--resource-group DsResourceGroup01 \
--location eastus \
--sku BASIC \
--allocation-method dynamic \
--version IPv4
```

## Create Basic Load Balancer

In this section, you configure dual frontend IP (IPv4 and IPv6) and the back-end address pool for the load balancer and then create a Basic Load Balancer.

### Create load balancer

Create the Basic Load Balancer with [az network lb create](#) named **dsLB** that includes a frontend pool named **dsLbFrontEnd\_v4**, a backend pool named **dsLbBackEndPool\_v4** that is associated with the IPv4 public IP address **dsPublicIP\_v4** that you created in the preceding step.

```
az network lb create \
--name dsLB \
--resource-group DsResourceGroup01 \
--sku Basic \
--location eastus \
--frontend-ip-name dsLbFrontEnd_v4 \
--public-ip-address dsPublicIP_v4 \
--backend-pool-name dsLbBackEndPool_v4
```

## Create IPv6 frontend

Create an IPv6 frontend IP with [az network lb frontend-ip create](#). The following example creates a frontend IP configuration named *dsLbFrontEnd\_v6* and attaches the *dsPublicIP\_v6* address:

```
az network lb frontend-ip create \
--lb-name dsLB \
--name dsLbFrontEnd_v6 \
--resource-group DsResourceGroup01 \
--public-ip-address dsPublicIP_v6
```

## Configure IPv6 back-end address pool

Create a IPv6 back-end address pools with [az network lb address-pool create](#). The following example creates back-end address pool named *dsLbBackEndPool\_v6* to include VMs with IPv6 NIC configurations:

```
az network lb address-pool create \
--lb-name dsLB \
--name dsLbBackEndPool_v6 \
--resource-group DsResourceGroup01
```

## Create a health probe

Create a health probe with [az network lb probe create](#) to monitor the health of the virtual machines.

```
az network lb probe create -g DsResourceGroup01 --lb-name dsLB -n dsProbe --protocol tcp --port 3389
```

## Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port.

Create a load balancer rule with [az network lb rule create](#). The following example creates load balancer rules named *dsLBrule\_v4* and *dsLBrule\_v6* and balances traffic on *TCP* port *80* to the IPv4 and IPv6 frontend IP configurations:

```
az network lb rule create \
--lb-name dsLB \
--name dsLBrule_v4 \
--resource-group DsResourceGroup01 \
--frontend-ip-name dsLbFrontEnd_v4 \
--protocol Tcp \
--frontend-port 80 \
--backend-port 80 \
--probe-name dsProbe \
--backend-pool-name dsLbBackEndPool_v4

az network lb rule create \
--lb-name dsLB \
--name dsLBrule_v6 \
--resource-group DsResourceGroup01 \
--frontend-ip-name dsLbFrontEnd_v6 \
--protocol Tcp \
--frontend-port 80 \
--backend-port 80 \
--probe-name dsProbe \
--backend-pool-name dsLbBackEndPool_v6
```

## Create network resources

Before you deploy some VMs, you must create supporting network resources - availability set, network security group, virtual network, and virtual NICs.

### Create an availability set

To improve the availability of your app, place your VMs in an availability set.

Create an availability set with [az vm availability-set create](#). The following example creates an availability set named *dsAVset*.

```
az vm availability-set create \
--name dsAVset \
--resource-group DsResourceGroup01 \
--location eastus \
--platform-fault-domain-count 2 \
--platform-update-domain-count 2
```

### Create network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNET.

#### Create a network security group

Create a network security group with [az network nsg create](#)

```
az network nsg create \
--name dsNSG1 \
--resource-group DsResourceGroup01 \
--location eastus
```

#### Create a network security group rule for inbound and outbound connections

Create a network security group rule to allow RDP connections through port 3389, internet connection through port 80, and for outbound connections with [az network nsg rule create](#).

```
Create inbound rule for port 3389
az network nsg rule create \
--name allowRdpIn \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 100 \
--description "Allow Remote Desktop In" \
--access Allow \
--protocol "*" \
--direction Inbound \
--source-address-prefixes "*" \
--source-port-ranges "*" \
--destination-address-prefixes "*" \
--destination-port-ranges 3389

Create inbound rule for port 80
az network nsg rule create \
--name allowHTTPIn \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 200 \
--description "Allow HTTP In" \
--access Allow \
--protocol "*" \
--direction Inbound \
--source-address-prefixes "*" \
--source-port-ranges 80 \
--destination-address-prefixes "*" \
--destination-port-ranges 80

Create outbound rule

az network nsg rule create \
--name allowAllOut \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 300 \
--description "Allow All Out" \
--access Allow \
--protocol "*" \
--direction Outbound \
--source-address-prefixes "*" \
--source-port-ranges "*" \
--destination-address-prefixes "*" \
--destination-port-ranges "*"
```

## Create a virtual network

Create a virtual network with [az network vnet create](#). The following example creates a virtual network named *dsVNET* with subnets *dsSubNET\_v4* and *dsSubNET\_v6*:

```
Create the virtual network
az network vnet create \
--name dsVNET \
--resource-group DsResourceGroup01 \
--location eastus \
--address-prefixes "10.0.0.0/16" "fd00:db8:deca::/48"

Create a single dual stack subnet

az network vnet subnet create \
--name dsSubNET \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--address-prefixes "10.0.0.0/24" "fd00:db8:deca:deed::/64" \
--network-security-group dsNSG1
```

## Create NICs

Create virtual NICs for each VM with [az network nic create](#). The following example creates a virtual NIC for each VM. Each NIC has two IP configurations (1 IPv4 config, 1 IPv6 config). You create the IPV6 configuration with [az network nic ip-config create](#).

```

Create NICs
az network nic create \
--name dsNIC0 \
--resource-group DsResourceGroup01 \
--network-security-group dsNSG1 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv4 \
--lb-address-pools dsLbBackEndPool_v4 \
--lb-name dsLB \
--public-ip-address dsVM0_remote_access

az network nic create \
--name dsNIC1 \
--resource-group DsResourceGroup01 \
--network-security-group dsNSG1 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv4 \
--lb-address-pools dsLbBackEndPool_v4 \
--lb-name dsLB \
--public-ip-address dsVM1_remote_access

Create IPV6 configurations for each NIC

az network nic ip-config create \
--name dsIp6Config_NIC0 \
--nic-name dsNIC0 \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name dsLB

az network nic ip-config create \
--name dsIp6Config_NIC1 \
--nic-name dsNIC1 \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name dsLB

```

## Create virtual machines

Create the VMs with [az vm create](#). The following example creates two VMs and the required virtual network components if they do not already exist.

Create virtual machine *dsVM0* as follows:

```

az vm create \
--name dsVM0 \
--resource-group DsResourceGroup01 \
--nics dsNIC0 \
--size Standard_A2 \
--availability-set dsAVset \
--image MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest

```

Create virtual machine *dsVM1* as follows:

```
az vm create \
--name dsVM1 \
--resource-group DsResourceGroup01 \
--nics dsNIC1 \
--size Standard_A2 \
--availability-set dsAVset \
--image MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest
```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *dsVnet*.
2. When **myVirtualNetwork** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *dsVnet*. The dual stack virtual network shows the two NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *dsSubnet*.

The screenshot shows the Azure portal's 'Overview' page for a virtual network named 'dsVnet'. The left sidebar lists various settings and logs. The main pane shows the virtual network's configuration, including its resource group ('DsRG02'), location ('West US'), and address space ('10.0.0.0/16, 1 more'). It also displays DNS servers ('Azure provided DNS service'). The 'Connected devices' section lists four entries: two Network interfaces (DsVM0 and DsVM1) and two IP addresses (10.0.0.4 and 10.0.0.5), all associated with the 'DualStackSubnet'.

## Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group, VM, and all related resources.

```
az group delete --name DsResourceGroup01
```

## Next steps

In this article, you created a Basic Load Balancer with a dual frontend IP configuration (IPv4 and IPv6). You also created a two virtual machines that included NICs with dual IP configurations (IPV4 + IPV6) that were added to the back-end pool of the load balancer. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy an IPv6 dual stack application with Basic Load Balancer in Azure - Template

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides a list of IPv6 configuration tasks with the portion of the Azure Resource Manager VM template that applies to. Use the template described in this article to deploy a dual stack (IPv4 + IPv6) application with Basic Load Balancer that includes a dual stack virtual network with IPv4 and IPv6 subnets, a Basic Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, network security group, and public IPs.

To deploy a dual stack (IPV4 + IPV6) application using Standard Load Balancer, see [Deploy an IPv6 dual stack application with Standard Load Balancer - Template](#).

## Required configurations

Search for the template sections in the template to see where they should occur.

### IPv6 addressSpace for the virtual network

Template section to add:

```
"addressSpace": {
 "addressPrefixes": [
 "[variables('vnetv4AddressRange')]",
 "[variables('vnetv6AddressRange')]"
]
}
```

### IPv6 subnet within the IPv6 virtual network addressSpace

Template section to add:

```
{
 "name": "V6Subnet",
 "properties": {
 "addressPrefix": "[variables('subnetv6AddressRange')]"
 }
}
```

### IPv6 configuration for the NIC

Template section to add:

```
{
 "name": "ipconfig-v6",
 "properties": {
 "privateIPAllocationMethod": "Dynamic",
 "privateIPAddressVersion": "IPv6",
 "subnet": {
 "id": "[variables('v6-subnet-id')]"
 },
 "loadBalancerBackendAddressPools": [
 {
 "id": "
[concat(resourceId('Microsoft.Network/loadBalancers', 'loadBalancer'), '/backendAddressPools/LBBAP-v6')]"
 }
]
 }
}
```

## IPv6 network security group (NSG) rules

```
{
 "name": "default-allow-rdp",
 "properties": {
 "description": "Allow RDP",
 "protocol": "Tcp",
 "sourcePortRange": "33819-33829",
 "destinationPortRange": "5000-6000",
 "sourceAddressPrefix": "fd00:db8:deca:deed::/64",
 "destinationAddressPrefix": "fd00:db8:deca:deed::/64",
 "access": "Allow",
 "priority": 1003,
 "direction": "Inbound"
 }
}
```

## Conditional configuration

If you're using a network virtual appliance, add IPv6 routes in the Route Table. Otherwise, this configuration is optional.

```
{
 "type": "Microsoft.Network/routeTables",
 "name": "v6route",
 "apiVersion": "[variables('ApiVersion')]",
 "location": "[resourceGroup().location]",
 "properties": {
 "routes": [
 {
 "name": "v6route",
 "properties": {
 "addressPrefix": "fd00:db8:deca:deed::/64",
 "nextHopType": "VirtualAppliance",
 "nextHopIpAddress": "fd00:db8:ace:f00d::1"
 }
 }
]
 }
}
```

## Optional configuration

### IPv6 Internet access for the virtual network

```
{
 "name": "LBFE-v6",
 "properties": {
 "publicIPAddress": {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses', 'lbpublicip-v6')]"
 }
 }
}
```

### IPv6 Public IP addresses

```
{
 "apiVersion": "[variables('ApiVersion')]",
 "type": "Microsoft.Network/publicIPAddresses",
 "name": "lbpublicip-v6",
 "location": "[resourceGroup().location]",
 "properties": {
 "publicIPAllocationMethod": "Dynamic",
 "publicIPAddressVersion": "IPv6"
 }
}
```

## IPv6 Front end for Load Balancer

```
{
 "name": "LBFE-v6",
 "properties": {
 "publicIPAddress": {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses', 'lbpublicip-v6')]"
 }
 }
}
```

## IPv6 Back-end address pool for Load Balancer

```
"backendAddressPool": {
 "id": "[concat(resourceId('Microsoft.Network/loadBalancers', 'loadBalancer'),
 '/backendAddressPools/LBBAP-v6')]"
},
 "protocol": "Tcp",
 "frontendPort": 8080,
 "backendPort": 8080
},
 "name": "lbrule-v6"
```

## IPv6 load balancer rules to associate incoming and outgoing ports

```
{
 "name": "ipconfig-v6",
 "properties": {
 "privateIPAllocationMethod": "Dynamic",
 "privateIPAddressVersion": "IPv6",
 "subnet": {
 "id": "[variables('v6-subnet-id')]"
 },
 "loadBalancerBackendAddressPools": [
 {
 "id": "
[concat(resourceId('Microsoft.Network/loadBalancers', 'loadBalancer'), '/backendAddressPools/LBBAP-v6')]"
 }
]
 }
}
```

## Sample VM template JSON

To deploy an IPv6 dual stack application with Basic Load Balancer in Azure virtual network using Azure Resource Manager template, view sample template [here](#).

## Next steps

You can find details about pricing for [public IP addresses](#), [network bandwidth](#), or [Load Balancer](#).

# Deploy an IPv6 dual stack application in Azure - PowerShell

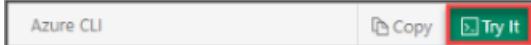
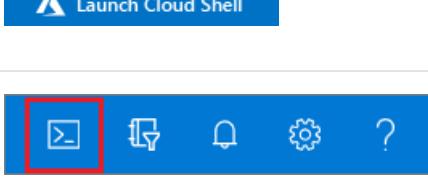
2/18/2021 • 9 minutes to read • [Edit Online](#)

This article shows you how to deploy a dual stack (IPv4 + IPv6) application using Standard Load Balancer in Azure that includes a dual stack virtual network and subnet, a Standard Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, network security group, and public IPs.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 6.9.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a resource group

Before you can create your dual-stack virtual network, you must create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named `myRGDualStack` in the `east us` location:

```
$rg = New-AzResourceGroup
-ResourceGroupName "dsRG1"
-Location "east us"
```

## Create IPv4 and IPv6 public IP addresses

To access your virtual machines from the Internet, you need IPv4 and IPv6 public IP addresses for the load balancer. Create public IP addresses with [New-AzPublicIpAddress](#). The following example creates IPv4 and IPv6 public IP address named *dsPublicIP\_v4* and *dsPublicIP\_v6* in the *dsRG1* resource group:

```
$PublicIP_v4 = New-AzPublicIpAddress
-Name "dsPublicIP_v4"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Static
-IpAddressVersion IPv4
-Sku Standard

$PublicIP_v6 = New-AzPublicIpAddress
-Name "dsPublicIP_v6"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Static
-IpAddressVersion IPv6
-Sku Standard
```

To access your virtual machines using a RDP connection, create a IPV4 public IP addresses for the virtual machines with [New-AzPublicIpAddress](#).

```
$RdpPublicIP_1 = New-AzPublicIpAddress
-Name "RdpPublicIP_1"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Static
-Sku Standard
-IpAddressVersion IPv4

$RdpPublicIP_2 = New-AzPublicIpAddress
-Name "RdpPublicIP_2"
-ResourceGroupName $rg.ResourceGroupName
-Location $rg.Location
-AllocationMethod Static
-Sku Standard
-IpAddressVersion IPv4
```

## Create Standard Load Balancer

In this section, you configure dual frontend IP (IPv4 and IPv6) and the back-end address pool for the load balancer and then create a Standard Load Balancer.

### Create front-end IP

Create a front-end IP with [New-AzLoadBalancerFrontendIpConfig](#). The following example creates IPv4 and IPv6 frontend IP configurations named *dsLbFrontEnd\_v4* and *dsLbFrontEnd\_v6*:

```
$frontendIPv4 = New-AzLoadBalancerFrontendIpConfig `
 -Name "dsLbFrontEnd_v4" `
 -PublicIpAddress $PublicIP_v4

$frontendIPv6 = New-AzLoadBalancerFrontendIpConfig `
 -Name "dsLbFrontEnd_v6" `
 -PublicIpAddress $PublicIP_v6
```

## Configure back-end address pool

Create a back-end address pool with [New-AzLoadBalancerBackendAddressPoolConfig](#). The VMs attach to this back-end pool in the remaining steps. The following example creates back-end address pools named *dsLbBackEndPool\_v4* and *dsLbBackEndPool\_v6* to include VMs with both IPv4 and IPv6 NIC configurations:

```
$backendPoolv4 = New-AzLoadBalancerBackendAddressPoolConfig `
 -Name "dsLbBackEndPool_v4"

$backendPoolv6 = New-AzLoadBalancerBackendAddressPoolConfig `
 -Name "dsLbBackEndPool_v6"
```

## Create a health probe

Use [Add-AzLoadBalancerProbeConfig](#) to create a health probe to monitor the health of the VMs.

```
$probe = New-AzLoadBalancerProbeConfig -Name MyProbe -Protocol tcp -Port 3389 -IntervalInSeconds 15 -
 ProbeCount 2
```

## Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port. To make sure only healthy VMs receive traffic, you can optionally define a health probe. Basic load balancer uses an IPv4 probe to assess health for both IPv4 and IPv6 endpoints on the VMs. Standard load balancer includes support for explicitly IPv6 health probes.

Create a load balancer rule with [Add-AzLoadBalancerRuleConfig](#). The following example creates load balancer rules named *dsLBrule\_v4* and *dsLBrule\_v6* and balances traffic on *TCP* port *80* to the IPv4 and IPv6 frontend IP configurations:

```
$lbrule_v4 = New-AzLoadBalancerRuleConfig `
 -Name "dsLBrule_v4" `
 -FrontendIpConfiguration $frontendIPv4 `
 -BackendAddressPool $backendPoolv4 `
 -Protocol Tcp `
 -FrontendPort 80 `
 -BackendPort 80 `
 -probe $probe

$lbrule_v6 = New-AzLoadBalancerRuleConfig `
 -Name "dsLBrule_v6" `
 -FrontendIpConfiguration $frontendIPv6 `
 -BackendAddressPool $backendPoolv6 `
 -Protocol Tcp `
 -FrontendPort 80 `
 -BackendPort 80 `
 -probe $probe
```

## Create load balancer

Create a Standard Load Balancer with [New-AzLoadBalancer](#). The following example creates a public Standard Load Balancer named *myLoadBalancer* using the IPv4 and IPv6 frontend IP configurations, backend pools, and load-balancing rules that you created in the preceding steps:

```
$lb = New-AzLoadBalancer `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "MyLoadBalancer" `
-Sku "Standard" `
-FrontendIpConfiguration $frontendIPv4,$frontendIPv6 `
-BackendAddressPool $backendPoolv4,$backendPoolv6 `
-LoadBalancingRule $lbrule_v4,$lbrule_v6 `
-Probe $probe
```

## Create network resources

Before you deploy some VMs and can test your balancer, you must create supporting network resources - availability set, network security group, virtual network, and virtual NICs.

### Create an availability set

To improve the high availability of your app, place your VMs in an availability set.

Create an availability set with [New-AzAvailabilitySet](#). The following example creates an availability set named *myAvailabilitySet*.

```
$avset = New-AzAvailabilitySet `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsAVset" `
-PlatformFaultDomainCount 2 `
-PlatformUpdateDomainCount 2 `
-Sku aligned
```

### Create network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNET.

#### Create a network security group rule for port 3389

Create a network security group rule to allow RDP connections through port 3389 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule1 = New-AzNetworkSecurityRuleConfig `
-Name 'myNetworkSecurityGroupRuleRDP' `
-Description 'Allow RDP' `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 100 `
-SourceAddressPrefix * `
-SourcePortRange * `
-DestinationAddressPrefix * `
-DestinationPortRange 3389
```

#### Create a network security group rule for port 80

Create a network security group rule to allow internet connections through port 80 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule2 = New-AzNetworkSecurityRuleConfig `
 -Name 'myNetworkSecurityGroupRuleHTTP' `
 -Description 'Allow HTTP' `
 -Access Allow `
 -Protocol Tcp `
 -Direction Inbound `
 -Priority 200 `
 -SourceAddressPrefix * `
 -SourcePortRange * `
 -DestinationAddressPrefix * `
 -DestinationPortRange 80
```

### Create a network security group

Create a network security group with [New-AzNetworkSecurityGroup](#).

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -Name "dsNSG1" `
 -SecurityRules $rule1,$rule2
```

### Create a virtual network

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *dsVnet* with *mySubnet*.

```
Create dual stack subnet
$subnet = New-AzVirtualNetworkSubnetConfig `
 -Name "dsSubnet" `
 -AddressPrefix "10.0.0.0/24","fd00:db8:deca:deed::/64"

Create the virtual network
$vnet = New-AzVirtualNetwork `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -Name "dsVnet" `
 -AddressPrefix "10.0.0.0/16","fd00:db8:deca::/48" `
 -Subnet $subnet
```

### Create NICs

Create virtual NICs with [New-AzNetworkInterface](#). The following example creates two virtual NICs both with IPv4 and IPv6 configurations. (One virtual NIC for each VM you create for your app in the following steps).

```

$Ip4Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp4Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv4 `
-LoadBalancerBackendAddressPool $backendPoolv4 `
-PublicIpAddress $RdpPublicIP_1

$Ip6Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp6Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv6 `
-LoadBalancerBackendAddressPool $backendPoolv6

$NIC_1 = New-AzNetworkInterface `
-Name "dsNIC1" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-NetworkSecurityGroupId $nsg.Id `
-IpConfiguration $Ip4Config,$Ip6Config

$Ip4Config=New-AzNetworkInterfaceIpConfig `
-Name dsIp4Config `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv4 `
-LoadBalancerBackendAddressPool $backendPoolv4 `
-PublicIpAddress $RdpPublicIP_2

$NIC_2 = New-AzNetworkInterface `
-Name "dsNIC2" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-NetworkSecurityGroupId $nsg.Id `
-IpConfiguration $Ip4Config,$Ip6Config

```

## Create virtual machines

Set an administrator username and password for the VMs with [Get-Credential](#):

```
$cred = get-credential -Message "DUAL STACK VNET SAMPLE: Please enter the Administrator credential to log into the VMs."
```

Now you can create the VMs with [New-AzVM](#). The following example creates two VMs and the required virtual network components if they do not already exist.

```

$vmsize = "Standard_A2"
$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "WindowsServer"
$imageSKU = "2019-Datacenter"

$vmName= "dsVM1"
$VMconfig1 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null | Set-AzVMOSDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id $NIC_1.Id 3> $null
$VM1 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig1

$vmName= "dsVM2"
$VMconfig2 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null | Set-AzVMOSDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id $NIC_2.Id 3> $null
$VM2 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig2

```

## Determine IP addresses of the IPv4 and IPv6 endpoints

Get all Network Interface Objects in the resource group to summarize the IP's used in this deployment with `get-AzNetworkInterface`. Also, get the Load Balancer's frontend addresses of the IPv4 and IPv6 endpoints with `get-AzpublicIpAddress`.

```

$rgName= "dsRG1"
$NICsInRG= get-AzNetworkInterface -resourceGroupName $rgName
write-host `nSummary of IPs in this Deployment:
write-host ****
foreach ($NIC in $NICsInRG) {

 $VMid= $NIC.virtualmachine.id
 $VMnamebits= $VMid.split("/")
 $VMname= $VMnamebits[($VMnamebits.count-1)]
 write-host `nPrivate IP addresses for $VMname
 $IPconfigsInNIC= $NIC.IPconfigurations
 foreach ($IPconfig in $IPconfigsInNIC) {

 $IPaddress= $IPconfig.privateipaddress
 write-host " "$IPaddress
 IF ($IPconfig.PublicIpAddress.ID) {

 $IDbits= ($IPconfig.PublicIpAddress.ID).split("/")
 $PipName= $IDbits[($IDbits.count-1)]
 $PipObject= get-azPublicIpAddress -name $PipName -resourceGroup $rgName
 write-host " RDP address: $PipObject.IpAddress
 }
 }
}

write-host `nPublic IP addresses on Load Balancer:
(get-AzpublicIpAddress -resourcegroupname $rgName | where { $_.name -notlike "RdpPublicIP*" }).IpAddress

```

The following figure shows a sample output that lists the private IPv4 and IPv6 addresses of the two VMs, and the frontend IPv4 and IPv6 IP addresses of the Load Balancer.

```
Summary of IPs in this Deployment:

```

#### Private IP addresses for DsVM0

```
10.0.0.4
RDP address: 40.118.190.180
ace:cab:deca:deed::4
```

#### Private IP addresses for DsVM1

```
10.0.0.5
RDP address: 40.118.190.195
ace:cab:deca:deed::5
```

#### Public IP addresses on Load Balancer:

```
40.118.190.251
```

```
2a01:111:f100:3000::a83e:19c3
```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *dsVnet*.
2. When **dsVnet** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *dsVnet*. The dual stack virtual network shows the two NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *dsSubnet*.

DEVICE	TYPE	IP ADDRESS	SUBNET
DsVM0	Network interface	10.0.0.4	DualStackSubnet
DsVM0	Network interface	ace:cab:deca:deed::4	DualStackSubnet
DsVM1	Network interface	10.0.0.5	DualStackSubnet
DsVM1	Network interface	ace:cab:deca:deed::5	DualStackSubnet

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name dsRG1
```

## Next steps

In this article, you created a Standard Load Balancer with a dual frontend IP configuration (IPv4 and IPv6). You also created a two virtual machines that included NICs with dual IP configurations (IPV4 + IPV6) that were added to the back-end pool of the load balancer. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy an IPv6 dual stack application in Azure virtual network - CLI

2/18/2021 • 8 minutes to read • [Edit Online](#)

This article shows you how to deploy a dual stack (IPv4 + IPv6) application using Standard Load Balancer in Azure that includes a dual stack virtual network with a dual stack subnet, a Standard Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, dual network security group rules, and dual public IPs.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).

 [Launch Cloud Shell](#)

- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.49 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

Before you can create your dual-stack virtual network, you must create a resource group with [az group create](#). The following example creates a resource group named *DsResourceGroup01* in the *eastus* location:

```
az group create \
--name DsResourceGroup01 \
--location eastus
```

## Create IPv4 and IPv6 public IP addresses for load balancer

To access your IPv4 and IPv6 endpoints on the Internet, you need IPv4 and IPv6 public IP addresses for the load balancer. Create a public IP address with [az network public-ip create](#). The following example creates IPv4 and IPv6 public IP address named *dsPublicIP\_v4* and *dsPublicIP\_v6* in the *DsResourceGroup01* resource group:

```
Create an IPV4 IP address
az network public-ip create \
--name dsPublicIP_v4 \
--resource-group DsResourceGroup01 \
--location eastus \
--sku STANDARD \
--allocation-method static \
--version IPv4

Create an IPV6 IP address
az network public-ip create \
--name dsPublicIP_v6 \
--resource-group DsResourceGroup01 \
--location eastus \
--sku STANDARD \
--allocation-method static \
--version IPv6
```

## Create public IP addresses for VMs

To remotely access your VMs on the internet, you need IPv4 public IP addresses for the VMs. Create a public IP address with [az network public-ip create](#).

```
az network public-ip create \
--name dsVM0_remote_access \
--resource-group DsResourceGroup01 \
--location eastus \
--sku Standard \
--allocation-method static \
--version IPv4

az network public-ip create \
--name dsVM1_remote_access \
--resource-group DsResourceGroup01 \
--location eastus \
--sku Standard \
--allocation-method static \
--version IPv4
```

## Create Standard Load Balancer

In this section, you configure dual frontend IP (IPv4 and IPv6) and the back-end address pool for the load balancer and then create a Standard Load Balancer.

### Create load balancer

Create the Standard Load Balancer with [az network lb create](#) named **dsLB** that includes a frontend pool named **dsLbFrontEnd\_v4**, a backend pool named **dsLbBackEndPool\_v4** that is associated with the IPv4 public IP address **dsPublicIP\_v4** that you created in the preceding step.

```
az network lb create \
--name dsLB \
--resource-group DsResourceGroup01 \
--sku Standard \
--location eastus \
--frontend-ip-name dsLbFrontEnd_v4 \
--public-ip-address dsPublicIP_v4 \
--backend-pool-name dsLbBackEndPool_v4
```

## Create IPv6 frontend

Create an IPv6 frontend IP with [az network lb frontend-ip create](#). The following example creates a frontend IP configuration named *dsLbFrontEnd\_v6* and attaches the *dsPublicIP\_v6* address:

```
az network lb frontend-ip create \
--lb-name dsLB \
--name dsLbFrontEnd_v6 \
--resource-group DsResourceGroup01 \
--public-ip-address dsPublicIP_v6
```

## Configure IPv6 back-end address pool

Create a IPv6 back-end address pools with [az network lb address-pool create](#). The following example creates back-end address pool named *dsLbBackEndPool\_v6* to include VMs with IPv6 NIC configurations:

```
az network lb address-pool create \
--lb-name dsLB \
--name dsLbBackEndPool_v6 \
--resource-group DsResourceGroup01
```

## Create a health probe

Create a health probe with [az network lb probe create](#) to monitor the health of the virtual machines.

```
az network lb probe create -g DsResourceGroup01 --lb-name dsLB -n dsProbe --protocol tcp --port 3389
```

## Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port.

Create a load balancer rule with [az network lb rule create](#). The following example creates load balancer rules named *dsLBRule\_v4* and *dsLBRule\_v6* and balances traffic on *TCP* port *80* to the IPv4 and IPv6 frontend IP configurations:

```
az network lb rule create \
--lb-name dsLB \
--name dsLBrule_v4 \
--resource-group DsResourceGroup01 \
--frontend-ip-name dsLbFrontEnd_v4 \
--protocol Tcp \
--frontend-port 80 \
--backend-port 80 \
--probe-name dsProbe \
--backend-pool-name dsLbBackEndPool_v4
```

```
az network lb rule create \
--lb-name dsLB \
--name dsLBrule_v6 \
--resource-group DsResourceGroup01 \
--frontend-ip-name dsLbFrontEnd_v6 \
--protocol Tcp \
--frontend-port 80 \
--backend-port 80 \
--probe-name dsProbe \
--backend-pool-name dsLbBackEndPool_v6
```

## Create network resources

Before you deploy some VMs, you must create supporting network resources - availability set, network security group, virtual network, and virtual NICs.

### Create an availability set

To improve the availability of your app, place your VMs in an availability set.

Create an availability set with [az vm availability-set create](#). The following example creates an availability set named *dsAVset*.

```
az vm availability-set create \
--name dsAVset \
--resource-group DsResourceGroup01 \
--location eastus \
--platform-fault-domain-count 2 \
--platform-update-domain-count 2
```

### Create network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNet.

#### Create a network security group

Create a network security group with [az network nsg create](#)

```
az network nsg create \
--name dsNSG1 \
--resource-group DsResourceGroup01 \
--location eastus
```

#### Create a network security group rule for inbound and outbound connections

Create a network security group rule to allow RDP connections through port 3389, internet connection through port 80, and for outbound connections with [az network nsg rule create](#).

```
Create inbound rule for port 3389
az network nsg rule create \
--name allowRdpIn \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 100 \
--description "Allow Remote Desktop In" \
--access Allow \
--protocol "*" \
--direction Inbound \
--source-address-prefixes "*" \
--source-port-ranges "*" \
--destination-address-prefixes "*" \
--destination-port-ranges 3389

Create inbound rule for port 80
az network nsg rule create \
--name allowHTTPIn \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 200 \
--description "Allow HTTP In" \
--access Allow \
--protocol "*" \
--direction Inbound \
--source-address-prefixes "*" \
--source-port-ranges 80 \
--destination-address-prefixes "*" \
--destination-port-ranges 80

Create outbound rule

az network nsg rule create \
--name allowAllOut \
--nsg-name dsNSG1 \
--resource-group DsResourceGroup01 \
--priority 300 \
--description "Allow All Out" \
--access Allow \
--protocol "*" \
--direction Outbound \
--source-address-prefixes "*" \
--source-port-ranges "*" \
--destination-address-prefixes "*" \
--destination-port-ranges "*"
```

## Create a virtual network

Create a virtual network with [az network vnet create](#). The following example creates a virtual network named *dsVNET* with subnets *dsSubNET\_v4* and *dsSubNET\_v6*:

```
Create the virtual network
az network vnet create \
--name dsVNET \
--resource-group DsResourceGroup01 \
--location eastus \
--address-prefixes "10.0.0.0/16" "fd00:db8:deca::/48"

Create a single dual stack subnet

az network vnet subnet create \
--name dsSubNET \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--address-prefixes "10.0.0.0/24" "fd00:db8:deca:deed::/64" \
--network-security-group dsNSG1
```

## Create NICs

Create virtual NICs for each VM with [az network nic create](#). The following example creates a virtual NIC for each VM. Each NIC has two IP configurations (1 IPv4 config, 1 IPv6 config). You create the IPV6 configuration with [az network nic ip-config create](#).

```

Create NICs
az network nic create \
--name dsNIC0 \
--resource-group DsResourceGroup01 \
--network-security-group dsNSG1 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv4 \
--lb-address-pools dsLbBackEndPool_v4 \
--lb-name dsLB \
--public-ip-address dsVM0_remote_access

az network nic create \
--name dsNIC1 \
--resource-group DsResourceGroup01 \
--network-security-group dsNSG1 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv4 \
--lb-address-pools dsLbBackEndPool_v4 \
--lb-name dsLB \
--public-ip-address dsVM1_remote_access

Create IPV6 configurations for each NIC

az network nic ip-config create \
--name dsIp6Config_NIC0 \
--nic-name dsNIC0 \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name dsLB

az network nic ip-config create \
--name dsIp6Config_NIC1 \
--nic-name dsNIC1 \
--resource-group DsResourceGroup01 \
--vnet-name dsVNET \
--subnet dsSubNet \
--private-ip-address-version IPv6 \
--lb-address-pools dsLbBackEndPool_v6 \
--lb-name dsLB

```

## Create virtual machines

Create the VMs with [az vm create](#). The following example creates two VMs and the required virtual network components if they do not already exist.

Create virtual machine *dsVM0* as follows:

```

az vm create \
--name dsVM0 \
--resource-group DsResourceGroup01 \
--nics dsNIC0 \
--size Standard_A2 \
--availability-set dsAVset \
--image MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest

```

Create virtual machine *dsVM1* as follows:

```
az vm create \
--name dsVM1 \
--resource-group DsResourceGroup01 \
--nics dsNIC1 \
--size Standard_A2 \
--availability-set dsAVset \
--image MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest
```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *dsVnet*.
2. When **myVirtualNetwork** appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *dsVnet*. The dual stack virtual network shows the two NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *dsSubnet*.

The screenshot shows the Azure portal's 'Overview' page for a virtual network named 'dsVnet'. The left sidebar lists various settings and monitoring options. The main pane displays the virtual network's configuration, including its resource group ('DsRG02'), location ('West US'), and address space ('10.0.0.0/16, 1 more'). It also shows DNS servers ('Azure provided DNS service'). Below this, there are sections for 'Connected devices' and 'Peering', which list four network interfaces (DsVM0, DsVM0, DsVM1, DsVM1) in the 'DualStackSubnet'.

## Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group, VM, and all related resources.

```
az group delete --name DsResourceGroup01
```

## Next steps

In this article, you created a Standard Load Balancer with a dual frontend IP configuration (IPv4 and IPv6). You also created a two virtual machines that included NICs with dual IP configurations (IPV4 + IPV6) that were added to the back-end pool of the load balancer. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy an IPv6 dual stack application in Azure virtual network - Template

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides a list of IPv6 configuration tasks with the portion of the Azure Resource Manager VM template that applies to. Use the template described in this article to deploy a dual stack (IPv4 + IPv6) application using Standard Load Balancer in Azure that includes a dual stack virtual network with IPv4 and IPv6 subnets, a Standard Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, network security group, and public IPs.

## Required configurations

Search for the template sections in the template to see where they should occur.

### IPv6 addressSpace for the virtual network

Template section to add:

```
"addressSpace": {
 "addressPrefixes": [
 "[variables('vnetv4AddressRange')]",
 "[variables('vnetv6AddressRange')]"
]
}
```

### IPv6 subnet within the IPv6 virtual network addressSpace

Template section to add:

```
{
 "name": "V6Subnet",
 "properties": {
 "addressPrefix": "[variables('subnetv6AddressRange')]"
 }
}
```

### IPv6 configuration for the NIC

Template section to add:

```
{
 "name": "ipconfig-v6",
 "properties": {
 "privateIPAllocationMethod": "Dynamic",
 "privateIPAddressVersion": "IPv6",
 "subnet": {
 "id": "[variables('v6-subnet-id')]"
 },
 "loadBalancerBackendAddressPools": [
 {
 "id": "
[concat(resourceId('Microsoft.Network/loadBalancers','loadBalancer'), '/backendAddressPools/LBBAP-v6')]"
 }
]
 }
}
```

### IPv6 network security group (NSG) rules

```
{
 "name": "default-allow-rdp",
 "properties": {
 "description": "Allow RDP",
 "protocol": "Tcp",
 "sourcePortRange": "33819-33829",
 "destinationPortRange": "5000-6000",
 "sourceAddressPrefix": "fd00:db8:deca:deed::/64",
 "destinationAddressPrefix": "fd00:db8:deca:deed::/64",
 "access": "Allow",
 "priority": 1003,
 "direction": "Inbound"
 }
}
```

## Conditional configuration

If you're using a network virtual appliance, add IPv6 routes in the Route Table. Otherwise, this configuration is optional.

```
{
 "type": "Microsoft.Network/routeTables",
 "name": "v6route",
 "apiVersion": "[variables('ApiVersion')]",
 "location": "[resourceGroup().location]",
 "properties": {
 "routes": [
 {
 "name": "v6route",
 "properties": {
 "addressPrefix": "fd00:db8:deca:deed::/64",
 "nextHopType": "VirtualAppliance",
 "nextHopIpAddress": "fd00:db8:ace:f00d::1"
 }
 }
]
 }
}
```

## Optional configuration

### IPv6 Internet access for the virtual network

```
{
 "name": "LBFE-v6",
 "properties": {
 "publicIPAddress": {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses','lbppublicip-v6')]"
 }
 }
}
```

### IPv6 Public IP addresses

```
{
 "apiVersion": "[variables('ApiVersion')]",
 "type": "Microsoft.Network/publicIPAddresses",
 "name": "lbppublicip-v6",
 "location": "[resourceGroup().location]",
 "sku": {
 "name": "Standard"
 },
 "properties": {
 "publicIPAllocationMethod": "Static",
 "publicIPAddressVersion": "IPv6"
 }
}
```

## IPv6 Front end for Load Balancer

```
{
 "name": "LBFE-v6",
 "properties": {
 "publicIPAddress": {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses','lbpublicip-v6')]"
 }
 }
}
```

## IPv6 Back-end address pool for Load Balancer

```
"backendAddressPool": {
 "id": "[concat(resourceId('Microsoft.Network/loadBalancers', 'loadBalancer'),
 '/backendAddressPools/LBBAP-v6')]"
},
 "protocol": "Tcp",
 "frontendPort": 8080,
 "backendPort": 8080
},
 "name": "lbrule-v6"
```

## IPv6 load balancer rules to associate incoming and outgoing ports

```
{
 "name": "ipconfig-v6",
 "properties": {
 "privateIPAllocationMethod": "Dynamic",
 "privateIPAddressVersion": "IPv6",
 "subnet": {
 "id": "[variables('v6-subnet-id')]"
 },
 "loadBalancerBackendAddressPools": [
 {
 "id": "
[concat(resourceId('Microsoft.Network/loadBalancers','loadBalancer'),'/backendAddressPools/LBBAP-v6')]"
 }
]
 }
}
```

## Sample VM template JSON

To deploy an IPv6 dual stack application in Azure virtual network using Azure Resource Manager template, view sample template [here](#).

## Next steps

You can find details about pricing for [public IP addresses](#), [network bandwidth](#), or [Load Balancer](#).

# Deploy an IPv6 dual stack application using Standard Internal Load Balancer in Azure - PowerShell (Preview)

2/18/2021 • 9 minutes to read • [Edit Online](#)

This article shows you how to deploy a dual stack (IPv4 + IPv6) application in Azure that includes a dual stack virtual network and subnet, a Standard Internal Load Balancer with dual (IPv4 + IPv6) front-end configurations, VMs with NICs that have a dual IP configuration, network security group, and public IPs.

The procedure to create an IPv6-capable Internal Load Balancer is nearly identical to the process for creating an Internet-facing IPv6 Load Balancer described [here](#). The only differences for creating an internal load balancer are in the front-end configuration as illustrated in the PowerShell example below:

```
$frontendIPv6 = New-AzLoadBalancerFrontendIpConfig `
 -Name "dsLbFrontEnd_v6" `
 -PrivateIpAddress "fd00:db8:deca:deed::100" `
 -PrivateIpAddressVersion "IPv6" `
 -Subnet $DsSubnet
```

The changes that make the above an internal load balancer front-end configuration are:

- The `PrivateIpAddressVersion` is specified as "IPv6"
- The `-PublicIpAddress` argument has been either omitted or replaced with `-PrivateIpAddress`. Note that the private address must be in the range of the Subnet IP space in which the internal load balancer will be deployed. If a static `-PrivateIpAddress` is omitted, the next free IPv6 address will be selected from the subnet in which the internal load Balancer is deployed.
- The dual stack subnet in which the internal load balancer will be deployed is specified with either a `-Subnet` or `-SubnetId` argument.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	

OPTION	EXAMPLE/LINK
Select the <b>Cloud Shell</b> button on the menu bar at the upper right in the <a href="#">Azure portal</a> .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 6.9.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Prerequisites

Before you deploy a dual stack application in Azure, you must configure your subscription for this preview feature using the following Azure PowerShell:

Register as follows:

```
Register-AzProviderFeature -FeatureName AllowIPv6VirtualNetwork -ProviderNamespace Microsoft.Network
Register-AzProviderFeature -FeatureName AllowIPv6CAOnStandardLB -ProviderNamespace Microsoft.Network
```

It takes up to 30 minutes for feature registration to complete. You can check your registration status by running the following Azure PowerShell command: Check on the registration as follows:

```
Get-AzProviderFeature -FeatureName AllowIPv6VirtualNetwork -ProviderNamespace Microsoft.Network
Get-AzProviderFeature -FeatureName AllowIPv6CAOnStandardLB -ProviderNamespace Microsoft.Network
```

After the registration is complete, run the following command:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.Network
```

## Create a resource group

Before you can create your dual-stack virtual network, you must create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *dsStd\_ILB\_RG* in the *east us* location:

```
$rg = New-AzResourceGroup `
-ResourceGroupName "dsStd_ILB_RG" `
-Location "east us"
```

## Create IPv4 and IPv6 public IP addresses

To access your virtual machines from the Internet, you need IPv4 and IPv6 public IP addresses for the VMs.

Create public IP addresses with [New-AzPublicIpAddress](#). The following example creates IPv4 and IPv6 public IP address named *RdpPublicIP\_1* and *RdpPublicIP\_2* in the *dsStdILB\_RG* resource group:

```
$RdpPublicIP_1 = New-AzPublicIpAddress `
-Name "RdpPublicIP_1" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-AllocationMethod Static `
-IpAddressVersion IPv4 `
-sku Standard

$RdpPublicIP_2 = New-AzPublicIpAddress `
-Name "RdpPublicIP_2" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-AllocationMethod Static `
-IpAddressVersion IPv4 `
-sku Standard
```

## Create the virtual network and the subnet

Create a virtual network using [New-AzVirtualNetwork](#) with dual stack a subnet configuration using [New-AzVirtualNetworkSubnetConfig](#). The following example creates a virtual network named *dsVnet* with *dsSubnet*.

```
Create dual stack subnet config
$DsSubnet = New-AzVirtualNetworkSubnetConfig `
-Name "dsSubnet" `
-AddressPrefix "10.0.0.0/24","fd00:db8:deca:deed::/64"

Create the virtual network
$vnet = New-AzVirtualNetwork `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsVnet" `
-AddressPrefix "10.0.0.0/16","fd00:db8:deca::/48" `
-Subnet $DsSubnet

#Refresh the fully populated subnet for use in load balancer frontend configuration
$DsSubnet = get-AzVirtualNetworkSubnetconfig -name dsSubnet -VirtualNetwork $vnet
```

## Create Standard Load Balancer

In this section, you configure dual frontend IP (IPv4 and IPv6) and the back-end address pool for the load balancer and then create a Standard Load Balancer.

### Create front-end IP

Create a front-end IP with [New-AzLoadBalancerFrontendIpConfig](#). The following example creates IPv4 and IPv6 front-end IP configurations named *dsLbFrontEnd\_v4* and *dsLbFrontEnd\_v6*:

```
$frontendIPv4 = New-AzLoadBalancerFrontendIpConfig `
-Name "dsLbFrontEnd_v4" `
-PrivateIpAddress "10.0.0.100" `
-PrivateIpAddressVersion "IPv4" `
-Subnet $DsSubnet

$frontendIPv6 = New-AzLoadBalancerFrontendIpConfig `
-Name "dsLbFrontEnd_v6" `
-PrivateIpAddress "fd00:db8:deca:deed::100" `
-PrivateIpAddressVersion "IPv6" `
-Subnet $DsSubnet
```

## Configure back-end address pool

Create a back-end address pool with [New-AzLoadBalancerBackendAddressPoolConfig](#). The VMs attach to this back-end pool in the remaining steps. The following example creates back-end address pools named *dsLbBackEndPool\_v4* and *dsLbBackEndPool\_v6* to include VMs with both IPv4 and IPv6 NIC configurations:

```
$backendPoolv4 = New-AzLoadBalancerBackendAddressPoolConfig -Name "dsLbBackEndPool_v4"

$backendPoolv6 = New-AzLoadBalancerBackendAddressPoolConfig -Name "dsLbBackEndPool_v6"
```

## Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port. To make sure only healthy VMs receive traffic, you can optionally define a health probe. Basic load balancer uses an IPv4 probe to assess health for both IPv4 and IPv6 endpoints on the VMs. Standard load balancer includes support for explicitly IPv6 health probes.

Create a load balancer rule with [Add-AzLoadBalancerRuleConfig](#). The following example creates load balancer rules named *dsLBrule\_v4* and *dsLBrule\_v6* and balances traffic on TCP port 80 to the IPv4 and IPv6 frontend IP configurations:

```
$lbrule_v4 = New-AzLoadBalancerRuleConfig `
-Name "dsLBrule_v4" `
-FrontendIpConfiguration $frontendIPv4 `
-BackendAddressPool $backendPoolv4 `
-Protocol Tcp `
-FrontendPort 80 `
-BackendPort 80

$lbrule_v6 = New-AzLoadBalancerRuleConfig `
-Name "dsLBrule_v6" `
-FrontendIpConfiguration $frontendIPv6 `
-BackendAddressPool $backendPoolv6 `
-Protocol Tcp `
-FrontendPort 80 `
-BackendPort 80
```

## Create load balancer

Create a Standard Load Balancer with [New-AzLoadBalancer](#). The following example creates a public Standard Load Balancer named *myInternalLoadBalancer* using the IPv4 and IPv6 frontend IP configurations, backend pools, and load-balancing rules that you created in the preceding steps:

```
$lb = New-AzLoadBalancer `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -Name "MyInternalLoadBalancer" `
 -Sku "Standard" `
 -FrontendIpConfiguration $frontendIPv4,$frontendIPv6 `
 -BackendAddressPool $backendPoolv4,$backendPoolv6 `
 -LoadBalancingRule $lbrule_v4,$lbrule_v6
```

## Create network resources

Before you deploy some VMs and can test your balancer, you must create supporting network resources - availability set, network security group, and virtual NICs.

### Create an availability set

To improve the high availability of your application, place your VMs in an availability set.

Create an availability set with [New-AzAvailabilitySet](#). The following example creates an availability set named *dsAVset*.

```
$avset = New-AzAvailabilitySet `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -Name "dsAVset" `
 -PlatformFaultDomainCount 2 `
 -PlatformUpdateDomainCount 2 `
 -Sku aligned
```

### Create network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNet.

#### Create a network security group rule for port 3389

Create a network security group rule to allow RDP connections through port 3389 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule1 = New-AzNetworkSecurityRuleConfig `
 -Name 'myNetworkSecurityGroupRuleRDP' `
 -Description 'Allow RDP' `
 -Access Allow `
 -Protocol Tcp `
 -Direction Inbound `
 -Priority 100 `
 -SourceAddressPrefix * `
 -SourcePortRange * `
 -DestinationAddressPrefix * `
 -DestinationPortRange 3389
```

#### Create a network security group rule for port 80

Create a network security group rule to allow internet connections through port 80 with [New-AzNetworkSecurityRuleConfig](#).

```
$rule2 = New-AzNetworkSecurityRuleConfig `
-Name 'myNetworkSecurityGroupRuleHTTP' `
-Description 'Allow HTTP' `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 200 `
-SourceAddressPrefix * `
-SourcePortRange 80 `
-DestinationAddressPrefix * `
-DestinationPortRange 80
```

### Create a network security group

Create a network security group with [New-AzNetworkSecurityGroup](#).

```
$nsg = New-AzNetworkSecurityGroup `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "dsNSG1" `
-SecurityRules $rule1,$rule2
```

### Create NICs

Create virtual NICs with [New-AzNetworkInterface](#). The following example creates two virtual NICs both with IPv4 and IPv6 configurations. (One virtual NIC for each VM you create for your app in the following steps).

```

Create the IPv4 configuration for NIC 1
$Ip4Config=New-AzNetworkInterfaceIpConfig `
 -Name dsIp4Config `
 -Subnet $vnet.subnets[0] `
 -PrivateIpAddressVersion IPv4 `
 -LoadBalancerBackendAddressPool $backendPoolv4 `
 -PublicIpAddress $RdpPublicIP_1

Create the IPv6 configuration
$Ip6Config=New-AzNetworkInterfaceIpConfig `
 -Name dsIp6Config `
 -Subnet $vnet.subnets[0] `
 -PrivateIpAddressVersion IPv6 `
 -LoadBalancerBackendAddressPool $backendPoolv6

Create NIC 1
$NIC_1 = New-AzNetworkInterface `
 -Name "dsNIC1" `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -NetworkSecurityGroupId $nsg.Id `
 -IpConfiguration $Ip4Config,$Ip6Config

Create the IPv4 configuration for NIC 2
$Ip4Config=New-AzNetworkInterfaceIpConfig `
 -Name dsIp4Config `
 -Subnet $vnet.subnets[0] `
 -PrivateIpAddressVersion IPv4 `
 -LoadBalancerBackendAddressPool $backendPoolv4 `
 -PublicIpAddress $RdpPublicIP_2

Create NIC 2 reusing the IPv6 configuration from NIC 1
$NIC_2 = New-AzNetworkInterface `
 -Name "dsNIC2" `
 -ResourceGroupName $rg.ResourceGroupName `
 -Location $rg.Location `
 -NetworkSecurityGroupId $nsg.Id `
 -IpConfiguration $Ip4Config,$Ip6Config

```

## Create virtual machines

Set an administrator username and password for the VMs with [Get-Credential](#):

```
$cred = get-credential -Message "DUAL STACK VNET SAMPLE: Please enter the Administrator credential to log into the VM's"
```

Now you can create the VMs with [New-AzVM](#). The following example creates two VMs and the required virtual network components if they do not already exist.

```

$vmsize = "Standard_A2"
$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "WindowsServer"
$imageSKU = "2019-Datacenter"

$vmName= "dsVM1"
$VMconfig1 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-
AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-
AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null
| Set-AzVMDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id
$NIC_1.Id 3> $null
$VM1 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig1

$vmName= "dsVM2"
$VMconfig2 = New-AzVMConfig -VMName $vmName -VMSize $vmsize -AvailabilitySetId $avset.Id 3> $null | Set-
AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent 3> $null | Set-
AzVMSourceImage -PublisherName $ImagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" 3> $null
| Set-AzVMDisk -Name "$vmName.vhd" -CreateOption fromImage 3> $null | Add-AzVMNetworkInterface -Id
$NIC_2.Id 3> $null
$VM2 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $VMconfig2

```

## View IPv6 dual stack virtual network in Azure portal

You can view the IPv6 dual stack virtual network in Azure portal as follows:

1. In the portal's search bar, enter *dsVnet*.
2. When *dsVnet* appears in the search results, select it. This launches the **Overview** page of the dual stack virtual network named *dsVnet*. The dual stack virtual network shows the two NICs with both IPv4 and IPv6 configurations located in the dual stack subnet named *dsSubnet*.

The screenshot shows the Azure portal interface for a virtual network named *dsVnet*. The left sidebar lists various settings and logs. The main pane shows the following details:

- Resource group:** *dsStd\_ILB\_RG*
- Location:** East US
- Subscription:** (change)
- Tags:** Click here to add tags
- Address space:** 10.0.0.0/16, 1 more
- DNS servers:** Azure provided DNS service

**Connected devices:**

DEVICE	TYPE	IP ADDRESS	SUBNET
dsNIC1	Network interface	10.0.0.4	dsSubnet
dsNIC1	Network interface	ace:cab:dec:a:deed::4	dsSubnet
dsNIC2	Network interface	10.0.0.5	dsSubnet
dsNIC2	Network interface	ace:cab:dec:a:deed::5	dsSubnet

### NOTE

The IPv6 for Azure virtual network is available in the Azure portal in read-only for this preview release.

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name dsStd_ILB_RG
```

## Next steps

In this article, you created a Standard Load Balancer with a dual frontend IP configuration (IPv4 and IPv6). You also created a two virtual machines that included NICs with dual IP configurations (IPV4 + IPV6) that were added to the back-end pool of the load balancer. To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#)

# Deploy virtual machine scale sets with IPv6 in Azure

11/2/2020 • 2 minutes to read • [Edit Online](#)

This article shows you how to deploy a dual stack (IPv4 + IPv6) Virtual Machine Scale Set with a dual stack external load balancer in an Azure virtual network. The process to create an IPv6-capable virtual machine scale set is nearly identical to the process for creating individual VMs described [here](#). You'll start with the steps that are similar to ones described for individual VMs:

1. Create IPv4 and IPv6 Public IPs.
2. Create a dual stack load balancer.
3. Create network security group (NSG) rules.

The only step that is different from individual VMs is creating the network interface (NIC) configuration that uses the virtual machine scale set resource: `networkProfile/networkInterfaceConfigurations`. The JSON structure is similar to that of the `Microsoft.Network/networkInterfaces` object used for individual VMs with the addition of setting the NIC and the IPv4 `IpConfiguration` as the primary interface using the `"primary": true` attribute as seen in the following example:

```

"networkProfile": {
 "networkInterfaceConfigurations": [
 {
 "name": "[variables('nicName')]",
 "properties": {
 "primary": true,
 "networkSecurityGroup": {
 "id": "[resourceId('Microsoft.Network/networkSecurityGroups', 'VmssNsg')]"
 },
 "ipConfigurations": [
 {
 "name": "[variables('ipConfigName')]",
 "properties": {
 "primary": true,
 "subnet": {
 "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets',
'MyvirtualNetwork', 'Mysubnet')]"
 },
 "privateIPAddressVersion": "IPv4",
 "publicIpAddressConfiguration": {
 "name": "pub1",
 "properties": {
 "idleTimeoutInMinutes": 15
 }
 },
 "loadBalancerBackendAddressPools": [
 {
 "id": "[resourceId('Microsoft.Network/loadBalancers/backendAddressPools',
'loadBalancer', 'bePool'))]"
 }
],
 "loadBalancerInboundNatPools": [
 {
 "id": "[resourceId('Microsoft.Network/loadBalancers/inboundNatPools',
'loadBalancer', 'natPool'))]"
 }
]
 }
 },
 {
 "name": "[variables('ipConfigNameV6')]",
 "properties": {
 "subnet": {
 "id": "
[resourceId('Microsoft.Network/virtualNetworks/subnets', 'MyvirtualNetwork', 'Mysubnet')]"
 },
 "privateIPAddressVersion": "IPv6",
 "loadBalancerBackendAddressPools": [
 {
 "id": "[resourceId('Microsoft.Network/loadBalancers/backendAddressPools',
'loadBalancer', 'bePoolv6'))]"
 }
],
 "loadBalancerInboundNatPools": [
 {
 "id": "
[resourceId('Microsoft.Network/loadBalancers/inboundNatPools', 'loadBalancer',
'natPoolv6'))]"
 }
]
 }
 }
]
 }
]
}

```

## Sample virtual machine scale set template JSON

To deploy a dual stack (IPv4 + IPv6) Virtual Machine Scale Set with dual stack external Load Balancer and virtual

network view sample template [here](#).

## Next steps

To learn more about IPv6 support in Azure virtual networks, see [What is IPv6 for Azure Virtual Network?](#).

# Reserve public IPv6 address prefix

11/2/2020 • 2 minutes to read • [Edit Online](#)

IPv6 for Azure Virtual Network (VNet) enables you to host applications in Azure with IPv6 and IPv4 connectivity both within a virtual network and to and from the Internet. In addition to reserving individual IPv6 addresses, you can reserve contiguous ranges of Azure IPv6 addresses (known as IP Prefix) for your use. This article describes how to create IPv6 public IP addresses and address ranges using Azure PowerShell and CLI.

## Create a single reserved IPv6 public IP

### Using Azure PowerShell

You can create a single reserved (static) IPv6 Public IP address using Azure PowerShell with [New-AzPublicIpAddress](#) as follows:

```
$myOwnIPv6Address = New-AzPublicIpAddress `
-name PIPv6_WestUS `
-ResourceGroup MyRG `
-Location "West US" `
-Sku Standard `
-allocationMethod static `
-IpAddressVersion IPv6
```

### Using Azure CLI

You can create a single reserved (static) IPv6 Public IP address Azure CLI with [az network public-ip create](#) as follows:

```
az network public-ip create \
--name dsPublicIP_v6 \
--resource-group UpgradeInPlace_CLI_RG1 \
--location WestUS \
--sku Standard \
--allocation-method static \
--version IPv6
```

## Create a reserved IPv6 prefix (range)

To reserve an IPv6 prefix, add the IP address family of IPv6 to the same command used for creating IPv4 prefixes. The following commands create a prefix of size /125 ( 8 IPv6 addresses).

### Using Azure PowerShell

You can create a public IPv6 address using Azure CLI with [az network public-ip create](#) as follows:

```
$myOwnIPv6Prefix = New-AzPublicIpPrefix `
-name IPv6PrefixWestUS `
-ResourceGroupName MyRG `
-Location "West US" `
-Sku Standard `
-IpAddressVersion IPv6 `
-PrefixLength 125
```

### Using Azure CLI

You can create a public IPv6 address using Azure CLI as follows:

```
az network public-ip prefix create \
--name IPv6PrefixWestUS \
--resource-group MyRG \
--location WestUS \
--version IPv6 \
--length 125
```

## Allocate a public IP address from a reserved IPv6 Prefix

### Using Azure PowerShell

You create a static IPv6 Public IP from a reserved prefix by adding the `-PublicIpPrefix` argument when creating the public IP using Azure PowerShell. The following example assumes that a prefix was created and stored in a PowerShell variable named: `$myOwnIPv6Prefix`.

```
$MyIPv6PublicIPFromMyReservedPrefix = New-AzPublicIpAddress \
-name PIPv6_fromPrefix `
-ResourceGroup DsStdLb04 `
-Location "West Central US" `
-Sku Standard `
-allocationMethod static `
-IpAddressVersion IPv6 `
-PublicIpPrefix $myOwnIPv6Prefix
```

### Using Azure CLI

The following example assumes that a prefix was created and stored in a CLI variable named: `IPv6PrefixWestUS`.

```
az network public-ip create \
--name dsPublicIP_v6 \
--resource-group UpgradeInPlace_CLI_RG1 \
--location WestUS \
--sku Standard \
--allocation-method static \
--version IPv6 \
--public-ip-prefix IPv6PrefixWestUS
```

## Next steps

- Learn more about [IPv6 address prefix](#).
- Learn more about [IPv6 addresses](#).

# Quickstart: Create a public IP address using the Azure portal

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to create a public IP address resource using the Azure portal. For more information on which resources this can be associated to, the difference between Basic and Standard SKU, and other related information, please see [Public IP addresses](#). For this example, we will focus on IPv4 addresses only; for more information on IPv6 addresses, see [IPv6 for Azure VNet](#).

- [Standard SKU](#)
- [Basic SKU](#)

Use the following steps to create a standard zone-redundant public IP address named `myStandardZRPublicIP`.

1. Sign in to the [Azure portal](#).
2. Select **Create a resource**.
3. In the search box, type *Public IP address*.
4. In the search results, select **Public IP address**. Next, in the **Public IP address** page, select **Create**.
5. On the **Create public IP address** page, enter or select the following information:

SETTING	VALUE
IP Version	Select IPv4
SKU	Select <b>Standard</b>
Tier (if shown*)	Select <b>Regional</b>
Name	Enter <code>myStandardZRPublicIP</code>
IP address assignment	Note this will be locked as "Static"
Idle Timeout (minutes)	Leave the value at 4
DNS name label	Leave the value as blank
Subscription	Select your subscription.
Resource group	Select <b>Create new</b> , enter <code>myResourceGroup</code> , then select <b>OK</b>
Location	Select <b>East US 2</b>
Availability Zone	Select <b>Zone-Redundant</b> , <b>No Zone</b> , or pick specific Zone (see note below)

Note that these are only valid selections in regions with [Availability Zones](#). (You can also select a specific zone in

these regions, though it will not be resilient to zonal failure.) For more information on availability zones, see [Availability zones overview](#).

\* = Tier relates to the [Cross-Region Load Balancer](#) functionality, currently in Preview.

## Additional information

For more details on the individual fields listed above, please see [Manage public IP addresses](#).

## Next steps

- Associate a [public IP address to a Virtual Machine](#)
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).

# Quickstart: Create a public IP address using Azure PowerShell

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article shows you how to create a public IP address resource using Azure PowerShell. For more information on which resources this can be associated to, the difference between Basic and Standard SKU, and other related information, please see [Public IP addresses](#). For this example, we will focus on IPv4 addresses only; for more information on IPv6 addresses, see [IPv6 for Azure VNet](#).

## Prerequisites

- Azure PowerShell installed locally or Azure Cloud Shell

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the Copy button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 5.4.1 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you're running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with `New-AzResourceGroup` named `myResourceGroup` in the `eastus2` location.

```
Variables for the command
$rg = 'myResourceGroup'
$loc = 'eastus2'

New-AzResourceGroup -Name $rg -Location $loc
```

## Create public IP

- [Standard SKU - Using zones](#)
- [Standard SKU - No zones](#)
- [Basic SKU](#)

### NOTE

The following command works for Az.Network module version 4.5.0 or later. For more information about the Powershell modules currently being used, please refer to the [PowerShellGet documentation](#).

Use `New-AzPublicIpAddress` to create a standard zone-redundant public IP address named `myStandardZRPublicIP` in `myResourceGroup`.

```
Variables for the command
$rg = 'myResourceGroup'
$loc = 'eastus2'
$pubIP = 'myStandardZRPublicIP'
$sku = 'Standard'
$alloc = 'Static'
$zone = 1,2,3

New-AzPublicIpAddress -ResourceGroupName $rg -Name $pubIP -Location $loc -AllocationMethod $alloc -SKU $sku
-zone $zone
```

### IMPORTANT

For Az.Network modules older than 4.5.0, run the command above without specifying a zone parameter to create a zone-redundant IP address.

In order to create a standard zonal public IP address in Zone 2 named `myStandardZonalPublicIP` in `myResourceGroup`, use the following command:

```
Variables for the command
$rg = 'myResourceGroup'
$loc = 'eastus2'
$pubIP = 'myStandardZonalPublicIP'
$sku = 'Standard'
$alloc = 'Static'
$zone = 2

New-AzPublicIpAddress -ResourceGroupName $rg -Name $pubIP -Location $loc -AllocationMethod $alloc -SKU $sku
-zone $zone
```

Note that the above options for zones are only valid selections in regions with [Availability Zones](#).

## Additional information

For more details on the individual variables listed above, please see [Manage public IP addresses](#).

## Next steps

- Associate a [public IP address to a Virtual Machine](#)
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).

# Quickstart: Create a public IP address using Azure CLI

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to create a public IP address resource using Azure CLI. For more information on which resources this can be associated to, the difference between Basic and Standard SKU, and other related information, please see [Public IP addresses](#). For this example, we will focus on IPv4 addresses only; for more information on IPv6 addresses, see [IPv6 for Azure VNet](#).

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with `az group create` named `myResourceGroup` in the `eastus2` location.

```
az group create \
 --name myResourceGroup \
 --location eastus2
```

## Create public IP

- [Standard SKU - Using zones](#)
- [Standard SKU - No zones](#)
- [Basic SKU](#)

#### NOTE

The following command works for API version 2020-08-01 or later. For more information about the API version currently being used, please refer to [Resource Providers and Types](#).

Use [az network public-ip create](#) to create a standard zone-redundant public IP address named **myStandardZRPublicIP** in **myResourceGroup**.

```
az network public-ip create \
--resource-group myResourceGroup \
--name myStandardZRPublicIP \
--sku Standard \
--zone 1 2 3
```

#### IMPORTANT

For versions of the API older than 2020-08-01, run the command above without specifying a zone parameter to create a zone-redundant IP address.

In order to create a standard zonal public IP address in Zone 2 named **myStandardZonalPublicIP** in **myResourceGroup**, use the following command:

```
az network public-ip create \
--resource-group myResourceGroupLB \
--name myStandardZonalPublicIP \
--sku Standard \
--zone 2
```

Note that the above options for zones are only valid selections in regions with [Availability Zones](#).

## Additional information

For more details on the individual variables listed above, please see [Manage public IP addresses](#).

## Next steps

- Associate a [public IP address to a Virtual Machine](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).

# Manage public IP addresses

2/18/2021 • 12 minutes to read • [Edit Online](#)

Learn about a public IP address and how to create, change, and delete one. A public IP address is a resource with its own configurable settings. Assigning a public IP address to an Azure resource that supports public IP addresses enables:

- Inbound communication from the Internet to the resource, such as Azure Virtual Machines (VM), Azure Application Gateways, Azure Load Balancers, Azure VPN Gateways, and others. You can still communicate with some resources, such as VMs, from the Internet, if a VM doesn't have a public IP address assigned to it, as long as the VM is part of a load balancer back-end pool, and the load balancer is assigned a public IP address. To determine whether a resource for a specific Azure service can be assigned a public IP address, or whether it can be communicated with through the public IP address of a different Azure resource, see the documentation for the service.
- Outbound connectivity to the Internet using a predictable IP address. For example, a virtual machine can communicate outbound to the Internet without a public IP address assigned to it, but its address is network address translated by Azure to an unpredictable public address, by default. Assigning a public IP address to a resource enables you to know which IP address is used for the outbound connection. Though predictable, the address can change, depending on the assignment method chosen. For more information, see [Create a public IP address](#). To learn more about outbound connections from Azure resources, see [Understand outbound connections](#).

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with your Azure account.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.31 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Permissions](#).

Public IP addresses have a nominal charge. To view the pricing, read the [IP address pricing](#) page.

## Create a public IP address

For instructions on how to Create Public IP addresses using the Portal, PowerShell, or CLI -- please refer to the following pages:

- [Create public IP addresses - portal](#)
- [Create public IP addresses - PowerShell](#)
- [Create public IP addresses - Azure CLI](#)

### NOTE

Though the portal provides the option to create two public IP address resources (one IPv4 and one IPv6), the PowerShell and CLI commands create one resource with an address for one IP version or the other. If you want two public IP address resources, one for each IP version, you must run the command twice, specifying different names and IP versions for the public IP address resources.

For additional detail on the specific attributes of a Public IP address during creation, see the table below.

SETTING	REQUIRED?	DETAILS
IP Version	Yes	Select IPv4 or IPv6 or Both. Selecting Both will result in 2 Public IP addresses being created- 1 IPv4 address and 1 IPv6 address. Learn more about <a href="#">IPv6 in Azure VNETs</a> .

Setting	Required?	Details
SKU	Yes	All public IP addresses created before the introduction of SKUs are <b>Basic</b> SKU public IP addresses. You cannot change the SKU after the public IP address is created. A standalone virtual machine, virtual machines within an availability set, or virtual machine scale sets can use Basic or Standard SKUs. Mixing SKUs between virtual machines within availability sets or scale sets or standalone VMs is not allowed. <b>Basic</b> SKU: If you are creating a public IP address in a region that supports availability zones, the <b>Availability zone</b> setting is set to <i>None</i> by default. Basic Public IPs do not support Availability zones. <b>Standard</b> SKU: A Standard SKU public IP can be associated to a virtual machine or a load balancer front end. If you're creating a public IP address in a region that supports availability zones, the <b>Availability zone</b> setting is set to <i>Zone-redundant</i> by default. For more information about availability zones, see the <b>Availability zone</b> setting. The standard SKU is required if you associate the address to a Standard load balancer. To learn more about standard load balancers, see <a href="#">Azure load balancer standard SKU</a> . When you assign a standard SKU public IP address to a virtual machine's network interface, you must explicitly allow the intended traffic with a <a href="#">network security group</a> . Communication with the resource fails until you create and associate a network security group and explicitly allow the desired traffic.
Tier	Yes	Indicates if the IP address is associated with a region ( <b>Regional</b> ) or is "anycast" from multiple regions ( <b>Global</b> ). Note that a "Global Tier" IP is preview functionality for Standard IPs, and currently only utilized for the Cross-Region Load Balancer.
Name	Yes	The name must be unique within the resource group you select.

Setting	Required?	Details
IP address assignment	Yes	<p><b>Dynamic:</b> Dynamic addresses are assigned only after a public IP address is associated to an Azure resource, and the resource is started for the first time. Dynamic addresses can change if they're assigned to a resource, such as a virtual machine, and the virtual machine is stopped (deallocated), and then restarted. The address remains the same if a virtual machine is rebooted or stopped (but not deallocated). Dynamic addresses are released when a public IP address resource is dissociated from a resource it is associated to. <b>Static:</b> Static addresses are assigned when a public IP address is created. Static addresses are not released until a public IP address resource is deleted. If the address is not associated to a resource, you can change the assignment method after the address is created. If the address is associated to a resource, you may not be able to change the assignment method. If you select <i>/IPv6</i> for the <b>IP version</b>, the assignment method must be <i>Dynamic</i> for Basic SKU. Standard SKU addresses are <i>Static</i> for both IPv4 and IPv6.</p>
Idle timeout (minutes)	No	How many minutes to keep a TCP or HTTP connection open without relying on clients to send keep-alive messages. If you select IPv6 for <b>IP Version</b> , this value can't be changed.

Setting	Required?	Details
DNS name label	No	<p>Must be unique within the Azure location you create the name in (across all subscriptions and all customers). Azure automatically registers the name and IP address in its DNS so you can connect to a resource with the name. Azure appends a default subnet such as <i>location.cloudapp.azure.com</i> (where location is the location you select) to the name you provide, to create the fully qualified DNS name. If you choose to create both address versions, the same DNS name is assigned to both the IPv4 and IPv6 addresses. Azure's default DNS contains both IPv4 A and IPv6 AAAA name records and responds with both records when the DNS name is looked up. The client chooses which address (IPv4 or IPv6) to communicate with. Instead of, or in addition to, using the DNS name label with the default suffix, you can use the Azure DNS service to configure a DNS name with a custom suffix that resolves to the public IP address. For more information, see <a href="#">Use Azure DNS with an Azure public IP address</a>.</p>
Name (Only visible if you select IP Version of Both)	Yes, if you select IP Version of Both	The name must be different than the name you enter for the first Name in this list. If you choose to create both an IPv4 and an IPv6 address, the portal creates two separate public IP address resources, one with each IP address version assigned to it.
IP address assignment (Only visible if you select IP Version of Both)	Yes, if you select IP Version of Both	Same restrictions as IP Address Assignment above
Subscription	Yes	Must exist in the same <a href="#">subscription</a> as the resource to which you'll associate the Public IP's.
Resource group	Yes	Can exist in the same, or different, <a href="#">resource group</a> as the resource to which you'll associate the Public IP's.
Location	Yes	Must exist in the same <a href="#">location</a> , also referred to as region, as the resource to which you'll associate the Public IP's.

SETTING	REQUIRED?	DETAILS
Availability zone	No	This setting only appears if you select a supported location. For a list of supported locations, see <a href="#">Availability zones overview</a> . If you selected the <b>Basic</b> SKU, <i>None</i> is automatically selected for you. If you prefer to guarantee a specific zone, you may select a specific zone. Either choice is not zone-redundant. If you selected the <b>Standard</b> SKU: Zone-redundant is automatically selected for you and makes your data path resilient to zone failure. If you prefer to guarantee a specific zone, which is not resilient to zone failure, you may select a specific zone.

## View, modify settings for, or delete a public IP address

- **View/List:** To review settings for a Public IP, including the SKU, address, any applicable association (e.g. Virtual Machine NIC, Load Balancer Frontend).
- **Modify:** To modify settings using the information in step 4 of [create a public IP address](#), such as the idle timeout, DNS name label, or assignment method. (For the full process of upgrading a Public IP SKU from Basic to Standard, see [Upgrade Azure public IP addresses](#).)

### WARNING

To change the assignment for a Public IP address from static to dynamic, you must first dissociate the address from any applicable IP configurations (see **Delete** section). Also note, when you change the assignment method from static to dynamic, you lose the IP address that was assigned to the public IP address. While the Azure public DNS servers maintain a mapping between static or dynamic addresses and any DNS name label (if you defined one), a dynamic IP address can change when the virtual machine is started after being in the stopped (deallocated) state. To prevent the address from changing, assign a static IP address.

OPERATION	AZURE PORTAL	AZURE POWERSHELL	AZURE CLI
View	In the <b>Overview</b> section of a Public IP	<a href="#">Get-AzPublicIpAddress</a> to retrieve a public IP address object and view its settings	<a href="#">az network public-ip show</a> to show settings
List	Under the <b>Public IP addresses</b> category	<a href="#">Get-AzPublicIpAddress</a> to retrieve one or more public IP address objects and view its settings	<a href="#">az network public-ip list</a> to list public IP addresses
Modify	For an IP that is dissociated, select <b>Configuration</b> to modify idle timeout, DNS name label, or change assignment of Basic IP from Static to Dynamic	<a href="#">Set-AzPublicIpAddress</a> to update settings	<a href="#">az network public-ip update</a> to update

- **Delete:** Deletion of Public IPs requires that the Public IP object not be associated to any IP configuration or Virtual Machine NIC. See the table below for more details.

RESOURCE	AZURE PORTAL	AZURE POWERSHELL	AZURE CLI
Virtual Machine	Select <b>Dissociate</b> to dissociate the IP address from the NIC configuration, then select <b>Delete</b> .	<a href="#">Set-AzPublicIpAddress</a> to dissociate the IP address from the NIC configuration; <a href="#">Remove-AzPublicIpAddress</a> to delete	<a href="#">az network public-ip update --remove</a> to dissociate the IP address from the NIC configuration; <a href="#">az network public-ip delete</a> to delete
Load Balancer Frontend	Navigate to an unused Public IP address and select <b>Associate</b> and pick the Load Balancer with the relevant Front End IP Configuration to replace it (then the old IP can be deleted using same method as for VM)	<a href="#">Set-AzLoadBalancerFrontendIpConfig</a> to associate new Frontend IP config with Public Load Balancer; <a href="#">Remove-AzPublicIpAddress</a> to delete; can also use <a href="#">Remove-AzLoadBalancerFrontendIpConfig</a> to remove Frontend IP Config if there are more than one	<a href="#">az network lb frontend-ip update</a> to associate new Frontend IP config with Public Load Balancer; <a href="#">Remove-AzPublicIpAddress</a> to delete; can also use <a href="#">az network lb frontend-ip delete</a> to remove Frontend IP Config if there are more than one
Firewall	N/A	<a href="#">Deallocate()</a> to deallocate firewall and remove all IP configurations	<a href="#">az network firewall ip-config delete</a> to remove IP (but must use PowerShell to deallocate first)

## Virtual Machine Scale Sets

When using a virtual machine scale set with Public IPs, there are not separate Public IP objects associated with the individual virtual machine instances. However, a Public IP Prefix object [can be used to generate the instance IPs](#).

To list the Public IPs on a virtual machine scale set, you can use PowerShell ([Get-AzPublicIpAddress -VirtualMachineScaleSetName](#)) or CLI ([az vmss list-instance-public-ips](#)).

For more information, see [Networking for Azure virtual machine scale sets](#).

## Assign a public IP address

Learn how to assign a public IP address to the following resources:

- A Windows or [Linux](#) Virtual Machine (when creating), or to an existing [Virtual Machine](#)
- [Public Load Balancer](#)
- [Application Gateway](#)
- [Site-to-site connection using a VPN Gateway](#)
- [Virtual Machine Scale Set](#)

## Permissions

To perform tasks on public IP addresses, your account must be assigned to the [network contributor](#) role or to a [custom](#) role that is assigned the appropriate actions listed in the following table:

ACTION	NAME
<a href="#">Microsoft.Network/publicIPAddresses/read</a>	Read a public IP address

ACTION	NAME
Microsoft.Network/publicIPAddresses/write	Create or update a public IP address
Microsoft.Network/publicIPAddresses/delete	Delete a public IP address
Microsoft.Network/publicIPAddresses/join/action	Associate a public IP address to a resource

## Next steps

- Create a public IP address using [PowerShell](#) or [Azure CLI](#) sample scripts, or using Azure [Resource Manager templates](#)
- Create and assign [Azure Policy definitions](#) for public IP addresses

# Create, change, or delete a public IP address prefix

2/18/2021 • 6 minutes to read • [Edit Online](#)

Learn about a public IP address prefix and how to create, change, and delete one. A public IP address prefix is a contiguous range of addresses based on the number of public IP addresses you specify. The addresses are assigned to your subscription. When you create a public IP address resource, you can assign a static public IP address from the prefix and associate the address to virtual machines, load balancers, or other resources, to enable internet connectivity. If you're not familiar with public IP address prefixes, see [Public IP address prefix overview](#)

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with your Azure account.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.41 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI 2.0](#). If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Permissions](#).

Public IP address prefixes have a charge. For details, see [pricing](#).

## Create a public IP address prefix

1. At the top, left corner of the portal, select + **Create a resource**.
2. Enter *public ip prefix* in the *Search the Marketplace* box. When **Public IP address prefix** appears in the search results, select it.
3. Under **Public IP address prefix**, select **Create**.
4. Enter, or select values for the following settings, under **Create public IP address prefix**, then select

## Create:

SETTING	REQUIRED?	DETAILS
Subscription	Yes	Must exist in the same <a href="#">subscription</a> as the resource you want to associate the public IP address to.
Resource group	Yes	Can exist in the same, or different, <a href="#">resource group</a> as the resource you want to associate the public IP address to.
Name	Yes	The name must be unique within the resource group you select.
Region	Yes	Must exist in the same <a href="#">region</a> as the public IP addresses you'll assign addresses from the range.
Prefix size	Yes	The size of the prefix you need. A /28 or 16 IP addresses is the default.

## Commands

TOOL	COMMAND
CLI	<a href="#">az network public-ip prefix create</a>
PowerShell	<a href="#">New-AzPublicIpPrefix</a>

### NOTE

In regions with availability zones, you can use PowerShell or CLI commands to create a public IP address prefix as either: non-zonal, associated with a specific zone, or to use zone-redundancy. For API version 2020-08-01 or later, if a zone parameter is not provided, a non-zonal public IP address prefix is created. For versions of the API older than 2020-08-01, a zone-redundant public IP address prefix is created.

## Create a static public IP address from a prefix

Once you create a prefix, you must create static IP addresses from the prefix. In order to do this, follow steps below.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *public ip address prefix*. When **Public IP address prefixes** appear in the search results, select it.
2. Select the prefix you want to create public IPs from.
3. When it appears in the search results, select it and click on **+ Add IP address** in the Overview section.
4. Enter or select values for the following settings under **Create public IP address**. Since a prefix is for Standard SKU, IPv4, and static, you only need to provide the following information:

SETTING	REQUIRED?	DETAILS
Name	Yes	The name of the public IP address must be unique within the resource group you select.
Idle timeout (minutes)	No	How many minutes to keep a TCP or HTTP connection open without relying on clients to send keep-alive messages.
DNS name label	No	Must be unique within the Azure region you create the name in (across all subscriptions and all customers). Azure automatically registers the name and IP address in its DNS so you can connect to a resource with the name. Azure appends a default subnet such as <i>location.cloudapp.azure.com</i> (where location is the location you select) to the name you provide, to create the fully qualified DNS name. For more information, see <a href="#">Use Azure DNS with an Azure public IP address</a> .

Alternatively you may use the CLI and PS commands below with the --public-ip-prefix (CLI) and -PublicIpPrefix (PS) parameters, to create a Public IP address resource.

TOOL	COMMAND
CLI	<a href="#">az network public-ip create</a>
PowerShell	<a href="#">New-AzPublicIpAddress</a>

## View or delete a prefix

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *public ip address prefix*. When **Public IP address prefixes** appear in the search results, select it.
2. Select the name of the public IP address prefix that you want to view, change settings for, or delete from the list.
3. Complete one of the following options, depending on whether you want to view, delete, or change the public IP address prefix.
  - **View:** The **Overview** section shows key settings for the public IP address prefix, such as prefix.
  - **Delete:** To delete the public IP address prefix, select **Delete** in the **Overview** section. If addresses within the prefix are associated to public IP address resources, you must first delete the public IP address resources. See [delete a public IP address](#).

## Commands

TOOL	COMMAND

TOOL	COMMAND
CLI	<a href="#">az network public-ip prefix list</a> to list public IP addresses, <a href="#">az network public-ip prefix show</a> to show settings; <a href="#">az network public-ip prefix update</a> to update; <a href="#">az network public-ip prefix delete</a> to delete
PowerShell	<a href="#">Get-AzPublicIpPrefix</a> to retrieve a public IP address object and view its settings, <a href="#">Set-AzPublicIpPrefix</a> to update settings; <a href="#">Remove-AzPublicIpPrefix</a> to delete

## Permissions

To perform tasks on public IP address prefixes, your account must be assigned to the [network contributor](#) role or to a [custom](#) role that is assigned the appropriate actions listed in the following table:

ACTION	NAME
Microsoft.Network/publicIPPrefixes/read	Read a public IP address prefix
Microsoft.Network/publicIPPrefixes/write	Create or update a public IP address prefix
Microsoft.Network/publicIPPrefixes/delete	Delete a public IP address prefix
Microsoft.Network/publicIPPrefixes/join/action	Create a public IP address from a prefix

## Next steps

- Learn about scenarios and benefits of using a [public IP prefix](#)

# Upgrade public IP addresses

2/18/2021 • 4 minutes to read • [Edit Online](#)

Azure public IP addresses are created with a SKU--either Basic or Standard--which determines aspects of their functionality (including allocation method, feature support, and which resources they can be associated with).

The following scenarios are reviewed in this article:

- How to upgrade a Basic SKU public IP to a Standard SKU public IP (using PowerShell or CLI)
- How to migrate a Classic Azure Reserved IP to an Azure Resource Manager Basic SKU public IP

## Upgrade public IP address from Basic to Standard SKU

In order to upgrade a public IP, it must not be associated with any resource (see [this page](#) for more information about how to disassociate public IPs).

### IMPORTANT

Public IPs upgraded from Basic to Standard SKU continue to have no guaranteed [availability zones](#). Please ensure this is kept in mind when choosing which resources to associate the IP address with.

- [Basic to Standard - PowerShell](#)
- [Basic to Standard - CLI](#)

The following example assumes previous creation of a Basic SKU public IP, using the example given on [this page](#) with a Basic public IP `myBasicPublicIP` in `myResourceGroup`.

In order to upgrade the IP, simply execute the commands below using PowerShell. Note if the IP address is already statically allocated, that section can be skipped.

```
Variables for the command
$rg = 'myResourceGroup'
$name = 'myBasicPublicIP'
$newsku = 'Standard'
$pubIP = Get-AzPublicIpAddress -name $name -ResourceGroupName $rg

This section is only needed if the Basic IP is not already set to Static
$pubIP.PublicIpAllocationMethod = 'Static'
Set-AzPublicIpAddress -PublicIpAddress $pubIP

This section is for conversion to Standard
$pubIP.Sku.Name = $newsku
Set-AzPublicIpAddress -PublicIpAddress $pubIP
```

## Upgrade (migrate) a classic Reserved IP to a static public IP

To benefit from the new capabilities in Azure Resource Manager, you can migrate existing public static IP address--called Reserved IPs--from the Classic model to the modern Azure Resource Manager model. The migrated public IP will be a Basic SKU type.

- [Reserved to Basic - PowerShell](#)
- [Reserved to Basic - CLI](#)

The following example assumes previous creation of a classic Azure Reserved IP **myReservedIP** in **myResourceGroup**. Another prerequisite for migration is to ensure the Azure Resource Manager subscription has registered for migration. This is covered in detail on Steps 3 and 4 of [this page](#).

In order to migrate the Reserved IP, execute the commands below using PowerShell. Note if the IP address is not associated with any service (below there is a service named **myService**), that step can be skipped.

```
Variables for the command
$name = 'myReservedIP'

This section is only needed if the Reserved IP is not already disassociated from any Cloud Services
$serviceName = 'myService'
Remove-AzureReservedIPAssociation -ReservedIPName $name -ServiceName $service

$validate = Move-AzureReservedIP -ReservedIPName $name -Validate
$validate.ValidationMessages
```

The previous command displays any warnings and errors that block migration. If validation is successful, you can proceed with the following steps to Prepare and Commit the migration:

```
Move-AzureReservedIP -ReservedIPName $name -Prepare
Move-AzureReservedIP -ReservedIPName $name -Commit
```

A new resource group in Azure Resource Manager is created using the name of the migrated Reserved IP (in the example above, it would be resource group **myReservedIP-Migrated**).

## Limitations

- In order to upgrade a Basic Public IP, it cannot be associated with any Azure resource. Please review [this page](#) for more information on how to disassociate public IPs. Similarly, in order to migrate a Reserved IP, it cannot be associated with any Cloud Service. Please review [this page](#) for more information on how to disassociate reserved IPs.
- Public IPs upgraded from Basic to Standard SKU will continue to have no **availability zones** and therefore cannot be associated with an Azure resource that is either zone-redundant or zonal. Note this only applies to regions that offer availability zones.
- You cannot downgrade from Standard to Basic.

## Next Steps

- Learn more about [public IP addresses](#) in Azure, including the difference between the SKU types, as well as [public IP address settings](#).
- Learn how to [Upgrade Azure Public Load Balancers from Basic to Standard](#).
- Understand [Classic Azure Reserved IPs](#) and [Migration of Classic resources to Azure Resource Manager](#).

# Add, change, or remove IP addresses for an Azure network interface

2/18/2021 • 15 minutes to read • [Edit Online](#)

Learn how to add, change, and remove public and private IP addresses for a network interface. Private IP addresses assigned to a network interface enable a virtual machine to communicate with other resources in an Azure virtual network and connected networks. A private IP address also enables outbound communication to the Internet using an unpredictable IP address. A [Public IP address](#) assigned to a network interface enables inbound communication to a virtual machine from the Internet. The address also enables outbound communication from the virtual machine to the Internet using a predictable IP address. For details, see [Understanding outbound connections in Azure](#).

If you need to create, change, or delete a network interface, read the [Manage a network interface](#) article. If you need to add network interfaces to or remove network interfaces from a virtual machine, read the [Add or remove network interfaces](#) article.

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Complete the following tasks before completing steps in any section of this article:

- If you don't already have an Azure account, sign up for a [free trial account](#).
- If using the portal, open <https://portal.azure.com>, and log in with your Azure account.
- If using PowerShell commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running PowerShell from your computer. 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. This tutorial requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.
- If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This tutorial requires the Azure CLI version 2.0.31 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with, must be assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in [Network interface permissions](#).

## Add IP addresses

You can add as many [private](#) and [public IPv4](#) addresses as necessary to a network interface, within the limits listed in the [Azure limits](#) article. You can add a private IPv6 address to one [secondary IP configuration](#) (as long as

there are no existing secondary IP configurations) for an existing network interface. Each network interface may have at most one IPv6 private address. You can optionally add a public IPv6 address to an IPv6 network interface configuration. See [IPv6](#) for details about using IPv6 addresses.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface you want to add an IPv4 address for from the list.
3. Under **SETTINGS**, select **IP configurations**.
4. Under **IP configurations**, select **+ Add**.
5. Specify the following, then select **OK**:

SETTING	REQUIRED?	DETAILS
Name	Yes	Must be unique for the network interface
Type	Yes	Since you're adding an IP configuration to an existing network interface, and each network interface must have a <a href="#">primary</a> IP configuration, your only option is <a href="#">Secondary</a> .
Private IP address assignment method	Yes	<a href="#">Dynamic</a> : Azure assigns the next available address for the subnet address range the network interface is deployed in. <a href="#">Static</a> : You assign an unused address for the subnet address range the network interface is deployed in.
Public IP address	No	<a href="#">Disabled</a> : No public IP address resource is currently associated to the IP configuration. <a href="#">Enabled</a> : Select an existing IPv4 Public IP address, or create a new one. To learn how to create a public IP address, read the <a href="#">Public IP addresses</a> article.

6. Manually add secondary private IP addresses to the virtual machine operating system by completing the instructions in the [Assign multiple IP addresses to virtual machine operating systems](#) article. See [private IP addresses](#) for special considerations before manually adding IP addresses to a virtual machine operating system. Do not add any public IP addresses to the virtual machine operating system.

## Commands

TOOL	COMMAND
CLI	<a href="#">az network nic ip-config create</a>
PowerShell	<a href="#">Add-AzNetworkInterfaceIpConfig</a>

## Change IP address settings

You may need to change the assignment method of an IPv4 address, change the static IPv4 address, or change the public IP address assigned to a network interface. If you're changing the private IPv4 address of a secondary IP configuration associated with a secondary network interface in a virtual machine (learn more about [primary and secondary network interfaces](#)), place the virtual machine into the stopped (deallocated) state before completing the following steps:

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface that you want to view or change IP address settings for from the list.
3. Under **SETTINGS**, select **IP configurations**.
4. Select the IP configuration you want to modify from the list.
5. Change the settings, as desired, using the information about the settings in step 5 of [Add an IP configuration](#).
6. Select **Save**.

#### NOTE

If the primary network interface has multiple IP configurations and you change the private IP address of the primary IP configuration, you must manually reassign the primary and secondary IP addresses to the network interface within Windows (not required for Linux). To manually assign IP addresses to a network interface within an operating system, see [Assign multiple IP addresses to virtual machines](#). For special considerations before manually adding IP addresses to a virtual machine operating system, see [private](#) IP addresses. Do not add any public IP addresses to the virtual machine operating system.

## Commands

TOOL	COMMAND
CLI	<a href="#">az network nic ip-config update</a>
PowerShell	<a href="#">Set-AzNetworkInterfaceIpConfig</a>

## Remove IP addresses

You can remove [private](#) and [public](#) IP addresses from a network interface, but a network interface must always have at least one private IPv4 address assigned to it.

1. In the box that contains the text *Search resources* at the top of the Azure portal, type *network interfaces*. When **network interfaces** appear in the search results, select it.
2. Select the network interface that you want to remove IP addresses from the list.
3. Under **SETTINGS**, select **IP configurations**.
4. Right-select a [secondary](#) IP configuration (you cannot delete the [primary](#) configuration) that you want to delete, select **Delete**, then select **Yes**, to confirm the deletion. If the configuration had a public IP address resource associated to it, the resource is dissociated from the IP configuration, but the resource is not deleted.

## Commands

TOOL	COMMAND
CLI	<a href="#">az network nic ip-config delete</a>
PowerShell	<a href="#">Remove-AzNetworkInterfaceIpConfig</a>

# IP configurations

Private and (optionally) public IP addresses are assigned to one or more IP configurations assigned to a network interface. There are two types of IP configurations:

## Primary

Each network interface is assigned one primary IP configuration. A primary IP configuration:

- Has a private IPv4 address assigned to it. You cannot assign a private IPv6 address to a primary IP configuration.
- May also have a public IPv4 address assigned to it. You cannot assign a public IPv6 address to a primary (IPv4) IP configuration.

## Secondary

In addition to a primary IP configuration, a network interface may have zero or more secondary IP configurations assigned to it. A secondary IP configuration:

- Must have a private IPv4 or IPv6 address assigned to it. If the address is IPv6, the network interface can only have one secondary IP configuration. If the address is IPv4, the network interface may have multiple secondary IP configurations assigned to it. To learn more about how many private and public IPv4 addresses can be assigned to a network interface, see the [Azure limits](#) article.
- May also have a public IPv4 or IPv6 address assigned to it. Assigning multiple IPv4 addresses to a network interface is helpful in scenarios such as:
  - Hosting multiple websites or services with different IP addresses and TLS/SSL certificates on a single server.
  - A virtual machine serving as a network virtual appliance, such as a firewall or load balancer.
  - The ability to add any of the private IPv4 addresses for any of the network interfaces to an Azure Load Balancer back-end pool. In the past, only the primary IPv4 address for the primary network interface could be added to a back-end pool. To learn more about how to load balance multiple IPv4 configurations, see the [Load balancing multiple IP configurations](#) article.
  - The ability to load balance one IPv6 address assigned to a network interface. To learn more about how to load balance to a private IPv6 address, see the [Load balance IPv6 addresses](#) article.

# Address types

You can assign the following types of IP addresses to an [IP configuration](#):

## Private

Private IPv4 or IPv6 addresses enable a virtual machine to communicate with other resources in a virtual network or other connected networks.

By default, the Azure DHCP servers assign the private IPv4 address for the [primary IP configuration](#) of the Azure network interface to the network interface within the virtual machine operating system. Unless necessary, you should never manually set the IP address of a network interface within the virtual machine's operating system.

### WARNING

If the IPv4 address set as the primary IP address of a network interface within a virtual machine's operating system is ever different than the private IPv4 address assigned to the primary IP configuration of the primary network interface attached to a virtual machine within Azure, you lose connectivity to the virtual machine.

There are scenarios where it's necessary to manually set the IP address of a network interface within the virtual machine's operating system. For example, you must manually set the primary and secondary IP addresses of a Windows operating system when adding multiple IP addresses to an Azure virtual machine. For a Linux virtual

machine, you may only need to manually set the secondary IP addresses. See [Add IP addresses to a VM operating system](#) for details. If you ever need to change the address assigned to an IP configuration, it's recommended that you:

1. Ensure that the virtual machine is receiving an address from the Azure DHCP servers. Once you have, change the assignment of the IP address back to DHCP within the operating system and restart the virtual machine.
2. Stop (deallocate) the virtual machine.
3. Change the IP address for the IP configuration within Azure.
4. Start the virtual machine.
5. [Manually configure](#) the secondary IP addresses within the operating system (and also the primary IP address within Windows) to match what you set within Azure.

By following the previous steps, the private IP address assigned to the network interface within Azure, and within a virtual machine's operating system, remain the same. To keep track of which virtual machines within your subscription that you've manually set IP addresses within an operating system for, consider adding an Azure [tag](#) to the virtual machines. You might use "IP address assignment: Static", for example. This way, you can easily find the virtual machines within your subscription that you've manually set the IP address for within the operating system.

In addition to enabling a virtual machine to communicate with other resources within the same, or connected virtual networks, a private IP address also enables a virtual machine to communicate outbound to the Internet. Outbound connections are source network address translated by Azure to an unpredictable public IP address. To learn more about Azure outbound Internet connectivity, read the [Azure outbound Internet connectivity](#) article. You cannot communicate inbound to a virtual machine's private IP address from the Internet. If your outbound connections require a predictable public IP address, associate a public IP address resource to a network interface.

## Public

Public IP addresses assigned through a public IP address resource enable inbound connectivity to a virtual machine from the Internet. Outbound connections to the Internet use a predictable IP address. See [Understanding outbound connections in Azure](#) for details. You may assign a public IP address to an IP configuration, but aren't required to. If you don't assign a public IP address to a virtual machine by associating a public IP address resource, the virtual machine can still communicate outbound to the Internet. In this case, the private IP address is source network address translated by Azure to an unpredictable public IP address. To learn more about public IP address resources, see [Public IP address resource](#).

There are limits to the number of private and public IP addresses that you can assign to a network interface. For details, read the [Azure limits](#) article.

### NOTE

Azure translates a virtual machine's private IP address to a public IP address. As a result, a virtual machine's operating system is unaware of any public IP address assigned to it, so there is no need to ever manually assign a public IP address within the operating system.

## Assignment methods

Public and private IP addresses are assigned using one of the following assignment methods:

### Dynamic

Dynamic private IPv4 and IPv6 (optionally) addresses are assigned by default.

- **Public only:** Azure assigns the address from a range unique to each Azure region. To learn which ranges are assigned to each region, see [Microsoft Azure Datacenter IP Ranges](#). The address can change when a virtual

machine is stopped (deallocated), then started again. You cannot assign a public IPv6 address to an IP configuration using either assignment method.

- **Private only:** Azure reserves the first four addresses in each subnet address range, and doesn't assign the addresses. Azure assigns the next available address to a resource from the subnet address range. For example, if the subnet's address range is 10.0.0.0/16, and addresses 10.0.0.4-10.0.0.14 are already assigned (.0-.3 are reserved), Azure assigns 10.0.0.15 to the resource. Dynamic is the default allocation method. Once assigned, dynamic IP addresses are only released if a network interface is deleted, assigned to a different subnet within the same virtual network, or the allocation method is changed to static, and a different IP address is specified. By default, Azure assigns the previous dynamically assigned address as the static address when you change the allocation method from dynamic to static.

## Static

You can (optionally) assign a public or private static IPv4 or IPv6 address to an IP configuration. To learn more about how Azure assigns static public IPv4 addresses, see [Public IP addresses](#).

- **Public only:** Azure assigns the address from a range unique to each Azure region. You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds. The address doesn't change until the public IP address resource it's assigned to is deleted, or the assignment method is changed to dynamic. If the public IP address resource is associated to an IP configuration, it must be dissociated from the IP configuration before changing its assignment method.
- **Private only:** You select and assign an address from the subnet's address range. The address you assign can be any address within the subnet address range that is not one of the first four addresses in the subnet's address range and is not currently assigned to any other resource in the subnet. Static addresses are only released if a network interface is deleted. If you change the allocation method to static, Azure dynamically assigns the previously assigned dynamic IP address as the static address, even if the address isn't the next available address in the subnet's address range. The address also changes if the network interface is assigned to a different subnet within the same virtual network, but to assign the network interface to a different subnet, you must first change the allocation method from static to dynamic. Once you've assigned the network interface to a different subnet, you can change the allocation method back to static, and assign an IP address from the new subnet's address range.

## IP address versions

You can specify the following versions when assigning addresses:

### IPv4

Each network interface must have one [primary](#) IP configuration with an assigned [private IPv4](#) address. You can add one or more [secondary](#) IP configurations that each have an IPv4 private and (optionally) an IPv4 [public](#) IP address.

### IPv6

You can assign zero or one private [IPv6](#) address to one secondary IP configuration of a network interface. The network interface cannot have any existing secondary IP configurations. Each network interface may have at most one IPv6 private address. You can optionally add a public IPv6 address to an IPv6 network interface configuration.

**NOTE**

Though you can create a network interface with an IPv6 address using the portal, you can't add an existing network interface to a new, or existing virtual machine, using the portal. Use PowerShell or the Azure CLI to create a network interface with a private IPv6 address, then attach the network interface when creating a virtual machine. You cannot attach a network interface with a private IPv6 address assigned to it to an existing virtual machine. You cannot add a private IPv6 address to an IP configuration for any network interface attached to a virtual machine using any tools (portal, CLI, or PowerShell).

You can't assign a public IPv6 address to a primary or secondary IP configuration.

## SKUs

A public IP address is created with the basic or standard SKU. For more information about SKU differences, see [Manage public IP addresses](#).

**NOTE**

When you assign a standard SKU public IP address to a virtual machine's network interface, you must explicitly allow the intended traffic with a [network security group](#). Communication with the resource fails until you create and associate a network security group and explicitly allow the desired traffic.

## Next steps

To create a virtual machine with different IP configurations, read the following articles:

TASK	TOOL
Create a VM with multiple network interfaces	<a href="#">CLI</a> , <a href="#">PowerShell</a>
Create a single NIC VM with multiple IPv4 addresses	<a href="#">CLI</a> , <a href="#">PowerShell</a>
Create a single NIC VM with a private IPv6 address (behind an Azure Load Balancer)	<a href="#">CLI</a> , <a href="#">PowerShell</a> , <a href="#">Azure Resource Manager template</a>

# Configure routing preference for a public IP address using the Azure portal

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to configure [routing preference](#) via ISP network (**Internet** option) for a public IP address. After creating the public IP address, you can associate it with the following Azure resources for inbound and outbound traffic to the internet:

- Virtual machine
- Virtual machine scale set
- Azure Kubernetes Service (AKS)
- Internet-facing load balancer
- Application Gateway
- Azure Firewall

By default, the routing preference for public IP address is set to the Microsoft global network for all Azure services and can be associated with any Azure service.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

If you don't have an Azure subscription, create a [free account](#) now.

## Create a public IP address with a routing preference

1. Sign in to the [Azure portal](#).
2. Select **Create a resource**.
3. In the search box, type *Public IP address*.
4. In the search results, select **Public IP address**. Next, in the **Public IP address** page, select **Create**.
5. In the **Routing preference** options, select **Internet**.

## Create public IP address

IP Version \* ⓘ

IPv4  IPv6  Both

SKU \* ⓘ

Basic  Standard

### IPv4 IP Address Configuration

Name \*

MyPublicIP

IP address assignment

Dynamic  Static

Routing preference

[Microsoft network](#) [Internet](#)

Idle timeout (minutes) \* ⓘ

#### NOTE

Public IP addresses are created with an IPv4 or IPv6 address. However, routing preference only supports IPV4 currently.

You can associate the above created public IP address with a [Windows](#) or [Linux](#) virtual machine. Use the CLI section on the tutorial page: [Associate a public IP address to a virtual machine](#) to associate the public IP to your VM. You can also associate the public IP address created above with an [Azure Load Balancer](#), by assigning it to the load balancer **frontend** configuration. The public IP address serves as a load-balanced virtual IP address (VIP).

## Next steps

- Learn more about [public IP with routing preference](#).
- [Configure routing preference for a VM](#).
- [Configure routing preference for a public IP address using the PowerShell](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).

# Configure routing preference for a public IP address using Azure PowerShell

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article shows you how to configure routing preference via ISP network (**Internet** option) for a public IP address using Azure PowerShell. After creating the public IP address, you can associate it with the following Azure resources for inbound and outbound traffic to the internet:

- Virtual machine
- Virtual machine scale set
- Azure Kubernetes Service (AKS)
- Internet-facing load balancer
- Application Gateway
- Azure Firewall

By default, the routing preference for public IP address is set to the Microsoft global network for all Azure services and can be associated with any Azure service.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

If you don't have an Azure subscription, create a [free account](#) now.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 6.9.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a resource group

Create a resource group with [New-AzResourceGroup](#). This example creates a resource group named `myResourceGroup` in the `eastus` location:

```
$rg = New-AzResourceGroup -Name myResourceGroup -Location EastUS
```

## Create a Public IP with Internet routing preference

The following command creates a new public IP with a routing preference type as `Internet` in the `East US` Azure region:

```
$iptagtype="RoutingPreference"
$tagName = "Internet"
$ipTag = New-AzPublicIpTag -IpTagType $iptagtype -Tag $tagName
attach the tag
$publicIp = New-AzPublicIpAddress `
-Name "MyPublicIP" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-IpTag $ipTag `
-AllocationMethod Static `
-Sku Standard `
-IpAddressVersion IPv4
```

You can associate the above created public IP address with a [Windows](#) or [Linux](#) virtual machine. Use the CLI section on the tutorial page: [Associate a public IP address to a virtual machine](#) to associate the Public IP to your VM. You can also associate the public IP address created above with an [Azure Load Balancer](#), by assigning it to the load balancer **frontend** configuration. The public IP address serves as a load-balanced virtual IP address (VIP).

## Clean up resources

If no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

## Next steps

- Learn more about [routing preference in public IP addresses](#).

- Configure routing preference for a VM using the Azure PowerShell.

# Configure routing preference for a public IP address using Azure CLI

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to configure routing preference via ISP network (**Internet** option) for a public IP address using Azure CLI. After creating the public IP address, you can associate it with the following Azure resources for inbound and outbound traffic to the internet:

- Virtual machine
- Virtual machine scale set
- Azure Kubernetes Service (AKS)
- Internet-facing load balancer
- Application Gateway
- Azure Firewall

By default, the routing preference for public IP address is set to the Microsoft global network for all Azure services and can be associated with any Azure service.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.49 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

Create a resource group with the [az group create](#) command. The following example creates a resource group in

the East US Azure region:

```
az group create --name myResourceGroup --location eastus
```

## Create a public IP address

Create a Public IP Address with routing preference of **Internet** type using command [az network public-ip create](#), with the format as shown below.

The following command creates a new public IP with **Internet** routing preference in the East US Azure region.

```
az network public-ip create \
--name MyRoutingPrefIP \
--resource-group MyResourceGroup \
--location eastus \
--ip-tags 'RoutingPreference=Internet' \
--sku STANDARD \
--allocation-method static \
--version IPv4
```

### NOTE

Currently, routing preference only supports IPV4 public IP addresses.

You can associate the above created public IP address with a [Windows](#) or [Linux](#) virtual machine. Use the CLI section on the tutorial page: [Associate a public IP address to a virtual machine](#) to associate the Public IP to your VM. You can also associate the public IP address created above with with an [Azure Load Balancer](#), by assigning it to the load balancer **frontend** configuration. The public IP address serves as a load-balanced virtual IP address (VIP).

## Next steps

- Learn more about [routing preference in public IP addresses](#).
- [Configure routing preference for a VM using the Azure CLI](#).

# Configure routing preference for a VM using the Azure portal

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to configure routing preference for a virtual machine. Internet bound traffic from the VM will be routed via the ISP network when you choose **Internet** as your routing preference option . The default routing is via the Microsoft global network.

This article shows you how to create a virtual machine with a public IP that is set to route traffic via the public internet using the Azure portal.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Sign in to Azure

Sign in to the [Azure portal](#).

## Create a virtual machine

1. Select + **Create a resource** found on the upper, left corner of the Azure portal.
2. Select **Compute**, and then select **Windows Server 2016 VM**, or another operating system of your choosing.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

SETTING	VALUE
Name	myVM
User name	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the <a href="#">defined complexity requirements</a> .
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> and select <b>myResourceGroup</b> .
Location	Select <b>East US</b>

4. Select a size for the VM and then select **Select**.
5. Under **Networking** tab, click on **Create new** for **Public IP address**.

6. Enter *myPublicIpAddress*, select sku as **Standard**, and then select routing preference **Internet** and then hit **ok**, as shown in the following picture:

7. Select a port, or no ports under **Select public inbound ports**. Portal 3389 is selected, to enable remote access to the Windows Server virtual machine from the internet. Opening port 3389 from the internet is not recommended for production workloads.

8. Accept the remaining default settings and select **OK**.

9. On the **Summary** page, select **Create**. The virtual machine takes a few minutes to deploy.

10. Once the virtual machine is deployed, enter *myPublicIpAddress* in the search box at the top of the portal. When **myPublicIpAddress** appears in the search results, select it.

11. You can view the public IP address that is assigned, and that the address is assigned to the **myVM** virtual machine, as shown in the following picture:

12. Select **Networking**, then click on nic **mynic** and then select the public ip address to confirm that the routing preference is assigned as **Internet**.

Home > Resource groups > myResourceGroup > myVM | Networking > mynic > MyPublicIP

MyPublicIP  
Public IP address

Search (Ctrl+ /) <>

Associate Dissociate Move Delete Refresh

Resource group (change) : myResourceGroup

Location : East US

SKU : Standard

Subscription (change) :

IP address : 20.47.108.1

Subscription ID :

DNS name : -

Tags (change) : Click here to add tags

Associated to : mynic

Tags : See more

Routing preference : Internet

## Clean up resources

When no longer needed, delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME**: and select **Delete**.

## Next steps

- Learn more about [public IP with routing preference](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).

# Configure routing preference for a VM using Azure PowerShell

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article shows you how to configure routing preference for a virtual machine. Internet bound traffic from the VM will be routed via the ISP network when you choose **Internet** as your routing preference option . The default routing is via the Microsoft global network.

This article shows you how to create a virtual machine with a public IP that is set to route traffic via the ISP network using Azure PowerShell.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Create a resource group

1. If using the Cloud Shell, skip to step 2. Open a command session and sign into Azure with `Connect-AzAccount` .
2. Create a resource group with the [New-AzResourceGroup](#) command. The following example creates a resource group in the East US Azure region:

```
$rg = New-AzResourceGroup -Name MyResourceGroup -Location EastUS
```

## Create a public IP address

To access your virtual machines from the Internet, you need a public IP addresses. Create public IP addresses with [New-AzPublicIpAddress](#). The following example creates a IPv4 public IP address named *MyPublicIP* routing preference type *Internet* in the *MyResourceGroup* resource group in *East US* region:

```
$iptagtype="RoutingPreference"
$tagName = "Internet"
$ipTag = New-AzPublicIpTag -IpTagType $iptagtype -Tag $tagName
attach the tag
$publicIp = New-AzPublicIpAddress `
-Name "MyPublicIP" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-IpTag $ipTag `
-AllocationMethod Static `
-Sku Standard `
-IpAddressVersion IPv4
```

## Create network resources

Before you deploy a VM, you must create supporting network resources - network security group, virtual network, and virtual NIC.

## Create a network security group

Create a network security group with [New-AzNetworkSecurityGroup](#). The following example creates a NSG named *myNSG*

```
$nsg = New-AzNetworkSecurityGroup `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "myNSG"
```

## Create a virtual network

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVNET* with *mySubNet*.

### Create a subnet

```
$subnet = New-AzVirtualNetworkSubnetConfig `
-Name "mySubnet" `
-AddressPrefix "10.0.0.0/24"
```

```
Create a virtual network
$vnet = New-AzVirtualNetwork `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-Name "myVNET" `
-AddressPrefix "10.0.0.0/16" `
-Subnet $subnet
```

## Create a NIC

Create virtual NICs with [New-AzNetworkInterface](/powershell/module/az.network/new-aznetworkinterface). The following example creates a virtual NIC.

```
Create an IP Config
$ipconfig=New-AzNetworkInterfaceIpConfig `
-Name myIpConfig `
-Subnet $vnet.subnets[0] `
-PrivateIpAddressVersion IPv4 `
-PublicIpAddress $publicIp

Create a NIC
$nic = New-AzNetworkInterface `
-Name "mynic" `
-ResourceGroupName $rg.ResourceGroupName `
-Location $rg.Location `
-NetworkSecurityGroupId $nsg.Id `
-IpConfiguration $ipconfig
```

## Create a virtual machine

Set an administrator username and password for the VMs with [Get-Credential](#):

```
$cred = get-credential -Message "Routing Preference SAMPLE: Please enter the Administrator credential to
log into the VM."
```

Now you can create the VM with [New-AzVM](#). The following example creates two VMs and the required virtual

network components if they do not already exist.

```
$vmsize = "Standard_A2"
$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "WindowsServer"
$imageSKU = "2019-Datacenter"

$vmName= "myVM"
$vmconfig = New-AzVMConfig -VMName $vmName -VMSize $vmsize | Set-AzVMOperatingSystem -Windows -ComputerName
$vmName -Credential $cred -ProvisionVMAgent -EnableAutoUpdate | Set-AzVMSourceImage -PublisherName
$imagePublisher -Offer $imageOffer -Skus $imageSKU -Version "latest" | Set-AzVMOSDisk -Name "$vmName.vhd" -
CreateOption "FromImage" | Add-AzVMNetworkInterface -Id $nic.Id
$VM1 = New-AzVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $vmconfig
```

## Allow network traffic to the VM

Before you can connect to the public IP address from the internet, ensure that you have the necessary ports open in any network security group that you might have associated to the network interface, the subnet the network interface is in, or both. You can view the effective security rules for a network interface and its subnet using the [Portal](#), [CLI](#), or [PowerShell](#).

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name MyResourceGroup
```

## Next steps

- Learn more about [routing preference in public IP addresses](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about [public IP address settings](#).

# Configure routing preference for a VM using Azure CLI

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article shows you how to configure routing preference for a virtual machine. Internet bound traffic from the VM will be routed via the ISP network when you choose **Internet** as your routing preference option . The default routing is via the Microsoft global network.

This article shows you how to create a virtual machine with a public IP that is set to route traffic via the public internet using Azure CLI.

## IMPORTANT

Routing preference is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Create a resource group

1. If using the Cloud Shell, skip to step 2. Open a command session and sign into Azure with `az login`.
2. Create a resource group with the `az group create` command. The following example creates a resource group in the East US Azure region:

```
az group create --name myResourceGroup --location eastus
```

## Create a public IP address

To access your virtual machines from the Internet, you need to create a public IP address. Create a public IP address with `az network public-ip create`. The following example creates a public ip of routing preference type *Internet* in the *East US* region:

```
az network public-ip create \
--name MyRoutingPrefIP \
--resource-group MyResourceGroup \
--location eastus \
--ip-tags 'RoutingPreference=Internet' \
--sku STANDARD \
--allocation-method static \
--version IPv4
```

## Create network resources

Before you deploy a VM, you must create supporting network resources - network security group, virtual network, and virtual NIC.

### Create a network security group

Create a network security group for the rules that will govern inbound and outbound communication in your VNet with `az network nsg create`

```
az network nsg create \
--name myNSG \
--resource-group MyResourceGroup \
--location eastus
```

## Create a virtual network

Create a virtual network with [az network vnet create](#). The following example creates a virtual network named *myVNET* with subnets *mySubNet*.

```
Create a virtual network
az network vnet create \
--name myVNET \
--resource-group MyResourceGroup \
--location eastus

Create a subnet
az network vnet subnet create \
--name mySubNet \
--resource-group MyResourceGroup \
--vnet-name myVNET \
--address-prefixes "10.0.0.0/24" \
--network-security-group myNSG
```

## Create a NIC

Create a virtual NIC for the VM with [az network nic create](#). The following example creates a virtual NIC, which will be attached to the VM.

```
Create a NIC
az network nic create \
--name mynic \
--resource-group MyResourceGroup \
--network-security-group myNSG \
--vnet-name myVNET \
--subnet mySubNet \
--private-ip-address-version IPv4 \
--public-ip-address MyRoutingPrefIP
```

## Create a virtual machine

Create a VM with [az vm create](#). The following example creates a windows server 2019 VM and the required virtual network components if they do not already exist.

```
az vm create \
--name myVM \
--resource-group MyResourceGroup \
--nics mynic \
--size Standard_A2 \
--image MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest \
--admin-username myUserName
```

## Clean up resources

When no longer needed, you can use [az group delete](#) to remove the resource group and all of the resources it contains:

```
az group delete --name myResourceGroup --yes
```

## Next steps

- Learn more about [routing preference in public IP addresses](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about [public IP address settings](#).

# Tutorial: Filter network traffic with a network security group using the Azure portal

2/18/2021 • 7 minutes to read • [Edit Online](#)

You can filter network traffic inbound to and outbound from a virtual network subnet with a network security group. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet. In this tutorial, you learn how to:

- Create a network security group and security rules
- Create a virtual network and associate a network security group to a subnet
- Deploy virtual machines (VM) into a subnet
- Test traffic filters

If you prefer, you can complete this tutorial using the [Azure CLI](#) or [PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create a virtual network

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Select **Networking**, and then select **Virtual network**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	myVirtualNetwork
Address space	10.0.0.0/16
Subscription	Select your subscription.
Resource group	Select <b>Create new</b> and enter <i>myResourceGroup</i> .
Location	Select <b>East US</b> .
Subnet- Name	mySubnet
Subnet - Address range	10.0.0.0/24

## Create application security groups

An application security group enables you to group together servers with similar functions, such as web servers.

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. In the **Search the Marketplace** box, enter *Application security group*. When **Application security group** appears in the search results, select it, select **Application security group** again under **Everything**, and then select **Create**.
3. Enter, or select, the following information, and then select **Create**:

SETTING	VALUE
Name	myAsgWebServers
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> and then select <b>myResourceGroup</b> .
Location	East US

4. Complete step 3 again, specifying the following values:

SETTING	VALUE
Name	myAsgMgmtServers
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> and then select <b>myResourceGroup</b> .
Location	East US

## Create a network security group

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Select **Networking**, and then select **Network security group**.
3. Enter, or select, the following information, and then select **Create**:

SETTING	VALUE
Name	myNsg
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> and then select <b>myResourceGroup</b> .
Location	East US

## Associate network security group to subnet

1. In the *Search resources, services, and docs* box at the top of the portal, begin typing *myNsg*. When **myNsg** appears in the search results, select it.

2. Under SETTINGS, select Subnets and then select + Associate, as shown in the following picture:

The screenshot shows the 'myNsg - Subnets' page in the Azure portal. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, SETTINGS (Inbound security rules, Outbound security rules, Network interfaces), and Subnets. The 'Subnets' link is highlighted with a red box. On the right, there's a search bar labeled 'Search subnets' and a table with columns 'NAME' and 'ADDRESS RANGE'. A red box highlights the '+ Associate' button at the top right.

3. Under Associate subnet, select Virtual network and then select myVirtualNetwork. Select Subnet, select mySubnet, and then select OK.

## Create security rules

1. Under SETTINGS, select Inbound security rules and then select + Add, as shown in the following picture:

The screenshot shows the 'myNsg - Inbound security rules' page. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Inbound security rules. The 'Inbound security rules' link is highlighted with a red box. On the right, there's a table with columns PRIORITY, NAME, PORT, PROTOCOL, SOURCE, DESTINATION, and ACTION. A red box highlights the '+ Add' button at the top right.

2. Create a security rule that allows ports 80 and 443 to the myAsgWebServers application security group. Under Add inbound security rule, enter, or select the following values, accept the remaining defaults, and then select Add:

SETTING	VALUE
Destination	Select Application security group, and then select myAsgWebServers for Application security group.

SETTING	VALUE
Destination port ranges	Enter 80,443
Protocol	Select TCP
Name	Allow-Web-All

3. Complete step 2 again, using the following values:

SETTING	VALUE
Destination	Select Application security group, and then select <b>myAsgMgmtServers</b> for Application security group.
Destination port ranges	Enter 3389
Protocol	Select TCP
Priority	Enter 110
Name	Allow-RDP-All

In this tutorial, RDP (port 3389) is exposed to the internet for the VM that is assigned to the *myAsgMgmtServers* application security group. For production environments, instead of exposing port 3389 to the internet, it's recommended that you connect to Azure resources that you want to manage using a VPN or private network connection.

Once you've completed steps 1-3, review the rules you created. Your list should look like the list in the following picture:

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
100	Allow-Web-All	80,443	TCP	Any	myAsgWebServers	Allow
110	Allow-RDP-All	3389	TCP	Any	myAsgMgmtServ...	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalanc...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

## Create virtual machines

Create two VMs in the virtual network.

### Create the first VM

- On the Azure portal menu or from the Home page, select **Create a resource**.
- Select **Compute**, and then select **Windows Server 2016 Datacenter**.
- Enter, or select, the following information, and accept the defaults for the remaining settings:

SETTING	VALUE
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> and select <b>myResourceGroup</b> .
Name	myVmWeb
Location	Select <b>East US</b> .
User name	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the <a href="#">defined complexity requirements</a> .

4. Select a size for the VM and then select **Select**.

5. Under **Networking**, select the following values, and accept the remaining defaults:

SETTING	VALUE
Virtual network	Select <b>myVirtualNetwork</b> .
NIC network security group	Select <b>None</b> .

6. Select **Review + Create** at the bottom, left corner, select **Create** to start VM deployment.

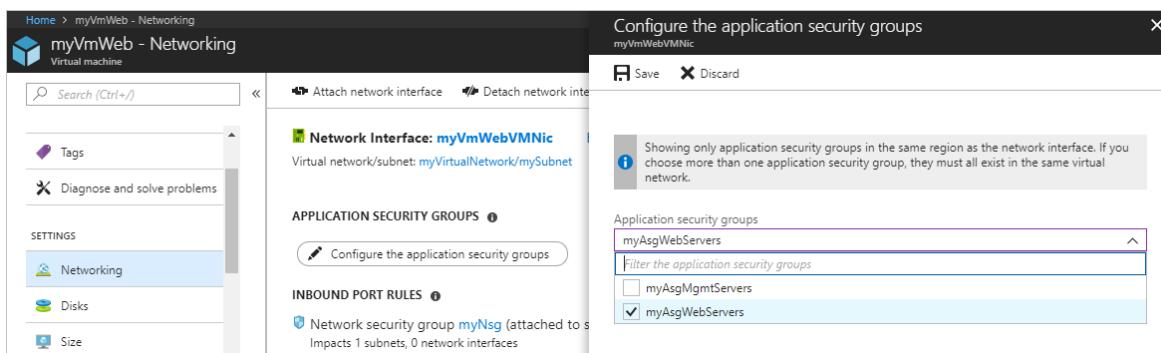
### Create the second VM

Complete steps 1-6 again, but in step 3, name the VM *myVmMgmt*. The VM takes a few minutes to deploy. Do not continue to the next step until the VM is deployed.

## Associate network interfaces to an ASG

When the portal created the VMs, it created a network interface for each VM, and attached the network interface to the VM. Add the network interface for each VM to one of the application security groups you created previously:

1. In the *Search resources, services, and docs* box at the top of the portal, begin typing *myVmWeb*. When the **myVmWeb** VM appears in the search results, select it.
2. Under **SETTINGS**, select **Networking**. Select **Configure the application security groups**, select **myAsgWebServers** for **Application security groups**, and then select **Save**, as shown in the following picture:



3. Complete steps 1 and 2 again, searching for the **myVmMgmt** VM and selecting the **myAsgMgmtServers** ASG.

## Test traffic filters

1. Connect to the *myVmMgmt* VM. Enter *myVmMgmt* in the search box at the top of the portal. When **myVmMgmt** appears in the search results, select it. Select the **Connect** button.
2. Select **Download RDP file**.
3. Open the downloaded rdp file and 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.
4. Select **OK**.
5. 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.

The connection succeeds, because port 3389 is allowed inbound from the internet to the *myAsgMgmtServers* application security group that the network interface attached to the *myVmMgmt* VM is in.

6. Connect to the *myVmWeb* VM from the *myVmMgmt* VM by entering the following command in a PowerShell session:

```
mstsc /v:myVmWeb
```

You are able to connect to the *myVmWeb* VM from the *myVmMgmt* VM because VMs in the same virtual network can communicate with each other over any port, by default. You can't however, create a remote desktop connection to the *myVmWeb* VM from the internet, because the security rule for the *myAsgWebServers* doesn't allow port 3389 inbound from the internet and inbound traffic from the Internet is denied to all resources, by default.

7. To install Microsoft IIS on the *myVmWeb* VM, enter the following command from a PowerShell session on the *myVmWeb* VM:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

8. After the IIS installation is complete, disconnect from the *myVmWeb* VM, which leaves you in the *myVmMgmt* VM remote desktop connection.
9. Disconnect from the *myVmMgmt* VM.
10. In the *Search resources, services, and docs* box at the top of the Azure portal, begin typing *myVmWeb* from your computer. When **myVmWeb** appears in the search results, select it. Note the **Public IP address** for your VM. The address shown in the following picture is 137.135.84.74, but your address is different:

11. To confirm that you can access the *myVmWeb* web server from the internet, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`. You see the IIS welcome screen, because port 80 is allowed inbound from the internet to the *myAsgWebServers* application security group that the network interface attached to the *myVmWeb* VM is in.

## Clean up resources

When no longer needed, delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

## Next steps

In this tutorial, you created a network security group and associated it to a virtual network subnet. To learn more about network security groups, see [Network security group overview](#) and [Manage a network security group](#).

Azure routes traffic between subnets by default. You may instead, choose to route traffic between subnets through a VM, serving as a firewall, for example. To learn how to create a route table, advance to the next tutorial.

[Create a route table](#)

# Filter network traffic with a network security group using PowerShell

2/18/2021 • 8 minutes to read • [Edit Online](#)

You can filter network traffic inbound to and outbound from a virtual network subnet with a network security group. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet. In this article, you learn how to:

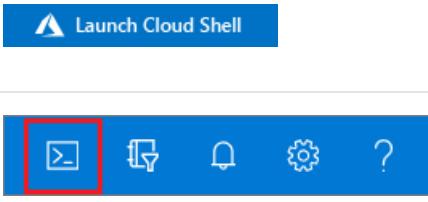
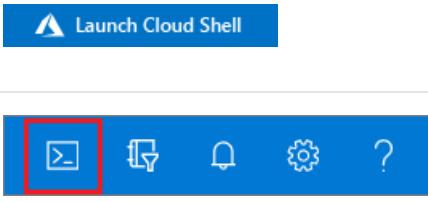
- Create a network security group and security rules
- Create a virtual network and associate a network security group to a subnet
- Deploy virtual machines (VM) into a subnet
- Test traffic filters

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the <a href="#">Azure portal</a> .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

# Create a network security group

A network security group contains security rules. Security rules specify a source and destination. Sources and destinations can be application security groups.

## Create application security groups

First create a resource group for all the resources created in this article with [New-AzResourceGroup](#). The following example creates a resource group in the *eastus* location:

```
New-AzResourceGroup -ResourceGroupName myResourceGroup -Location EastUS
```

Create an application security group with [New-AzApplicationSecurityGroup](#). An application security group enables you to group servers with similar port filtering requirements. The following example creates two application security groups.

```
$webAsg = New-AzApplicationSecurityGroup `
-ResourceGroupName myResourceGroup `
-Name myAsgWebServers `
-Location eastus

$mgmtAsg = New-AzApplicationSecurityGroup `
-ResourceGroupName myResourceGroup `
-Name myAsgMgmtServers `
-Location eastus
```

## Create security rules

Create a security rule with [New-AzNetworkSecurityRuleConfig](#). The following example creates a rule that allows traffic inbound from the internet to the *myWebServers* application security group over ports 80 and 443:

```
$webRule = New-AzNetworkSecurityRuleConfig `
-Name "Allow-Web-All" `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 100 `
-SourceAddressPrefix Internet `
-SourcePortRange * `
-DestinationApplicationSecurityGroupId $webAsg.id `
-DestinationPortRange 80,443
```

The following example creates a rule that allows traffic inbound from the internet to the *\*myMgmtServers\** application security group over port 3389:

```
$mgmtRule = New-AzNetworkSecurityRuleConfig `
-Name "Allow-RDP-All" `
-Access Allow `
-Protocol Tcp `
-Direction Inbound `
-Priority 110 `
-SourceAddressPrefix Internet `
-SourcePortRange * `
-DestinationApplicationSecurityGroupId $mgmtAsg.id `
-DestinationPortRange 3389
```

In this article, RDP (port 3389) is exposed to the internet for the *myAsgMgmtServers* VM. For production environments, instead of exposing port 3389 to the internet, it's recommended that you connect to Azure resources that you want to manage using a [VPN](#) or [private](#) network connection.

## Create a network security group

Create a network security group with [New-AzNetworkSecurityGroup](#). The following example creates a network security group named *myNsg*:

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName myResourceGroup `
 -Location eastus `
 -Name myNsg `
 -SecurityRules $webRule,$mgmtRule
```

## Create a virtual network

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual named *myVirtualNetwork*:

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

Create a subnet configuration with [New-AzVirtualNetworkSubnetConfig](#), and then write the subnet configuration to the virtual network with [Set-AzVirtualNetwork](#). The following example adds a subnet named *mySubnet* to the virtual network and associates the *myNsg* network security group to it:

```
Add-AzVirtualNetworkSubnetConfig `
 -Name mySubnet `
 -VirtualNetwork $virtualNetwork `
 -AddressPrefix "10.0.2.0/24" `
 -NetworkSecurityGroup $nsg
$virtualNetwork | Set-AzVirtualNetwork
```

## Create virtual machines

Before creating the VMs, retrieve the virtual network object with the subnet with [Get-AzVirtualNetwork](#):

```
$virtualNetwork = Get-AzVirtualNetwork `
 -Name myVirtualNetwork `
 -Resourcegroupname myResourceGroup
```

Create a public IP address for each VM with [New-AzPublicIpAddress](#):

```
$publicIpWeb = New-AzPublicIpAddress `
 -AllocationMethod Dynamic `
 -ResourceGroupName myResourceGroup `
 -Location eastus `
 -Name myVmWeb

$publicIpMgmt = New-AzPublicIpAddress `
 -AllocationMethod Dynamic `
 -ResourceGroupName myResourceGroup `
 -Location eastus `
 -Name myVmMgmt
```

Create two network interfaces with [New-AzNetworkInterface](#), and assign a public IP address to the network interface. The following example creates a network interface, associates the *myVmWeb* public IP address to it,

and makes it a member of the *myAsgWebServers* application security group:

```
$webNic = New-AzNetworkInterface `
 -Location eastus `
 -Name myVmWeb `
 -ResourceGroupName myResourceGroup `
 -SubnetId $virtualNetwork.Subnets[0].Id `
 -ApplicationSecurityGroupId $webAsg.Id `
 -PublicIpAddressId $publicIpWeb.Id
```

The following example creates a network interface, associates the *myVmMgmt* public IP address to it, and makes it a member of the *myAsgMgmtServers* application security group:

```
$mgmtNic = New-AzNetworkInterface `
 -Location eastus `
 -Name myVmMgmt `
 -ResourceGroupName myResourceGroup `
 -SubnetId $virtualNetwork.Subnets[0].Id `
 -ApplicationSecurityGroupId $mgmtAsg.Id `
 -PublicIpAddressId $publicIpMgmt.Id
```

Create two VMs in the virtual network so you can validate traffic filtering in a later step.

Create a VM configuration with [New-AzVMConfig](#), then create the VM with [New-AzVM](#). The following example creates a VM that will serve as a web server. The `-AsJob` option creates the VM in the background, so you can continue to the next step:

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

$webVmConfig = New-AzVMConfig `
 -VMName myVmWeb `
 -VMSize Standard_DS1_V2 | `
Set-AzVMOperatingSystem -Windows `
 -ComputerName myVmWeb `
 -Credential $cred | `
Set-AzVMSourceImage `
 -PublisherName MicrosoftWindowsServer `
 -Offer WindowsServer `
 -Skus 2016-Datacenter `
 -Version latest | `
Add-AzVMNetworkInterface `
 -Id $webNic.Id
New-AzVM `
 -ResourceGroupName myResourceGroup `
 -Location eastus `
 -VM $webVmConfig `
 -AsJob
```

Create a VM to serve as a management server:

```

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

Create the web server virtual machine configuration and virtual machine.
$mgmtVmConfig = New-AzVMConfig `

 -VMName myVmMgmt `

 -VMSize Standard_DS1_V2 | `

Set-AzVMOperatingSystem -Windows `

 -ComputerName myVmMgmt `

 -Credential $cred | `

Set-AzVMSourceImage `

 -PublisherName MicrosoftWindowsServer `

 -Offer WindowsServer `

 -Skus 2016-Datacenter `

 -Version latest | `

Add-AzVMNetworkInterface `

 -Id $mgmtNic.Id

New-AzVM `

 -ResourceGroupName myResourceGroup `

 -Location eastus `

 -VM $mgmtVmConfig

```

The virtual machine takes a few minutes to create. Don't continue with the next step until Azure finishes creating the VM.

## Test traffic filters

Use [Get-AzPublicIpAddress](#) to return the public IP address of a VM. The following example returns the public IP address of the *myVmMgmt* VM:

```

Get-AzPublicIpAddress `

 -Name myVmMgmt `

 -ResourceGroupName myResourceGroup `

 | Select IpAddress

```

Use the following command to create a remote desktop session with the *myVmMgmt* VM from your local computer. Replace `<publicIpAddress>` with the IP address returned from the previous command.

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

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), then select **OK**. You may receive a certificate warning during the sign-in process. Select **Yes** to proceed with the connection.

The connection succeeds, because port 3389 is allowed inbound from the internet to the *myAsgMgmtServers* application security group that the network interface attached to the *myVmMgmt* VM is in.

Use the following command to create a remote desktop connection to the *myVmWeb* VM, from the *myVmMgmt* VM, with the following command, from PowerShell:

```
mstsc /v:myvmWeb
```

The connection succeeds because a default security rule within each network security group allows traffic over all ports between all IP addresses within a virtual network. You can't create a remote desktop connection to the *myVmWeb* VM from the internet because the security rule for the *myAsgWebServers* doesn't allow port 3389

inbound from the internet.

Use the following command to install Microsoft IIS on the *myVmWeb* VM from PowerShell:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

After the IIS installation is complete, disconnect from the *myVmWeb* VM, which leaves you in the *myVmMgmt* VM remote desktop connection. To view the IIS welcome screen, open an internet browser and browse to <http://myVmWeb>.

Disconnect from the *myVmMgmt* VM.

On your computer, enter the following command from PowerShell to retrieve the public IP address of the *myVmWeb* server:

```
Get-AzPublicIpAddress
-Name myVmWeb
-ResourceGroupName myResourceGroup
| Select IpAddress
```

To confirm that you can access the *myVmWeb* web server from outside of Azure, open an internet browser on your computer and browse to <http://<public-ip-address-from-previous-step>>. The connection succeeds, because port 80 is allowed inbound from the internet to the *myAsgWebServers* application security group that the network interface attached to the *myVmWeb* VM is in.

## Clean up resources

When no longer needed, you can use [Remove-AzResourceGroup](#) to remove the resource group and all of the resources it contains:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

## Next steps

In this article, you created a network security group and associated it to a virtual network subnet. To learn more about network security groups, see [Network security group overview](#) and [Manage a network security group](#).

Azure routes traffic between subnets by default. You may instead, choose to route traffic between subnets through a VM, serving as a firewall, for example. To learn how, see [Create a route table](#).

# Filter network traffic with a network security group using the Azure CLI

2/18/2021 • 6 minutes to read • [Edit Online](#)

You can filter network traffic inbound to and outbound from a virtual network subnet with a network security group. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet. In this article, you learn how to:

- Create a network security group and security rules
- Create a virtual network and associate a network security group to a subnet
- Deploy virtual machines (VM) into a subnet
- Test traffic filters

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a network security group

A network security group contains security rules. Security rules specify a source and destination. Sources and destinations can be application security groups.

### Create application security groups

First create a resource group for all the resources created in this article with `az group create`. The following example creates a resource group in the *eastus* location:

```
az group create \
--name myResourceGroup \
--location eastus
```

Create an application security group with `az network asg create`. An application security group enables you to group servers with similar port filtering requirements. The following example creates two application security

groups.

```
az network asg create \
--resource-group myResourceGroup \
--name myAsgWebServers \
--location eastus

az network asg create \
--resource-group myResourceGroup \
--name myAsgMgmtServers \
--location eastus
```

## Create a network security group

Create a network security group with [az network nsg create](#). The following example creates a network security group named *myNsg*:

```
Create a network security group
az network nsg create \
--resource-group myResourceGroup \
--name myNsg
```

## Create security rules

Create a security rule with [az network nsg rule create](#). The following example creates a rule that allows traffic inbound from the internet to the *myWebServers* application security group over ports 80 and 443:

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsg \
--name Allow-Web-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 100 \
--source-address-prefix Internet \
--source-port-range "*" \
--destination-asgs "myAsgWebServers" \
--destination-port-range 80 443
```

The following example creates a rule that allows traffic inbound from the Internet to the *myMgmtServers* application security group over port 22:

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsg \
--name Allow-SSH-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 110 \
--source-address-prefix Internet \
--source-port-range "*" \
--destination-asgs "myAsgMgmtServers" \
--destination-port-range 22
```

In this article, SSH (port 22) is exposed to the internet for the *myAsgMgmtServers* VM. For production environments, instead of exposing port 22 to the internet, it's recommended that you connect to Azure resources that you want to manage using a [VPN](#) or [private](#) network connection.

## Create a virtual network

Create a virtual network with [az network vnet create](#). The following example creates a virtual named *myVirtualNetwork*.

```
az network vnet create \
--name myVirtualNetwork \
--resource-group myResourceGroup \
--address-prefixes 10.0.0.0/16
```

Add a subnet to a virtual network with [az network vnet subnet create](#). The following example adds a subnet named *mySubnet* to the virtual network and associates the *myNsg* network security group to it:

```
az network vnet subnet create \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name mySubnet \
--address-prefix 10.0.0.0/24 \
--network-security-group myNsg
```

## Create virtual machines

Create two VMs in the virtual network so you can validate traffic filtering in a later step.

Create a VM with [az vm create](#). The following example creates a VM that will serve as a web server. The `--asgs myAsgWebServers` option causes Azure to make the network interface it creates for the VM a member of the *myAsgWebServers* application security group.

The `--nsg ""` option is specified to prevent Azure from creating a default network security group for the network interface Azure creates when it creates the VM. To streamline this article, a password is used. Keys are typically used in production deployments. If you use keys, you must also configure SSH agent forwarding for the remaining steps. For more information, see the documentation for your SSH client. Replace `<replace-with-your-password>` in the following command with a password of your choosing.

```
adminPassword=<replace-with-your-password>

az vm create \
--resource-group myResourceGroup \
--name myVmWeb \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet mySubnet \
--nsg "" \
--asgs myAsgWebServers \
--admin-username azureuser \
--admin-password $adminPassword
```

The VM takes a few minutes to create. After the VM is created, output similar to the following example is returned:

```
{
 "fqdns": "",
 "id": "/subscriptions/00000000-0000-0000-0000-
0000000000/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVmWeb",
 "location": "eastus",
 "macAddress": "00-0D-3A-23-9A-49",
 "powerState": "VM running",
 "privateIpAddress": "10.0.0.4",
 "publicIpAddress": "13.90.242.231",
 "resourceGroup": "myResourceGroup"
}
```

Take note of the **publicIpAddress**. This address is used to access the VM from the internet in a later step. Create a VM to serve as a management server:

```
az vm create \
 --resource-group myResourceGroup \
 --name myVmMgmt \
 --image UbuntuLTS \
 --vnet-name myVirtualNetwork \
 --subnet mySubnet \
 --nsg "" \
 --asgs myAsgMgmtServers \
 --admin-username azureuser \
 --admin-password $adminPassword
```

The VM takes a few minutes to create. After the VM is created, note the **publicIpAddress** in the returned output. This address is used to access the VM in the next step. Don't continue with the next step until Azure finishes creating the VM.

## Test traffic filters

Use the command that follows to create an SSH session with the *myVmMgmt* VM. Replace *<publicIpAddress>* with the public IP address of your VM. In the example above, the IP address is *13.90.242.231*.

```
ssh azureuser@<publicIpAddress>
```

When prompted for a password, enter the password you entered in [Create VMs](#).

The connection succeeds, because port 22 is allowed inbound from the Internet to the *myAsgMgmtServers* application security group that the network interface attached to the *myVmMgmt* VM is in.

Use the following command to SSH to the *myVmWeb* VM from the *myVmMgmt* VM:

```
ssh azureuser@myVmWeb
```

The connection succeeds because a default security rule within each network security group allows traffic over all ports between all IP addresses within a virtual network. You can't SSH to the *myVmWeb* VM from the Internet because the security rule for the *myAsgWebServers* doesn't allow port 22 inbound from the Internet.

Use the following commands to install the nginx web server on the *myVmWeb* VM:

```
Update package source
sudo apt-get -y update

Install NGINX
sudo apt-get -y install nginx
```

The *myVmWeb* VM is allowed outbound to the Internet to retrieve nginx because a default security rule allows all outbound traffic to the Internet. Exit the *myVmWeb* SSH session, which leaves you at the

`username@myVmMgmt:~$` prompt of the *myVmMgmt* VM. To retrieve the nginx welcome screen from the *myVmWeb* VM, enter the following command:

```
curl myVmWeb
```

Logout of the *myVmMgmt* VM. To confirm that you can access the *myVmWeb* web server from outside of Azure, enter `curl <publicIpAddress>` from your own computer. The connection succeeds, because port 80 is allowed inbound from the Internet to the *myAsgWebServers* application security group that the network interface attached to the *myVmWeb* VM is in.

## Clean up resources

When no longer needed, use [az group delete](#) to remove the resource group and all of the resources it contains.

```
az group delete --name myResourceGroup --yes
```

## Next steps

In this article, you created a network security group and associated it to a virtual network subnet. To learn more about network security groups, see [Network security group overview](#) and [Manage a network security group](#).

Azure routes traffic between subnets by default. You may instead, choose to route traffic between subnets through a VM, serving as a firewall, for example. To learn how, see [Create a route table](#).

# Quickstart: Create a Private Endpoint using the Azure portal

11/2/2020 • 5 minutes to read • [Edit Online](#)

Get started with Azure Private Link by using a Private Endpoint to connect securely to an Azure web app.

In this quickstart, you'll create a private endpoint for an Azure web app and deploy a virtual machine to test the private connection.

Private endpoints can be created for different kinds of Azure services, such as Azure SQL and Azure Storage.

## Prerequisites

- An Azure account with an active subscription. [Create an account for free](#).
- An Azure Web App with a **PremiumV2-tier** or higher app service plan deployed in your Azure subscription.
  - For more information and an example, see [Quickstart: Create an ASP.NET Core web app in Azure](#).
  - For a detailed tutorial on creating a web app and an endpoint, see [Tutorial: Connect to a web app using an Azure Private Endpoint](#).

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create a virtual network and bastion host

In this section, you'll create a virtual network, subnet, and bastion host.

The bastion host will be used to connect securely to the virtual machine for testing the private endpoint.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network** or search for **Virtual network** in the search box.
2. In **Create virtual network**, enter or select this information in the **Basics** tab:

SETTING	VALUE
<b>Project Details</b>	
Subscription	Select your Azure subscription
Resource Group	Select <b>CreatePrivateEndpointQS-rg</b>
<b>Instance details</b>	
Name	Enter <b>myVNet</b>
Region	Select <b>West Europe</b> .

3. Select the **IP Addresses** tab or select the **Next: IP Addresses** button at the bottom of the page.
4. In the **IP Addresses** tab, enter this information:

SETTING	VALUE
IPv4 address space	Enter 10.1.0.0/16

5. Under **Subnet name**, select the word **default**.

6. In **Edit subnet**, enter this information:

SETTING	VALUE
Subnet name	Enter mySubnet
Subnet address range	Enter 10.1.0.0/24

7. Select **Save**.

8. Select the **Security** tab.

9. Under **BastionHost**, select **Enable**. Enter this information:

SETTING	VALUE
Bastion name	Enter myBastionHost
AzureBastionSubnet address space	Enter 10.1.1.0/24
Public IP Address	Select <b>Create new</b> . For <b>Name</b> , enter <b>myBastionIP</b> . Select <b>OK</b> .

10. Select the **Review + create** tab or select the **Review + create** button.

11. Select **Create**.

## Create a virtual machine

In this section, you'll create a virtual machine that will be used to test the private endpoint.

1. On the upper-left side of the portal, select **Create a resource > Compute > Virtual machine** or search for **Virtual machine** in the search box.

2. In **Create a virtual machine**, type or select the values in the **Basics** tab:

SETTING	VALUE
<b>Project Details</b>	
Subscription	Select your Azure subscription
Resource Group	Select <b>CreatePrivateEndpointQS-rg</b>
<b>Instance details</b>	
Virtual machine name	Enter <b>myVM</b>

SETTING	VALUE
Region	Select <b>West Europe</b> .
Availability Options	Select <b>No infrastructure redundancy required</b>
Image	Select <b>Windows Server 2019 Datacenter - Gen1</b>
Azure Spot instance	Select <b>No</b>
Size	Choose VM size or take default setting
<b>Administrator account</b>	
Username	Enter a username
Password	Enter a password
Confirm password	Reenter password

3. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

4. In the Networking tab, select or enter:

SETTING	VALUE
<b>Network interface</b>	
Virtual network	<b>myVNet</b>
Subnet	<b>mySubnet</b>
Public IP	Select <b>None</b> .
NIC network security group	<b>Basic</b>
Public inbound ports	Select <b>None</b> .

5. Select **Review + create**.

6. Review the settings, and then select **Create**.

## Create a Private Endpoint

In this section, you'll create a Private Endpoint for the web app you created in the prerequisites section.

1. On the upper-left side of the screen in the portal, select **Create a resource > Networking > Private Link**, or in the search box enter **Private Link**.
2. Select **Create**.
3. In **Private Link Center**, select **Private endpoints** in the left-hand menu.
4. In **Private endpoints**, select **+ Add**.
5. In the **Basics** tab of **Create a private endpoint**, enter, or select this information:

SETTING	VALUE
<b>Project details</b>	
Subscription	Select your subscription.
Resource group	Select <b>CreatePrivateEndpointQS-rg</b> . You created this resource group in the previous section.
<b>Instance details</b>	
Name	Enter <b>myPrivateEndpoint</b> .
Region	Select <b>West Europe</b> .

6. Select the **Resource** tab or the **Next: Resource** button at the bottom of the page.

7. In **Resource**, enter or select this information:

SETTING	VALUE
Connection method	Select <b>Connect to an Azure resource in my directory</b> .
Subscription	Select your subscription.
Resource type	Select <b>Microsoft.Web/sites</b> .
Resource	Select <your-web-app-name>. Select the name of the web app you created in the prerequisites.
Target sub-resource	Select <b>sites</b> .

8. Select the **Configuration** tab or the **Next: Configuration** button at the bottom of the screen.

9. In **Configuration**, enter or select this information:

SETTING	VALUE
<b>Networking</b>	
Virtual network	Select <b>myVNet</b> .
Subnet	Select <b>mySubnet</b> .
<b>Private DNS integration</b>	
Integrate with private DNS zone	Leave the default of <b>Yes</b> .
Subscription	Select your subscription.
Private DNS zones	Leave the default of ( <b>New</b> ) <b>privatelink.azurewebsites.net</b> .

10. Select **Review + create**.

11. Select **Create**.

## Test connectivity to private endpoint

In this section, you'll use the virtual machine you created in the previous step to connect to the web app across the private endpoint.

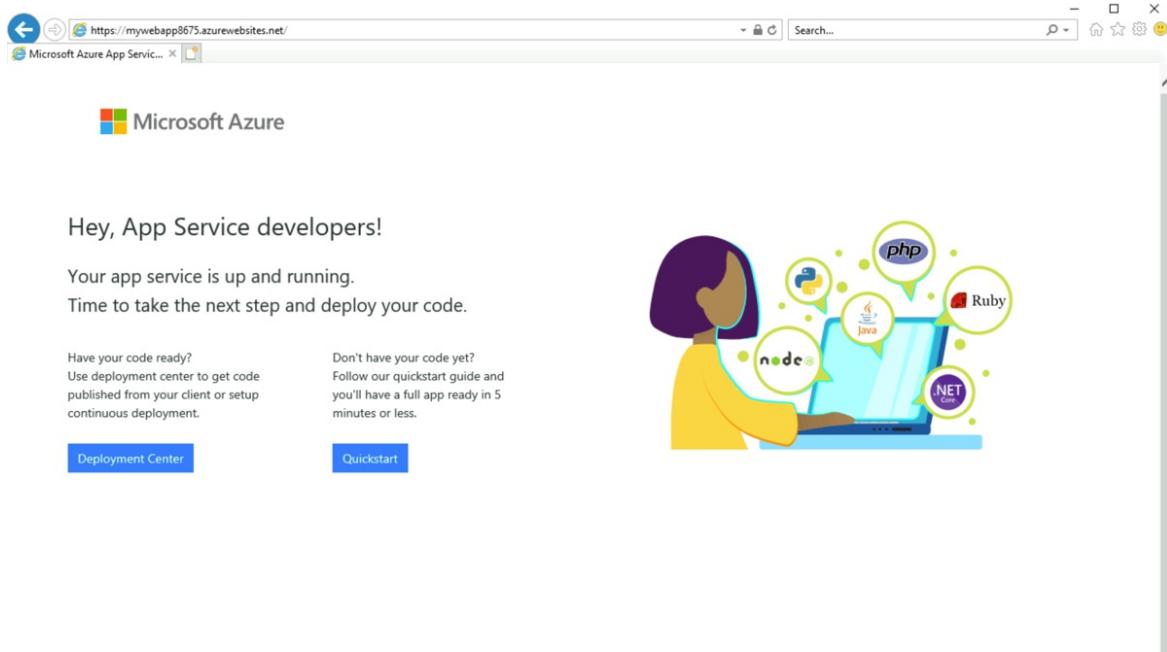
1. Select **Resource groups** in the left-hand navigation pane.
2. Select **CreatePrivateEndpointQS-rg**.
3. Select **myVM**.
4. On the overview page for **myVM**, select **Connect** then **Bastion**.
5. Select the blue **Use Bastion** button.
6. Enter the username and password that you entered during the virtual machine creation.
7. Open Windows PowerShell on the server after you connect.
8. Enter `nslookup <your-webapp-name>.azurewebsites.net`. Replace **<your-webapp-name>** with the name of the web app you created in the previous steps. You'll receive a message similar to what is displayed below:

```
Server: UnKnown
Address: 168.63.129.16

Non-authoritative answer:
Name: mywebapp8675.privatelink.azurewebsites.net
Address: 10.1.0.5
Aliases: mywebapp8675.azurewebsites.net
```

A private IP address of 10.1.0.5 is returned for the web app name. This address is in the subnet of the virtual network you created previously.

9. In the bastion connection to **myVM**, open Internet Explorer.
10. Enter the url of your web app, <https://<your-webapp-name>.azurewebsites.net>.
11. You'll receive the default web app page if your application hasn't been deployed:



12. Close the connection to myVM.

## Clean up resources

If you're not going to continue to use this application, delete the virtual network, virtual machine, and web app with the following steps:

1. From the left-hand menu, select **Resource groups**.
2. Select **CreatePrivateEndpointQS-rg**.
3. Select **Delete resource group**.
4. Enter **CreatePrivateEndpointQS-rg** in **TYPE THE RESOURCE GROUP NAME**.
5. Select **Delete**.

## Next steps

In this quickstart, you created a:

- Virtual network and bastion host.
- Virtual machine.
- Private endpoint for an Azure Web App.

You used the virtual machine to test connectivity securely to the web app across the private endpoint.

For more information on the services that support a private endpoint, see:

[Private Link availability](#)

# Use PowerShell to create a Private Endpoint

2/18/2021 • 5 minutes to read • [Edit Online](#)

Get started with Azure Private Link by using a Private Endpoint to connect securely to an Azure web app.

In this quickstart, you'll create a private endpoint for an Azure web app and deploy a virtual machine to test the private connection.

Private endpoints can be created for different kinds of Azure services, such as Azure SQL and Azure Storage.

## Prerequisites

- An Azure account with an active subscription. [Create an account for free](#).
- An Azure Web App with a **PremiumV2**-tier or higher app service plan deployed in your Azure subscription.
  - For more information and an example, see [Quickstart: Create an ASP.NET Core web app in Azure](#).
  - For a detailed tutorial on creating a web app and an endpoint, see [Tutorial: Connect to a web app using an Azure Private Endpoint](#).

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 5.4.1 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you're running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with [New-AzResourceGroup](#):

```
New-AzResourceGroup -Name 'CreatePrivateEndpointQS-rg' -Location 'eastus'
```

## Create a virtual network and bastion host

In this section, you'll create a virtual network, subnet, and bastion host.

The bastion host will be used to connect securely to the virtual machine for testing the private endpoint.

Create a virtual network and bastion host with:

- [New-AzVirtualNetwork](#)
- [New-AzPublicIpAddress](#)
- [New-AzBastion](#)

```

Create backend subnet config.
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name myBackendSubnet -AddressPrefix 10.0.0.0/24

Create Azure Bastion subnet.
$bastsubnetConfig = New-AzVirtualNetworkSubnetConfig -Name AzureBastionSubnet -AddressPrefix 10.0.1.0/24

Create the virtual network.
$parameters1 = @{
 Name = 'MyVNet'
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Location = 'eastus'
 AddressPrefix = '10.0.0.0/16'
 Subnet = $subnetConfig, $bastsubnetConfig
}
$vnet = New-AzVirtualNetwork @parameters1

Create public IP address for bastion host.
$parameters2 = @{
 Name = 'myBastionIP'
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Location = 'eastus'
 Sku = 'Standard'
 AllocationMethod = 'Static'
}
$publicip = New-AzPublicIpAddress @parameters2

Create bastion host
$parameters3 = @{
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Name = 'myBastion'
 PublicIpAddress = $publicip
 VirtualNetwork = $vnet
}
New-AzBastion @parameters3

```

It can take a few minutes for the Azure Bastion host to deploy.

## Create test virtual machine

In this section, you'll create a virtual machine that will be used to test the private endpoint.

Create the virtual machine with:

- [Get-Credential](#)
- [New-AzNetworkInterface](#)
- [New-AzVM](#)
- [New-AzVMConfig](#)
- [Set-AzVMOperatingSystem](#)
- [Set-AzVMSourceImage](#)
- [Add-AzVMNetworkInterface](#)

```

Set credentials for server admin and password.
$cred = Get-Credential

Command to get virtual network configuration.
$vnet = Get-AzVirtualNetwork -Name myVNet -ResourceGroupName CreatePrivateEndpointQS-rg

Command to create network interface for VM
$parameters1 = @{
 Name = 'myNicVM'
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Location = 'eastus'
 Subnet = $vnet.Subnets[0]
}
$nicVM = New-AzNetworkInterface @parameters1

Create a virtual machine configuration.##
$parameters2 = @{
 VMName = 'myVM'
 VMSize = 'Standard_DS1_v2'
}
$parameters3 = @{
 ComputerName = 'myVM'
 Credential = $cred
}
$parameters4 = @{
 PublisherName = 'MicrosoftWindowsServer'
 Offer = 'WindowsServer'
 Skus = '2019-Datacenter'
 Version = 'latest'
}
$vmConfig =
New-AzVMConfig @parameters2 | Set-AzVMOperatingSystem -Windows @parameters3 | Set-AzVMSourceImage
@parameters4 | Add-AzVMNetworkInterface -Id $nicVM.Id

Create the virtual machine
New-AzVM -ResourceGroupName 'CreatePrivateEndpointQS-rg' -Location 'eastus' -VM $vmConfig

```

## Create private endpoint

In this section, you'll create the private endpoint and connection using:

- [New-AzPrivateLinkServiceConnection](#)
- [New-AzPrivateEndpoint](#)

```

Place web app into variable. Replace <webapp-resource-group-name> with the resource group of your webapp.
##
Replace <your-webapp-name> with your webapp name
$webapp = Get-AzWebApp -ResourceGroupName <webapp-resource-group-name> -Name <your-webapp-name>

Create private endpoint connection.
$parameters1 = @{
 Name = 'myConnection'
 PrivateLinkServiceId = $webapp.ID
 GroupID = 'sites'
}
$privateEndpointConnection = New-AzPrivateLinkServiceConnection @parameters1

Place virtual network into variable.
$vnet = Get-AzVirtualNetwork -ResourceGroupName 'CreatePrivateEndpointQS-rg' -Name 'myVNet'

Disable private endpoint network policy
$vnet.Subnets[0].PrivateEndpointNetworkPolicies = "Disabled"
$vnet | Set-AzVirtualNetwork

Create private endpoint
$parameters2 = @{
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Name = 'myPrivateEndpoint'
 Location = 'eastus'
 Subnet = $vnet.Subnets[0]
 PrivateLinkServiceConnection = $privateEndpointConnection
}
New-AzPrivateEndpoint @parameters2

```

## Configure the private DNS zone

In this section you'll create and configure the private DNS zone using:

- [New-AzPrivateDnsZone](#)
- [New-AzPrivateDnsVirtualNetworkLink](#)
- [New-AzPrivateDnsZoneConfig](#)
- [New-AzPrivateDnsZoneGroup](#)

```

Place virtual network into variable.
$vnet = Get-AzVirtualNetwork -ResourceGroupName 'CreatePrivateEndpointQS-rg' -Name 'myVNet'

Create private dns zone.
$parameters1 = @{
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 Name = 'privatelink.azurewebsites.net'
}
$zone = New-AzPrivateDnsZone @parameters1

Create dns network link.
$parameters2 = @{
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 ZoneName = 'privatelink.azurewebsites.net'
 Name = 'myLink'
 VirtualNetworkId = $vnet.Id
}
$link = New-AzPrivateDnsVirtualNetworkLink @parameters2

Create DNS configuration
$parameters3 = @{
 Name = 'privatelink.azurewebsites.net'
 PrivateDnsZoneId = $zone.ResourceId
}
$config = New-AzPrivateDnsZoneConfig @parameters3

Create DNS zone group.
$parameters4 = @{
 ResourceGroupName = 'CreatePrivateEndpointQS-rg'
 PrivateEndpointName = 'myPrivateEndpoint'
 Name = 'myZoneGroup'
 PrivateDnsZoneConfig = $config
}
New-AzPrivateDnsZoneGroup @parameters4

```

## Test connectivity to private endpoint

In this section, you'll use the virtual machine you created in the previous step to connect to the SQL server across the private endpoint.

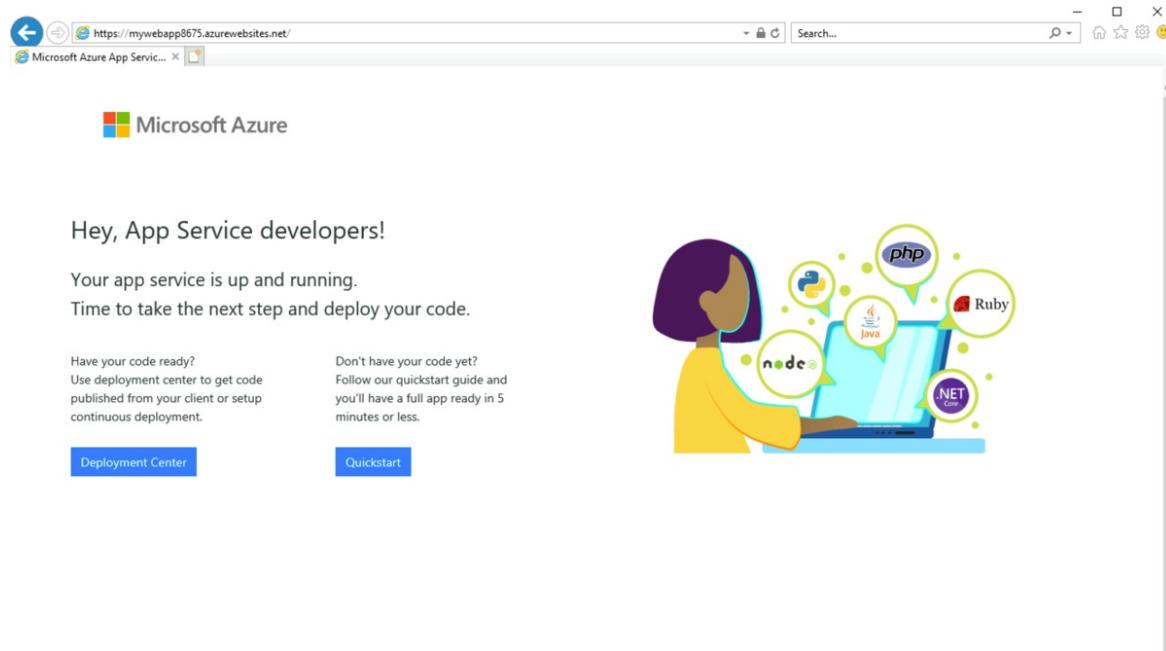
1. Sign in to the [Azure portal](#)
2. Select **Resource groups** in the left-hand navigation pane.
3. Select **CreatePrivateEndpointQS-rg**.
4. Select **myVM**.
5. On the overview page for **myVM**, select **Connect** then **Bastion**.
6. Select the blue **Use Bastion** button.
7. Enter the username and password that you entered during the virtual machine creation.
8. Open Windows PowerShell on the server after you connect.
9. Enter `nslookup <your-webapp-name>.azurewebsites.net`. Replace `<your-webapp-name>` with the name of the web app you created in the previous steps. You'll receive a message similar to what is displayed below:

```
Server: Unknown
Address: 168.63.129.16

Non-authoritative answer:
Name: mywebapp8675.privatelink.azurewebsites.net
Address: 10.0.0.5
Aliases: mywebapp8675.azurewebsites.net
```

A private IP address of 10.0.0.5 is returned for the web app name. This address is in the subnet of the virtual network you created previously.

10. In the bastion connection to **myVM**, open Internet Explorer.
11. Enter the url of your web app, <https://mywebapp8675.azurewebsites.net>.
12. You'll receive the default web app page if your application hasn't been deployed:



13. Close the connection to **myVM**.

## Clean up resources

When you're done using the private endpoint and the VM, use [Remove-AzResourceGroup](#) to remove the resource group and all the resources it has:

```
Remove-AzResourceGroup -Name CreatePrivateEndpointQS-rg -Force
```

## Next steps

In this quickstart, you created a:

- Virtual network and bastion host.
- Virtual machine.
- Private endpoint for an Azure Web App.

You used the virtual machine to test connectivity securely to the web app across the private endpoint.

For more information on the services that support a private endpoint, see:

Private Link availability

# Quickstart: Create a Private Endpoint using Azure CLI

2/18/2021 • 5 minutes to read • [Edit Online](#)

Get started with Azure Private Link by using a Private Endpoint to connect securely to an Azure web app.

In this quickstart, you'll create a private endpoint for an Azure web app and deploy a virtual machine to test the private connection.

Private endpoints can be created for different kinds of Azure services, such as Azure SQL and Azure Storage.

## Prerequisites

- An Azure account with an active subscription. [Create an account for free](#).
- An Azure Web App with a **PremiumV2**-tier or higher app service plan deployed in your Azure subscription.
  - For more information and an example, see [Quickstart: Create an ASP.NET Core web app in Azure](#).
  - For a detailed tutorial on creating a web app and an endpoint, see [Tutorial: Connect to a web app using an Azure Private Endpoint](#).
- Sign in to the Azure portal and check that your subscription is active by running `az login`.
- Check your version of the Azure CLI in a terminal or command window by running `az --version`. For the latest version, see the [latest release notes](#).
  - If you don't have the latest version, update your installation by following the [installation guide for your operating system or platform](#).

## Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with `az group create`:

- Named `CreatePrivateEndpointQS-rg`.
- In the `eastus` location.

```
az group create \
 --name CreatePrivateEndpointQS-rg \
 --location eastus
```

## Create a virtual network and bastion host

In this section, you'll create a virtual network, subnet, and bastion host.

The bastion host will be used to connect securely to the virtual machine for testing the private endpoint.

Create a virtual network with `az network vnet create`

- Named `myVNet`.
- Address prefix of `10.0.0.0/16`.
- Subnet named `myBackendSubnet`.
- Subnet prefix of `10.0.0.0/24`.

- In the **CreatePrivateEndpointQS-rg** resource group.
- Location of **eastus**.

```
az network vnet create \
--resource-group CreatePrivateEndpointQS-rg \
--location eastus \
--name myVNet \
--address-prefixes 10.0.0.0/16 \
--subnet-name myBackendSubnet \
--subnet-prefixes 10.0.0.0/24
```

Update the subnet to disable private endpoint network policies for the private endpoint with [az network vnet subnet update](#):

```
az network vnet subnet update \
--name myBackendSubnet \
--resource-group CreatePrivateEndpointQS-rg \
--vnet-name myVNet \
--disable-private-endpoint-network-policies true
```

Use [az network public-ip create](#) to create a public ip address for the bastion host:

- Create a standard zone redundant public IP address named **myBastionIP**.
- In **CreatePrivateEndpointQS-rg**.

```
az network public-ip create \
--resource-group CreatePrivateEndpointQS-rg \
--name myBastionIP \
--sku Standard
```

Use [az network vnet subnet create](#) to create a bastion subnet:

- Named **AzureBastionSubnet**.
- Address prefix of **10.0.1.0/24**.
- In virtual network **myVNet**.
- In resource group **CreatePrivateEndpointQS-rg**.

```
az network vnet subnet create \
--resource-group CreatePrivateEndpointQS-rg \
--name AzureBastionSubnet \
--vnet-name myVNet \
--address-prefixes 10.0.1.0/24
```

Use [az network bastion create](#) to create a bastion host:

- Named **myBastionHost**.
- In **CreatePrivateEndpointQS-rg**.
- Associated with public IP **myBastionIP**.
- Associated with virtual network **myVNet**.
- In **eastus** location.

```
az network bastion create \
--resource-group CreatePrivateEndpointQS-rg \
--name myBastionHost \
--public-ip-address myBastionIP \
--vnet-name myVNet \
--location eastus
```

It can take a few minutes for the Azure Bastion host to deploy.

## Create test virtual machine

In this section, you'll create a virtual machine that will be used to test the private endpoint.

Create a VM with [az vm create](#). When prompted, provide a password to be used as the credentials for the VM:

- Named **myVM**.
- In **CreatePrivateEndpointQS-rg**.
- In network **myVNet**.
- In subnet **myBackendSubnet**.
- Server image **Win2019Datacenter**.

```
az vm create \
--resource-group CreatePrivateEndpointQS-rg \
--name myVM \
--image Win2019Datacenter \
--public-ip-address "" \
--vnet-name myVNet \
--subnet myBackendSubnet \
--admin-username azureuser
```

## Create private endpoint

In this section, you'll create the private endpoint.

Use [az webapp list](#) to place the resource ID of the Web app you previously created into a shell variable.

Use [az network private-endpoint create](#) to create the endpoint and connection:

- Named **myPrivateEndpoint**.
- In resource group **CreatePrivateEndpointQS-rg**.
- In virtual network **myVNet**.
- In subnet **myBackendSubnet**.
- Connection named **myConnection**.
- Your webapp <webapp-resource-group-name>.

```
id=$(az webapp list \
 --resource-group <webapp-resource-group-name> \
 --query '[][.id]' \
 --output tsv)

az network private-endpoint create \
 --name myPrivateEndpoint \
 --resource-group CreatePrivateEndpointQS-rg \
 --vnet-name myVNet --subnet myBackendSubnet \
 --private-connection-resource-id $id \
 --group-id sites \
 --connection-name myConnection
```

## Configure the private DNS zone

In this section, you'll create and configure the private DNS zone using [az network private-dns zone create](#).

You'll use [az network private-dns link vnet create](#) to create the virtual network link to the dns zone.

You'll create a dns zone group with [az network private-endpoint dns-zone-group create](#).

- Zone named **privatelink.azurewebsites.net**
- In virtual network **myVNet**.
- In resource group **CreatePrivateEndpointQS-rg**.
- DNS link named **myDNSLink**.
- Associated with **myPrivateEndpoint**.
- Zone group named **MyZoneGroup**.

```
az network private-dns zone create \
 --resource-group CreatePrivateEndpointQS-rg \
 --name "privatelink.azurewebsites.net"

az network private-dns link vnet create \
 --resource-group CreatePrivateEndpointQS-rg \
 --zone-name "privatelink.azurewebsites.net" \
 --name MyDNSLink \
 --virtual-network myVNet \
 --registration-enabled false

az network private-endpoint dns-zone-group create \
 --resource-group CreatePrivateEndpointQS-rg \
 --endpoint-name myPrivateEndpoint \
 --name MyZoneGroup \
 --private-dns-zone "privatelink.azurewebsites.net" \
 --zone-name webapp
```

## Test connectivity to private endpoint

In this section, you'll use the virtual machine you created in the previous step to connect to the SQL server across the private endpoint.

1. Sign in to the [Azure portal](#)
2. Select **Resource groups** in the left-hand navigation pane.
3. Select **CreatePrivateEndpointQS-rg**.
4. Select **myVM**.

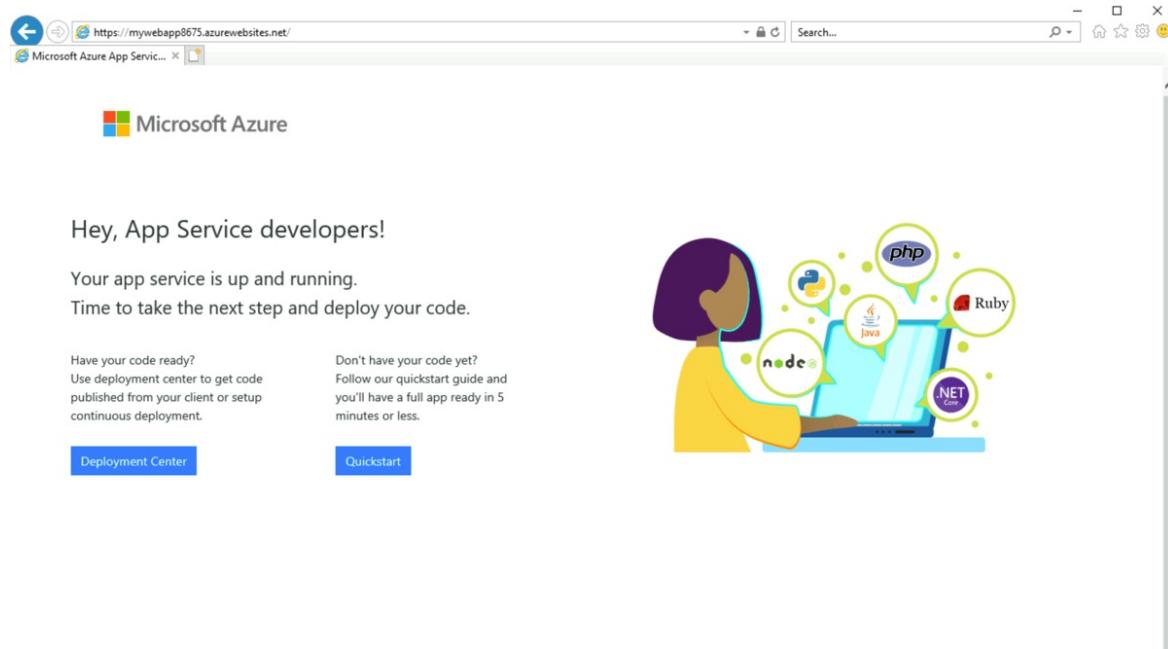
5. On the overview page for **myVM**, select **Connect** then **Bastion**.
6. Select the blue **Use Bastion** button.
7. Enter the username and password that you entered during the virtual machine creation.
8. Open Windows PowerShell on the server after you connect.
9. Enter `nslookup <your-webapp-name>.azurewebsites.net`. Replace `<your-webapp-name>` with the name of the web app you created in the previous steps. You'll receive a message similar to what is displayed below:

```
Server: UnKnown
Address: 168.63.129.16

Non-authoritative answer:
Name: mywebapp8675.privatelink.azurewebsites.net
Address: 10.0.0.5
Aliases: mywebapp8675.azurewebsites.net
```

A private IP address of 10.0.0.5 is returned for the web app name. This address is in the subnet of the virtual network you created previously.

10. In the bastion connection to **myVM**, open Internet Explorer.
11. Enter the url of your web app, <https://<your-webapp-name>.azurewebsites.net>.
12. You'll receive the default web app page if your application hasn't been deployed:



13. Close the connection to **myVM**.

## Clean up resources

When you're done using the private endpoint and the VM, use `az group delete` to remove the resource group and all the resources it has:

```
az group delete \
--name CreatePrivateEndpointQS-rg
```

## Next steps

In this quickstart, you created a:

- Virtual network and bastion host.
- Virtual machine.
- Private endpoint for an Azure Web App.

You used the virtual machine to test connectivity securely to the web app across the private endpoint.

For more information on the services that support a private endpoint, see:

[Private Link availability](#)

# Tutorial: Restrict network access to PaaS resources with virtual network service endpoints using the Azure portal

2/18/2021 • 10 minutes to read • [Edit Online](#)

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network. In this tutorial, you learn how to:

- Create a virtual network with one subnet
- Add a subnet and enable a service endpoint
- Create an Azure resource and allow network access to it from only a subnet
- Deploy a virtual machine (VM) to each subnet
- Confirm access to a resource from a subnet
- Confirm access is denied to a resource from a subnet and the internet

If you prefer, you can complete this tutorial using the [Azure CLI](#) or [Azure PowerShell](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Log in to Azure

Log in to the Azure portal at <https://portal.azure.com>.

## Create a virtual network

1. Select + **Create a resource** on the upper, left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual networks**.
3. Click + **Add** and enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select <b>Create new</b> and enter <i>myResourceGroup</i> .
Name	Enter <i>myVirtualNetwork</i>
Region	Select <b>(US) East US</b>

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

**Project details**

Subscription \*

Resource group \*

**Instance details**

Name \*

Region \*

**Buttons:** Review + create, < Previous, Next : IP Addresses >, Download a template for automation

4. Click **Next: IP addresses >**

SETTING	VALUE
IPv4Address space	Leave as default
Subnet name	Click <b>default</b> and change the name from "default" to "Public"
Subnet Address Range	Leave as default

5. Click **Next: Security >**

SETTING	VALUE
BastionHost	Disable
DDoS protection	Disable
Firewall	Disable

6. When complete, click **Review and create**.

7. If the validation checks pass, click **Create**.

8. Wait for the deployment to finish, then click **Go to resource** or move on to the next section.

## Enable a service endpoint

Service endpoints are enabled per service, per subnet. To create a subnet and enable a service endpoint for the subnet:

- If you are not already on the virtual network resource page, you can search for the newly created network in the **Search resources, services, and docs** box at the top of the portal, enter *myVirtualNetwork*, and select it from the list.

2. In the **Settings** menu (left), select **Subnets**, and then select **+ Subnet**, as shown:

The screenshot shows the Azure portal interface for managing virtual networks. On the left, under 'Virtual networks', the 'myVirtualNetwork' network is selected. The 'Subnets' option is highlighted with a red box. On the right, a modal window titled 'Add subnet' is open. It has fields for 'Name' (set to 'Private'), 'Subnet address range' (set to '10.2.1.0/24'), and a dropdown for 'Services' which is set to 'Microsoft.Storage'. At the bottom of the modal are 'Save' and 'Cancel' buttons.

3. Under **Add subnet**, select or enter the following information, and then select **OK**:

SETTING	VALUE
Name	Private
Address range	Leave as default
Service endpoints	Select <b>Microsoft.Storage</b>
Service endpoint policies	Leave default as 0

#### Caution

Before enabling a service endpoint for an existing subnet that has resources in it, see [Change subnet settings](#).

4. Click **Save**, then close the Subnet window on the right. The newly created subnet should appear the list.

## Restrict network access for a subnet

By default, all virtual machine instances in a subnet can communicate with all resources. You can limit communication to and from all resources in a subnet by creating a network security group, and associating it to the subnet:

1. Select **All services** in the upper left corner of the Azure portal.
2. Select **Networking**, and then select (or search for) **Network security groups**.
3. From the **Network security groups** page, click **+ Add**.
4. Enter the following information

SETTING	VALUE
Subscription	Select your subscription

SETTING	VALUE
Resource group	Select <i>myResourceGroup</i> from the list
Name	Enter <b>myNsgPrivate</b>
Location	Select <b>East US</b>

5. Click **Review + create**, and when the validation check is passed, click **Create**.
6. After the network security group is created, click **Go to resource** or search for *myNsgPrivate*.
7. Under **Settings** on the left, select **Outbound security rules**.
8. Select **+ Add**.
9. Create a rule that allows outbound communication to the Azure Storage service. Enter, or select, the following information, and then select **Add**:

SETTING	VALUE
Source	Select <b>VirtualNetwork</b>
Source port ranges	*
Destination	Select <b>Service Tag</b>
Destination service tag	Select <b>Storage</b>
Destination port ranges	Leave default as <i>8080</i>
Protocol	Any
Action	Allow
Priority	100
Name	Rename to <b>Allow-Storage-All</b>

10. Create another outbound security rule that denies communication to the internet. This rule overrides a default rule in all network security groups that allows outbound internet communication. Complete steps 6-9 from above using the following values:

SETTING	VALUE
Source	Select <b>VirtualNetwork</b>
Source port ranges	*
Destination	Select <b>Service Tag</b>
Destination service tag	Select <b>Internet</b>
Destination port ranges	*

SETTING	VALUE
Protocol	Any
Action	Change default to <i>Deny</i>
Priority	110
Name	Change to <i>Deny-Internet-All</i>

11. Create an *inbound security rule* that allows Remote Desktop Protocol (RDP) traffic to the subnet from anywhere. The rule overrides a default security rule that denies all inbound traffic from the internet. Remote desktop connections are allowed to the subnet so that connectivity can be tested in a later step.

12. Under **Settings**, select **Inbound security rules**.

13. Select + **Add** and use the following values:

SETTING	VALUE
Source	Any
Source port ranges	*
Destination	Select <b>VirtualNetwork</b>
Destination port ranges	Change to <i>3389</i>
Protocol	Any
Action	Allow
Priority	120
Name	Change to <i>Allow-RDP-All</i>

#### WARNING

RDP port 3389 is exposed to the Internet. This is only recommended for testing. For *Production environments*, we recommend using a VPN or private connection.

1. Under **Settings**, select **Subnets**.
2. Click + **Associate**.
3. Under **Virtual network**, select **myVirtualNetwork**.
4. Under **Subnet**, select **Private**, and then select **Ok**.

## Restrict network access to a resource

The steps required to restrict network access to resources created through Azure services, which are enabled for service endpoints will vary across services. See the documentation for individual services for specific steps for each service. The remainder of this tutorial includes steps to restrict network access for an Azure Storage account, as an example.

## Create a storage account

1. Select + **Create a resource** on the upper, left corner of the Azure portal.
2. Enter "Storage account" in the search bar, and select it from the drop-down menu.
3. Click + **Add**.
4. Enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select <i>myResourceGroup</i>
Storage Account Name	Enter a name that is unique across all Azure locations, between 3-24 characters in length, using only numbers and lower-case letters.
Location	Select <b>(US) East US</b>
Performance	Standard
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)

5. Select **Create + review**, and when validation checks have passed, click **Create**.

### NOTE

The deployment may take a couple of minutes to complete.

6. After the storage account is created, click **Go to resource**

## Create a file share in the storage account

1. Go to your storage account overview page.
2. Select the **File Shares** app icon, then click + **File share**.

SETTING	VALUE
Name	my-file-share
Quota	'Set to maximum'

The screenshot shows the Azure Storage account 'rhodiumstoracct2' with the 'File shares' blade open. On the left, there's a navigation menu with options like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data transfer, Events, and Storage Explorer (preview). Under Settings, there are links for Access keys, Geo-replication, CORS, Configuration, Encryption, Shared access signature, and Networking. The main area shows 'File share settings' with Active Directory set to 'Not configured', Soft delete to 'Disabled', and Share capacity to '5 TiB'. A search bar for file shares by prefix is present. Below it, a table lists file shares with columns for Name, Modified, and Tier. A message says 'You don't have any file shares yet. Click '+ File share' to get started.' On the right, a modal window titled 'New file share' is displayed, asking for a 'Name' (with 'my-file-share' entered) and a 'Tier' (set to 'Premium'). At the bottom of the modal are 'Create' and 'Discard' buttons.

3. Click **Create**.
4. The file share should be shown in the Azure window, if not click **Refresh**

#### **Restrict network access to a subnet**

By default, storage accounts accept network connections from clients in any network, including the internet. You can restrict network access from the internet, and all other subnets in all virtual networks (except the *Private* subnet in the *myVirtualNetwork* virtual network.) To restrict network access to a subnet:

1. Under **Settings** for your (uniquely named) storage account, select **Networking**.
2. Select **Selected networks**.
3. Select **+ Add existing virtual network**.
4. Under **Add networks**, select the following values, and then select **Add**:

SETTING	VALUE
Subscription	Select your subscription
Virtual networks	<b>myVirtualNetwork</b>
Subnets	<b>Private</b>

The screenshot shows the Microsoft Azure Storage account Networking settings page. The left sidebar has a 'Networking' tab highlighted with a red box. In the main area, there's a 'Virtual networks' section with a 'Add existing virtual network' button highlighted by a red box. A modal window titled 'Add networks' is overlaid, showing a list of virtual networks. One specific entry, 'myVirtualNetwork', is highlighted with a red box.

5. Click **Add** and, then immediately click the **Save** icon to save the changes.
6. Under **Settings** for the storage account, select **Access keys**, as shown in the following image:

The screenshot shows the Microsoft Azure Storage account Access keys page. The left sidebar has an 'Access keys' tab highlighted with a red box. In the main area, there's a 'Show keys' button highlighted by a red box. The page lists two access keys: 'key1' and 'key2', each with its key value and connection string.

7. Click **Show keys** and note the **Key** values, as you'll have to manually enter key1 in a later step when mapping the file share to a drive letter in a VM.

## Create virtual machines

To test network access to a storage account, deploy a VM to each subnet.

### Create the first virtual machine

1. From the "Search resources ..." bar, search for **Virtual machines**.
2. Select **+ Add > Virtual machine**.
3. Enter the following information:

SETTING	VALUE
Subscription	Select your subscription
Resource group	Select **myResourceGroup, which was created earlier.
Virtual machine name	Enter <i>myVmPublic</i>
Region	(US) East US
Availability options	Availability zone
Availability zone	1
Image	Windows Server 2019 Datacenter - Gen1
Size	Select the VM Instance size you want to use
Username	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the <a href="#">defined complexity requirements</a> .
Public inbound ports	Allow selected ports
Select inbound ports	Leave default set to <i>RDP (3389)</i>

Microsoft Azure

Search resources, services, and docs (G+)

Home > Virtual machines >

## Create a virtual machine

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*  Resource group \*  [Create new](#)

**Instance details**

Virtual machine name \*  Region \*  Availability options  Availability zone \*  Image \*  [See all images](#)

Azure Spot instance

Size \*  [See all sizes](#)

**Administrator account**

Username \*  Password \*  Confirm password \*

**Inbound port rules**

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \*  Allow selected ports  None

Select inbound ports \*

**Warning:** This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

**Review + create** < Previous Next : Disks >

4. Select the **Networking** tab and then select **myVirtualNetwork**.
5. Select the **Public** subnet.
6. Under **NIC Network Security Group**, select **Advanced**. The portal automatically creates a network security group for you that allows port 3389, which you'll need open to connect to the virtual machine in a later step.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Virtual machines >

## Create a virtual machine ⊕

[Learn more](#)

### Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network \* (i) myVirtualNetwork Create new

Subnet \* (i) Public Manage subnet configuration

Public IP (i) (new) myVmPublic-ip Create new

NIC network security group (i)  None  Basic  Advanced

Configure network security group \* (i) (new) myVmPublic-nsg Create new

Accelerated networking (i)  On  Off  
The selected VM size does not support accelerated networking.

### Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution?  Yes  No

Review + create < Previous Next : Management >

7. Select **Review and create**, then **Create** and wait for the deployment to finish.
8. Click **Go to resource**, or open the **Home > Virtual machines** page, and select the VM you just created *myVmPublic*, which should be started.

### Create the second virtual machine

1. Complete steps 1-8 again, but in step 3, name the virtual machine *myVmPrivate* and set **Public inbound port** to "None".
2. In step 4-5, select the **Private** subnet.

#### NOTE

The **NIC network security group** and **Public inbound ports** settings should mirror the image shown below, including the blue confirmation window stating: "all public internet traffic will be blocked by default".

**Microsoft Azure** Search resources, services, and docs (G+/-)

Home > Virtual machines >

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

**Network interface**

When creating a virtual machine, a network interface will be created for you.

Virtual network\* (myVirtualNetwork) [Create new](#)

Subnet\* (Private) [Manage subnet configuration](#)

Public IP (new) myVmPrivateip507 [Create new](#)

NIC network security group (None)  Basic  Advanced

Public inbound ports\* (None)  None  Allow selected ports

Select inbound ports Select one or more ports

ⓘ All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Accelerated networking (Off)  On  Off

The selected VM size does not support accelerated networking.

Load balancing

**Review + create** [< Previous](#) [Next : Management >](#)

**3. Select Review and create, then Create and wait for the deployment to finish.**

**WARNING**

Please do not continue to the next step until the deployment is complete.

**4. Wait for the confirmation window shown below and click Go to resource.**

Microsoft Azure

Home > CreateVm-MicrosoftWindowsServer.WindowsServer-201-20201209152720 | Overview

Deployment

Search (Ctrl+ /) <> Delete Cancel Redeploy Refresh

Overview We'd love your feedback! →

**Your deployment is complete**

Deployment name: CreateVm-MicrosoftWindowsServer.WindowsSe... Start time: 12/9/2020, 3:42:28 PM  
Subscription: C&L App Platforms Correlation ID: 102a0772-1851-4967-950e-7575b912e  
Resource group: myResourceGroup

Deployment details (Download)

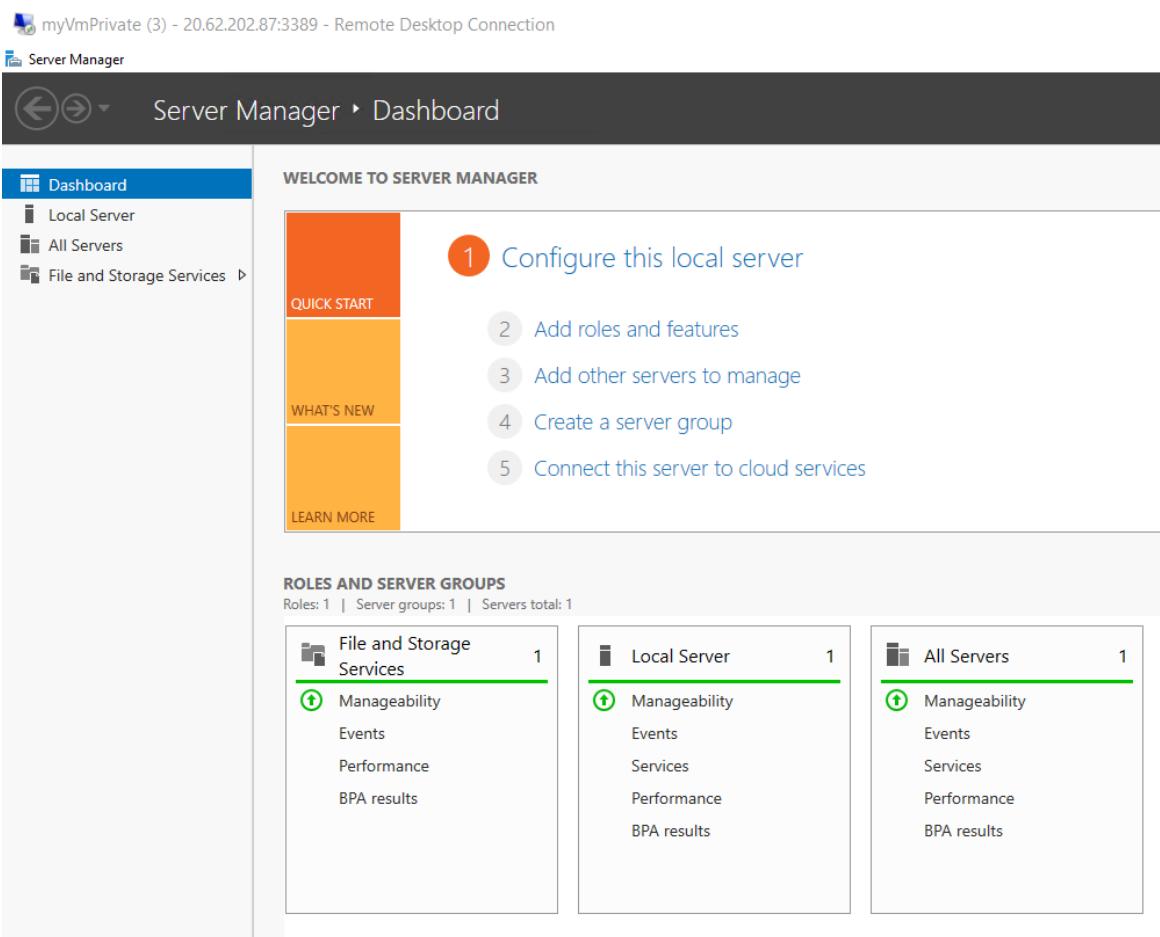
Next steps

- Enable Automange virtual machine best practices (preview) Recommended
- Setup auto-shutdown Recommended
- Monitor VM health, performance and network dependencies Recommended
- Run a script inside the virtual machine Recommended

Go to resource Create another VM

## Confirm access to storage account

1. Once the *myVmPrivate* VM has been created, click **Go to resource**.
2. Connect to the VM by selecting the **Connect > RDP**.
3. After selecting the **Connect** button, a Remote Desktop Protocol (.rdp) file is created. Click **Download RDP File** to download to your computer.
4. 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. For the email field, enter the "Administrator account: username" credentials you specified earlier.
5. Select **OK**.
6. 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. You should see the VM start as shown:



7. In the VM window, open a PowerShell CLI instance.
8. Using the script below, map the Azure file share to drive Z using PowerShell. Before running the commands that follow, replace `<storage-account-key>` and both `<storage-account-name>` fields with values you supplied and "d earlier in [Create a storage account](#).

```
$acctKey = ConvertTo-SecureString -String "<storage-account-key>" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\<storage-account-name>", $acctKey
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\<storage-account-name>.file.core.windows.net\my-file-share" -Credential $credential
```

PowerShell returns output similar to the following example output:

Name	Used (GB)	Free (GB)	Provider	Root
Z			FileSystem	\vnt.file.core.windows.net\my-f...

The Azure file share successfully mapped to the Z drive.

9. Close the remote desktop session to the *myVmPrivate* VM.

## Confirm access is denied to storage account

1. Enter *myVmPublic* in the **Search resources, services, and docs** box at the top of the portal.
2. When **myVmPublic** appears in the search results, select it.
3. Complete steps 1-8 above in [Confirm access to storage account](#) for the *myVmPublic* VM.

After a short wait, you receive a `New-PSDrive : Access is denied` error. Access is denied because the *myVmPublic* VM is deployed in the *Public* subnet. The *Public* subnet does not have a service endpoint enabled for Azure Storage. The storage account only allows network access from the *Private* subnet, not the *Public* subnet.

4. Close the remote desktop session to the *myVmPublic* VM.
5. Back in the Azure portal, go to the uniquely named storage account you created earlier.
6. Under File Service, select **File shares**, the *my-file-share*, created earlier.

7. You should receive the following error message:

The screenshot shows the Azure Storage File Share blade for the 'my-file-share' file share. The left sidebar lists 'Overview', 'Access Control (IAM)', 'Properties', 'Operations', 'Synchronizations', and 'Backup'. The 'Overview' tab is selected. In the center, there's a large 'Cloud with a slash' icon. Below it, the text 'This machine doesn't seem to have access.' is displayed. A note below states: 'This storage account is located in a VNET. Recent changes to "Firewalls and virtual networks" settings may not be in effect yet. If you expect this machine to be able to connect to the content of this file share, check that this machine is a part of the VNET or try waiting a few minutes for changes in settings to take effect, and then refresh this page.' There are 'Learn more' and 'Leave preview' buttons. At the bottom, a summary table shows details: Session ID: dd087e484e9e4d0899af8b25b129dd9e; Resource ID: /subscriptions/b96d505c-4e34-426c-9e93-7e5e8ff744b...; Extension: Microsoft\_Azure\_FileStorage; Content: FilesGridBladeV2; Error code: 403.

### NOTE

Access is denied, because your computer is not in the *Private* subnet of the *MyVirtualNetwork* virtual network.

## Clean up resources

When no longer needed, delete the resource group and all resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

## Next steps

In this tutorial, you enabled a service endpoint for a virtual network subnet. You learned that you can enable service endpoints for resources deployed from multiple Azure services. You created an Azure Storage account and restricted network access to the storage account to only resources within a virtual network subnet. To learn more about service endpoints, see [Service endpoints overview](#) and [Manage subnets](#).

If you have multiple virtual networks in your account, you may want to connect two virtual networks together so the resources within each virtual network can communicate with each other. To learn how to connect virtual networks, advance to the next tutorial.

[Connect virtual networks](#)

# Restrict network access to PaaS resources with virtual network service endpoints using PowerShell

2/18/2021 • 10 minutes to read • [Edit Online](#)

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network. In this article, you learn how to:

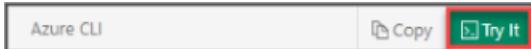
- Create a virtual network with one subnet
- Add a subnet and enable a service endpoint
- Create an Azure resource and allow network access to it from only a subnet
- Deploy a virtual machine (VM) to each subnet
- Confirm access to a resource from a subnet
- Confirm access is denied to a resource from a subnet and the internet

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a virtual network

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup*.

```
New-AzResourceGroup -ResourceGroupName myResourceGroup -Location EastUS
```

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVirtualNetwork* with the address prefix *10.0.0.0/16*.

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

Create a subnet configuration with [New-AzVirtualNetworkSubnetConfig](#). The following example creates a subnet configuration for a subnet named *Public*.

```
$subnetConfigPublic = Add-AzVirtualNetworkSubnetConfig `
 -Name Public `
 -AddressPrefix 10.0.0.0/24 `
 -VirtualNetwork $virtualNetwork
```

Create the subnet in the virtual network by writing the subnet configuration to the virtual network with [Set-AzVirtualNetwork](#):

```
$virtualNetwork | Set-AzVirtualNetwork
```

## Enable a service endpoint

You can enable service endpoints only for services that support service endpoints. View service endpoint-enabled services available in an Azure location with [Get-AzVirtualNetworkAvailableEndpointService](#). The following example returns a list of service-endpoint-enabled services available in the *eastus* region. The list of services returned will grow over time as more Azure services become service endpoint enabled.

```
Get-AzVirtualNetworkAvailableEndpointService -Location eastus | Select Name
```

Create an additional subnet in the virtual network. In this example, a subnet named *Private* is created with a service endpoint for *Microsoft.Storage*.

```
$subnetConfigPrivate = Add-AzVirtualNetworkSubnetConfig `
 -Name Private `
 -AddressPrefix 10.0.1.0/24 `
 -VirtualNetwork $virtualNetwork `
 -ServiceEndpoint Microsoft.Storage

$virtualNetwork | Set-AzVirtualNetwork
```

## Restrict network access for a subnet

Create network security group security rules with [New-AzNetworkSecurityRuleConfig](#). The following rule allows outbound access to the public IP addresses assigned to the Azure Storage service:

```
$rule1 = New-AzNetworkSecurityRuleConfig `
 -Name Allow-Storage-All `
 -Access Allow `
 -DestinationAddressPrefix Storage `
 -DestinationPortRange * `
 -Direction Outbound `
 -Priority 100 `
 -Protocol * `
 -SourceAddressPrefix VirtualNetwork `
 -SourcePortRange *
```

The following rule denies access to all public IP addresses. The previous rule overrides this rule, due to its higher priority, which allows access to the public IP addresses of Azure Storage.

```
$rule2 = New-AzNetworkSecurityRuleConfig `
 -Name Deny-Internet-All `
 -Access Deny `
 -DestinationAddressPrefix Internet `
 -DestinationPortRange * `
 -Direction Outbound `
 -Priority 110 `
 -Protocol * `
 -SourceAddressPrefix VirtualNetwork `
 -SourcePortRange *
```

The following rule allows Remote Desktop Protocol (RDP) traffic inbound to the subnet from anywhere. Remote desktop connections are allowed to the subnet, so that you can confirm network access to a resource in a later step.

```
$rule3 = New-AzNetworkSecurityRuleConfig `
 -Name Allow-RDP-All `
 -Access Allow `
 -DestinationAddressPrefix VirtualNetwork `
 -DestinationPortRange 3389 `
 -Direction Inbound `
 -Priority 120 `
 -Protocol * `
 -SourceAddressPrefix * `
 -SourcePortRange *
```

Create a network security group with [New-AzNetworkSecurityGroup](#). The following example creates a network security group named *myNsgPrivate*.

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName myResourceGroup `
 -Location EastUS `
 -Name myNsgPrivate `
 -SecurityRules $rule1,$rule2,$rule3
```

Associate the network security group to the *Private* subnet with [Set-AzVirtualNetworkSubnetConfig](#) and then write the subnet configuration to the virtual network. The following example associates the *myNsgPrivate* network security group to the *Private* subnet:

```
Set-AzVirtualNetworkSubnetConfig `
 -VirtualNetwork $VirtualNetwork `
 -Name Private `
 -AddressPrefix 10.0.1.0/24 `
 -ServiceEndpoint Microsoft.Storage `
 -NetworkSecurityGroup $nsg

$virtualNetwork | Set-AzVirtualNetwork
```

## Restrict network access to a resource

The steps necessary to restrict network access to resources created through Azure services enabled for service endpoints varies across services. See the documentation for individual services for specific steps for each service. The remainder of this article includes steps to restrict network access for an Azure Storage account, as an example.

### Create a storage account

Create an Azure storage account with [New-AzStorageAccount](#). Replace <replace-with-your-unique-storage-account-name> with a name that is unique across all Azure locations, between 3-24 characters in length, using only numbers and lower-case letters.

```
$storageAcctName = '<replace-with-your-unique-storage-account-name>'

New-AzStorageAccount `
 -Location EastUS `
 -Name $storageAcctName `
 -ResourceGroupName myResourceGroup `
 -SkuName Standard_LRS `
 -Kind StorageV2
```

After the storage account is created, retrieve the key for the storage account into a variable with [Get-AzStorageAccountKey](#):

```
$storageAcctKey = (Get-AzStorageAccountKey `
 -ResourceGroupName myResourceGroup `
 -AccountName $storageAcctName).Value[0]
```

The key is used to create a file share in a later step. Enter \$storageAcctKey and note the value, as you'll also need to manually enter it in a later step when you map the file share to a drive in a VM.

### Create a file share in the storage account

Create a context for your storage account and key with [New-AzStorageContext](#). The context encapsulates the storage account name and account key:

```
$storageContext = New-AzStorageContext $storageAcctName $storageAcctKey
```

Create a file share with [New-AzStorageShare](#):

```
$share = New-AzStorageShare my-file-share -Context $storageContext
```

### Deny all network access to a storage account

By default, storage accounts accept network connections from clients in any network. To limit access to selected networks, change the default action to *Deny* with [Update-AzStorageAccountNetworkRuleSet](#). Once network access is denied, the storage account is not accessible from any network.

```
Update-AzStorageAccountNetworkRuleSet `
-ResourceGroupName "myresourcegroup" `
-Name $storageAcctName `
-DefaultAction Deny
```

### Enable network access from a subnet

Retrieve the created virtual network with [Get-AzVirtualNetwork](#) and then retrieve the private subnet object into a variable with [Get-AzVirtualNetworkSubnetConfig](#):

```
$privateSubnet = Get-AzVirtualNetwork `
-ResourceGroupName "myResourceGroup" `
-Name "myVirtualNetwork" `
| Get-AzVirtualNetworkSubnetConfig `
-Name "Private"
```

Allow network access to the storage account from the *Private* subnet with [Add-AzStorageAccountNetworkRule](#).

```
Add-AzStorageAccountNetworkRule `
-ResourceGroupName "myresourcegroup" `
-Name $storageAcctName `
-VirtualNetworkResourceId $privateSubnet.Id
```

## Create virtual machines

To test network access to a storage account, deploy a VM to each subnet.

### Create the first virtual machine

Create a virtual machine in the *Public* subnet with [New-AzVM](#). 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-AzVm `
-ResourceGroupName "myResourceGroup" `
-Location "East US" `
-VirtualNetworkName "myVirtualNetwork" `
-SubnetName "Public" `
-Name "myVmPublic" `
-AsJob
```

Output similar to the following example output is returned:

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

## Create the second virtual machine

Create a virtual machine in the *Private* subnet:

```
New-AzVm `
-ResourceGroupName "myResourceGroup" `
-Location "East US" `
-VirtualNetworkName "myVirtualNetwork" `
-SubnetName "Private" `
-Name "myVmPrivate"
```

It takes a few minutes for Azure to create the VM. Do not continue to the next step until Azure finishes creating the VM and returns output to PowerShell.

## Confirm access to storage account

Use [Get-AzPublicIpAddress](#) to return the public IP address of a VM. The following example returns the public IP address of the *myVmPrivate* VM:

```
Get-AzPublicIpAddress `
-Name myVmPrivate `
-ResourceGroupName myResourceGroup `
| SelectIpAddress
```

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

```
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.

On the *myVmPrivate* VM, map the Azure file share to drive Z using PowerShell. Before running the commands that follow, replace `<storage-account-key>` and `<storage-account-name>` with values from you supplied or retrieved in [Create a storage account](#).

```
$acctKey = ConvertTo-SecureString -String "<storage-account-key>" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\<storage-account-name>", $acctKey
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\\<storage-account-name>.file.core.windows.net\my-file-share" -Credential $credential
```

PowerShell returns output similar to the following example output:

Name	Used (GB)	Free (GB)	Provider	Root
Z			FileSystem	\vnt.file.core.windows.net\my-f...

The Azure file share successfully mapped to the Z drive.

Confirm that the VM has no outbound connectivity to any other public IP addresses:

```
ping bing.com
```

You receive no replies, because the network security group associated to the *Private* subnet does not allow outbound access to public IP addresses other than the addresses assigned to the Azure Storage service.

Close the remote desktop session to the *myVmPrivate* VM.

## Confirm access is denied to storage account

Get the public IP address of the *myVmPublic* VM:

```
Get-AzPublicIpAddress `
-Name myVmPublic `
-ResourceGroupName myResourceGroup `
| SelectIpAddress
```

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

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

On the *myVmPublic* VM, attempt to map the Azure file share to drive Z. Before running the commands that follow, replace `<storage-account-key>` and `<storage-account-name>` with values from you supplied or retrieved in [Create a storage account](#).

```
$acctKey = ConvertTo-SecureString -String "<storage-account-key>" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\<storage-account-name>", $acctKey
New-PSDrive -Name Z -PSPrinter FileSystem -Root "\\\<storage-account-name>.file.core.windows.net\my-file-share" -Credential $credential
```

Access to the share is denied, and you receive a `New-PSDrive : Access is denied` error. Access is denied because the *myVmPublic* VM is deployed in the *Public* subnet. The *Public* subnet does not have a service endpoint enabled for Azure Storage, and the storage account only allows network access from the *Private* subnet, not the *Public* subnet.

Close the remote desktop session to the *myVmPublic* VM.

From your computer, attempt to view the file shares in the storage account with the following command:

```
Get-AzStorageFile `
-ShareName my-file-share `
-Context $storageContext
```

Access is denied, and you receive a *Get-AzStorageFile : The remote server returned an error: (403) Forbidden. HTTP Status Code: 403 - HTTP Error Message: This request is not authorized to perform this operation* error,

because your computer is not in the *Private* subnet of the *MyVirtualNetwork* virtual network.

## Clean up resources

When no longer needed, you can use [Remove-AzResourceGroup](#) to remove the resource group and all of the resources it contains:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

## Next steps

In this article, you enabled a service endpoint for a virtual network subnet. You learned that service endpoints can be enabled for resources deployed with multiple Azure services. You created an Azure Storage account and limited network access to the storage account to only resources within a virtual network subnet. To learn more about service endpoints, see [Service endpoints overview](#) and [Manage subnets](#).

If you have multiple virtual networks in your account, you may want to connect two virtual networks together so the resources within each virtual network can communicate with each other. To learn how, see [Connect virtual networks](#).

# Restrict network access to PaaS resources with virtual network service endpoints using the Azure CLI

2/18/2021 • 9 minutes to read • [Edit Online](#)

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network. In this article, you learn how to:

- Create a virtual network with one subnet
- Add a subnet and enable a service endpoint
- Create an Azure resource and allow network access to it from only a subnet
- Deploy a virtual machine (VM) to each subnet
- Confirm access to a resource from a subnet
- Confirm access is denied to a resource from a subnet and the internet

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a virtual network

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with `az group create`. The following example creates a resource group named `myResourceGroup` in the `eastus` location.

```
az group create \
--name myResourceGroup \
--location eastus
```

Create a virtual network with one subnet with [az network vnet create](#).

```
az network vnet create \
--name myVirtualNetwork \
--resource-group myResourceGroup \
--address-prefix 10.0.0.0/16 \
--subnet-name Public \
--subnet-prefix 10.0.0.0/24
```

## Enable a service endpoint

You can enable service endpoints only for services that support service endpoints. View service endpoint-enabled services available in an Azure location with [az network vnet list-endpoint-services](#). The following example returns a list of service-endpoint-enabled services available in the *eastus* region. The list of services returned will grow over time, as more Azure services become service endpoint enabled.

```
az network vnet list-endpoint-services \
--location eastus \
--out table
```

Create an additional subnet in the virtual network with [az network vnet subnet create](#). In this example, a service endpoint for *Microsoft.Storage* is created for the subnet:

```
az network vnet subnet create \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name Private \
--address-prefix 10.0.1.0/24 \
--service-endpoints Microsoft.Storage
```

## Restrict network access for a subnet

Create a network security group with [az network nsg create](#). The following example creates a network security group named *myNsgPrivate*.

```
az network nsg create \
--resource-group myResourceGroup \
--name myNsgPrivate
```

Associate the network security group to the *Private* subnet with [az network vnet subnet update](#). The following example associates the *myNsgPrivate* network security group to the *Private* subnet:

```
az network vnet subnet update \
--vnet-name myVirtualNetwork \
--name Private \
--resource-group myResourceGroup \
--network-security-group myNsgPrivate
```

Create security rules with [az network nsg rule create](#). The rule that follows allows outbound access to the public

IP addresses assigned to the Azure Storage service:

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Allow-Storage-All \
--access Allow \
--protocol "*" \
--direction Outbound \
--priority 100 \
--source-address-prefix "VirtualNetwork" \
--source-port-range "*" \
--destination-address-prefix "Storage" \
--destination-port-range "*"
```

Each network security group contains several [default security rules](#). The rule that follows overrides a default security rule that allows outbound access to all public IP addresses. The `--destination-address-prefix "Internet"` option denies outbound access to all public IP addresses. The previous rule overrides this rule, due to its higher priority, which allows access to the public IP addresses of Azure Storage.

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Deny-Internet-All \
--access Deny \
--protocol "*" \
--direction Outbound \
--priority 110 \
--source-address-prefix "VirtualNetwork" \
--source-port-range "*" \
--destination-address-prefix "Internet" \
--destination-port-range "*"
```

The following rule allows SSH traffic inbound to the subnet from anywhere. The rule overrides a default security rule that denies all inbound traffic from the internet. SSH is allowed to the subnet so that connectivity can be tested in a later step.

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Allow-SSH-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 120 \
--source-address-prefix "*" \
--source-port-range "*" \
--destination-address-prefix "VirtualNetwork" \
--destination-port-range "22"
```

## Restrict network access to a resource

The steps necessary to restrict network access to resources created through Azure services enabled for service endpoints varies across services. See the documentation for individual services for specific steps for each service. The remainder of this article includes steps to restrict network access for an Azure Storage account, as an example.

### Create a storage account

Create an Azure storage account with [az storage account create](#). Replace

<replace-with-your-unique-storage-account-name> with a name that is unique across all Azure locations, between 3-24 characters in length, using only numbers and lower-case letters.

```
storageAcctName=<replace-with-your-unique-storage-account-name>

az storage account create \
--name $storageAcctName \
--resource-group myResourceGroup \
--sku Standard_LRS \
--kind StorageV2
```

After the storage account is created, retrieve the connection string for the storage account into a variable with [az storage account show-connection-string](#). The connection string is used to create a file share in a later step.

```
saConnectionString=$(az storage account show-connection-string \
--name $storageAcctName \
--resource-group myResourceGroup \
--query 'connectionString' \
--out tsv)
```

View the contents of the variable and note the value for **AccountKey** returned in the output, because it's used in a later step.

```
echo $saConnectionString
```

### Create a file share in the storage account

Create a file share in the storage account with [az storage share create](#). In a later step, this file share is mounted to confirm network access to it.

```
az storage share create \
--name my-file-share \
--quota 2048 \
--connection-string $saConnectionString > /dev/null
```

### Deny all network access to a storage account

By default, storage accounts accept network connections from clients in any network. To limit access to selected networks, change the default action to *Deny* with [az storage account update](#). Once network access is denied, the storage account is not accessible from any network.

```
az storage account update \
--name $storageAcctName \
--resource-group myResourceGroup \
--default-action Deny
```

### Enable network access from a subnet

Allow network access to the storage account from the *Private* subnet with [az storage account network-rule add](#).

```
az storage account network-rule add \
--resource-group myResourceGroup \
--account-name $storageAcctName \
--vnet-name myVirtualNetwork \
--subnet Private
```

# Create virtual machines

To test network access to a storage account, deploy a VM to each subnet.

## Create the first virtual machine

Create a VM in the *Public* subnet with `az vm create`. If SSH keys do not already exist in a default key location, the command creates them. To use a specific set of keys, use the `--ssh-key-value` option.

```
az vm create \
--resource-group myResourceGroup \
--name myVmPublic \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet Public \
--generate-ssh-keys
```

The VM takes a few minutes to create. After the VM is created, the Azure CLI shows information similar to the following example:

```
{
 "fqdns": "",
 "id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVmPublic",
 "location": "eastus",
 "macAddress": "00-0D-3A-23-9A-49",
 "powerState": "VM running",
 "privateIpAddress": "10.0.0.4",
 "publicIpAddress": "13.90.242.231",
 "resourceGroup": "myResourceGroup"
}
```

Take note of the `publicIpAddress` in the returned output. This address is used to access the VM from the internet in a later step.

## Create the second virtual machine

```
az vm create \
--resource-group myResourceGroup \
--name myVmPrivate \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet Private \
--generate-ssh-keys
```

The VM takes a few minutes to create. After creation, take note of the `publicIpAddress` in the output returned. This address is used to access the VM from the internet in a later step.

## Confirm access to storage account

SSH into the `myVmPrivate` VM. Replace `<publicIpAddress>` with the public IP address of your `myVmPrivate` VM.

```
ssh <publicIpAddress>
```

Create a folder for a mount point:

```
sudo mkdir /mnt/MyAzureFileShare
```

Mount the Azure file share to the directory you created. Before running the following command, replace `<storage-account-name>` with the account name and `<storage-account-key>` with the key you retrieved in [Create a storage account](#).

```
sudo mount --types cifs //<storage-account-name>.file.core.windows.net/my-file-share /mnt/MyAzureFileShare -o vers=3.0,username=<storage-account-name>,password=<storage-account-key>,dir_mode=0777,file_mode=0777,serverino
```

You receive the `user@myVmPrivate:~$` prompt. The Azure file share successfully mounted to `/mnt/MyAzureFileShare`.

Confirm that the VM has no outbound connectivity to any other public IP addresses:

```
ping bing.com -c 4
```

You receive no replies, because the network security group associated to the *Private* subnet does not allow outbound access to public IP addresses other than the addresses assigned to the Azure Storage service.

Exit the SSH session to the *myVmPrivate* VM.

## Confirm access is denied to storage account

Use the following command to create an SSH session with the *myVmPublic* VM. Replace `<publicIpAddress>` with the public IP address of your *myVmPublic* VM:

```
ssh <publicIpAddress>
```

Create a directory for a mount point:

```
sudo mkdir /mnt/MyAzureFileShare
```

Attempt to mount the Azure file share to the directory you created. This article assumes you deployed the latest version of Ubuntu. If you are using earlier versions of Ubuntu, see [Mount on Linux](#) for additional instructions about mounting file shares. Before running the following command, replace `<storage-account-name>` with the account name and `<storage-account-key>` with the key you retrieved in [Create a storage account](#):

```
sudo mount --types cifs //<storage-account-name>.file.core.windows.net/my-file-share /mnt/MyAzureFileShare --options vers=3.0,username=<storage-account-name>,password=<storage-account-key>,dir_mode=0777,file_mode=0777,serverino
```

Access is denied, and you receive a `mount error(13): Permission denied` error, because the *myVmPublic* VM is deployed within the *Public* subnet. The *Public* subnet does not have a service endpoint enabled for Azure Storage, and the storage account only allows network access from the *Private* subnet, not the *Public* subnet.

Exit the SSH session to the *myVmPublic* VM.

From your computer, attempt to view the shares in your storage account with `az storage share list`. Replace `<account-name>` and `<account-key>` with the storage account name and key from [Create a storage account](#):

```
az storage share list \
--account-name <account-name> \
--account-key <account-key>
```

Access is denied and you receive a *This request is not authorized to perform this operation* error, because your computer is not in the *Private* subnet of the *MyVirtualNetwork* virtual network.

## Clean up resources

When no longer needed, use [az group delete](#) to remove the resource group and all of the resources it contains.

```
az group delete --name myResourceGroup --yes
```

## Next steps

In this article, you enabled a service endpoint for a virtual network subnet. You learned that service endpoints can be enabled for resources deployed with multiple Azure services. You created an Azure Storage account and limited network access to the storage account to only resources within a virtual network subnet. To learn more about service endpoints, see [Service endpoints overview](#) and [Manage subnets](#).

If you have multiple virtual networks in your account, you may want to connect two virtual networks together so the resources within each virtual network can communicate with each other. To learn how, see [Connect virtual networks](#).

# Create, change, or delete service endpoint policy using the Azure portal

11/2/2020 • 3 minutes to read • [Edit Online](#)

Service endpoint policies enable you to filter virtual network traffic to specific Azure resources, over service endpoints. If you're not familiar with service endpoint policies, see [service endpoint policies overview](#) to learn more.

In this tutorial, you learn how to:

- Create a service endpoint policy
- Create a service endpoint policy definition
- Create a virtual network with a subnet
- Associate a service endpoint policy to a subnet

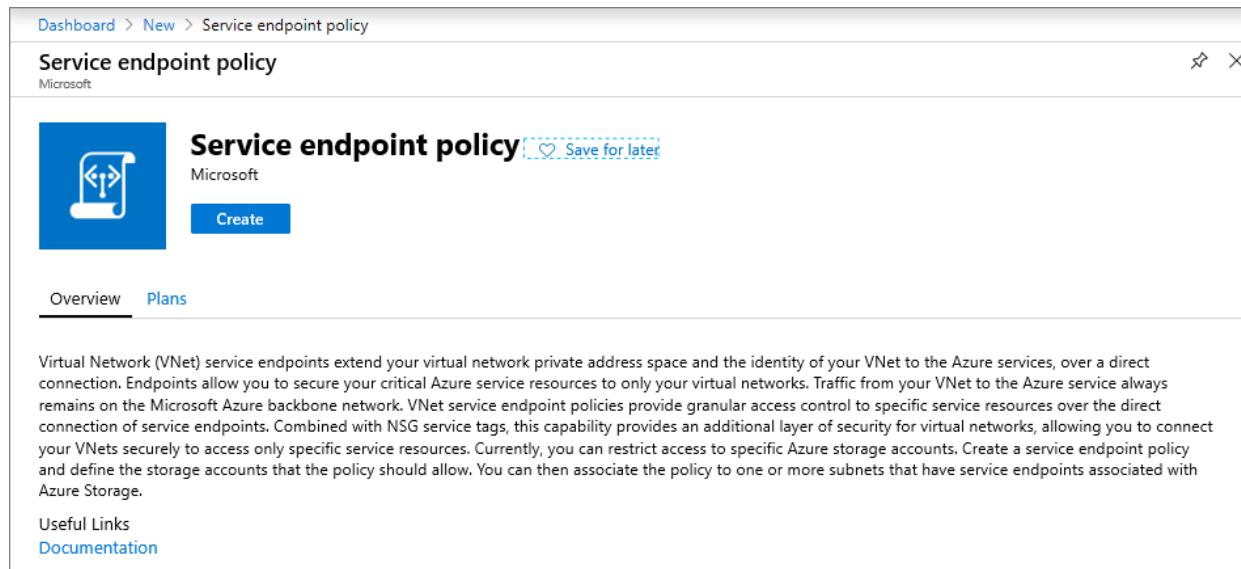
If you don't have an Azure subscription, create a [free account](#) before you begin.

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create a service endpoint policy

1. Select + **Create a resource** on the upper, left corner of the Azure portal.
2. In search pane, type "service endpoint policy" and select **Service endpoint policy** and then select **Create**.



Dashboard > New > Service endpoint policy

## Service endpoint policy

Microsoft

 **Service endpoint policy** Save for later

**Create**

[Overview](#) [Plans](#)

Virtual Network (VNet) service endpoints extend your virtual network private address space and the identity of your VNet to the Azure services, over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your VNet to the Azure service always remains on the Microsoft Azure backbone network. VNet service endpoint policies provide granular access control to specific service resources over the direct connection of service endpoints. Combined with NSG service tags, this capability provides an additional layer of security for virtual networks, allowing you to connect your VNets securely to access only specific service resources. Currently, you can restrict access to specific Azure storage accounts. Create a service endpoint policy and define the storage accounts that the policy should allow. You can then associate the policy to one or more subnets that have service endpoints associated with Azure Storage.

Useful Links  
[Documentation](#)

3. Enter, or select, the following information in **Basics**

- Subscription : Select your subscription for policy
- Resource group : Select **Create new** and enter *myResourceGroup*
- Name : *myEndpointPolicy*
- Location : Central US

Home > MySEP-RG > New > Service endpoint policy > Create a service endpoint policy

## Create a service endpoint policy

**Basics** Policy definitions Tags Review + create

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

**Project details**

Subscription \*

Resource group \*  MySEP-RG [Create new](#)

**Instance details**

Name \*  MySEP ✓

Location \*  (US) Central US

[Review + create](#) [Next : Policy definitions >](#) [Download a template for automation](#)

4. Select + Add under Resources and enter or select the following information in Add a resource pane

- Service : Only **Microsoft.Storage** is available with Service Endpoint Policies
- Scope : Select one out of **Single Account**, **All accounts in subscription** and **All accounts in resource group**
- Subscription : Select your subscription for storage account. Policy and storage accounts can be in different subscriptions.
- Resource group : Select your resource group. Required, if scope is set as, "All accounts in resource group" or "Single account".
- Resource : Select your Azure Storage resource under the selected Subscription or Resource Group
- Click on **Add** button at bottom to finish adding the resource

**Create a service endpoint policy**

**Basics** Policy definitions Tags Review + create

This policy will allow access only to the azure service resources listed

**Resources** + Add a resource

Service	Allowed Resources	Re
Add a resource to get started		

Service \* Microsoft.Storage

Scope \* Single account

Subscription \*

Resource group \* MySEP-RG

Resource \* myteststgacc123

Review + create < Previous Next : Tags > Add Close

- Add more resources by repeating the above steps as needed

5. Optional: Enter or select, the following information in **Tags**:

- Key : Select your key for the policy. Ex: Dept
- Value : Enter value pair for the key. Ex: Finance

6. Select **Review + Create**. Validate the information and Click **Create**. To make further edits, click **Previous**.

**Create a service endpoint policy**

Validation passed

**Basics** Policy definitions Tags Review + create

**Basics**

Subscription: NON-PROD - Testing - EAPv2  
Resource group: MySEP-RG  
Region: (US) Central US  
Name: MySEP

**Resources**

Microsoft.Storage: myteststgacc123 (Storage account)

For this policy to take effect, you will need to associate it to one or more subnets that have virtual network service endpoints. Please visit a virtual network in (US) Central US region and then select the subnets to which you would like to associate this policy.

Create < Previous Download a template for automation

## View endpoint policies

1. In the *All services* box in the portal, begin typing *service endpoint policies*. Select **Service Endpoint**

## Policies.

- Under Subscriptions, select your subscription and resource group, as shown in the following picture

The screenshot shows the 'Service endpoint policies' blade in the Azure portal. It displays a table with one row of data. The columns are: Name, Type, Resource group, Location, and Subscription. The single entry is 'MySEP' under 'Service endpoint policy' in the 'MySEP-RG' resource group, located in 'Central US' of the 'NON-PROD - Tesing - E...' subscription.

Name	Type	Resource group	Location	Subscription
MySEP	Service endpoint policy	MySEP-RG	Central US	NON-PROD - Tesing - E...

- Select the policy and click on **Policy Definitions** to view or add more policy definitions.

The screenshot shows the 'MySEP - Policy definitions' blade. On the left is a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Settings, Policy definitions, Associated subnets, Properties, Locks, Export template, Support + troubleshooting, and New support request. The 'Policy definitions' option is selected. The main area shows a table with columns: Service, Allowed Resources, Resource Group, Subscription name, and Subscription ID. One entry is listed: Microsoft.Storage with Microsoft.Storage as the service, myteststgacc123 (Sto...) as the allowed resource, MySEP-RG as the resource group, and MySEP-RG as the subscription.

Service	Allowed Resources	Resource Group	Subscription name	Subscription ID
Microsoft.Storage	myteststgacc123 (Sto...)	MySEP-RG	MySEP-RG	MySEP-RG

- Select **Associated subnets** to view the subnets the policy is associated. If no subnet is associated yet, follow the instructions in the next step.

The screenshot shows the 'MySEP - Associated subnets' blade. The left navigation menu includes Overview, Activity log, Access control (IAM), Tags, Settings, Policy definitions, and Associated subnets. The 'Associated subnets' option is selected. The main area contains a table with columns: Virtual network | Subnet, Resource group, and Service endpoints. A message at the top says 'Begin with associating subnet with this service endpoint policy.' There is a '+ Edit subnet association' button and a 'Refresh' button above the table.

Virtual network   Subnet	Resource group	Service endpoints
Begin with associating subnet with this service endpoint policy.		

- Associate a policy to a subnet

## WARNING

Ensure that all the resources accessed from the subnet are added to the policy definition before associating the policy to the given subnet. Once the policy is associated, only access to the *allow listed* resources will be allowed over service endpoints.

Also ensure that no managed Azure services exist in the subnet that is being associated to the service endpoint policy

- Before you can associate a policy to a subnet, you have to create a virtual network and subnet. Please refer to the [Create a Virtual Network](#) article for help with this.
- Once you have the virtual network and subnet are setup, you need to configure Virtual Network Service Endpoints for Azure Storage. On the Virtual Network blade, select **Service endpoints**, and in the next pane select **Microsoft.Storage** and under **Subnets** select the desired VNet or Subnet
- Now, you can either choose to select the Service Endpoint Policy from the drop-down in the above pane if you have already created Service Endpoint policies before configuring Service Endpoint for the Subnet as shown below

The screenshot shows the Azure portal interface. On the left, the navigation menu for 'myVNet - Service endpoints' is visible, with 'Service endpoints' selected. The main area displays a table with columns 'Service' and 'Subnet', showing 'No service endpoints.' A modal window titled 'Add service endpoints' is open on the right. In the 'Service' dropdown, 'Microsoft.Storage' is selected. Under 'Service endpoint policies', 'MySEP' is chosen. A note at the bottom of this section states: 'With service endpoints, the source IP address for service traffic from this subnet will switch from using public IPv4 addresses to using private IPv4 address. Existing IP firewall rules using Azure public IP addresses will stop working with this switch. Please ensure IP firewall rules allow for this switch before setting up service endpoints. You may also experience temporary interruption to service traffic from this subnet while configuring service endpoints.' At the bottom of the modal is a blue 'Add' button.

- OR if you are associating Service Endpoint policies after Service Endpoints are already configured, you can choose to associate the subnet from within the Service Endpoint Policy blade by navigating to the **Associated Subnets** pane as shown below

The screenshot shows the Azure portal interface for managing a Service Endpoint Policy (MySEP). On the left, there's a sidebar with navigation links like Overview, Activity log, Access control (IAM), Tags, Policy definitions, Associated subnets (which is selected), Properties, Locks, and Export template. The main area is titled 'Edit subnet association' for 'MySEP'. It shows a dropdown for 'Virtual network' set to 'myVnet'. Below it, a table lists subnets and their associated service endpoints. One subnet, 'mySubnet1', is checked and associated with 'Microsoft.Storage'. A note at the top right says: 'Select subnets from this virtual network to which the policy will be applied. Unselecting a subnet will disassociate the policy from that subnet. You can only apply this policy to subnets with service endpoints that support the services in this policy.' At the bottom is a blue 'Apply' button.

#### WARNING

Access to Azure Storage resources in all regions will be restricted as per Service Endpoint Policy from this subnet.

## Next steps

In this tutorial, you created a service endpoint policy and associated it to a subnet. To learn more about service endpoint policies, see [service endpoint policies overview](#).

# Manage data exfiltration to Azure Storage accounts with Virtual network service endpoint policies using Azure PowerShell

2/18/2021 • 9 minutes to read • [Edit Online](#)

Virtual network service endpoint policies enable you to apply access control on Azure Storage accounts from within a virtual network over service endpoints. This is a key to securing your workloads, managing what storage accounts are allowed and where data exfiltration is allowed. In this article, you learn how to:

- Create a virtual network.
- Add a subnet and enable service endpoint for Azure Storage.
- Create two Azure Storage accounts and allow network access to it from the subnet created above.
- Create a service endpoint policy to allow access only to one of the storage accounts.
- Deploy a virtual machine (VM) to the subnet.
- Confirm access to the allowed storage account from the subnet.
- Confirm access is denied to the non-allowed storage account from the subnet.

## NOTE

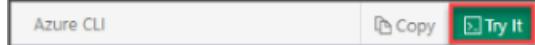
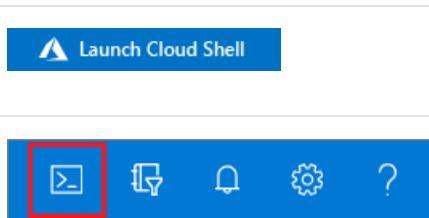
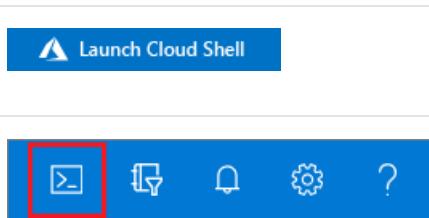
This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

## Create a virtual network

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup*.

```
New-AzResourceGroup `
-ResourceGroupName myResourceGroup `
-Location EastUS
```

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVirtualNetwork* with the address prefix *10.0.0.0/16*.

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

## Enable a service endpoint

Create a subnet in the virtual network. In this example, a subnet named *Private* is created with a service endpoint for *Microsoft.Storage*.

```
$subnetConfigPrivate = Add-AzVirtualNetworkSubnetConfig `
-Name Private `
-AddressPrefix 10.0.0.0/24 `
-VirtualNetwork $virtualNetwork `
-ServiceEndpoint Microsoft.Storage

$virtualNetwork | Set-AzVirtualNetwork
```

## Restrict network access for the subnet

Create network security group security rules with [New-AzNetworkSecurityRuleConfig](#). The following rule allows outbound access to the public IP addresses assigned to the Azure Storage service:

```
$rule1 = New-AzNetworkSecurityRuleConfig `
 -Name Allow-Storage-All `
 -Access Allow `
 -DestinationAddressPrefix Storage `
 -DestinationPortRange * `
 -Direction Outbound `
 -Priority 100 -Protocol * `
 -SourceAddressPrefix VirtualNetwork `
 -SourcePortRange *
```

The following rule denies access to all public IP addresses. The previous rule overrides this rule, due to its higher priority, which allows access to the public IP addresses of Azure Storage.

```
$rule2 = New-AzNetworkSecurityRuleConfig `
 -Name Deny-Internet-All `
 -Access Deny `
 -DestinationAddressPrefix Internet `
 -DestinationPortRange * `
 -Direction Outbound `
 -Priority 110 -Protocol * `
 -SourceAddressPrefix VirtualNetwork `
 -SourcePortRange *
```

The following rule allows Remote Desktop Protocol (RDP) traffic inbound to the subnet from anywhere. Remote desktop connections are allowed to the subnet, so that you can confirm network access to a resource in a later step.

```
$rule3 = New-AzNetworkSecurityRuleConfig `
 -Name Allow-RDP-All `
 -Access Allow `
 -DestinationAddressPrefix VirtualNetwork `
 -DestinationPortRange 3389 `
 -Direction Inbound `
 -Priority 120 `
 -Protocol * `
 -SourceAddressPrefix * `
 -SourcePortRange *
```

Create a network security group with [New-AzNetworkSecurityGroup](#). The following example creates a network security group named *myNsgPrivate*.

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName myResourceGroup `
 -Location EastUS `
 -Name myNsgPrivate `
 -SecurityRules $rule1,$rule2,$rule3
```

Associate the network security group to the *Private* subnet with [Set-AzVirtualNetworkSubnetConfig](#) and then write the subnet configuration to the virtual network. The following example associates the *myNsgPrivate* network security group to the *Private* subnet:

```
Set-AzVirtualNetworkSubnetConfig `
 -VirtualNetwork $VirtualNetwork `
 -Name Private `
 -AddressPrefix 10.0.0.0/24 `
 -ServiceEndpoint Microsoft.Storage `
 -NetworkSecurityGroup $nsg

$virtualNetwork | Set-AzVirtualNetwork
```

## Restrict network access to Azure Storage accounts

The steps necessary to restrict network access to resources created through Azure services enabled for service endpoints varies across services. See the documentation for individual services for specific steps for each service. The remainder of this article includes steps to restrict network access for an Azure Storage account, as an example.

### Create two storage accounts

Create an Azure storage account with [New-AzStorageAccount](#).

```
$storageAcctName1 = 'allowedaccount'

New-AzStorageAccount `
 -Location EastUS `
 -Name $storageAcctName1 `
 -ResourceGroupName myResourceGroup `
 -SkuName Standard_LRS `
 -Kind StorageV2
```

After the storage account is created, retrieve the key for the storage account into a variable with [Get-AzStorageAccountKey](#):

```
$storageAcctKey1 = (Get-AzStorageAccountKey -ResourceGroupName myResourceGroup -AccountName
$storageAcctName1).Value[0]
```

The key is used to create a file share in a later step. Enter `$storageAcctKey` and note the value, as you'll also need to manually enter it in a later step when you map the file share to a drive in a VM.

Now repeat the above steps to create a second storage account.

```
$storageAcctName2 = 'notallowedaccount'

New-AzStorageAccount `
 -Location EastUS `
 -Name $storageAcctName2 `
 -ResourceGroupName myResourceGroup `
 -SkuName Standard_LRS `
 -Kind StorageV2
```

Also retrieve the storage account key from this account for using later to create a file share.

```
$storageAcctKey2 = (Get-AzStorageAccountKey -ResourceGroupName myResourceGroup -AccountName
$storageAcctName2).Value[0]
```

### Create a file share in each of the storage account

Create a context for your storage account and key with [New-AzStorageContext](#). The context encapsulates the

storage account name and account key:

```
$storageContext1 = New-AzStorageContext $storageAcctName1 $storageAcctKey1
$storageContext2 = New-AzStorageContext $storageAcctName2 $storageAcctKey2
```

Create a file share with [New-AzStorageShare](#):

```
$share1 = New-AzStorageShare my-file-share -Context $storageContext1
$share2 = New-AzStorageShare my-file-share -Context $storageContext2
```

### Deny all network access to a storage accounts

By default, storage accounts accept network connections from clients in any network. To limit access to selected networks, change the default action to *Deny* with [Update-AzStorageAccountNetworkRuleSet](#). Once network access is denied, the storage account is not accessible from any network.

```
Update-AzStorageAccountNetworkRuleSet `
-ResourceGroupName myresourcegroup `
-Name $storageAcctName1 `
-DefaultAction Deny

Update-AzStorageAccountNetworkRuleSet `
-ResourceGroupName myresourcegroup `
-Name $storageAcctName2 `
-DefaultAction Deny
```

### Enable network access only from the VNet subnet

Retrieve the created virtual network with [Get-AzVirtualNetwork](#) and then retrieve the private subnet object into a variable with [Get-AzVirtualNetworkSubnetConfig](#):

```
$privateSubnet = Get-AzVirtualNetwork `
-ResourceGroupName myResourceGroup `
-Name myVirtualNetwork `
| Get-AzVirtualNetworkSubnetConfig -Name Private
```

Allow network access to the storage account from the *Private* subnet with [Add-AzStorageAccountNetworkRule](#).

```
Add-AzStorageAccountNetworkRule `
-ResourceGroupName myresourcegroup `
-Name $storageAcctName1 `
-VirtualNetworkResourceId $privateSubnet.Id

Add-AzStorageAccountNetworkRule `
-ResourceGroupName myresourcegroup `
-Name $storageAcctName2 `
-VirtualNetworkResourceId $privateSubnet.Id
```

## Apply policy to allow access to valid storage account

To make sure the users in the virtual network can only access the Azure Storage accounts that are safe and allowed, you can create a Service endpoint policy with the list of allowed storage accounts in the definition. This policy is then applied to the virtual network subnet which is connected to storage via service endpoints.

### Create a service endpoint policy

This section creates the policy definition with the list of allowed resources for access over service endpoint

Retrieve the resource ID for the first (allowed) storage account

```
$resourceId = (Get-AzStorageAccount -ResourceGroupName myresourcegroup -Name $storageAcctName1).id
```

Create the policy definition to allow the above resource

```
$policyDefinition = New-AzServiceEndpointPolicyDefinition -Name mypolicydefinition `
-Description "Service Endpoint Policy Definition" `
-Service "Microsoft.Storage" `
-ServiceResource $resourceId
```

Create the service endpoint policy using the policy definition created above

```
$sepolicy = New-AzServiceEndpointPolicy -ResourceGroupName myresourcegroup `
-Name mysepolicy -Location EastUS
-ServiceEndpointPolicyDefinition $policyDefinition
```

### Associate the service endpoint policy to the virtual network subnet

After creating the service endpoint policy, you'll associate it with the target subnet with the service endpoint configuration for Azure Storage.

```
Set-AzVirtualNetworkSubnetConfig -VirtualNetwork $VirtualNetwork `
-Name Private `
-AddressPrefix 10.0.0.0/24 `
-NetworkSecurityGroup $nsg `
-ServiceEndpoint Microsoft.Storage `
-ServiceEndpointPolicy $sepolicy

$virtualNetwork | Set-AzVirtualNetwork
```

## Validate access restriction to Azure Storage accounts

### Deploy the virtual machine

To test network access to a storage account, deploy a VM in the subnet.

Create a virtual machine in the *Private* subnet with [New-AzVM](#). 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-AzVm -ResourceGroupName myresourcegroup `
-Location "East US" `
-VirtualNetworkName myVirtualNetwork `
-SubnetName Private `
-Name "myVMPrivate" -AsJob
```

Output similar to the following example output is returned:

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

### Confirm access to the *allowed* storage account

Use [Get-AzPublicIpAddress](#) to return the public IP address of a VM. The following example returns the public IP address of the *myVmPrivate* VM:

```
Get-AzPublicIpAddress
-Name myVmPrivate
-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:

```
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.

On the *myVmPrivate* VM, map the Azure file share from allowed storage account to drive Z using PowerShell.

```
$acctKey = ConvertTo-SecureString -String $storageAcctKey1 -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList ("Azure\allowedaccount"),
$acctKey
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\allowedaccount.file.core.windows.net\my-file-share" -
Credential $credential
```

PowerShell returns output similar to the following example output:

Name	Used (GB)	Free (GB)	Provider	Root
Z	-----	-----	-----	\\allowedaccount.file.core.windows.net\my-f...

The Azure file share successfully mapped to the Z drive.

Close the remote desktop session to the *myVmPrivate* VM.

#### Confirm access is denied to **non-allowed** storage account

On the same *myVmPrivate* VM, attempt to map the Azure file share to drive X.

```
$acctKey = ConvertTo-SecureString -String $storageAcctKey1 -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\notallowedaccount",
$acctKey
New-PSDrive -Name X -PSProvider FileSystem -Root "\\notallowedaccount.file.core.windows.net\my-file-share" -
Credential $credential
```

Access to the share is denied, and you receive a `New-PSDrive : Access is denied` error. Access is denied because the storage account *notallowedaccount* is not in the allowed resources list in the service endpoint policy.

Close the remote desktop session to the *myVmPublic* VM.

## Clean up resources

When no longer needed, you can use [Remove-AzResourceGroup](#) to remove the resource group and all of the resources it contains:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

## Next steps

In this article, you applied a service endpoint policy over an Azure virtual network service endpoint to Azure Storage. You created Azure Storage accounts and limited network access to only certain storage accounts (and thus denied others) from a virtual network subnet. To learn more about service endpoint policies, see [Service endpoints policies overview](#).

# Manage data exfiltration to Azure Storage accounts with virtual network service endpoint policies using the Azure CLI

2/18/2021 • 8 minutes to read • [Edit Online](#)

Virtual network service endpoint policies enable you to apply access control on Azure Storage accounts from within a virtual network over service endpoints. This is a key to securing your workloads, managing what storage accounts are allowed and where data exfiltration is allowed. In this article, you learn how to:

- Create a virtual network and add a subnet.
- Enable service endpoint for Azure Storage.
- Create two Azure Storage accounts and allow network access to it from the subnet created above.
- Create a service endpoint policy to allow access only to one of the storage accounts.
- Deploy a virtual machine (VM) to the subnet.
- Confirm access to the allowed storage account from the subnet.
- Confirm access is denied to the non-allowed storage account from the subnet.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a virtual network

Before creating a virtual network, you have to create a resource group for the virtual network, and all other resources created in this article. Create a resource group with [az group create](#). The following example creates a resource group named *myResourceGroup* in the *eastus* location.

```
az group create \
 --name myResourceGroup \
 --location eastus
```

Create a virtual network with one subnet with [az network vnet create](#).

```
az network vnet create \
--name myVirtualNetwork \
--resource-group myResourceGroup \
--address-prefix 10.0.0.0/16 \
--subnet-name Private \
--subnet-prefix 10.0.0.0/24
```

## Enable a service endpoint

In this example, a service endpoint for *Microsoft.Storage* is created for the subnet *Private*.

```
az network vnet subnet create \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name Private \
--address-prefix 10.0.0.0/24 \
--service-endpoints Microsoft.Storage
```

## Restrict network access for a subnet

Create a network security group with [az network nsg create](#). The following example creates a network security group named *myNsgPrivate*.

```
az network nsg create \
--resource-group myResourceGroup \
--name myNsgPrivate
```

Associate the network security group to the *Private* subnet with [az network vnet subnet update](#). The following example associates the *myNsgPrivate* network security group to the *Private* subnet:

```
az network vnet subnet update \
--vnet-name myVirtualNetwork \
--name Private \
--resource-group myResourceGroup \
--network-security-group myNsgPrivate
```

Create security rules with [az network nsg rule create](#). The rule that follows allows outbound access to the public IP addresses assigned to the Azure Storage service:

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Allow-Storage-All \
--access Allow \
--protocol "*" \
--direction Outbound \
--priority 100 \
--source-address-prefix "VirtualNetwork" \
--source-port-range "*" \
--destination-address-prefix "Storage" \
--destination-port-range "*"
```

Each network security group contains several [default security rules](#). The rule that follows overrides a default security rule that allows outbound access to all public IP addresses. The `destination-address-prefix "Internet"`

option denies outbound access to all public IP addresses. The previous rule overrides this rule, due to its higher priority, which allows access to the public IP addresses of Azure Storage.

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Deny-Internet-All \
--access Deny \
--protocol "*" \
--direction Outbound \
--priority 110 \
--source-address-prefix "VirtualNetwork" \
--source-port-range "*" \
--destination-address-prefix "Internet" \
--destination-port-range "*"
```

The following rule allows SSH traffic inbound to the subnet from anywhere. The rule overrides a default security rule that denies all inbound traffic from the internet. SSH is allowed to the subnet so that connectivity can be tested in a later step.

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNsgPrivate \
--name Allow-SSH-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 120 \
--source-address-prefix "*" \
--source-port-range "*" \
--destination-address-prefix "VirtualNetwork" \
--destination-port-range "22"
```

## Restrict network access to Azure Storage accounts

This section lists steps to restrict network access for an Azure Storage account from the given subnet in a Virtual network via service endpoint.

### Create a storage account

Create two Azure storage accounts with [az storage account create](#).

```
storageAcctName1="allowedstorageacc"

az storage account create \
--name $storageAcctName1 \
--resource-group myResourceGroup \
--sku Standard_LRS \
--kind StorageV2

storageAcctName2="notallowedstorageacc"

az storage account create \
--name $storageAcctName2 \
--resource-group myResourceGroup \
--sku Standard_LRS \
--kind StorageV2
```

After the storage accounts are created, retrieve the connection string for the storage accounts into a variable with [az storage account show-connection-string](#). The connection string is used to create a file share in a later

step.

```
saConnectionString1=$(az storage account show-connection-string \
--name $storageAcctName1 \
--resource-group myResourceGroup \
--query 'connectionString' \
--out tsv)

saConnectionString2=$(az storage account show-connection-string \
--name $storageAcctName2 \
--resource-group myResourceGroup \
--query 'connectionString' \
--out tsv)
```

View the contents of the variable and note the value for **AccountKey** returned in the output, because it's used in a later step.

```
echo $saConnectionString1

echo $saConnectionString2
```

### Create a file share in the storage account

Create a file share in the storage account with [az storage share create](#). In a later step, this file share is mounted to confirm network access to it.

```
az storage share create \
--name my-file-share \
--quota 2048 \
--connection-string $saConnectionString1 > /dev/null

az storage share create \
--name my-file-share \
--quota 2048 \
--connection-string $saConnectionString2 > /dev/null
```

### Deny all network access to the storage account

By default, storage accounts accept network connections from clients in any network. To limit access to selected networks, change the default action to *Deny* with [az storage account update](#). Once network access is denied, the storage account is not accessible from any network.

```
az storage account update \
--name $storageAcctName1 \
--resource-group myResourceGroup \
--default-action Deny

az storage account update \
--name $storageAcctName2 \
--resource-group myResourceGroup \
--default-action Deny
```

### Enable network access from virtual network subnet

Allow network access to the storage account from the *Private* subnet with [az storage account network-rule add](#).

```
az storage account network-rule add \
--resource-group myResourceGroup \
--account-name $storageAcctName1 \
--vnet-name myVirtualNetwork \
--subnet Private

az storage account network-rule add \
--resource-group myResourceGroup \
--account-name $storageAcctName2 \
--vnet-name myVirtualNetwork \
--subnet Private
```

## Apply policy to allow access to valid storage account

Azure Service Endpoint policies are only available for Azure Storage. So, we'll be enabling Service Endpoint for *Microsoft.Storage* on this subnet for this example setup.

Service endpoint policies are applied over service endpoints. We will start by creating a service endpoint policy. We will then create the policy definitions under this policy for Azure Storage accounts to be approved for this subnet

Create a service endpoint policy

```
az network service-endpoint policy create \
--resource-group myResourceGroup \
--name mysepolicy \
--location eastus
```

Save the resource URI for the allowed storage account in a variable. Before executing the command below, replace <your-subscription-id> with actual value of your subscription ID.

```
$serviceResourceId="/subscriptions/<your-subscription-
id>/resourceGroups/myResourceGroup/providers/Microsoft.Storage/storageAccounts/allowedstorageacc"
```

Create & add a policy definition for allowing the above Azure Storage account to the service endpoint policy

```
az network service-endpoint policy-definition create \
--resource-group myResourceGroup \
--policy-name mysepolicy \
--name mypolicydefinition \
--service "Microsoft.Storage" \
--service-resources $serviceResourceId
```

And update the virtual network subnet to associate with it the service endpoint policy created in the previous step

```
az network vnet subnet update \
--vnet-name myVirtualNetwork \
--resource-group myResourceGroup \
--name Private \
--service-endpoints Microsoft.Storage \
--service-endpoint-policy mysepolicy
```

## Validate access restriction to Azure Storage accounts

## Create the virtual machine

To test network access to a storage account, deploy a VM to the subnet.

Create a VM in the *Private* subnet with [az vm create](#). If SSH keys do not already exist in a default key location, the command creates them. To use a specific set of keys, use the `--ssh-key-value` option.

```
az vm create \
--resource-group myResourceGroup \
--name myVmPrivate \
--image UbuntuLTS \
--vnet-name myVirtualNetwork \
--subnet Private \
--generate-ssh-keys
```

The VM takes a few minutes to create. After creation, take note of the **publicIpAddress** in the output returned. This address is used to access the VM from the internet in a later step.

## Confirm access to storage account

SSH into the *myVmPrivate* VM. Replace `<publicIpAddress>` with the public IP address of your *myVmPrivate* VM.

```
ssh <publicIpAddress>
```

Create a folder for a mount point:

```
sudo mkdir /mnt/MyAzureFileShare1
```

Mount the Azure file share to the directory you created. Before executing the command below, replace `<storage-account-key>` with value of *AccountKey* from `$saConnectionString1`.

```
sudo mount --types cifs //allowedstorageacc.file.core.windows.net/my-file-share /mnt/MyAzureFileShare1 --
options vers=3.0,username=allowedstorageacc,password=<storage-account-
key>,dir_mode=0777,file_mode=0777,serverino
```

You receive the `user@myVmPrivate:~$` prompt. The Azure file share successfully mounted to `/mnt/MyAzureFileShare`.

## Confirm access is denied to storage account

From the same VM *myVmPrivate*, create a directory for a mount point:

```
sudo mkdir /mnt/MyAzureFileShare2
```

Attempt to mount the Azure file share from storage account *notallowedstorageacc* to the directory you created. This article assumes you deployed the latest version of Ubuntu. If you are using earlier versions of Ubuntu, see [Mount on Linux](#) for additional instructions about mounting file shares.

Before executing the command below, replace `<storage-account-key>` with value of *AccountKey* from `$saConnectionString2`.

```
sudo mount --types cifs //notallowedstorageacc.file.core.windows.net/my-file-share /mnt/MyAzureFileShare2 --
options vers=3.0,username=notallowedstorageacc,password=<storage-account-
key>,dir_mode=0777,file_mode=0777,serverino
```

Access is denied, and you receive a `mount error(13): Permission denied` error, because this storage account is

not in the allow list of the service endpoint policy we applied to the subnet.

Exit the SSH session to the *myVmPublic* VM.

## Clean up resources

When no longer needed, use [az group delete](#) to remove the resource group and all of the resources it contains.

```
az group delete --name myResourceGroup --yes
```

## Next steps

In this article, you applied a service endpoint policy over an Azure virtual network service endpoint to Azure Storage. You created Azure Storage accounts and limited network access to only certain storage accounts (and thus denied others) from a virtual network subnet. To learn more about service endpoint policies, see [Service endpoints policies overview](#).

# Quickstart: Create and configure Azure DDoS Protection Standard

2/18/2021 • 4 minutes to read • [Edit Online](#)

Get started with Azure DDoS Protection Standard by using the Azure portal.

A DDoS protection plan defines a set of virtual networks that have DDoS protection standard enabled, across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions to the same plan.

In this quickstart, you'll create a DDoS protection plan and link it to a virtual network.

## Prerequisites

- If you don't have an Azure subscription, create a [free account](#) before you begin.
- Sign in to the Azure portal at <https://portal.azure.com>. Ensure that your account is assigned to the [network contributor](#) role or to a [custom role](#) that is assigned the appropriate actions listed in the how-to guide on [Permissions](#).

## Create a DDoS protection plan

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Search the term **DDoS**. When **DDoS protection plan** appears in the search results, select it.
3. Select **Create**.
4. Enter or select the following values, then select **Create**:

SETTING	VALUE
Name	Enter <i>MyDdosProtectionPlan</i> .
Subscription	Select your subscription.
Resource group	Select <b>Create new</b> and enter <i>MyResourceGroup</i> .
Location	Enter <i>East US</i> .

## Enable DDoS protection for a virtual network

### Enable DDoS protection for a new virtual network

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter or select the following values, accept the remaining defaults, and then select **Create**:

SETTING	VALUE
Name	Enter <i>MyVnet</i> .
Subscription	Select your subscription.
Resource group	Select <b>Use existing</b> , and then select <b>MyResourceGroup</b>
Location	Enter <i>East US</i>
DDoS Protection Standard	Select <b>Enable</b> . The plan you select can be in the same, or different subscription than the virtual network, but both subscriptions must be associated to the same Azure Active Directory tenant.

You cannot move a virtual network to another resource group or subscription when DDoS Standard is enabled for the virtual network. If you need to move a virtual network with DDoS Standard enabled, disable DDoS Standard first, move the virtual network, and then enable DDoS standard. After the move, the auto-tuned policy thresholds for all the protected public IP addresses in the virtual network are reset.

#### Enable DDoS protection for an existing virtual network

1. Create a DDoS protection plan by completing the steps in [Create a DDoS protection plan](#), if you don't have an existing DDoS protection plan.
2. Select **Create a resource** in the upper left corner of the Azure portal.
3. Enter the name of the virtual network that you want to enable DDoS Protection Standard for in the **Search resources, services, and docs** box at the top of the portal. When the name of the virtual network appears in the search results, select it.
4. Select **DDoS protection**, under **SETTINGS**.
5. Select **Standard**. Under **DDoS protection plan**, select an existing DDoS protection plan, or the plan you created in step 1, and then select **Save**. The plan you select can be in the same, or different subscription than the virtual network, but both subscriptions must be associated to the same Azure Active Directory tenant.

#### Enable DDoS protection for all virtual networks

This [policy](#) will detect any virtual networks in a defined scope that do not have DDoS Protection Standard enabled, then optionally create a remediation task that will create the association to protect the VNet. For detailed step-by-step instructions on how to deploy this policy, see <https://aka.ms/ddosvnetpolicy-techcommunity>.

## Validate and test

First, check the details of your DDoS protection plan:

1. Select **All services** on the top, left of the portal.
2. Enter **DDoS** in the **Filter** box. When **DDoS protection plans** appear in the results, select it.
3. Select your DDoS protection plan from the list.

The *MyVnet* virtual network should be listed.

#### View protected resources

Under **Protected resources**, you can view your protected virtual networks and public IP addresses, or add more virtual networks to your DDoS protection plan:

The screenshot shows the Azure portal interface for managing DDoS protection plans. On the left, a sidebar navigation includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Protected resources' (which is selected), 'Properties', 'Locks', 'Monitoring', 'Alerts', 'Metrics', 'Automation', and 'Support + troubleshooting'. The main content area is titled 'ddosplanav | Protected resources' and shows a table of protected resources. The table has columns for 'Public IP address', 'Virtual network', 'Application gateway', and 'Subscription'. The data is as follows:

Public IP address	Virtual network	Application gateway	Subscription
myAppGatewayDDoSCTest	DDoSAndCompromiseProtectiontest	myAppGatewayDDoSCTS	Azure DDoS Protection Demo Sub
newddosappgwip	NewDDoSvNET	NewDDoSAppGW	Azure DDoS Protection Demo Sub
ddosbillingsappgw-ip	ddos-test	ddos-billingappgw	Azure DDoS Protection Demo Sub
appgw-wcu	appgw-vnet-wcu	appgw-wcu	Azure DDoS Protection Demo Sub

At the bottom of the table, there are buttons for 'Previous', 'Page 1 of 1', and 'Next'.

## Clean up resources

You can keep your resources for the next tutorial. If no longer needed, delete the *MyResourceGroup* resource group. When you delete the resource group, you also delete the DDoS protection plan and all its related resources. If you don't intend to use this DDoS protection plan, you should remove resources to avoid unnecessary charges.

### WARNING

This action is irreversible.

1. In the Azure portal, search for and select **Resource groups**, or select **Resource groups** from the Azure portal menu.
2. Filter or scroll down to find the *MyResourceGroup* resource group.
3. Select the resource group, then select **Delete resource group**.
4. Type the resource group name to verify, and then select **Delete**.

To disable DDoS protection for a virtual network:

1. Enter the name of the virtual network you want to disable DDoS protection standard for in the **Search resources, services, and docs** box at the top of the portal. When the name of the virtual network appears in the search results, select it.
2. Select **Under DDoS Protection Standard**, select **Disable**.

If you want to delete a DDoS protection plan, you must first dissociate all virtual networks from it.

## Next steps

To learn how to view and configure telemetry for your DDoS protection plan, continue to the tutorials.

[View and configure DDoS protection telemetry](#)

# Partnering with Azure DDoS Protection Standard

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes partnering opportunities enabled by the Azure DDoS Protection Standard. This article is designed to help product managers and business development roles understand the investment paths and provide insight into the partnering value propositions.

## Background

Distributed denial of service (DDoS) attacks are one of the top availability and security concerns voiced by customers moving their applications to the cloud. With extortion and hacktivism being the common motivations behind DDoS attacks, they have been consistently increasing in type, scale, and frequency of occurrence as they are relatively easy and cheap to launch.

Azure DDoS Protection provides countermeasures against the most sophisticated DDoS threats, leveraging the global scale of Azure networking. The service provides enhanced DDoS mitigation capabilities for applications and resources deployed in virtual networks.

Technology partners can protect their customers' resources natively with Azure DDoS Protection Standard to address the availability and reliability concerns due to DDoS attacks.

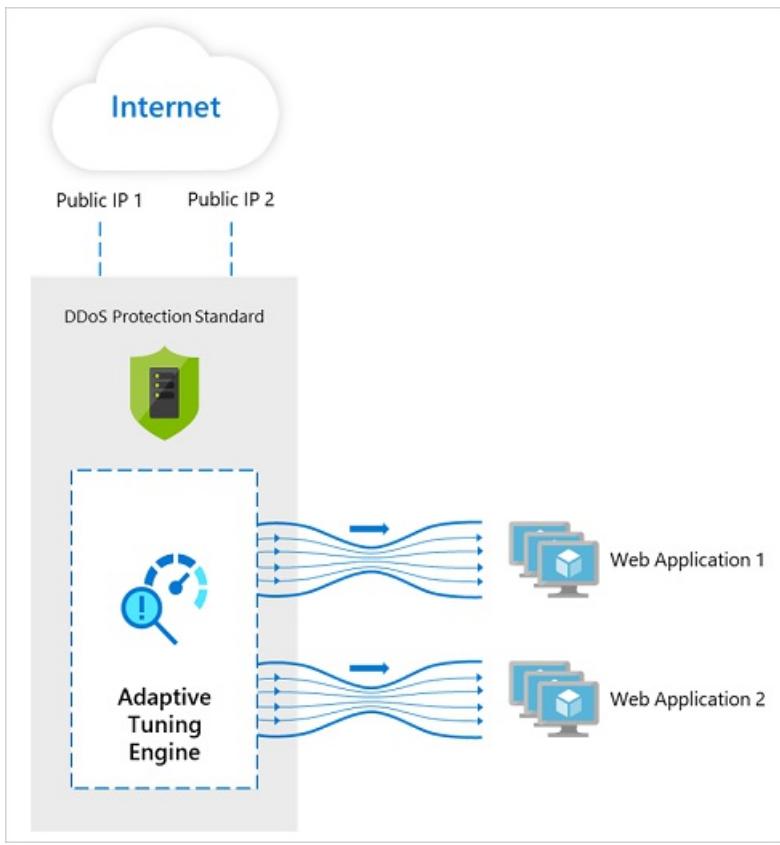
## Introduction to Azure DDoS Protection Standard

Azure DDoS Protection Standard provides enhanced DDoS mitigation capabilities against Layer 3 and Layer 4 DDoS attacks. The following are the key features of DDoS Protection Standard service.

### Adaptive real-time tuning

For every protected application, Azure DDoS Protection Standard automatically tunes the DDoS mitigation policy thresholds based on the application's traffic profile patterns. The service accomplishes this customization by using two insights:

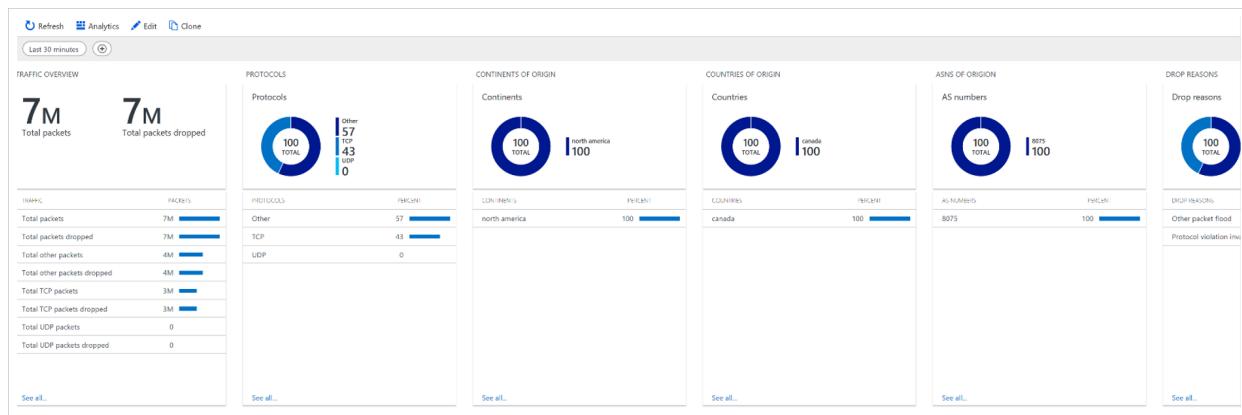
- Automatic learning of per-customer (per-IP) traffic patterns for Layer 3 and 4.
- Minimizing false positives, considering that the scale of Azure allows it to absorb a significant amount of traffic.



## Attack analytics, telemetry, monitoring, and alerting

Azure DDoS Protection identifies and mitigates DDoS attacks without any user intervention.

- If the protected resource is in the subscription covered under Azure Security Center, DDoS Protection Standard automatically sends an alert to Security Center whenever a DDoS attack is detected and mitigated against the protected application.
- Alternatively, to get notified when there's an active mitigation for a protected public IP, you can [configure an alert](#) on the metric Under DDoS attack or not.
- You can additionally choose to create alerts for the other DDoS metrics and [configure attack telemetry](#) to understand the scale of the attack, traffic being dropped, attack vectors, top contributors, and other details.



## DDoS rapid response (DRR)

DDoS Protection Standard customers have access to [Rapid Response team](#) during an active attack. DRR can help with attack investigation during an attack as well as post-attack analysis.

## SLA guarantee and cost protection

DDoS Protection Standard service is covered by a 99.99% SLA, and cost protection provides resource credits for scale out during a documented attack. For more information, see [SLA for Azure DDoS Protection](#).

# Featured partner scenarios

The following are key benefits you can derive by integrating with the Azure DDoS Protection Standard:

- Partners' offered services (load balancer, web application firewall, firewall, etc.) to their customers are automatically protected (white labeled) by Azure DDoS Protection Standard in the back end.
- Partners have access to Azure DDoS Protection Standard attack analytics and telemetry that they can integrate with their own products, offering a unified customer experience.
- Partners have access to DDoS rapid response support even in the absence of Azure rapid response, for DDoS related issues.
- Partners' protected applications are backed by a DDoS SLA guarantee and cost protection in the event of DDoS attacks.

## Technical integration overview

Azure DDoS Protection Standard partnering opportunities are made available via Azure portal, APIs, and CLI/PS.

### Integrate with DDoS Protection Standard

The following steps are required for partners to configure integration with Azure DDoS Protection Standard:

- Create a DDoS Protection Plan in your desired (partner) subscription. For step-by-step instructions, see [Create a DDoS Standard Protection plan](#).

**NOTE**

Only 1 DDoS Protection Plan needs to be created for a given tenant.
- Deploy a service with public endpoint in your (partner) subscriptions, such as load balancer, firewalls, and web application firewall.
- Enable Azure DDoS Protection Standard on the virtual network of the service that has public endpoints using DDoS Protection Plan created in the first step. For step-by-step instructions, see [Enable DDoS Standard Protection plan](#)

### IMPORTANT

After Azure DDoS Protection Standard is enabled on a virtual network, all public IPs within that virtual network are automatically protected. The origin of these public IPs can be either within Azure (client subscription) or outside of Azure.

- Optionally, integrate Azure DDoS Protection Standard telemetry and attack analytics in your application-specific customer-facing dashboard. For more information about using telemetry, see [View and configure DDoS protection telemetry](#).

### Onboarding guides and technical documentation

- [Azure DDoS Protection product page](#)
- [Azure DDoS Protection documentation](#)
- [Azure DDoS Protection API reference](#)
- [Azure virtual network API reference](#)

### Get help

- If you have questions about application, service, or product integrations with Azure DDoS Protection Standard, reach out to the [Azure security community](#).
- Follow discussions on [Stack Overflow](#).

## **Get to market**

- The primary program for partnering with Microsoft is the [Microsoft Partner Network](#). – Microsoft Graph Security integrations fall into the [MPN Independent Software Vendor \(ISV\)](#) track.
- [Microsoft Intelligent Security Association](#) is the program specifically for Microsoft Security Partners to help enrich your security products and improve customer discoverability of your integrations to Microsoft Security products.

## **Next steps**

View existing partner integrations:

- [Barracuda WAF-as-a-service](#)
- [Azure Cloud WAF from Radware](#)

# Create, change, or delete a network security group

2/18/2021 • 15 minutes to read • [Edit Online](#)

Security rules in network security groups enable you to filter the type of network traffic that can flow in and out of virtual network subnets and network interfaces. To learn more about network security groups, see [Network security group overview](#). Next, complete the [Filter network traffic](#) tutorial to gain some experience with network security groups.

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

If you don't have one, set up an Azure account with an active subscription. [Create an account for free](#). Complete one of these tasks before starting the remainder of this article:

- **Portal users:** Sign in to the [Azure portal](#) with your Azure account.
- **PowerShell users:** Either run the commands in the [Azure Cloud Shell](#), or run PowerShell from your computer. 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. In the Azure Cloud Shell browser tab, find the **Select environment** dropdown list, then pick **PowerShell** if it isn't already selected.

If you're running PowerShell locally, use Azure PowerShell module version 1.0.0 or later. Run

```
Get-Module -ListAvailable Az.Network
```

 to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). Run `Connect-AzAccount` to create a connection with Azure.

- **Azure Command-line interface (CLI) users:** Either run the commands in the [Azure Cloud Shell](#), or run the CLI from your computer. Use Azure CLI version 2.0.28 or later if you're running the Azure CLI locally. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). Run `az login` to create a connection with Azure.

The account you log into, or connect to Azure with must be assigned to the [Network contributor role](#) or to a [Custom role](#) that's assigned the appropriate actions listed in [Permissions](#).

## Work with network security groups

You can create, [view all](#), [view details of](#), [change](#), and [delete](#) a network security group. You can also [associate](#) or [dissociate](#) a network security group from a network interface or subnet.

### Create a network security group

There's a limit to how many network security groups you can create for each Azure location and subscription. To learn more, see [Azure subscription and service limits, quotas, and constraints](#).

1. On the [Azure portal](#) menu or from the Home page, select **Create a resource**.
2. Select **Networking**, then select **Network security group**.

3. In the **Create network security group** page, under the **Basics** tab, set values for the following settings:

SETTING	ACTION
Subscription	Choose your subscription.
Resource group	Choose an existing resource group, or select <b>Create new</b> to create a new resource group.
Name	Enter a unique text string within a resource group.
Region	Choose the location you want.

4. Select **Review + create**.

5. After you see the **Validation passed** message, select **Create**.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg create</a>
PowerShell	<a href="#">New-AzNetworkSecurityGroup</a>

#### View all network security groups

Go to the [Azure portal](#) to view your network security groups. Search for and select **Network security groups**. The list of network security groups appears for your subscription.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg list</a>
PowerShell	<a href="#">Get-AzNetworkSecurityGroup</a>

#### View details of a network security group

1. Go to the [Azure portal](#) to view your network security groups. Search for and select **Network security groups**.
2. Select the name of your network security group.

In the menu bar of the network security group, under **Settings**, you can view the **Inbound security rules**, **Outbound security rules**, **Network interfaces**, and **Subnets** that the network security group is associated to.

Under **Monitoring**, you can enable or disable **Diagnostic settings**. Under **Support + troubleshooting**, you can view **Effective security rules**. To learn more, see [Diagnostic logging for a network security group](#) and [Diagnose a VM network traffic filter problem](#).

To learn more about the common Azure settings listed, see the following articles:

- [Activity log](#)
- [Access control \(IAM\)](#)
- [Tags](#)

- [Locks](#)
- [Automation script](#)

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg show</a>
PowerShell	<a href="#">Get-AzNetworkSecurityGroup</a>

### Change a network security group

1. Go to the [Azure portal](#) to view your network security groups. Search for and select **Network security groups**.

2. Select the name of the network security group you want to change.

The most common changes are to [add a security rule](#), [remove a rule](#), and [associate or dissociate a network security group to or from a subnet or network interface](#).

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg update</a>
PowerShell	<a href="#">Set-AzNetworkSecurityGroup</a>

### Associate or dissociate a network security group to or from a subnet or network interface

To associate a network security group to, or dissociate a network security group from a network interface, see [Associate a network security group to, or dissociate a network security group from a network interface](#). To associate a network security group to, or dissociate a network security group from a subnet, see [Change subnet settings](#).

### Delete a network security group

If a network security group is associated to any subnets or network interfaces, it can't be deleted. Dissociate a network security group from all subnets and network interfaces before attempting to delete it.

1. Go to the [Azure portal](#) to view your network security groups. Search for and select **Network security groups**.
2. Select the name of the network security group you want to delete.
3. In the network security group's toolbar, select **Delete**. Then select **Yes** in the confirmation dialog box.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg delete</a>
PowerShell	<a href="#">Remove-AzNetworkSecurityGroup</a>

## Work with security rules

A network security group contains zero or more security rules. You can create, [view all](#), [view details of](#), [change](#), and [delete](#) a security rule.

## Create a security rule

There's a limit to how many rules per network security group you can create for each Azure location and subscription. To learn more, see [Azure subscription and service limits, quotas, and constraints](#).

1. Go to the [Azure portal](#) to view your network security groups. Search for and select **Network security groups**.
2. Select the name of the network security group you want to add a security rule to.
3. In the network security group's menu bar, choose **Inbound security rules** or **Outbound security rules**.

Several existing rules are listed, including some you may not have added. When you create a network security group, several default security rules are created in it. To learn more, see [default security rules](#). You can't delete default security rules, but you can override them with rules that have a higher priority.

4. Select **Add**. Select or add values for the following settings, and then select **OK**:

SETTING	VALUE	DETAILS
Source	<p>One of:</p> <ul style="list-style-type: none"><li>• Any</li><li>• IP Addresses</li><li>• Service Tag (inbound security rule) or VirtualNetwork (outbound security rule)</li><li>• Application security group</li></ul>	<p>If you choose <b>IP Addresses</b>, you must also specify <b>Source IP addresses/CIDR ranges</b>.</p> <p>If you choose <b>Service Tag</b>, you may also pick a <b>Source service tag</b>.</p> <p>If you choose <b>Application security group</b>, you must also pick an existing application security group. If you choose <b>Application security group</b> for both <b>Source</b> and <b>Destination</b>, the network interfaces within both application security groups must be in the same virtual network.</p>

Setting	Value	Details
Source IP addresses/CIDR ranges	A comma-delimited list of IP addresses and Classless Interdomain Routing (CIDR) ranges	<p>This setting appears if you change <b>Source</b> to <b>IP Addresses</b>. You must specify a single value or comma-separated list of multiple values. An example of multiple values is <code>10.0.0.0/16, 192.188.1.1</code>.</p> <p>There are limits to the number of values you can specify. For more details, see <a href="#">Azure limits</a>.</p> <p>If the IP address you specify is assigned to an Azure VM, specify its private IP address, not its public IP address. Azure processes security rules after it translates the public IP address to a private IP address for inbound security rules, but before it translates a private IP address to a public IP address for outbound rules. To learn more about public and private IP addresses in Azure, see <a href="#">IP address types</a>.</p>
Source service tag	A service tag from the dropdown list	This optional setting appears if you set <b>Source</b> to <b>Service Tag</b> for an inbound security rule. A service tag is a predefined identifier for a category of IP addresses. To learn more about available service tags, and what each tag represents, see <a href="#">Service tags</a> .
Source application security group	An existing application security group	This setting appears if you set <b>Source</b> to <b>Application security group</b> . Select an application security group that exists in the same region as the network interface. Learn how to <a href="#">create an application security group</a> .
Source port ranges	<p>One of:</p> <ul style="list-style-type: none"> <li>• A single port, such as <code>80</code></li> <li>• A range of ports, such as <code>1024-65535</code></li> <li>• A comma-separated list of single ports and/or port ranges, such as <code>80, 1024-65535</code></li> <li>• An asterisk (<code>*</code>) to allow traffic on any port</li> </ul>	This setting specifies the ports on which the rule allows or denies traffic. There are limits to the number of ports you can specify. For more details, see <a href="#">Azure limits</a> .

Setting	Value	Details
Destination	<p>One of:</p> <ul style="list-style-type: none"> <li>• Any</li> <li>• IP Addresses</li> <li>• Service Tag (outbound security rule) or VirtualNetwork (inbound security rule)</li> <li>• Application security group</li> </ul>	<p>If you choose <b>IP addresses</b>, then also specify <b>Destination IP addresses/CIDR ranges</b>.</p> <p>If you choose <b>VirtualNetwork</b>, traffic is allowed to all IP addresses within the virtual network's address space. <b>VirtualNetwork</b> is a service tag.</p> <p>If you select <b>Application security group</b>, you must then select an existing application security group. Learn how to <a href="#">create an application security group</a>.</p>
Destination IP addresses/CIDR ranges	A comma-delimited list of IP addresses and CIDR ranges	<p>This setting appears if you change <b>Destination</b> to <b>IP Addresses</b>. Similar to <b>Source and Source IP addresses/CIDR ranges</b>, you can specify single or multiple addresses or ranges. There are limits to the number you can specify. For more details, see <a href="#">Azure limits</a>.</p> <p>If the IP address you specify is assigned to an Azure VM, ensure that you specify its private IP, not its public IP address. Azure processes security rules after it translates the public IP address to a private IP address for inbound security rules, but before Azure translates a private IP address to a public IP address for outbound rules. To learn more about public and private IP addresses in Azure, see <a href="#">IP address types</a>.</p>
Destination service tag	A service tag from the dropdown list	This optional setting appears if you change <b>Destination</b> to <b>Service Tag</b> for an outbound security rule. A service tag is a predefined identifier for a category of IP addresses. To learn more about available service tags, and what each tag represents, see <a href="#">Service tags</a> .

Setting	Value	Details
Destination application security group	An existing application security group	This setting appears if you set <b>Destination to Application security group</b> . Select an application security group that exists in the same region as the network interface. Learn how to <a href="#">create an application security group</a> .
Destination port ranges	<p>One of:</p> <ul style="list-style-type: none"> <li>• A single port, such as <code>80</code></li> <li>• A range of ports, such as <code>1024-65535</code></li> <li>• A comma-separated list of single ports and/or port ranges, such as <code>80, 1024-65535</code></li> <li>• An asterisk (<code>*</code>) to allow traffic on any port</li> </ul>	As with <b>Source port ranges</b> , you can specify single or multiple ports and ranges. There are limits to the number you can specify. For more details, see <a href="#">Azure limits</a> .
Protocol	Any, TCP, UDP, or ICMP	You may restrict the rule to the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP). The default is for the rule to apply to all protocols.
Action	Allow or Deny	This setting specifies whether this rule allows or denies access for the supplied source and destination configuration.
Priority	A value between 100 and 4096 that's unique for all security rules within the network security group	Azure processes security rules in priority order. The lower the number, the higher the priority. We recommend that you leave a gap between priority numbers when you create rules, such as 100, 200, and 300. Leaving gaps makes it easier to add rules in the future, so that you can give them higher or lower priority than existing rules.
Name	A unique name for the rule within the network security group	The name can be up to 80 characters. It must begin with a letter or number, and it must end with a letter, number, or underscore. The name may contain only letters, numbers, underscores, periods, or hyphens.
Description	A text description	You may optionally specify a text description for the security rule. The description cannot be longer than 140 characters.

## Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg rule create</a>
PowerShell	<a href="#">New-AzNetworkSecurityRuleConfig</a>

## View all security rules

A network security group contains zero or more rules. To learn more about the information listed when viewing rules, see [Network security group overview](#).

1. Go to the [Azure portal](#) to view the rules of a network security group. Search for and select **Network security groups**.
2. Select the name of the network security group that you want to view the rules for.
3. In the network security group's menu bar, choose **Inbound security rules** or **Outbound security rules**.

The list contains any rules you've created and the network security group's [default security rules](#).

### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg rule list</a>
PowerShell	<a href="#">Get-AzNetworkSecurityRuleConfig</a>

## View details of a security rule

1. Go to the [Azure portal](#) to view the rules of a network security group. Search for and select **Network security groups**.
2. Select the name of the network security group that you want to view the details of a rule for.
3. In the network security group's menu bar, choose **Inbound security rules** or **Outbound security rules**.
4. Select the rule you want to view details for. For an explanation of all settings, see [Security rule settings](#).

### NOTE

This procedure only applies to a custom security rule. It doesn't work if you choose a default security rule.

### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg rule show</a>
PowerShell	<a href="#">Get-AzNetworkSecurityRuleConfig</a>

## Change a security rule

1. Complete the steps in [View details of a security rule](#).
2. Change the settings as needed, and then select **Save**. For an explanation of all settings, see [Security rule settings](#).

**NOTE**

This procedure only applies to a custom security rule. You aren't allowed to change a default security rule.

**Commands**

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg rule update</a>
PowerShell	<a href="#">Set-AzNetworkSecurityRuleConfig</a>

**Delete a security rule**

1. Complete the steps in [View details of a security rule](#).
2. Select **Delete**, and then select **Yes**.

**NOTE**

This procedure only applies to a custom security rule. You aren't allowed to delete a default security rule.

**Commands**

TOOL	COMMAND
Azure CLI	<a href="#">az network nsg rule delete</a>
PowerShell	<a href="#">Remove-AzNetworkSecurityRuleConfig</a>

## Work with application security groups

An application security group contains zero or more network interfaces. To learn more, see [application security groups](#). All network interfaces in an application security group must exist in the same virtual network. To learn how to add a network interface to an application security group, see [Add a network interface to an application security group](#).

**Create an application security group**

1. On the [Azure portal](#) menu or from the **Home** page, select **Create a resource**.
2. In the search box, enter *Application security group*.
3. In the **Application security group** page, select **Create**.
4. In the **Create an application security group** page, under the **Basics** tab, set values for the following settings:

SETTING	ACTION
Subscription	Choose your subscription.
Resource group	Choose an existing resource group, or select <b>Create new</b> to create a new resource group.
Name	Enter a unique text string within a resource group.

SETTING	ACTION
Region	Choose the location you want.

5. Select **Review + create**.

6. Under the **Review + create** tab, after you see the **Validation passed** message, select **Create**.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network asg create</a>
PowerShell	<a href="#">New-AzApplicationSecurityGroup</a>

#### View all application security groups

Go to the [Azure portal](#) to view your application security groups. Search for and select **Application security groups**. The Azure portal displays a list of your application security groups.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network asg list</a>
PowerShell	<a href="#">Get-AzApplicationSecurityGroup</a>

#### View details of a specific application security group

1. Go to the [Azure portal](#) to view an application security group. Search for and select **Application security groups**.
2. Select the name of the application security group that you want to view the details of.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network asg show</a>
PowerShell	<a href="#">Get-AzApplicationSecurityGroup</a>

#### Change an application security group

1. Go to the [Azure portal](#) to view an application security group. Search for and select **Application security groups**.
2. Select the name of the application security group that you want to change.
3. Select **change** next to the setting that you want to modify. For example, you can add or remove **Tags**, or you can change the **Resource group** or **Subscription**.

#### NOTE

You can't change the location.

In the menu bar, you can also select **Access control (IAM)**. In the **Access control (IAM)** page, you can

assign or remove permissions to the application security group.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network asg update</a>
PowerShell	No PowerShell cmdlet

#### Delete an application security group

You can't delete an application security group if it contains any network interfaces. To remove all network interfaces from the application security group, either change the network interface settings or delete the network interfaces. To learn more, see [Add to or remove from application security groups](#) or [Delete a network interface](#).

1. Go to the [Azure portal](#) to manage your application security groups. Search for and select **Application security groups**.
2. Select the name of the application security group that you want to delete.
3. Select **Delete**, and then select **Yes** to delete the application security group.

#### Commands

TOOL	COMMAND
Azure CLI	<a href="#">az network asg delete</a>
PowerShell	<a href="#">Remove-AzApplicationSecurityGroup</a>

## Permissions

To do tasks on network security groups, security rules, and application security groups, your account must be assigned to the [Network contributor](#) role or to a [Custom role](#) that's assigned the appropriate permissions as listed in the following tables:

#### Network security group

ACTION	NAME
Microsoft.Network/networkSecurityGroups/read	Get network security group
Microsoft.Network/networkSecurityGroups/write	Create or update network security group
Microsoft.Network/networkSecurityGroups/delete	Delete network security group
Microsoft.Network/networkSecurityGroups/join/action	Associate a network security group to a subnet or network interface

#### Network security group rule

ACTION	NAME
Microsoft.Network/networkSecurityGroups/securityRules/read	Get rule

ACTION	NAME
Microsoft.Network/networkSecurityGroups/securityRules/write	Create or update rule
Microsoft.Network/networkSecurityGroups/securityRules/delete	Delete rule

## Application security group

ACTION	NAME
Microsoft.Network/applicationSecurityGroups/joinIpConfiguration/action	Join an IP configuration to an application security group
Microsoft.Network/applicationSecurityGroups/joinNetworkSecurityRule/action	Join a security rule to an application security group
Microsoft.Network/applicationSecurityGroups/read	Get an application security group
Microsoft.Network/applicationSecurityGroups/write	Create or update an application security group
Microsoft.Network/applicationSecurityGroups/delete	Delete an application security group

## Next steps

- Create a network or application security group using [PowerShell](#) or [Azure CLI](#) sample scripts, or [Azure Resource Manager templates](#)
- Create and assign [Azure Policy definitions](#) for virtual networks

# Virtual appliance scenario

11/2/2020 • 8 minutes to read • [Edit Online](#)

A common scenario among larger Azure customer is the need to provide a two-tiered application exposed to the Internet, while allowing access to the back tier from an on-premises datacenter. This document will walk you through a scenario using User Defined Routes (UDR), a VPN Gateway, and network virtual appliances to deploy a two-tier environment that meets the following requirements:

- Web application must be accessible from the public Internet only.
- Web server hosting the application must be able to access a backend application server.
- All traffic from the Internet to the web application must go through a firewall virtual appliance. This virtual appliance will be used for Internet traffic only.
- All traffic going to the application server must go through a firewall virtual appliance. This virtual appliance will be used for access to the backend end server, and access coming in from the on-premises network via a VPN Gateway.
- Administrators must be able to manage the firewall virtual appliances from their on-premises computers, by using a third firewall virtual appliance used exclusively for management purposes.

This is a standard perimeter network (also knowns as DMZ) scenario with a DMZ and a protected network. Such scenario can be constructed in Azure by using NSGs,firewall virtual appliances, or a combination of both. The table below shows some of the pros and cons between NSGs and firewall virtual appliances.

	PROS	CONS
NSG	No cost. Integrated into Azure RBAC. Rules can be created in Azure Resource Manager templates.	Complexity could vary in larger environments.
Firewall	Full control over data plane. Central management through firewall console.	Cost of firewall appliance. Not integrated with Azure RBAC.

The solution below uses firewall virtual appliances to implement a perimeter network (DMZ)/protected network scenario.

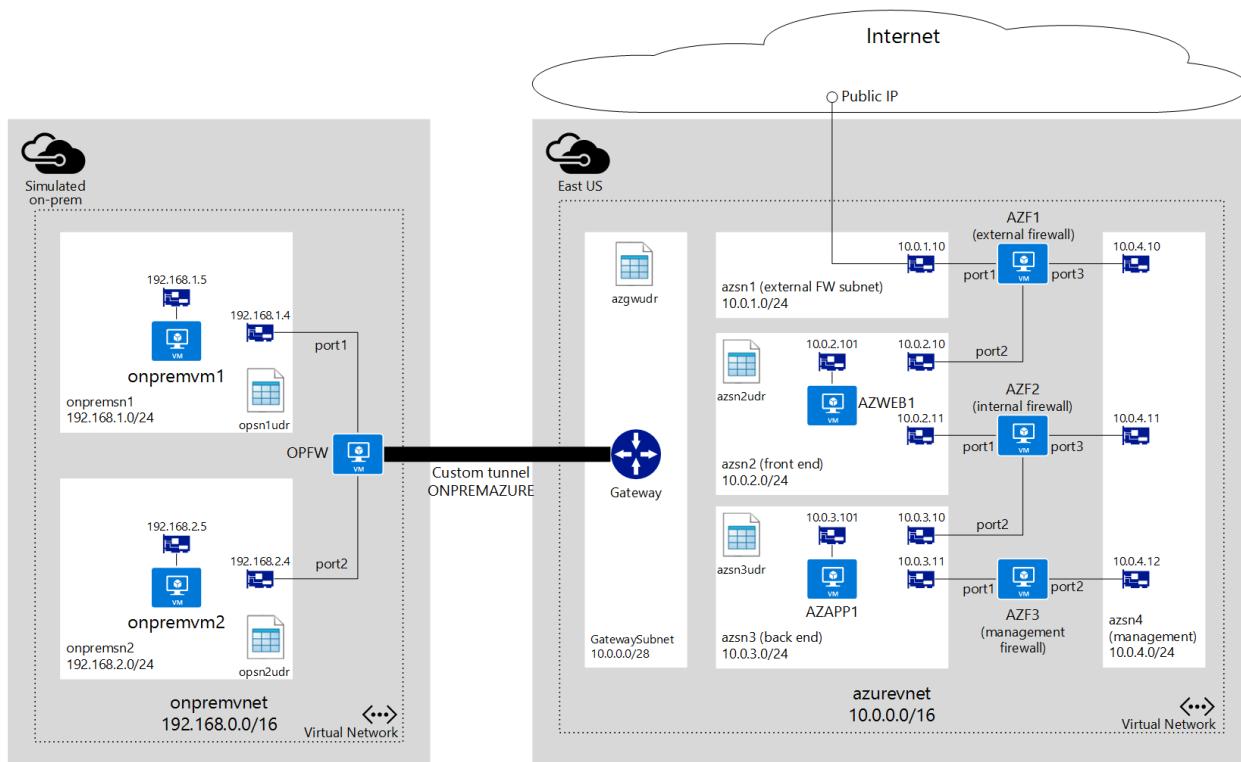
## Considerations

You can deploy the environment explained above in Azure using different features available today, as follows.

- **Virtual network (VNet).** An Azure VNet acts in similar fashion to an on-premises network, and can be segmented into one or more subnets to provide traffic isolation, and separation of concerns.
- **Virtual appliance.** Several partners provide virtual appliances in the Azure Marketplace that can be used for the three firewalls described above.
- **User Defined Routes (UDR).** Route tables can contain UDRs used by Azure networking to control the flow of packets within a VNet. These route tables can be applied to subnets. One of the newest features in Azure is the ability to apply a route table to the GatewaySubnet, providing the ability to forward all traffic coming into the Azure VNet from a hybrid connection to a virtual appliance.
- **IP Forwarding.** By default, the Azure networking engine forward packets to virtual network interface cards (NICs) only if the packet destination IP address matches the NIC IP address. Therefore, if a UDR defines that a

packet must be sent to a given virtual appliance, the Azure networking engine would drop that packet. To ensure the packet is delivered to a VM (in this case a virtual appliance) that is not the actual destination for the packet, you need to enable IP Forwarding for the virtual appliance.

- **Network Security Groups (NSGs).** The example below does not make use of NSGs, but you could use NSGs applied to the subnets and/or NICs in this solution to further filter the traffic in and out of those subnets and NICs.



In this example there is a subscription that contains the following:

- 2 resource groups, not shown in the diagram.
  - **ONPREMRG.** Contains all resources necessary to simulate an on-premises network.
  - **AZURERG.** Contains all resources necessary for the Azure virtual network environment.
- A VNet named **onpremvnet** used to mimic an on-premises datacenter segmented as listed below.
  - **onpremsn1.** Subnet containing a virtual machine (VM) running Ubuntu to mimic an on-premises server.
  - **onpremsn2.** Subnet containing a VM running Ubuntu to mimic an on-premises computer used by an administrator.
- There is one firewall virtual appliance named **OPFW** on **onpremvnet** used to maintain a tunnel to **azurevnet**.
- A VNet named **azurevnet** segmented as listed below.
  - **azsn1.** External firewall subnet used exclusively for the external firewall. All Internet traffic will come in through this subnet. This subnet only contains a NIC linked to the external firewall.
  - **azsn2.** Front end subnet hosting a VM running as a web server that will be accessed from the Internet.
  - **azsn3.** Backend subnet hosting a VM running a backend application server that will be accessed by the front end web server.
  - **azsn4.** Management subnet used exclusively to provide management access to all firewall virtual appliances. This subnet only contains a NIC for each firewall virtual appliance used in the solution.
  - **GatewaySubnet.** Azure hybrid connection subnet required for ExpressRoute and VPN Gateway to provide connectivity between Azure VNets and other networks.
- There are 3 firewall virtual appliances in the **azurevnet** network.

- AZF1. External firewall exposed to the public Internet by using a public IP address resource in Azure. You need to ensure you have a template from the Marketplace, or directly from your appliance vendor, that provisions a 3-NIC virtual appliance.
- AZF2. Internal firewall used to control traffic between **azsn2** and **azsn3**. This is also a 3-NIC virtual appliance.
- AZF3. Management firewall accessible to administrators from the on-premises datacenter, and connected to a management subnet used to manage all firewall appliances. You can find 2-NIC virtual appliance templates in the Marketplace, or request one directly from your appliance vendor.

## User Defined Routing (UDR)

Each subnet in Azure can be linked to a UDR table used to define how traffic initiated in that subnet is routed. If no UDRs are defined, Azure uses default routes to allow traffic to flow from one subnet to another. To better understand UDRs, visit [What are User Defined Routes and IP Forwarding](#).

To ensure communication is done through the right firewall appliance, based on the last requirement above, you need to create the following route table containing UDRs in **azurevnet**.

### **azgwudr**

In this scenario, the only traffic flowing from on-premises to Azure will be used to manage the firewalls by connecting to **AZF3**, and that traffic must go through the internal firewall, **AZF2**. Therefore, only one route is necessary in the **GatewaySubnet** as shown below.

DESTINATION	NEXT HOP	EXPLANATION
10.0.4.0/24	10.0.3.11	Allows on-premises traffic to reach management firewall <b>AZF3</b>

### **azsn2udr**

DESTINATION	NEXT HOP	EXPLANATION
10.0.3.0/24	10.0.2.11	Allows traffic to the backend subnet hosting the application server through <b>AZF2</b>
0.0.0.0/0	10.0.2.10	Allows all other traffic to be routed through <b>AZF1</b>

### **azsn3udr**

DESTINATION	NEXT HOP	EXPLANATION
10.0.2.0/24	10.0.3.10	Allows traffic to <b>azsn2</b> to flow from app server to the webserver through <b>AZF2</b>

You also need to create route tables for the subnets in **onpremvnet** to mimic the on-premises datacenter.

### **onpremsn1udr**

DESTINATION	NEXT HOP	EXPLANATION
192.168.2.0/24	192.168.1.4	Allows traffic to <b>onpremsn2</b> through <b>OPFW</b>

## onpremsn2udr

DESTINATION	NEXT HOP	EXPLANATION
10.0.3.0/24	192.168.2.4	Allows traffic to the backed subnet in Azure through <b>OPFW</b>
192.168.1.0/24	192.168.2.4	Allows traffic to <b>onpremsn1</b> through <b>OPFW</b>

## IP Forwarding

UDR and IP Forwarding are features that you can use in combination to allow virtual appliances to be used to control traffic flow in an Azure VNet. A virtual appliance is nothing more than a VM that runs an application used to handle network traffic in some way, such as a firewall or a NAT device.

This virtual appliance VM must be able to receive incoming traffic that is not addressed to itself. To allow a VM to receive traffic addressed to other destinations, you must enable IP Forwarding for the VM. This is an Azure setting, not a setting in the guest operating system. Your virtual appliance still needs to run some type of application to handle the incoming traffic, and route it appropriately.

To learn more about IP Forwarding, visit [What are User Defined Routes and IP Forwarding](#).

As an example, imagine you have the following setup in an Azure vnet:

- Subnet **onpremsn1** contains a VM named **onpremvm1**.
- Subnet **onpremsn2** contains a VM named **onpremvm2**.
- A virtual appliance named **OPFW** is connected to **onpremsn1** and **onpremsn2**.
- A user defined route linked to **onpremsn1** specifies that all traffic to **onpremsn2** must be sent to **OPFW**.

At this point, if **onpremvm1** tries to establish a connection with **onpremvm2**, the UDR will be used and traffic will be sent to **OPFW** as the next hop. Keep in mind that the actual packet destination is not being changed, it still says **onpremvm2** is the destination.

Without IP Forwarding enabled for **OPFW**, the Azure virtual networking logic will drop the packets, since it only allows packets to be sent to a VM if the VM's IP address is the destination for the packet.

With IP Forwarding, the Azure virtual network logic will forward the packets to **OPFW**, without changing its original destination address. **OPFW** must handle the packets and determine what to do with them.

For the scenario above to work, you must enable IP Forwarding on the NICs for **OPFW**, **AZF1**, **AZF2**, and **AZF3** that are used for routing (all NICs except the ones linked to the management subnet).

## Firewall Rules

As described above, IP Forwarding only ensures packets are sent to the virtual appliances. Your appliance still needs to decide what to do with those packets. In the scenario above, you will need to create the following rules in your appliances:

### OPFW

**OPFW** represents an on-premises device containing the following rules:

- **Route:** All traffic to 10.0.0.0/16 (**azurevnet**) must be sent through tunnel **ONPREMAZURE**.
- **Policy:** Allow all bidirectional traffic between **port2** and **ONPREMAZURE**.

### AZF1

**AZF1** represents an Azure virtual appliance containing the following rules:

- **Policy:** Allow all bidirectional traffic between **port1** and **port2**.

## AZF2

AZF2 represents an Azure virtual appliance containing the following rules:

- **Route:** All traffic to 10.0.0.0/16 (**onpremvnet**) must be sent to the Azure gateway IP address (i.e. 10.0.0.1) through **port1**.
- **Policy:** Allow all bidirectional traffic between **port1** and **port2**.

## Network Security Groups (NSGs)

In this scenario, NSGs are not being used. However, you could apply NSGs to each subnet to restrict incoming and outgoing traffic. For instance, you could apply the following NSG rules to the external FW subnet.

### Incoming

- Allow all TCP traffic from the Internet to port 80 on any VM in the subnet.
- Deny all other traffic from the Internet.

### Outgoing

- Deny all traffic to the Internet.

## High level steps

To deploy this scenario, follow the high level steps below.

1. Login to your Azure Subscription.
2. If you want to deploy a VNet to mimic the on-premises network, provision the resources that are part of **ONPREMRG**.
3. Provision the resources that are part of **AZURERG**.
4. Provision the tunnel from **onpremvnet** to **azurevnet**.
5. Once all resources are provisioned, sign in to **onprevm2** and ping 10.0.3.101 to test connectivity between **onpremsn2** and **azsn3**.

# Create a virtual machine with a static public IP address using the Azure portal

2/18/2021 • 2 minutes to read • [Edit Online](#)

A public IP address enables you to communicate to a virtual machine from the internet.

Assign a static public IP address, rather than a dynamic address, to ensure that the address never changes.

## Sign in to Azure

Sign in to the [Azure portal](#).

## Create a virtual machine

1. On the upper-left side of the portal, select **Create a resource > Compute > Virtual machine** or search for **Virtual machine** in the search box.
2. In **Create a virtual machine**, type or select the values in the **Basics** tab:

SETTING	VALUE
<b>Project Details</b>	
Subscription	Select your Azure subscription
Resource Group	Select <b>Create new</b> . In <b>Name</b> , enter <b>myResourceGroup</b> . Select <b>OK</b> .
<b>Instance details</b>	
Virtual machine name	Enter <b>myVM</b>
Region	Select <b>East US</b>
Availability Options	Select <b>No infrastructure redundancy required</b>
Image	Select <b>Windows Server 2019 Datacenter - Gen1</b>
Azure Spot instance	Select <b>No</b>
Size	Choose VM size or take default setting
<b>Administrator account</b>	
Username	Enter a username
Password	Enter a password

SETTING	VALUE
Confirm password	Reenter password

3. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

4. In the Networking tab, select or enter:

SETTING	VALUE
Network interface	
Virtual network	Accept the default network name.
Subnet	Accept the default subnet configuration.
Public IP	Select <b>Create new</b> . In <b>Create public IP address</b> , in name enter <b>myPublicIP</b> . For <b>SKU</b> , select <b>Standard</b> . <b>Assignment</b> , select <b>Static</b> . Select <b>OK</b> .
NIC network security group	Select <b>Basic</b>
Public inbound ports	Select <b>Allow selected ports</b> .
Select inbound ports	Select <b>RDP (3389)</b>

#### WARNING

Portal 3389 is selected, to enable remote access to the Windows Server virtual machine from the internet.

Opening port 3389 to the internet is not recommended to manage production workloads.

For secure access to Azure virtual machines, see [What is Azure Bastion?](#)

5. Select **Review + create**.

6. Review the settings, and then select **Create**.

## Clean up resources

When no longer needed, delete the resource group and all of the resources it contains:

1. Enter **myResourceGroup** in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter **myResourceGroup** for **TYPE THE RESOURCE GROUP NAME**: and select **Delete**.

## Next steps

See [Add, change, or remove IP addresses](#):

- To change a public IP address from dynamic to static.
- Work with private IP addresses.

Public IP addresses have a [nominal charge](#). There's a [limit](#) to the number of public IP addresses that you can use per subscription.

The SKU of the virtual machine's public IP address must match the public IP SKU of Azure Load Balancer when added to a backend pool. For details, see [Azure Load Balancer](#).

You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds.

- Learn more about [static public IP addresses](#).
- Learn more about [public IP addresses](#) in Azure.
- Learn more about all [public IP address settings](#).
- Learn more about [private IP addresses](#) and assigning a [static private IP address](#) to an Azure virtual machine.

# Create a virtual machine with a static public IP address using PowerShell

2/18/2021 • 3 minutes to read • [Edit Online](#)

You can create a virtual machine with a static public IP address. A public IP address enables you to communicate to a virtual machine from the internet. Assign a static public IP address, rather than a dynamic address, to ensure that the address never changes. Learn more about [static public IP addresses](#). To change a public IP address assigned to an existing virtual machine from dynamic to static, or to work with private IP addresses, see [Add, change, or remove IP addresses](#). Public IP addresses have a [nominal charge](#), and there is a [limit](#) to the number of public IP addresses that you can use per subscription.

## NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Create a virtual machine

You can complete the following steps from your local computer or by using the Azure Cloud Shell. To use your local computer, ensure you have the [Azure PowerShell installed](#). To use the Azure Cloud Shell, select **Try It** in the top right corner of any command box that follows. The Cloud Shell signs you into Azure.

1. If using the Cloud Shell, skip to step 2. Open a command session and sign into Azure with `Connect-AzAccount`.
2. Create a resource group with the `New-AzResourceGroup` command. The following example creates a resource group in the East US Azure region:

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS
```

3. Create a virtual machine with the `New-AzVM` command. The `-AllocationMethod "Static"` option assigns a static public IP address to the virtual machine. The following example creates a Windows Server virtual machine with a static, basic SKU public IP address named *myPublicIpAddress*. When prompted, provide a username and password to be used as the sign in credentials for the virtual machine:

```
New-AzVm `
-ResourceGroupName "myResourceGroup" `
-Name "myVM" `
-Location "East US" `
-PublicIpAddressName "myPublicIpAddress" `
-AllocationMethod "Static"
```

If the public IP address must be a standard SKU, you have to [create a public IP address](#), [create a network interface](#), [assign the public IP address to the network interface](#), and then [create a virtual machine with the network interface](#), in separate steps. Learn more about [Public IP address SKUs](#). If the virtual machine will be added to the back-end pool of a public Azure Load Balancer, the SKU of the virtual machine's public IP address must match the SKU of the load balancer's public IP address. For details, see [Azure Load Balancer](#).

4. View the public IP address assigned and confirm that it was created as a static address, with [Get-AzPublicIpAddress](#):

```
Get-AzPublicIpAddress `
-ResourceGroupName "myResourceGroup" `
-Name "myPublicIpAddress" `
| Select "IpAddress", "PublicIpAllocationMethod" `
| Format-Table
```

Azure assigned a public IP address from addresses used in the region you created the virtual machine in. You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds.

#### WARNING

Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary, and not until after reading [Add a private IP address to an operating system](#).

## Clean up resources

When no longer needed, you can use [Remove-AzResourceGroup](#) to remove the resource group and all of the resources it contains:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

## Next steps

- Learn more about [public IP addresses](#) in Azure
- Learn more about all [public IP address settings](#)
- Learn more about [private IP addresses](#) and assigning a [static private IP address](#) to an Azure virtual machine
- Learn more about creating [Linux](#) and [Windows](#) virtual machines

# Create a virtual machine with a static public IP address using the Azure CLI

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can create a virtual machine with a static public IP address. A public IP address enables you to communicate to a virtual machine from the internet. Assign a static public IP address, rather than a dynamic address, to ensure that the address never changes. Learn more about [static public IP addresses](#). To change a public IP address assigned to an existing virtual machine from dynamic to static, or to work with private IP addresses, see [Add, change, or remove IP addresses](#). Public IP addresses have a [nominal charge](#), and there is a [limit](#) to the number of public IP addresses that you can use per subscription.

## Create a virtual machine

You can complete the following steps from your local computer or by using the Azure Cloud Shell. To use your local computer, ensure you have the [Azure CLI installed](#). To use the Azure Cloud Shell, select **Try It** in the top right corner of any command box that follows. The Cloud Shell signs you into Azure.

1. If using the Cloud Shell, skip to step 2. Open a command session and sign into Azure with `az login`.
2. Create a resource group with the `az group create` command. The following example creates a resource group in the East US Azure region:

```
az group create --name myResourceGroup --location eastus
```

3. Create a virtual machine with the `az vm create` command. The `--public-ip-address-allocation=static` option assigns a static public IP address to the virtual machine. The following example creates an Ubuntu virtual machine with a static, basic SKU public IP address named *myPublicIpAddress*:

```
az vm create \
 --resource-group myResourceGroup \
 --name myVM \
 --image UbuntuLTS \
 --admin-username azureuser \
 --generate-ssh-keys \
 --public-ip-address myPublicIpAddress \
 --public-ip-address-allocation static
```

If the public IP address must be a standard SKU, add `--public-ip-sku Standard` to the previous command. Learn more about [Public IP address SKUs](#). If the virtual machine will be added to the back-end pool of a public Azure Load Balancer, the SKU of the virtual machine's public IP address must match the SKU of the load balancer's public IP address. For details, see [Azure Load Balancer](#).

4. View the public IP address assigned and confirm that it was created as a static, basic SKU address, with `az network public-ip show`:

```
az network public-ip show \
 --resource-group myResourceGroup \
 --name myPublicIpAddress \
 --query [ipAddress,publicIpAllocationMethod,sku] \
 --output table
```

Azure assigned a public IP address from addresses used in the region you created the virtual machine in. You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds.

#### WARNING

Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary, and not until after reading [Add a private IP address to an operating system](#).

## Clean up resources

When no longer needed, you can use `az group delete` to remove the resource group and all of the resources it contains:

```
az group delete --name myResourceGroup --yes
```

## Next steps

- Learn more about [public IP addresses](#) in Azure
- Learn more about all [public IP address settings](#)
- Learn more about [private IP addresses](#) and assigning a [static private IP address](#) to an Azure virtual machine
- Learn more about creating [Linux](#) and [Windows](#) virtual machines

# Associate a public IP address to a virtual machine

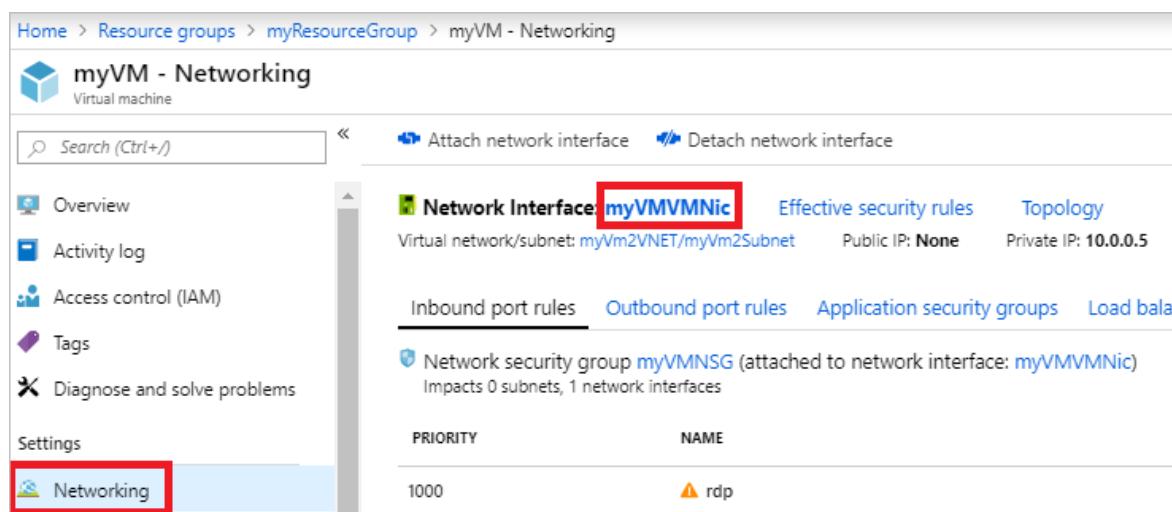
2/18/2021 • 10 minutes to read • [Edit Online](#)

In this article, you learn how to associate a public IP address to an existing virtual machine (VM). If you want to connect to a VM from the internet, the VM must have a public IP address associated to it. If you want to create a new VM with a public IP address, you can do so using the [Azure portal](#), the [Azure command-line interface \(CLI\)](#), or [PowerShell](#). Public IP addresses have a nominal fee. For details, see [pricing](#). There is a limit to the number of public IP addresses that you can use per subscription. For details, see [limits](#).

You can use the [Azure portal](#), the Azure [command-line interface \(CLI\)](#), or [PowerShell](#) to associate a public IP address to a VM.

## Azure portal

1. Sign in to the [Azure portal](#).
2. Browse to, or search for the virtual machine that you want to add the public IP address to and then select it.
3. Under **Settings**, select **Networking**, and then select the network interface you want to add the public IP address to, as shown in the following picture:



The screenshot shows the Azure portal interface for managing a virtual machine named 'myVM'. The top navigation bar shows the path: Home > Resource groups > myResourceGroup > myVM - Networking. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, and Networking (which is highlighted with a red box). The main content area is titled 'myVM - Networking' and shows a 'Network Interface' section with 'myVMVMNic' selected. It displays information such as the Virtual network/subnet: myVm2VNET/myVm2Subnet, Public IP: None, and Private IP: 10.0.0.5. Below this, there are tabs for Inbound port rules, Outbound port rules, Application security groups, and Load balancer. A note indicates that Network security group myVMNSG is attached to the network interface. At the bottom, there's a table for Inbound port rules with columns for Priority and Name, showing one rule for port 1000 named 'rdp'.

### NOTE

Public IP addresses are associated to network interfaces attached to a VM. In the previous picture, the VM only has one network interface. If the VM had multiple network interfaces, they would all appear, and you'd select the network interface you want to associate the public IP address to.

4. Select **IP configurations** and then select an IP configuration, as shown in the following picture:

The screenshot shows the Azure portal interface for managing network interfaces. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Settings, IP configurations (which is highlighted with a red box), DNS servers, Network security group, and Properties. The main area displays 'IP forwarding settings' (Enabled) and 'Virtual network' (myVm2VNET). Under 'IP configurations', it shows a single entry: 'Subnet' (myVm2Subnet (10.0.0.0/24)). Below this is a table with columns NAME, IP VERSION, TYPE, PRIVATE IP ADDRESS, and PUBLIC IP ADDRESS. One row is shown: ipconfigmyVM (IPv4, Primary, 10.0.0.5 (Dynamic), -). A red box highlights this row.

#### NOTE

Public IP addresses are associated to IP configurations for a network interface. In the previous picture, the network interface has one IP configuration. If the network interface had multiple IP configurations, they would all appear in the list, and you'd select the IP configuration that you want to associate the public IP address to.

5. Select **Enabled**, then select **IP address (Configure required settings)**. Choose an existing public IP address, which automatically closes the **Choose public IP address** box. If you don't have any available public IP addresses listed, you need to create one. To learn how, see [Create a public IP address](#). Select **Save**, as shown in the picture that follows, and then close the box for the IP configuration.

This screenshot shows the 'Choose public IP address' dialog box. On the left, there's a sidebar with Save and Discard buttons. The main area lists public IP addresses: 'myVMPublicIP' under 'myResourceGroup'. A red box highlights the 'myVMPublicIP' entry. A message at the top right says: 'These are the public IP addresses in the selected subscription and location 'East US''. A 'Create new' button is also visible.

#### NOTE

The public IP addresses that appear are those that exist in the same region as the VM. If you have multiple public IP addresses created in the region, all will appear here. If any are grayed out, it's because the address is already associated to a different resource.

6. View the public IP address assigned to the IP configuration, as shown in the picture that follows. It may take a few seconds for an IP address to appear.

The screenshot shows the Azure portal's 'IP configurations' page for the 'myVMVMNic' network interface. On the left, a sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings', 'IP configurations' (which is selected), 'DNS servers', 'Network security group', and 'Properties'. The main pane displays 'IP forwarding settings' (disabled) and 'Virtual network' (myVm2VNET). Under 'IP configurations', it shows a table with one row:

NAME	IP VERSION	TYPE	PRIVATE IP ADDRESS	PUBLIC IP ADDRESS
ipconfig...	IPv4	Primary	10.0.0.5 (Dynamic)	52.179.3.114 (myVMPublicIP) ...

#### NOTE

The address is assigned from a pool of addresses used in each Azure region. To see a list of address pools used in each region, see [Microsoft Azure Datacenter IP Ranges](#). The address assigned can be any address in the pools used for the region. If you need the address to be assigned from a specific pool in the region, use a [Public IP address prefix](#).

7. [Allow network traffic to the VM](#) with security rules in a network security group.

## Azure CLI

Install the [Azure CLI](#), or use the Azure Cloud Shell. The Azure Cloud Shell is a free Bash shell that you can run directly within the Azure portal. It has the Azure CLI preinstalled and configured to use with your account. Select the **Try it** button in the CLI commands that follow. Selecting **Try it** invokes a Cloud Shell that you can sign in to your Azure account with.

1. If using the CLI locally in Bash, sign in to Azure with `az login`.
2. A public IP address is associated to an IP configuration of a network interface attached to a VM. Use the `az network nic-ip-config update` command to associate a public IP address to an IP configuration. The following example associates an existing public IP address named *myVMPublicIP* to the IP configuration named *ipconfigmyVM* of an existing network interface named *myVMVMNic* that exists in a resource group named *myResourceGroup*.

```
az network nic ip-config update \
--name ipconfigmyVM \
--nic-name myVMVMNic \
--resource-group myResourceGroup \
--public-ip-address myVMPublicIP
```

- If you don't have an existing public IP address, use the `az network public-ip create` command to create one. For example, the following command creates a public IP address named *myVMPublicIP* in a resource group named *myResourceGroup*.

```
az network public-ip create --name myVMPublicIP --resource-group myResourceGroup
```

#### NOTE

The previous command creates a public IP address with default values for several settings that you may want to customize. To learn more about all public IP address settings, see [Create a public IP address](#). The address is assigned from a pool of public IP addresses used for each Azure region. To see a list of address pools used in each region, see [Microsoft Azure Datacenter IP Ranges](#).

- If you don't know the name of a network interface attached to your VM, use the [az vm nic list](#) command to view them. For example, the following command lists the names of the network interfaces attached to a VM named *myVM* in a resource group named *myResourceGroup*:

```
az vm nic list --vm-name myVM --resource-group myResourceGroup
```

The output includes one or more lines that are similar to the following example:

```
"id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic",
```

In the previous example, *myVMVMNic* is the name of the network interface.

- If you don't know the name of an IP configuration for a network interface, use the [az network nic ip-config list](#) command to retrieve them. For example, the following command lists the names of the IP configurations for a network interface named *myVMVMNic* in a resource group named *myResourceGroup*:

```
az network nic ip-config list --nic-name myVMVMNic --resource-group myResourceGroup --out table
```

3. View the public IP address assigned to the IP configuration with the [az vm list-ip-addresses](#) command. The following example shows the IP addresses assigned to an existing VM named *myVM* in a resource group named *myResourceGroup*.

```
az vm list-ip-addresses --name myVM --resource-group myResourceGroup --out table
```

#### NOTE

The address is assigned from a pool of addresses used in each Azure region. To see a list of address pools used in each region, see [Microsoft Azure Datacenter IP Ranges](#). The address assigned can be any address in the pools used for the region. If you need the address to be assigned from a specific pool in the region, use a [Public IP address prefix](#).

4. [Allow network traffic to the VM](#) with security rules in a network security group.

## PowerShell

Install [PowerShell](#), or use the Azure Cloud Shell. The Azure Cloud Shell is a free shell that you can run directly within the Azure portal. It has PowerShell preinstalled and configured to use with your account. Select the [Try it](#) button in the PowerShell commands that follow. Selecting [Try it](#) invokes a Cloud Shell that you can sign in to your Azure account with.

1. If using PowerShell locally, sign in to Azure with `Connect-AzAccount`.
2. A public IP address is associated to an IP configuration of a network interface attached to a VM. Use the `Get-AzVirtualNetwork` and `Get-AzVirtualNetworkSubnetConfig` commands to get the virtual network and subnet that the network interface is in. Next, use the `Get-AzNetworkInterface` command to get a network interface and the `Get-AzPublicIpAddress` command to get an existing public IP address. Then use the `Set-AzNetworkInterfaceIpConfig` command to associate the public IP address to the IP configuration and the `Set-AzNetworkInterface` command to write the new IP configuration to the network interface.

The following example associates an existing public IP address named *myVMPublicIP* to the IP configuration named *ipconfigmyVM* of an existing network interface named *myVMVNic* that exists in a subnet named *myVMSubnet* in a virtual network named *myVMVNet*. All resources are in a resource group named *myResourceGroup*.

```
$vnet = Get-AzVirtualNetwork -Name myVMVNet -ResourceGroupName myResourceGroup
$subnet = Get-AzVirtualNetworkSubnetConfig -Name myVMSubnet -VirtualNetwork $vnet
$nic = Get-AzNetworkInterface -Name myVMVNic -ResourceGroupName myResourceGroup
$pip = Get-AzPublicIpAddress -Name myVMPublicIP -ResourceGroupName myResourceGroup
$nic | Set-AzNetworkInterfaceIpConfig -Name ipconfigmyVM -PublicIPAddress $pip -Subnet $subnet
$nic | Set-AzNetworkInterface
```

- If you don't have an existing public IP address, use the `New-AzPublicIpAddress` command to create one. For example, the following command creates a *dynamic* public IP address named *myVMPublicIP* in a resource group named *myResourceGroup* in the *eastus* region.

```
New-AzPublicIpAddress -Name myVMPublicIP -ResourceGroupName myResourceGroup -AllocationMethod Dynamic -Location eastus
```

#### NOTE

The previous command creates a public IP address with default values for several settings that you may want to customize. To learn more about all public IP address settings, see [Create a public IP address](#). The address is assigned from a pool of public IP addresses used for each Azure region. To see a list of address pools used in each region, see [Microsoft Azure Datacenter IP Ranges](#).

- If you don't know the name of a network interface attached to your VM, use the `Get-AzVM` command to view them. For example, the following command lists the names of the network interfaces attached to a VM named *myVM* in a resource group named *myResourceGroup*:

```
$vm = Get-AzVM -name myVM -ResourceGroupName myResourceGroup
$vm.NetworkProfile
```

The output includes one or more lines that are similar to the example that follows. In the example output, *myVMVNic* is the name of the network interface.

```
"id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVNic",
```

- If you don't know the name of the virtual network or subnet that the network interface is in, use the `Get-AzNetworkInterface` command to view the information. For example, the following command gets the virtual network and subnet information for a network interface named *myVMVNic* in a resource group named *myResourceGroup*.

```
$nic = Get-AzNetworkInterface -Name myVMVNINic -ResourceGroupName myResourceGroup
$ipConfigs = $nic.IpConfigurations
$ipConfigs.Subnet | Select Id
```

The output includes one or more lines that are similar to the example that follows. In the example output, *myVMVNET* is the name of the virtual network and *myVMSubnet* is the name of the subnet.

```
"/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/virtualNetworks/myVMVNET/subnets/myVMSubnet",
```

- If you don't know the name of an IP configuration for a network interface, use the [Get-AzNetworkInterface](#) command to retrieve them. For example, the following command lists the names of the IP configurations for a network interface named *myVMVNINic* in a resource group named *myResourceGroup*.

```
$nic = Get-AzNetworkInterface -Name myVMVNINic -ResourceGroupName myResourceGroup
$nic.IPConfigurations
```

The output includes one or more lines that are similar to the example that follows. In the example output, *ipconfigmyVM* is the name of an IP configuration.

```
Id : /subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVNINic/ipConfigurations/ipconfigmyVM
```

3. View the public IP address assigned to the IP configuration with the [Get-AzPublicIpAddress](#) command. The following example shows the address assigned to a public IP address named *myVMPublicIP* in a resource group named *myResourceGroup*.

```
Get-AzPublicIpAddress -Name myVMPublicIP -ResourceGroupName myResourceGroup | Select IPAddress
```

If you don't know the name of the public IP address assigned to an IP configuration, run the following commands to get it:

```
$nic = Get-AzNetworkInterface -Name myVMVNINic -ResourceGroupName myResourceGroup
$nic.IPConfigurations
$address = $nic.IPConfigurations.PublicIpAddress
$address | Select Id
```

The output includes one or more lines that are similar to the example that follows. In the example output, *myVMPublicIP* is the name of the public IP address assigned to the IP configuration.

```
"/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myVMPublicIP"
```

**NOTE**

The address is assigned from a pool of addresses used in each Azure region. To see a list of address pools used in each region, see [Microsoft Azure Datacenter IP Ranges](#). The address assigned can be any address in the pools used for the region. If you need the address to be assigned from a specific pool in the region, use a [Public IP address prefix](#).

4. Allow network traffic to the VM with security rules in a network security group.

## Allow network traffic to the VM

Before you can connect to the public IP address from the internet, ensure that you have the necessary ports open in any network security group that you might have associated to the network interface, the subnet the network interface is in, or both. Though security groups filter traffic to the private IP address of the network interface, once inbound internet traffic arrives at the public IP address, Azure translates the public address to the private IP address, so if a network security group prevents the traffic flow, the communication with the public IP address fails. You can view the effective security rules for a network interface and its subnet using the [Portal](#), [CLI](#), or [PowerShell](#).

## Next steps

Allow inbound internet traffic to your VM with a network security group. To learn how to create a network security group, see [Work with network security groups](#). To learn more about network security groups, see [Security groups](#).

# Dissociate a public IP address from an Azure VM

2/18/2021 • 4 minutes to read • [Edit Online](#)

In this article, you learn how to dissociate a public IP address from an Azure virtual machine (VM).

You can use the [Azure portal](#), the Azure command-line interface (CLI), or [PowerShell](#) to dissociate a public IP address from a VM.

## Azure portal

1. Sign in to the [Azure portal](#).
2. Browse to, or search for the virtual machine that you want to disassociate the public IP address from and then select it.
3. In the VM page, select **Overview**, select the public IP address as shown in the following picture:

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with a tree view: 'myVM' (Virtual machine) is selected. Under 'myVM', the 'Overview' tab is highlighted with a red box. Other tabs include 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Networking', 'Disks', and 'Size'. The main content area displays various details about the VM, such as its resource group ('myResourceGroup'), status ('Unknown'), location ('East US'), subscription information, computer name ('myVM'), operating system ('Windows'), size ('Standard DS1 v2 (1 vcpus, 3.5 GiB memory)'), and tags. A specific row for 'Public IP address' is highlighted with a red box, showing the value '52.170.252.185'. To the right of this row, other fields like 'Private IP address' (192.168.1.4), 'Public IP address (IPv6)', 'Private IP address (IPv6)', 'Virtual network/subnet' (myVnet/mySubnet), 'DNS name' (myvm-0467f8.eastus.cloudapp.azure.com), and 'Scale Set' (N/A) are listed.

4. In the public IP address page, select **Overview**, and then select **Dissociate**, as shown in the following picture:

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with a tree view: 'myPublicIpAddress' (Public IP address) is selected. Under 'myPublicIpAddress', the 'Overview' tab is highlighted with a red box. Other tabs include 'Activity log', 'Access control (IAM)', 'Tags', and 'Settings'. The main content area displays details about the public IP address, such as its resource group ('myResourceGroup'), location ('East US'), subscription information, and tags. A prominent 'Dissociate' button is highlighted with a red box. Below the button, the IP address ('52.170.252.185'), DNS name ('myvm-0467f8.eastus.cloudapp.azure.com'), and associated VM ('myVM') are listed.

5. In **Dissociate public IP address**, select **Yes**.

## Azure CLI

Install the [Azure CLI](#), or use the Azure Cloud Shell. The Azure Cloud Shell is a free Bash shell that you can run directly within the Azure portal. It has the Azure CLI preinstalled and configured to use with your account. Select the **Try it** button in the CLI commands that follow. Selecting **Try it** invokes a Cloud Shell that you can sign in to your Azure account with.

1. If using the CLI locally in Bash, sign in to Azure with `az login`.
2. A public IP address is associated to an IP configuration of a network interface attached to a VM. Use the

`az network nic-ip-config update` command to dissociate a public IP address from an IP configuration. The following example dissociates a public IP address named *myVMPublicIP* from the IP configuration named *ipconfigmyVM* of an existing network interface named *myVMVMNic* that is attached to a VM named *myVM* in a resource group named *myResourceGroup*.

```
az network nic ip-config update \
--name ipconfigmyVM \
--resource-group myResourceGroup \
--nic-name myVMVMNic \
--remove PublicIpAddress
```

If you don't know the name of a network interface attached to your VM, use the `az vm nic list` command to view them. For example, the following command lists the names of the network interfaces attached to a VM named *myVM* in a resource group named *myResourceGroup*.

```
az vm nic list --vm-name myVM --resource-group myResourceGroup
```

The output includes one or more lines that are similar to the following example:

```
"id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic",
```

In the previous example, *myVMVMNic* is the name of the network interface.

- If you don't know the name of an IP configuration for a network interface, use the `az network nic ip-config list` command to retrieve them. For example, the following command lists the names of the public IP configurations for a network interface named *myVMVMNic* in a resource group named *myResourceGroup*.

```
az network nic ip-config list --nic-name myVMVMNic --resource-group myResourceGroup --out table
```

- If you don't know the name of a public IP configuration for a network interface, use the `az network nic ip-config show` command to retrieve them. For example, the following command lists the names of the public IP configurations for a network interface named *myVMVMNic* in a resource group named *myResourceGroup*.

```
az network nic ip-config show --name ipconfigmyVM --nic-name myVMVMNic --resource-group myResourceGroup --query publicIPAddress.id
```

## PowerShell

Install [PowerShell](#), or use the Azure Cloud Shell. The Azure Cloud Shell is a free shell that you can run directly within the Azure portal. It has PowerShell preinstalled and configured to use with your account. Select the **Try it** button in the PowerShell commands that follow. Selecting **Try it** invokes a Cloud Shell that you can sign in to your Azure account with.

1. If using PowerShell locally, sign in to Azure with `Connect-AzAccount`.
2. A public IP address is associated to an IP configuration of a network interface attached to a VM. Use the `Get-AzNetworkInterface` command to get a network interface. Set the Public IP address value to null and then use the `Set-AzNetworkInterface` command to write the new IP configuration to the network interface.

The following example dissociates a public IP address named *myVMPublicIP* from a network interface named *myVMVMNic* that is attached to a VM named *myVM*. All resources are in a resource group named *myResourceGroup*.

```
$nic = Get-AzNetworkInterface -Name myVMVMNic -ResourceGroup myResourceGroup
$nic.IpConfigurations.publicipaddress.id = $null
Set-AzNetworkInterface -NetworkInterface $nic
```

- If you don't know the name of a network interface attached to your VM, use the [Get-AzVM](#) command to view them. For example, the following command lists the names of the network interfaces attached to a VM named *myVM* in a resource group named *myResourceGroup*.

```
$vm = Get-AzVM -name myVM -ResourceGroupName myResourceGroup
$vm.NetworkProfile
```

The output includes one or more lines that are similar to the example that follows. In the example output, *myVMVMNic* is the name of the network interface.

```
"id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic",
```

- If you don't know the name of an IP configuration for a network interface, use the [Get-AzNetworkInterface](#) command to retrieve them. For example, the following command lists the names of the IP configurations for a network interface named *myVMVMNic* in a resource group named *myResourceGroup*.

```
$nic = Get-AzNetworkInterface -Name myVMVMNic -ResourceGroupName myResourceGroup
$nic.IPConfigurations.id
```

The output includes one or more lines that are similar to the example that follows. In the example output, *ipconfigmyVM* is the name of an IP configuration.

```
"id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic/i
pConfigurations/ipconfigmyVM"
```

## Next steps

- Learn how to [associate a public IP address to a VM](#).

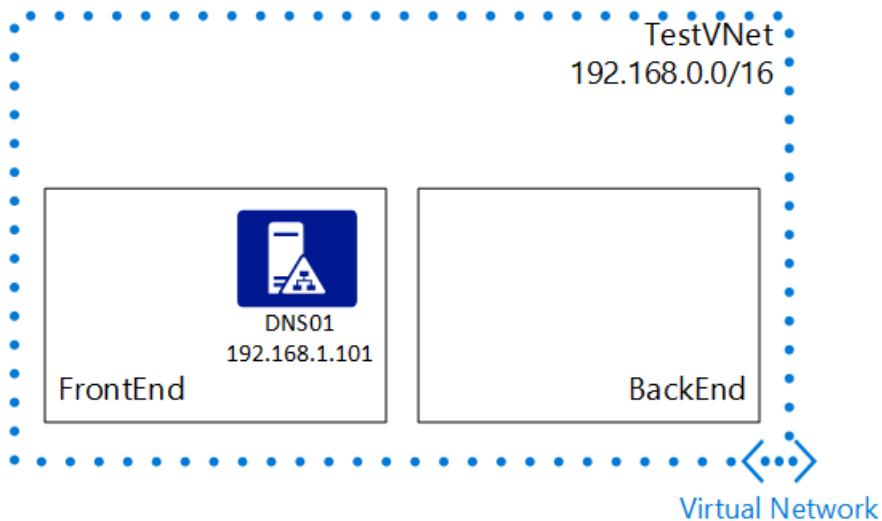
# Configure a private IP address for a VM using the Azure portal

11/2/2020 • 3 minutes to read • [Edit Online](#)

A virtual machine (VM) is automatically assigned a private IP address from a range that you specify, based on the subnet in which the VM is deployed. The VM retains the address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want a specific IP address from the subnet assigned to the VM, assign a static IP address.

## Scenario

To better illustrate how to configure a static IP address for a VM, this document uses this scenario:



In this scenario, you create a VM named *DNS01* in the *FrontEnd* subnet, and then set it to use a static IP address of *192.168.1.101*.

The following sample steps expect a simple environment to already be created. If you want to run the steps as they're displayed in this document, first [create a virtual network](#). However, in step 3, use these values instead:

SETTING	VALUE
Name	<i>TestVNet</i>
Address space	<i>192.168.0.0/16</i>
Resource group	<b>TestRG</b> (if necessary, select <b>Create new</b> to create it)
Subnet - Name	<i>FrontEnd</i>
Subnet - Address range	<i>192.168.1.0/24</i>

## Create a VM for testing static private IP addresses

When you create a VM in Resource Manager deployment mode, you can't set a static private IP address using the Azure portal. Instead, you create the VM first. Then you can set its private IP to be static.

To create a VM named *DNS01* in the *FrontEnd* subnet of a virtual network named *TestVNet*, follow these steps:

1. From the [Azure portal](#) menu, select **Create a resource**.

The screenshot shows the Microsoft Azure portal interface. On the left, there is a navigation sidebar with various service icons and links. A red box highlights the 'Create a resource' button, which is located at the top of the sidebar. The main content area displays several service tiles: Apps, Dev/Test Labs, All resources, CDN profiles, Service providers, Virtual networks, and Azure Cosmos DB. Below these tiles, there is a table with one row showing 'Subscription' and 'Last Viewed' (2 mo ago). At the bottom of the page, there are two promotional cards: 'Azure Monitor' (Monitor your apps and) and 'Security Center' (Secure your apps and).

2. Select **Compute > Virtual machine**.

The screenshot shows the Microsoft Azure 'New' blade. At the top, there's a search bar labeled 'Search resources, services, and docs (G+/-)'. Below it, a 'Home > New' breadcrumb path is shown. A large search bar says 'Search the Marketplace'. Under 'Azure Marketplace', there are two tabs: 'See all' and 'Featured'. The 'Featured' tab is selected. A red box highlights the 'Compute' category, which is also selected. To the right of the 'Compute' category, a red box highlights the 'Virtual machine' option, which is also selected. A cursor arrow points towards the 'Virtual machine' option.

3. In **Basics**, specify values for items as described in the following table. Then select **Next : Disks** and then **Next : Networking**.

ITEM	VALUE
Subscription	Your current subscription
Resource group	TestRG (select from dropdown list)
Virtual machine name	DNS01
Region	(US) East US
Image	Windows Server 2019 Datacenter
Size	VM Size of B1ls, Offering of Standard
Username	The user name of your administrator account
Password	The password for your administrator account's user name
Confirm password	The password again

## Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.

Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Contoso
Resource group *	TestRG
<a href="#">Create new</a>	

### Instance details

Virtual machine name *	DNS01
Region *	(US) East US
Availability options	No infrastructure redundancy required
Image *	Windows Server 2019 Datacenter
<a href="#">Browse all public and private images</a>	
Azure Spot instance	<input type="radio"/> Yes <input checked="" type="radio"/> No

Size *	<b>Standard B1ls</b> 1 vcpu, 0.5 GiB memory (\$7.15/month) <a href="#">Change size</a>
<b>Administrator account</b>	
Username *	adminuser
Password *	*****
Confirm password *	*****

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \*  None  Allow selected ports

Select inbound ports \* [RDP \(3389\)](#)

**⚠️** This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

[Review + create](#)

< Previous

Next : Disks >

4. In **Networking**, specify values for items as described in the following table, and then select **Next**.

ITEM	VALUE
Virtual network	TestVNet

ITEM	VALUE
Subnet	FrontEnd

Home > New > Create a virtual machine

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

**Network interface**

When creating a virtual machine, a network interface will be created for you.

**Virtual network \*** [①](#) TestVNet [▼](#) [Create new](#)

**Subnet \*** [①](#) FrontEnd (192.168.1.0/24) [▼](#) [Manage subnet configuration](#)

**Public IP** [①](#) (new) DNS01-ip [▼](#) [Create new](#)

**NIC network security group** [①](#)  None  Basic  Advanced

**Public inbound ports \*** [①](#)  None  Allow selected ports

**Select inbound ports \*** RDP (3389) [▼](#)

**Accelerated networking** [①](#)  On  Off

The selected VM size does not support accelerated networking.

**Load balancing**

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution?  Yes  No

[Review + create](#) [◀ Previous](#) [Next : Management >](#)

5. In **Management**, under **Diagnostics storage account**, choose **vnetstorage**. If that storage account

doesn't appear in the list, select **Create new**, specify a Name of *vnetstorage*, and select OK. Finally, select **Review + create**.

Home > New > Create a virtual machine

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Configure monitoring and management options for your VM.

**Azure Security Center**

Azure Security Center provides unified security management and advanced threat protection across hybrid cloud workloads.

[Learn more](#)

✓ Your subscription is protected by Azure Security Center standard plan.

**Monitoring**

Enable detailed monitoring (preview)  On  Off

Boot diagnostics  On  Off

OS guest diagnostics  On  Off

Diagnostics storage account \*  vnetstorage  Create new

**Identity**

System assigned managed identity  On  Off

**Azure Active Directory**

Login with AAD credentials (Preview)  On  Off

**Auto-shutdown**

Enable auto-shutdown  On  Off

**Backup**

Enable backup  On  Off

**Review + create** < Previous Next : Advanced >

6. In **Review + create**, review the overview information, and then select **Create**.

Create a virtual machine X✓ Validation passed[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

## PRODUCT DETAILS

Standard B1ls

Subscription credits apply (i)

by Microsoft

**0.0098 USD/hr**[Terms of use](#) | [Privacy policy](#)[Pricing for other VM sizes](#)

## TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

⚠ You have set RDP port(s) open to the internet. This is only recommended for testing. If you want to change this setting, go back to Basics tab.

## Basics

Subscription	Contoso
Resource group	TestRG
Virtual machine name	DNS01
Region	(US) East US
Availability options	No infrastructure redundancy required
Username	adminuser
Public inbound ports	RDP
Already have a Windows Server license?	No
Azure Spot	No

## Disks

OS disk type	Premium SSD
Use managed disks	Yes
Use ephemeral OS disk	No

## Networking

Virtual network ▼Create

&lt; Previous

Next &gt;

[Download a template for automation](#)

The following message appears once the VM is created.

 Your deployment is complete

Deployment name: CreateVm-MicrosoftWindowsServer.WindowsServer2019-1 Deployment ID: 01ed9780-2020-4444-bb7e-175e0eb932a2  
Subscription: Contoso Start time: 2/11/2020, 8:59:06 PM  
Resource group: TestRG Correlation ID: 01ed9780-2020-4444-bb7e-175e0eb932a2

Deployment details [\(Download\)](#)

Next steps

Setup auto-shutdown Recommended

Monitor VM health, performance and network dependencies Recommended

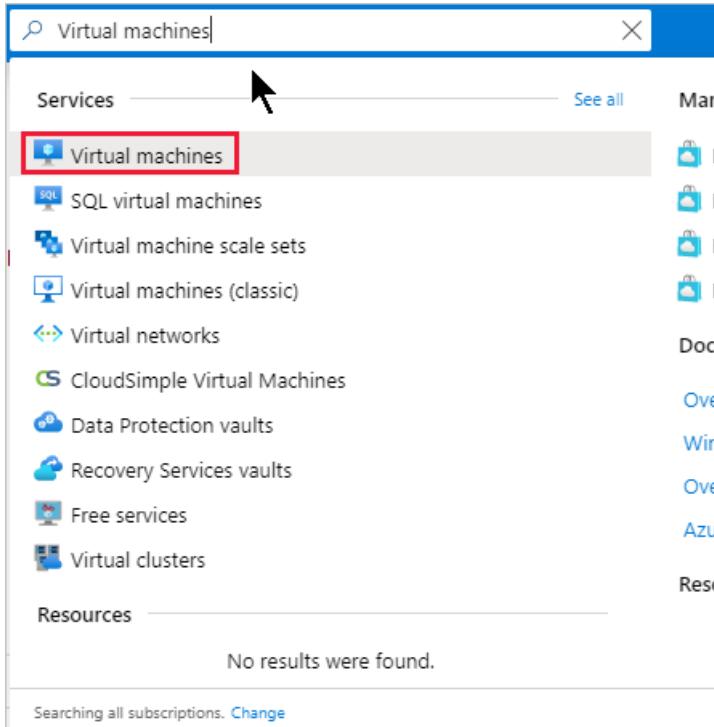
Run a script inside the virtual machine Recommended

[Go to resource](#) [Create another VM](#)

## Retrieve private IP address information for a VM

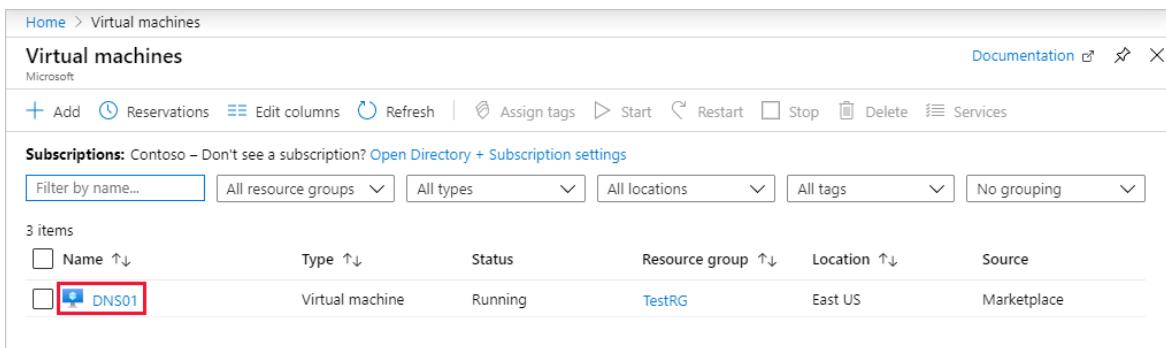
To view the private IP address information for your new VM:

1. Go to the [Azure portal](#) to find your VM. Search for and select **Virtual machines**.



The screenshot shows the Azure portal's search interface. A search bar at the top contains the text "Virtual machines". Below the search bar, there are two main sections: "Services" and "Resources". Under "Services", several items are listed: "Virtual machines" (which is highlighted with a red box), "SQL virtual machines", "Virtual machine scale sets", "Virtual machines (classic)", "Virtual networks", "CloudSimple Virtual Machines", "Data Protection vaults", "Recovery Services vaults", "Free services", and "Virtual clusters". Under "Resources", it says "No results were found." At the bottom of the page, there is a note "Searching all subscriptions. [Change](#)".

2. Select the name of your new VM (DNS01).



The screenshot shows the "Virtual machines" list page in the Azure portal. The title bar includes "Home > Virtual machines" and "Virtual machines Microsoft". The top navigation bar has links for "Add", "Reservations", "Edit columns", "Refresh", "Assign tags", "Start", "Restart", "Stop", "Delete", and "Services". Below this, there is a "Subscriptions" section showing "Contoso – Don't see a subscription? Open Directory + Subscription settings". There are several filter options: "Filter by name...", "All resource groups", "All types", "All locations", "All tags", and "No grouping". The main table displays 3 items, with a header row for "Name", "Type", "Status", "Resource group", "Location", and "Source". One item, "DNS01", is highlighted with a red box.

Name	Type	Status	Resource group	Location	Source
DNS01	Virtual machine	Running	TestRG	East US	Marketplace

3. Choose **Networking**, and select the sole network interface listed.

**Network Interface: dns01315**

Priority	Name	Port	Protocol	Source
300	RDP	3389	TCP	Any
65000	AllowVnetInBound	Any	Any	VirtualNetwork
65001	AllowAzureLoadBalanc...	Any	Any	AzureLoadBalancer
65500	DenyAllInBound	Any	Any	Any

4. Choose IP configurations, and select the IP configuration listed in the table.

Name	IP Version	Type	Private IP address
ipconfig1	IPv4	Primary	192.168.1.4 (Dynamic)

5. In Private IP address settings, under the TestVNet/FrontEnd virtual network/subnet, note the Assignment value (Dynamic or Static) and the IP address.

Home > Virtual machines > DNS01 - Networking > dns01315 - IP configurations > ipconfig1

## ipconfig1

dns01315

Save Discard

Public IP address settings

Public IP address

Disabled Enabled

\*IP address >  
DNS01-ip (52.170.118.57)

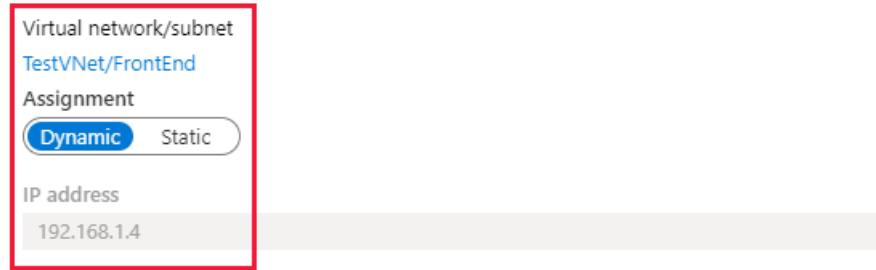
Private IP address settings

Virtual network/subnet  
TestVNet/FrontEnd

Assignment

Dynamic Static

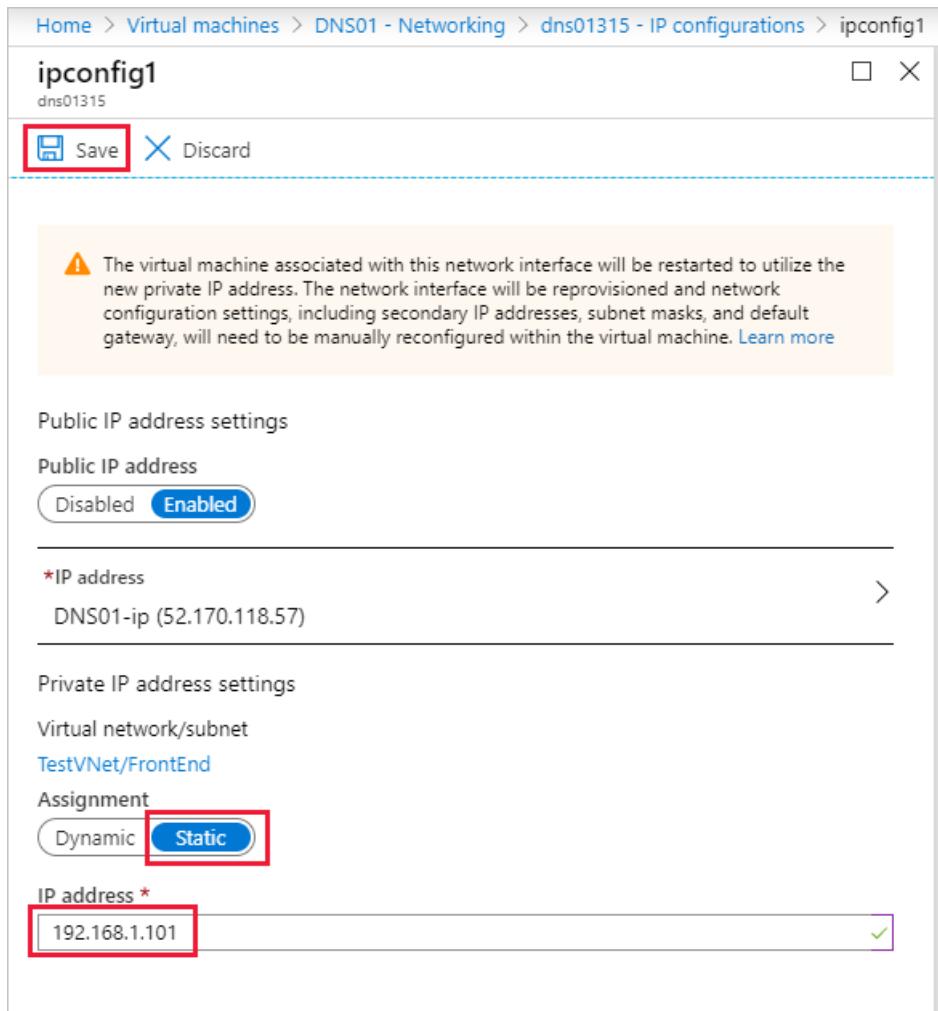
IP address  
192.168.1.4



## Add a static private IP address to an existing VM

To add a static private IP address to your new VM:

1. In the IP configuration page, set the assignment for your private IP address to **Static**.
2. Change your private **IP address** to **192.168.1.101**, and then select **Save**.



#### NOTE

If you notice after selecting **Save** that the assignment is still set to **Dynamic**, the IP address you typed is already in use. Try another IP address.

## Remove a static private IP address from a VM

To remove the static private IP address from your VM:

In the IP configuration page, set the assignment for your private IP address to **Dynamic**, and then select **Save**.

## Set IP addresses within the operating system

From within the operating system of a VM, you shouldn't statically assign the *private* IP that's assigned to the Azure VM. Only do static assignment of a private IP when it's necessary, such as when [assigning many IP addresses to VMs](#). If you manually set the private IP address within the operating system, make sure it matches the private IP address assigned to the Azure [network interface](#). Otherwise, you can lose connectivity to the VM. Learn more about [private IP address settings](#).

Also, you should never manually assign the *public* IP address assigned to an Azure virtual machine within the virtual machine's operating system.

## Next steps

Learn about managing [IP address settings](#).

# Create a virtual machine with a static private IP address using PowerShell

2/18/2021 • 3 minutes to read • [Edit Online](#)

You can create a virtual machine (VM) with a static private IP address. Assign a static private IP address, rather than a dynamic address, if you want to select which address from a subnet is assigned to a VM. Learn more about [static private IP addresses](#). To change a private IP address assigned to an existing VM from dynamic to static, or to work with public IP addresses, see [Add, change, or remove IP addresses](#).

## NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Create a virtual machine

You can complete the following steps from your local computer or by using the Azure Cloud Shell. To use your local computer, ensure you have the [Azure PowerShell installed](#). To use the Azure Cloud Shell, select **Try It** in the top right corner of any command box that follows. The Cloud Shell signs you into Azure.

1. If using the Cloud Shell, skip to step 2. Open a command session and sign into Azure with

```
Connect-AzAccount .
```

2. Create a resource group with the [New-AzResourceGroup](#) command. The following example creates a resource group in the East US Azure region:

```
$RgName = "myResourceGroup"
$Location = "eastus"
New-AzResourceGroup -Name $RgName -Location $Location
```

3. Create a subnet configuration and virtual network with the [New-AzVirtualNetworkSubnetConfig](#) and [New-AzVirtualNetwork](#) commands:

```
Create a subnet configuration
$SubnetConfig = New-AzVirtualNetworkSubnetConfig `
-Name MySubnet `
-AddressPrefix 10.0.0.0/24

Create a virtual network
$VNet = New-AzVirtualNetwork `
-ResourceGroupName $RgName `
-Location $Location `
-Name MyVNet `
-AddressPrefix 10.0.0.0/16 `
-Subnet $subnetConfig

Get the subnet object for use in a later step.
$Subnet = Get-AzVirtualNetworkSubnetConfig -Name $SubnetConfig.Name -VirtualNetwork $VNet
```

4. Create a network interface in the virtual network and assign a private IP address from the subnet to the

network interface with the [New-AzNetworkInterfaceIpConfig](#) and [New-AzNetworkInterface](#) commands:

```
$IpConfigName1 = "IPConfig-1"
$IpConfig1 = New-AzNetworkInterfaceIpConfig `
 -Name $IpConfigName1 `
 -Subnet $Subnet `
 -PrivateIpAddress 10.0.0.4 `
 -Primary

$NIC = New-AzNetworkInterface `
 -Name MyNIC `
 -ResourceGroupName $RgName `
 -Location $Location `
 -IpConfiguration $IpConfig1
```

5. Create a VM configuration with [New-AzVMConfig](#), and then create the VM with [New-AzVM](#). When prompted, provide a username and password to be used as the sign in credentials for the VM:

```
$VirtualMachine = New-AzVMConfig -VMName MyVM -VMSize "Standard_DS3"
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -ComputerName MyServerVM -
ProvisionVMAgent -EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName 'MicrosoftWindowsServer' -
Offer 'WindowsServer' -Skus '2012-R2-Datacenter' -Version latest
New-AzVM -ResourceGroupName $RgName -Location $Location -VM $VirtualMachine -Verbose
```

#### WARNING

Though you can add private IP address settings to the operating system, we recommend not doing so until after reading [Add a private IP address to an operating system](#).

#### IMPORTANT

To access the VM from the internet, you must assign a public IP address to the VM. You can also change a dynamic private IP address assignment to a static assignment. For details, see [Add or change IP addresses](#). Additionally, it's recommended that you limit the network traffic to your VM by associating a network security group to the network interface, the subnet you created the network interface in, or both. For details, see [Manage network security groups](#).

## Clean up resources

When no longer needed, you can use [Remove-AzResourceGroup](#) to remove the resource group and all of the resources it contains:

```
Remove-AzResourceGroup -Name myResourceGroup -Force
```

## Next steps

- Learn more about [private IP addresses](#) and assigning a [static private IP address](#) to an Azure virtual machine.
- Learn more about creating [Linux](#) and [Windows](#) virtual machines.

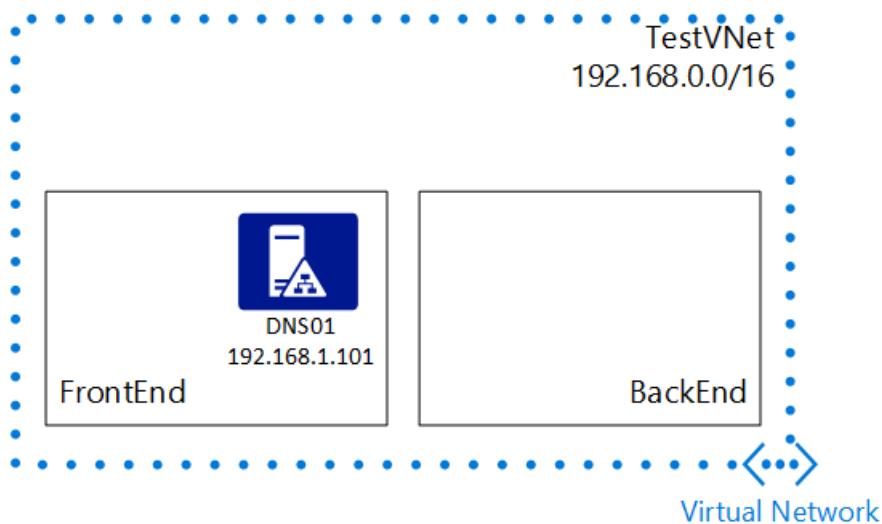
# Configure private IP addresses for a virtual machine using the Azure CLI

11/2/2020 • 4 minutes to read • [Edit Online](#)

A virtual machine (VM) is automatically assigned a private IP address from a range that you specify, based on the subnet in which the VM is deployed. The VM retains the address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want a specific IP address from the subnet assigned to the VM, assign a static IP address.

## Scenario

To better illustrate how to configure a static IP address for a VM, this document uses this scenario:



In this scenario, you create a VM named *DNS01* in the *FrontEnd* subnet, and then set it to use a static IP address of *192.168.1.101*.

### NOTE

The following sample Azure CLI commands expect an existing simple environment. If you want to run the commands as they are displayed in this document, first build the test environment described in [create a vnet](#).

## Specify a static private IP address when creating a VM

To create a VM named *DNS01* in the *FrontEnd* subnet of a VNet named *TestVNet* with a static private IP of *192.168.1.101*, complete the following steps:

1. If you haven't yet, install and configure the latest [Azure CLI](#) and log in to an Azure account using [az login](#).
2. Run the `az network nic create` command to create a NIC with a static private IP. The list shown after the output explains the parameters used.

```
az network nic create \
--resource-group TestRG \
--name TestNIC \
--location centralus \
--subnet FrontEnd \
--private-ip-address 192.168.1.101 \
--vnet-name TestVNet
```

Expected output:

```
{
 "newNIC": {
 "dnsSettings": {
 "appliedDnsServers": [],
 "dnsServers": []
 },
 "enableIPForwarding": false,
 "ipConfigurations": [
 {
 "etag": "W/\"<guid>\\"",
 "id": "/subscriptions/<guid>/resourceGroups/TestRG/providers/Microsoft.Network/networkInterfaces/TestNIC/ipConfigurations/ipconfig1",
 "name": "ipconfig1",
 "properties": {
 "primary": true,
 "privateIPAddress": "192.168.1.101",
 "privateIPAllocationMethod": "Static",
 "provisioningState": "Succeeded",
 "subnet": {
 "id": "/subscriptions/<guid>/resourceGroups/TestRG/providers/Microsoft.Network/virtualNetworks/TestVNet/subnets/FrontEnd",
 "resourceGroup": "TestRG"
 }
 },
 "resourceGroup": "TestRG"
 }
],
 "provisioningState": "Succeeded",
 "resourceGuid": "<guid>"
 }
}
```

Parameters:

- `--private-ip-address` : Static private IP address for the NIC.
- `--vnet-name` : Name of the VNet in which to create the NIC.
- `--subnet` : Name of the subnet in which to create the NIC.

3. Run the [azure vm create](#) command to create the VM using the public IP and NIC created previously. The list shown after the output explains the parameters used.

```
az vm create \
--resource-group TestRG \
--name DNS01 \
--location centralus \
--image Debian \
--admin-username adminuser \
--ssh-key-value ~/.ssh/id_rsa.pub \
--nics TestNIC
```

Expected output:

```
{
 "fqdns": "",
 "id":
 "/subscriptions/<guid>/resourceGroups/TestRG/providers/Microsoft.Compute/virtualMachines/DNS01",
 "location": "centralus",
 "macAddress": "00-0D-3A-92-C1-66",
 "powerState": "VM running",
 "privateIpAddress": "192.168.1.101",
 "publicIpAddress": "",
 "resourceGroup": "TestRG"
}
```

Parameters other than the basic `az vm create` parameters.

- `--nics` : Name of the NIC to which the VM is attached.

It's recommended that you do not statically assign the private IP assigned to the Azure virtual machine within the operating system of a VM, unless necessary, such as when [assigning multiple IP addresses to a Windows VM](#). If you do manually set the private IP address within the operating system, ensure that it is the same address as the private IP address assigned to the Azure [network interface](#), or you can lose connectivity to the virtual machine. Learn more about [private IP address](#) settings.

## Retrieve static private IP address information for a VM

Run the following Azure CLI command to observe the values for *Private IP alloc-method* and *Private IP address*:

```
az vm show -g TestRG -n DNS01 --show-details --query 'privateIps'
```

Expected output:

```
"192.168.1.101"
```

To display the specific IP information of the NIC for that VM, query the NIC specifically:

```
az network nic show \
-g testrg \
-n testnic \
--query 'ipConfigurations[0].{PrivateAddress:privateIpAddress,IPVer:privateIpAddressVersion,IpAllocMethod:p
rivateIpAllocationMethod,PublicAddress:publicIpAddress}'
```

The output is something like:

```
{
 "IPVer": "IPv4",
 "IpAllocMethod": "Static",
 "PrivateAddress": "192.168.1.101",
 "PublicAddress": null
}
```

## Remove a static private IP address from a VM

You cannot remove a static private IP address from a NIC in Azure CLI for Azure Resource Manager deployments. You must:

- Create a new NIC that uses a dynamic IP
- Set the NIC on the VM do the newly created NIC.

To change the NIC for the VM used in the previous commands, complete the following steps:

1. Run the **azure network nic create** command to create a new NIC using dynamic IP allocation with a new IP address. Because no IP address is specified, the allocation method is **Dynamic**.

```
az network nic create \
--resource-group TestRG \
--name TestNIC2 \
--location centralus \
--subnet FrontEnd \
--vnet-name TestVNet
```

Expected output:

```
{
 "newNIC": {
 "dnsSettings": {
 "appliedDnsServers": [],
 "dnsServers": []
 },
 "enableIPForwarding": false,
 "ipConfigurations": [
 {
 "etag": "W/\"<guid>\\"",
 "id": "/subscriptions/<guid>/resourceGroups/TestRG/providers/Microsoft.Network/networkInterfaces/TestNIC2/ipConfigurations/ipconfig1",
 "name": "ipconfig1",
 "properties": {
 "primary": true,
 "privateIPAddress": "192.168.1.4",
 "privateIPAllocationMethod": "Dynamic",
 "provisioningState": "Succeeded",
 "subnet": {
 "id": "/subscriptions/<guid>/resourceGroups/TestRG/providers/Microsoft.Network/virtualNetworks/TestVNet/subnets/FrontEnd",
 "resourceGroup": "TestRG"
 }
 },
 "resourceGroup": "TestRG"
 }
],
 "provisioningState": "Succeeded",
 "resourceGuid": "0808a61c-476f-4d08-98ee-0fa83671b010"
 }
}
```

2. Run the **azure vm set** command to change the NIC used by the VM.

```
az vm nic set --resource-group TestRG --vm-name DNS01 --nics TestNIC2
```

Expected output:

```
[
 {
 "id": "/subscriptions/0e220bf6-5caa-4e9f-8383-
51f16b6c109f/resourceGroups/TestRG/providers/Microsoft.Network/networkInterfaces/TestNIC3",
 "primary": true,
 "resourceGroup": "TestRG"
 }
]
```

**NOTE**

If the VM is large enough to have more than one NIC, run the **azure network nic delete** command to delete the old NIC.

## Next steps

Learn about managing [IP address settings](#).

# Configure private IP addresses for a virtual machine (Classic) using the Azure portal

2/18/2021 • 3 minutes to read • [Edit Online](#)

A virtual machine (VM) is automatically assigned a private IP address from a range that you specify, based on the subnet in which the VM is deployed. The VM retains the address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want a specific IP address from the subnet assigned to the VM, assign a static IP address.

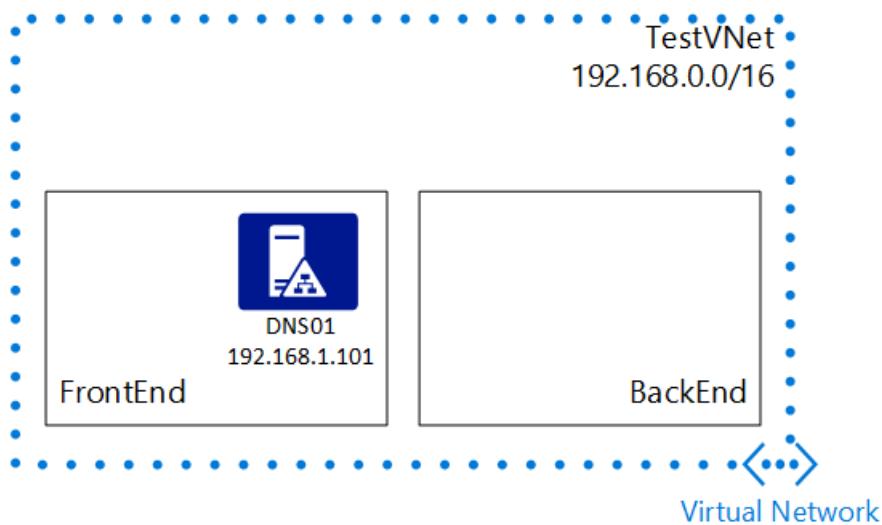
## IMPORTANT

Azure currently has two deployment models: Azure Resource Manager and classic. Make sure you understand [deployment models and tools](#) before you work with any Azure resource. To view the documentation for different tools, select the tabs at the beginning of this article.

This article covers the classic deployment model. You can also [manage a static private IP address in the Resource Manager deployment model](#).

## Scenario

To better illustrate how to configure a static IP address for a VM, this document uses this scenario:



In this scenario, you create a VM named *DNS01* in the *FrontEnd* subnet, and then set it to use a static IP address of *192.168.1.101*.

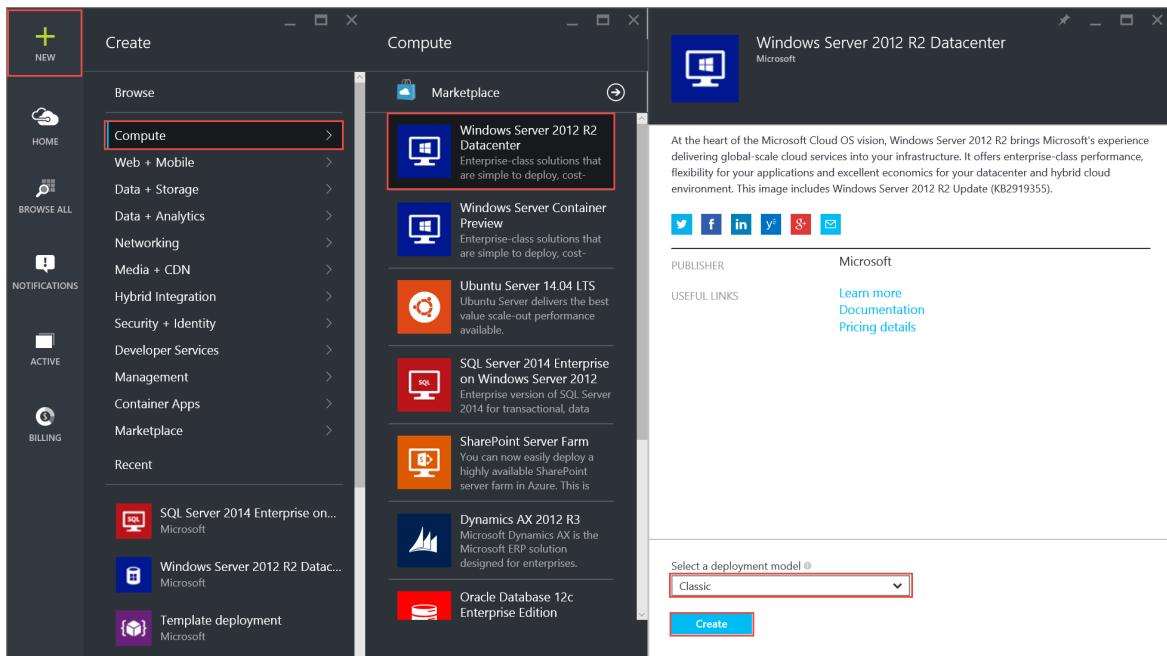
The sample steps that follow expect a simple environment already created. If you want to run the steps as they are displayed in this document, first build the test environment described in [create a vnet](#).

## How to specify a static private IP address when creating a VM

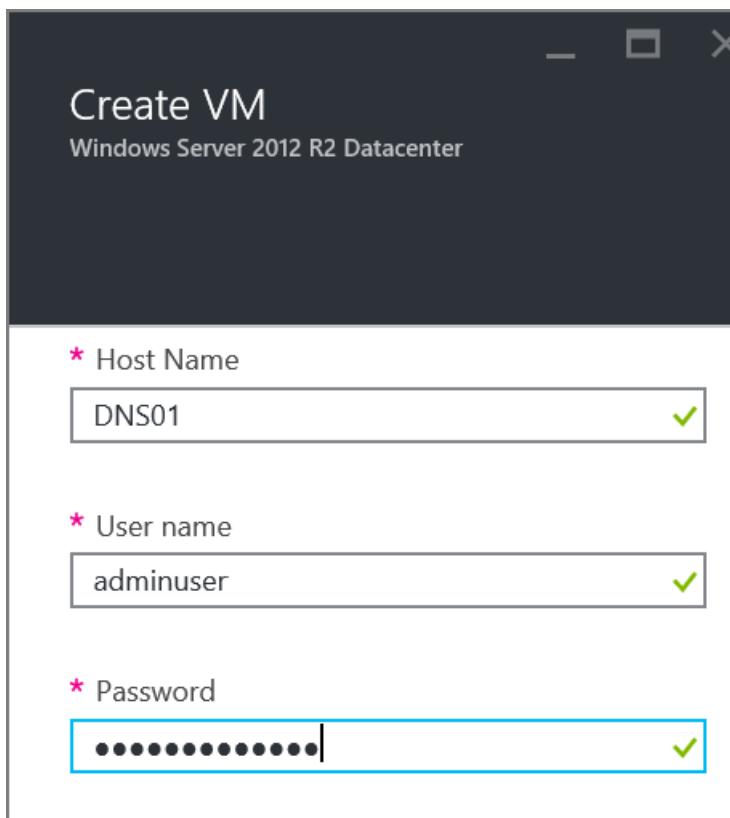
To create a VM named *DNS01* in the *FrontEnd* subnet of a VNet named *TestVNet* with a static private IP of *192.168.1.101*, complete the following steps:

1. From a browser, navigate to <https://portal.azure.com> and, if necessary, sign in with your Azure account.
2. Select NEW > Compute > Windows Server 2012 R2 Datacenter, notice that the Select a

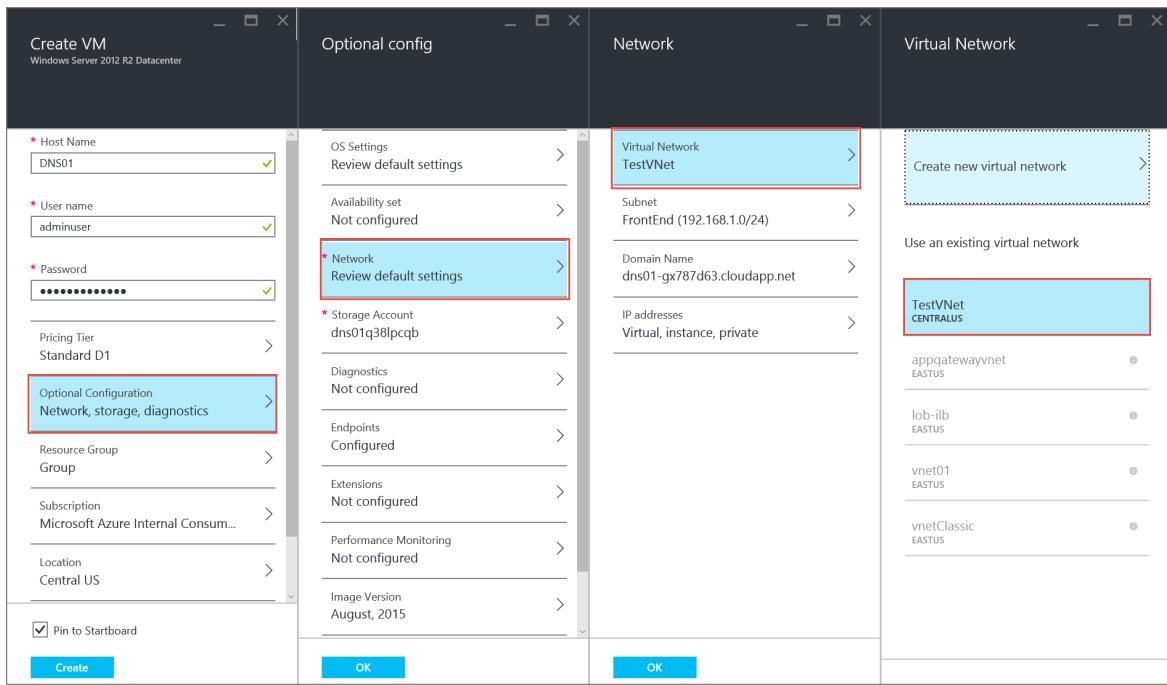
deployment model list already shows **Classic**, and then select **Create**.



3. Under **Create VM**, enter the name of the VM to be created (*DNS01* in the scenario), the local administrator account, and password.



4. Select **Optional Configuration > Network > Virtual Network**, and then select **TestVNet**. If **TestVNet** is not available, make sure you are using the *Central US* location and have created the test environment described at the beginning of this article.



5. Under **Network**, make sure the subnet currently selected is *FrontEnd*, then select **IP addresses**, under **IP address assignment** select **Static**, and then enter **192.168.1.101** for **IP Address** as seen below.

The screenshot shows two side-by-side configuration panels in the Azure portal.

**Network Panel:**

- Virtual Network: TestVNet
- Subnet: FrontEnd (192.168.1.0/24)
- Domain Name: dns01-gx787d63.cloudapp.net
- IP addresses: Virtual, instance, private (selected, highlighted with a blue background)

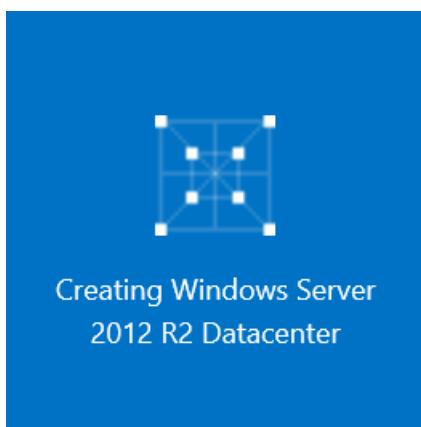
**IP addresses Panel:**

- Virtual IP address:**
  - Virtual IP address assignment: Dynamic (selected)
- Instance IP address:**
  - Instance IP address: On (selected)
  - Idle timeout (in minutes): 4
- Private IP address:**
  - IP address assignment: Static (selected)
  - IP Address\*: 192.168.1.101 (highlighted with a red border)

Both panels have an "OK" button at the bottom right.

6. Select OK under **IP addresses**, select OK under **Network**, and then select OK under **Optional config**.

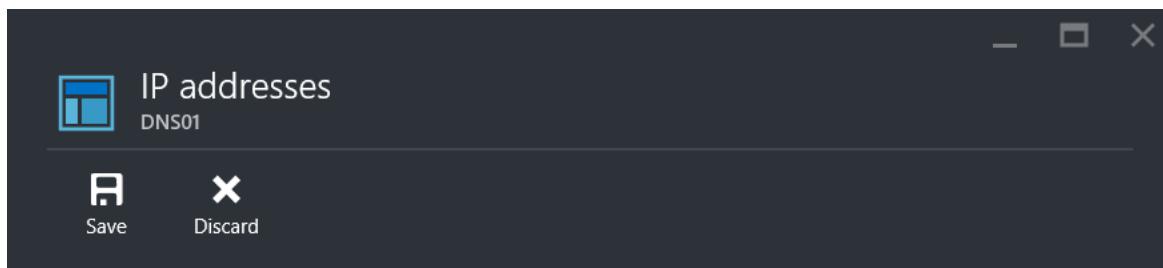
7. Under **Create VM**, select **Create**. Notice the tile below displayed in your dashboard:



How to retrieve static private IP address information for a VM

To view the static private IP address information for the VM created with the steps above, execute the steps below.

1. From the Azure portal, select **BROWSE ALL > Virtual machines (classic) > DNS01 > All settings > IP addresses** and notice the IP address assignment and IP address as seen below.



#### Virtual IP address

IP address assignment ⓘ	Dynamic
IP address	40.122.208.41

#### Instance IP address

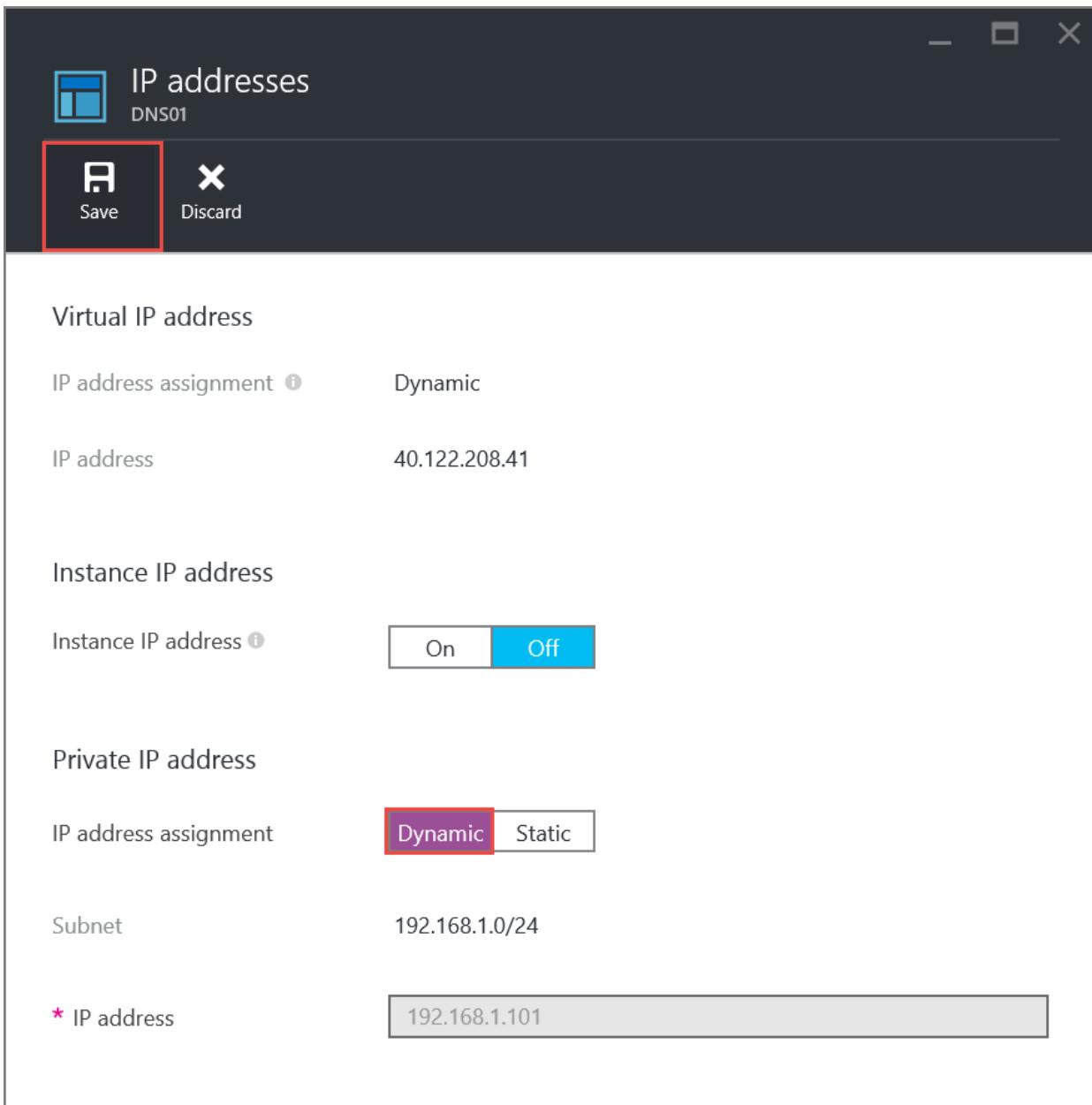
Instance IP address ⓘ	On	Off
-----------------------	----	-----

#### Private IP address

IP address assignment	Dynamic	Static
Subnet	192.168.1.0/24	
* IP address	192.168.1.101	✓

## How to remove a static private IP address from a VM

Under **IP addresses**, select **Dynamic** to the right of **IP address assignment**, select **Save**, and then select **Yes**, as shown in the following picture:



## How to add a static private IP address to an existing VM

1. Under **IP addresses**, shown previously, select **Static** to the right of **IP address assignment**.
2. Type *192.168.1.101* for **IP address**, select **Save**, and then select **Yes**.

## Set IP addresses within the operating system

It's recommended that you do not statically assign the private IP assigned to the Azure virtual machine within the operating system of a VM, unless necessary. If you do manually set the private IP address within the operating system, ensure that it is the same address as the private IP address assigned to the Azure VM, or you can lose connectivity to the virtual machine. You should never manually assign the public IP address assigned to an Azure virtual machine within the virtual machine's operating system.

## Next steps

- Learn about [reserved public IP](#) addresses.
- Learn about [instance-level public IP \(ILPIP\)](#) addresses.
- Consult the [Reserved IP REST APIs](#).

# Configure private IP addresses for a virtual machine (Classic) using PowerShell

2/18/2021 • 3 minutes to read • [Edit Online](#)

A virtual machine (VM) is automatically assigned a private IP address from a range that you specify, based on the subnet in which the VM is deployed. The VM retains the address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want a specific IP address from the subnet assigned to the VM, assign a static IP address.

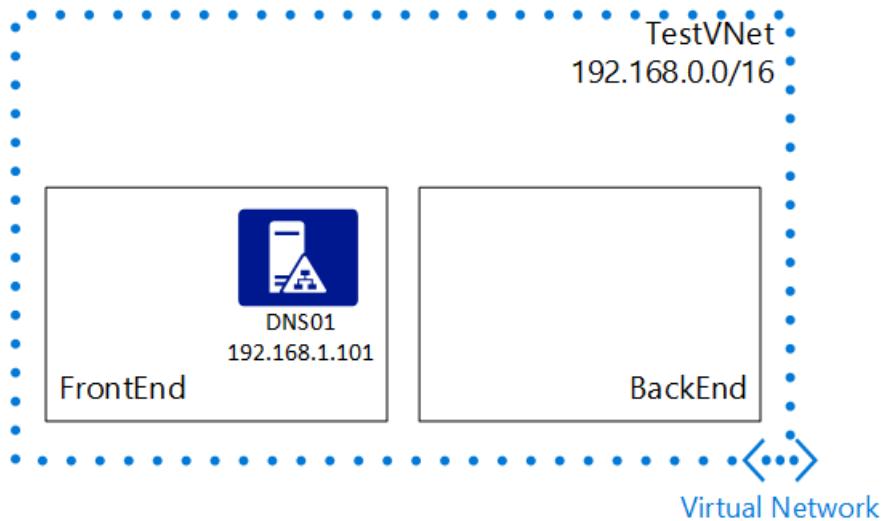
## IMPORTANT

Azure currently has two deployment models: Azure Resource Manager and classic. Make sure you understand [deployment models and tools](#) before you work with any Azure resource. To view the documentation for different tools, select the tabs at the beginning of this article.

This article covers the classic deployment model. You can also [manage a static private IP address in the Resource Manager deployment model](#).

## Scenario

To better illustrate how to configure a static IP address for a VM, this document uses this scenario:



In this scenario, you create a VM named *DNS01* in the *FrontEnd* subnet, and then set it to use a static IP address of *192.168.1.101*.

The sample PowerShell commands below expect a simple environment already created. If you want to run the commands as they are displayed in this document, first build the test environment described in [Create a VNet](#).

## How to verify if a specific IP address is available

To verify if the IP address *192.168.1.101* is available in a VNet named *TestVNet*, run the following PowerShell command and verify the value for *IsAvailable*.

```
Test-AzureStaticVNetIP -VNetName TestVNet -IPAddress 192.168.1.101
```

Expected output:

```
IsAvailable : True
AvailableAddresses : {}
OperationDescription : Test-AzureStaticVNetIP
OperationId : fd3097e1-5f4b-9cac-8afa-bba1e3492609
OperationStatus : Succeeded
```

## How to specify a static private IP address when creating a VM

The PowerShell script below creates a new cloud service named *TestService*, then retrieves an image from Azure, creates a VM named *DNS01* in the new cloud service using the retrieved image, sets the VM to be in a subnet named *FrontEnd*, and sets *192.168.1.7* as a static private IP address for the VM:

```
New-AzureService -ServiceName TestService -Location "Central US"
$image = Get-AzureVMImage | where {$_.ImageName -like "*RightImage-Windows-2012R2-x64*"}
New-AzureVMConfig -Name DNS01 -InstanceSize Small -ImageName $image.ImageName |
 Add-AzureProvisioningConfig -Windows -AdminUsername adminuser -Password MyP@ssw0rd!! |
 Set-AzureSubnet -SubnetNames FrontEnd |
 Set-AzureStaticVNetIP -IPAddress 192.168.1.7 |
New-AzureVM -ServiceName TestService -VNetName TestVNet
```

Expected output:

```
WARNING: No deployment found in service: 'TestService'.
OperationDescription OperationId OperationStatus
----- -----
New-AzureService fcf705f1-d902-011c-95c7-b690735e7412 Succeeded
New-AzureVM 3b99a86d-84f8-04e5-888e-b6fc3c73c4b9 Succeeded
```

## How to retrieve static private IP address information for a VM

To view the static private IP address information for the VM created with the script above, run the following PowerShell command and observe the values for *IpAddress*:

```
Get-AzureVM -Name DNS01 -ServiceName TestService
```

Expected output:

```

DeploymentName : TestService
Name : DNS01
Label :
VM : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.PersistentVM
InstanceState : Provisioning
IpAddress : 192.168.1.7
InstanceStateDetails: Windows is preparing your computer for first use...
PowerState : Started
InstanceErrorCode :
InstanceFaultDomain: 0
InstanceName : DNS01
InstanceUpgradeDomain: 0
InstanceSize : Small
HostName : rsR2-797
AvailabilitySetName:
DNSName : http://testservice000.cloudapp.net/
Status : Provisioning
GuestAgentStatus : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.GuestAgentStatus
ResourceExtensionStatusList: {Microsoft.Compute.BGInfo}
PublicIPAddress :
PublicIPName :
NetworkInterfaces : {}
ServiceName : TestService
OperationDescription: Get-AzureVM
OperationId : 34c1560a62f0901ab75cde4fed8e8bd1
OperationStatus : OK

```

## How to remove a static private IP address from a VM

To remove the static private IP address added to the VM in the script above, run the following PowerShell command:

```

Get-AzureVM -ServiceName TestService -Name DNS01 |
 Remove-AzureStaticVNetIP |
 Update-AzureVM

```

Expected output:

OperationDescription	OperationId	OperationStatus
Update-AzureVM	052fa6f6-1483-0ede-a7bf-14f91f805483	Succeeded

## How to add a static private IP address to an existing VM

To add a static private IP address to the VM created using the script above, run the following command:

```

Get-AzureVM -ServiceName TestService -Name DNS01 |
 Set-AzureStaticVNetIP -IPAddress 192.168.1.7 |
 Update-AzureVM

```

Expected output:

OperationDescription	OperationId	OperationStatus
Update-AzureVM	77d8cae2-87e6-0ead-9738-7c7dae9810cb	Succeeded

## Set IP addresses within the operating system

It's recommended that you do not statically assign the private IP assigned to the Azure virtual machine within the operating system of a VM, unless necessary. If you do manually set the private IP address within the operating system, ensure that it is the same address as the private IP address assigned to the Azure VM, or you can lose connectivity to the virtual machine. You should never manually assign the public IP address assigned to an Azure virtual machine within the virtual machine's operating system.

## Next steps

- Learn about [reserved public IP](#) addresses.
- Learn about [instance-level public IP \(ILPIP\)](#) addresses.
- Consult the [Reserved IP REST APIs](#).

# Assign multiple IP addresses to virtual machines using the Azure portal

2/18/2021 • 16 minutes to read • [Edit Online](#)

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it. Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any of the private IP addresses for any of the NICs to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary NIC could be added to a back-end pool. To learn more about how to load balance multiple IP configurations, read the [Load balancing multiple IP configurations](#) article.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. A public IP address resource has either a dynamic or static public IP address assigned to it. To learn more about IP addresses in Azure, read the [IP addresses in Azure](#) article.

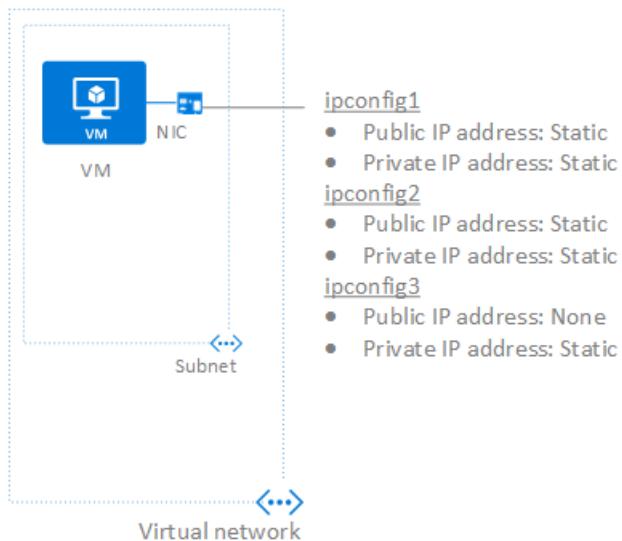
There is a limit to how many private IP addresses can be assigned to a NIC. There is also a limit to how many public IP addresses that can be used in an Azure subscription. See the [Azure limits](#) article for details.

This article explains how to create a virtual machine (VM) through the Azure Resource Manager deployment model using the Azure portal. Multiple IP addresses cannot be assigned to resources created through the classic deployment model. To learn more about Azure deployment models, read the [Understand deployment models](#) article.

## Scenario

A VM with a single NIC is created and connected to a virtual network. The VM requires three different *private* IP addresses and two *public* IP addresses. The IP addresses are assigned to the following IP configurations:

- **IPConfig-1:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-2:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-3:** Assigns a *static* private IP address and no public IP address.



The IP configurations are associated to the NIC when the NIC is created and the NIC is attached to the VM when the VM is created. The types of IP addresses used for the scenario are for illustration. You can assign whatever IP address and assignment types you require.

#### NOTE

Though the steps in this article assigns all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, read the [Create a VM with multiple NICs](#) article.

## Create a VM with multiple IP addresses

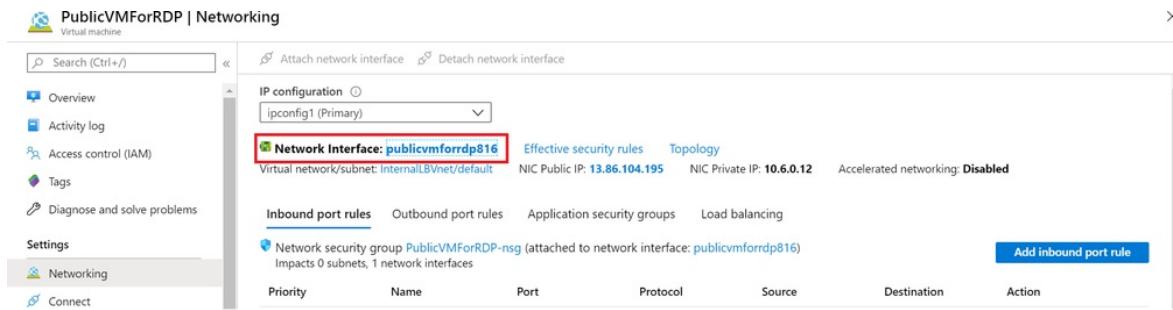
If you want to create a VM with multiple IP addresses, or a static private IP address, you must create it using PowerShell or the Azure CLI. To learn how, click the PowerShell or CLI options at the top of this article. You can create a VM with a single dynamic private IP address and (optionally) a single public IP address. Use the portal by following the steps in the [Create a Windows VM](#) or [Create a Linux VM](#) articles. After you create the VM, you can change the IP address type from dynamic to static and add additional IP addresses using the portal by following steps in the [Add IP addresses to a VM](#) section of this article.

## Add IP addresses to a VM

You can add private and public IP addresses to an Azure network interface by completing the steps that follow. The examples in the following sections assume that you already have a VM with the three IP configurations described in the [scenario](#), but it's not required.

### Core steps

1. Browse to the Azure portal at <https://portal.azure.com> and sign into it, if necessary.
2. In the portal, click **More services** > type *virtual machines* in the filter box, and then click **Virtual machines**.
3. In the **Virtual machines** pane, click the VM you want to add IP addresses to. Navigate to **Networking** Tab. Click **Network interface** on the page. As shown in the picture below:



4. In the **Network interface** pane, click the **IP configurations**.

5. In the pane that appears for the NIC you selected, click **IP configurations**. Click **Add**, complete the steps in one of sections that follow, based on the type of IP address you want to add, and then click **OK**.

### Add a private IP address

Complete the following steps to add a new private IP address:

1. Complete the steps in the [Core steps](#) section of this article and ensure you are on the **IP configurations** section of the VM Network Interface. Review the subnet shown as default (such as 10.0.0.0/24).
2. Click **Add**. In the **Add IP configuration** pane that appears, create an IP configuration named *IPConfig-4* with a new *Static* private IP address by picking a new number for the final octet, then click **OK**. (For the 10.0.0.0/24 subnet, an example IP would be *10.0.0.7*.)

#### NOTE

When adding a static IP address, you must specify an unused, valid address on the subnet the NIC is connected to. If the address you select is not available, the portal displays an X for the IP address and you must select a different one.

3. Once you click **OK**, the pane closes and you see the new IP configuration listed. Click **OK** to close the **Add IP configuration** pane.
4. You can click **Add** to add additional IP configurations, or close all open blades to finish adding IP addresses.
5. Add the private IP addresses to the VM operating system by completing the steps in the [Add IP addresses to a VM operating system](#) section of this article.

### Add a public IP address

A public IP address is added by associating a public IP address resource to either a new IP configuration or an existing IP configuration.

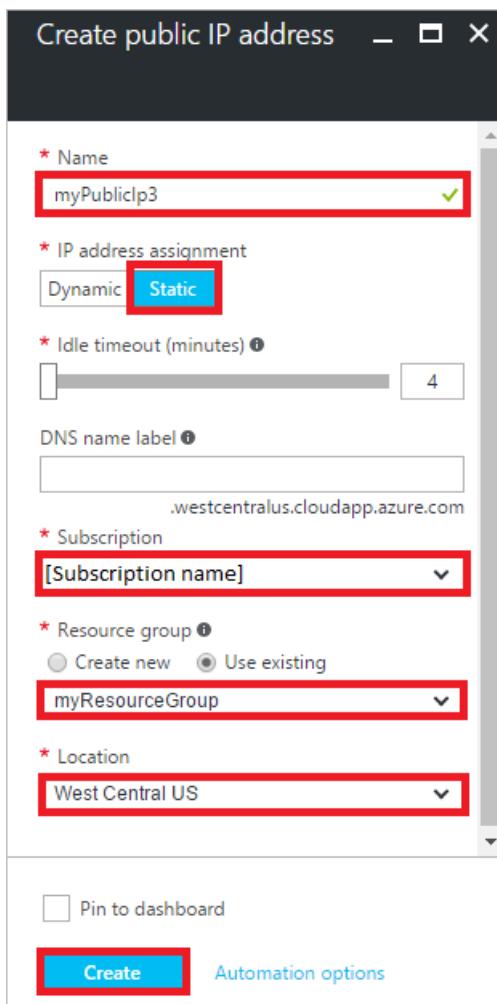
#### NOTE

Public IP addresses have a nominal fee. To learn more about IP address pricing, read the [IP address pricing](#) page. There is a limit to the number of public IP addresses that can be used in a subscription. To learn more about the limits, read the [Azure limits](#) article.

### Create a public IP address resource

A public IP address is one setting for a public IP address resource. If you have a public IP address resource that is not currently associated to an IP configuration that you want to associate to an IP configuration, skip the following steps and complete the steps in one of the sections that follow, as you require. If you don't have an available public IP address resource, complete the following steps to create one:

1. Browse to the Azure portal at <https://portal.azure.com> and sign into it, if necessary.
2. In the portal, click **Create a resource > Networking > Public IP address**.
3. In the **Create public IP address** pane that appears, enter a **Name**, select an **IP address assignment** type, a **Subscription**, a **Resource group**, and a **Location**, then click **Create**, as shown in the following picture:



4. Complete the steps in one of the sections that follow to associate the public IP address resource to an IP configuration.

#### **Associate the public IP address resource to a new IP configuration**

1. Complete the steps in the [Core steps](#) section of this article.
2. Click **Add**. In the **Add IP configuration** pane that appears, create an IP configuration named *IPConfig-4*. Enable the **Public IP address** and select an existing, available public IP address resource from the **Choose public IP address** pane that appears.

Once you've selected the public IP address resource, click **OK** and the pane closes. If you don't have an existing public IP address, you can create one by completing the steps in the [Create a public IP address resource](#) section of this article.

3. Review the new IP configuration. Even though a private IP address wasn't explicitly assigned, one was automatically assigned to the IP configuration, because all IP configurations must have a private IP address.
4. You can click **Add** to add additional IP configurations, or close all open blades to finish adding IP addresses.
5. Add the private IP address to the VM operating system by completing the steps for your operating

system in the [Add IP addresses to a VM operating system](#) section of this article. Do not add the public IP address to the operating system.

#### **Associate the public IP address resource to an existing IP configuration**

1. Complete the steps in the [Core steps](#) section of this article.
2. Click the IP configuration you want to add the public IP address resource to.
3. In the IPConfig pane that appears, click **IP address**.
4. In the **Choose public IP address** pane that appears, select a public IP address.
5. Click **Save** and the panes close. If you don't have an existing public IP address, you can create one by completing the steps in the [Create a public IP address resource](#) section of this article.
6. Review the new IP configuration.
7. You can click **Add** to add additional IP configurations, or close all open blades to finish adding IP addresses.  
Do not add the public IP address to the operating system.

## **Add IP addresses to a VM operating system**

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses (including the primary) that you added to the VM. Complete the steps that following for your VM operating system.

### **Windows Server**

- ▶ Expand

### **Linux (Ubuntu 14/16)**

- ▶ Expand

### **Linux (Ubuntu 18.04+)**

- ▶ Expand

### **Linux (Red Hat, CentOS, and others)**

- ▶ Expand

# Assign multiple IP addresses to virtual machines using PowerShell

2/18/2021 • 19 minutes to read • [Edit Online](#)

## NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it. Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any of the private IP addresses for any of the NICs to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary NIC could be added to a back-end pool. To learn more about how to load balance multiple IP configurations, read the [Load balancing multiple IP configurations](#) article.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. A public IP address resource has either a dynamic or static public IP address assigned to it. To learn more about IP addresses in Azure, read the [IP addresses in Azure](#) article.

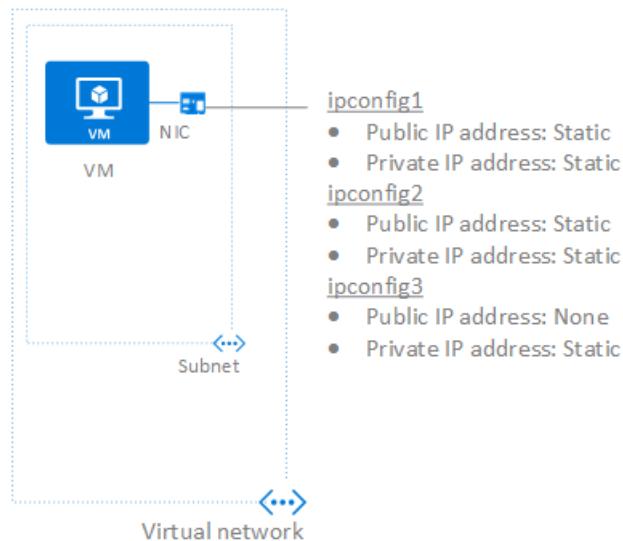
There is a limit to how many private IP addresses can be assigned to a NIC. There is also a limit to how many public IP addresses that can be used in an Azure subscription. See the [Azure limits](#) article for details.

This article explains how to create a virtual machine (VM) through the Azure Resource Manager deployment model using PowerShell. Multiple IP addresses cannot be assigned to resources created through the classic deployment model. To learn more about Azure deployment models, read the [Understand deployment models](#) article.

## Scenario

A VM with a single NIC is created and connected to a virtual network. The VM requires three different *private* IP addresses and two *public* IP addresses. The IP addresses are assigned to the following IP configurations:

- **IPConfig-1:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-2:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-3:** Assigns a *static* private IP address and no public IP address.



The IP configurations are associated to the NIC when the NIC is created and the NIC is attached to the VM when the VM is created. The types of IP addresses used for the scenario are for illustration. You can assign whatever IP address and assignment types you require.

#### NOTE

Though the steps in this article assigns all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, read the [Create a VM with multiple NICs](#) article.

## Create a VM with multiple IP addresses

The steps that follow explain how to create an example VM with multiple IP addresses, as described in the scenario. Change variable values as required for your implementation.

1. Open a PowerShell command prompt and complete the remaining steps in this section within a single PowerShell session. If you don't already have PowerShell installed and configured, complete the steps in the [How to install and configure Azure PowerShell](#) article.
2. Login to your account with the `Connect-AzAccount` command.
3. Replace *myResourceGroup* and *westus* with a name and location of your choosing. Create a resource group. A resource group is a logical container into which Azure resources are deployed and managed.

```
$RgName = "MyResourceGroup"
$Location = "westus"

New-AzResourceGroup `
-Name $RgName `
-Location $Location
```

4. Create a virtual network (VNet) and subnet in the same location as the resource group:

```

Create a subnet configuration
$SubnetConfig = New-AzVirtualNetworkSubnetConfig `
-Name MySubnet `
-AddressPrefix 10.0.0.0/24

Create a virtual network
$VNet = New-AzVirtualNetwork `
-ResourceGroupName $RgName `
-Location $Location `
-Name MyVNet `
-AddressPrefix 10.0.0.0/16 `
-Subnet $subnetConfig

Get the subnet object
$Subnet = Get-AzVirtualNetworkSubnetConfig -Name $SubnetConfig.Name -VirtualNetwork $VNet

```

5. Create a network security group (NSG) and a rule. The NSG secures the VM using inbound and outbound rules. In this case, an inbound rule is created for port 3389, which allows incoming remote desktop connections.

```

Create an inbound network security group rule for port 3389

$NSGRule = New-AzNetworkSecurityRuleConfig `
-Name MyNsgRuleRDP `
-Protocol Tcp `
-Direction Inbound `
-Priority 1000 `
-SourceAddressPrefix * `
-SourcePortRange * `
-DestinationAddressPrefix * `
-DestinationPortRange 3389 -Access Allow

Create a network security group
$NSG = New-AzNetworkSecurityGroup `
-ResourceGroupName $RgName `
-Location $Location `
-Name MyNetworkSecurityGroup `
-SecurityRules $NSGRule

```

6. Define the primary IP configuration for the NIC. Change 10.0.0.4 to a valid address in the subnet you created, if you didn't use the value defined previously. Before assigning a static IP address, it's recommended that you first confirm it's not already in use. Enter the command

`Test-AzPrivateIPAddressAvailability -IPAddress 10.0.0.4 -VirtualNetwork $VNet`. If the address is available, the output returns *True*. If it's not available, the output returns *False* and a list of addresses that are available.

In the following commands, **Replace <replace-with-your-unique-name> with the unique DNS name to use**. The name must be unique across all public IP addresses within an Azure region. This is an optional parameter. It can be removed if you only want to connect to the VM using the public IP address.

```

Create a public IP address
$PublicIP1 = New-AzPublicIpAddress `
-Name "MyPublicIP1" `
-ResourceGroupName $RgName `
-Location $Location `
-DomainNameLabel <replace-with-your-unique-name> `
-AllocationMethod Static

#Create an IP configuration with a static private IP address and assign the public IP address to it
$IpConfigName1 = "IPConfig-1"
$IpConfig1 = New-AzNetworkInterfaceIpConfig `
-Name $IpConfigName1 `
-Subnet $Subnet `
-PrivateIpAddress 10.0.0.4 `
-PublicIpAddress $PublicIP1 `
-Primary

```

When you assign multiple IP configurations to a NIC, one configuration must be assigned as the *-Primary*.

#### **NOTE**

Public IP addresses have a nominal fee. To learn more about IP address pricing, read the [IP address pricing](#) page. There is a limit to the number of public IP addresses that can be used in a subscription. To learn more about the limits, read the [Azure limits](#) article.

7. Define the secondary IP configurations for the NIC. You can add or remove configurations as necessary. Each IP configuration must have a private IP address assigned. Each configuration can optionally have one public IP address assigned.

```

Create a public IP address
$PublicIP2 = New-AzPublicIpAddress `
-Name "MyPublicIP2" `
-ResourceGroupName $RgName `
-Location $Location `
-AllocationMethod Static

#Create an IP configuration with a static private IP address and assign the public IP address to it
$IpConfigName2 = "IPConfig-2"
$IpConfig2 = New-AzNetworkInterfaceIpConfig `
-Name $IpConfigName2 `
-Subnet $Subnet `
-PrivateIpAddress 10.0.0.5 `
-PublicIpAddress $PublicIP2

$IpConfigName3 = "IpConfig-3"
$IpConfig3 = New-AzNetworkInterfaceIpConfig `
-Name $IPConfigName3 `
-Subnet $Subnet `
-PrivateIpAddress 10.0.0.6

```

8. Create the NIC and associate the three IP configurations to it:

```
$NIC = New-AzNetworkInterface `
-Name MyNIC `
-ResourceGroupName $RgName `
-Location $Location `
-NetworkSecurityGroupId $NSG.Id `
-IpConfiguration $IpConfig1,$IpConfig2,$IpConfig3
```

#### NOTE

Though all configurations are assigned to one NIC in this article, you can assign multiple IP configurations to every NIC attached to the VM. To learn how to create a VM with multiple NICs, read the [Create a VM with multiple NICs](#) article.

9. Create the VM by entering the following commands:

```
Define a credential object. When you run these commands, you're prompted to enter a username and password for the VM you're creating.
$cred = Get-Credential

Create a virtual machine configuration
$VmConfig = New-AzVMConfig `
-VMName MyVM `
-VMSize Standard_DS1_v2 | `
Set-AzVMOperatingSystem -Windows `
-ComputerName MyVM `
-Credential $cred | `
Set-AzVMSourceImage `
-PublisherName MicrosoftWindowsServer `
-Offer WindowsServer `
-Skus 2016-Datacenter `
-Version latest | `
Add-AzVMNetworkInterface `
-Id $NIC.Id

Create the VM
New-AzVM `
-ResourceGroupName $RgName `
-Location $Location `
-VM $VmConfig
```

10. Add the private IP addresses to the VM operating system by completing the steps for your operating system in the [Add IP addresses to a VM operating system](#) section of this article. Do not add the public IP addresses to the operating system.

## Add IP addresses to a VM

You can add private and public IP addresses to the Azure network interface by completing the steps that follow. The examples in the following sections assume that you already have a VM with the three IP configurations described in the [scenario](#) in this article, but it's not required that you do.

1. Open a PowerShell command prompt and complete the remaining steps in this section within a single PowerShell session. If you don't already have PowerShell installed and configured, complete the steps in the [How to install and configure Azure PowerShell](#) article.
2. Change the "values" of the following \$Variables to the name of the NIC you want to add IP address to and the resource group and location the NIC exists in:

```
$NicName = "MyNIC"
$RgName = "MyResourceGroup"
$Location = "westus"
```

If you don't know the name of the NIC you want to change, enter the following commands, then change the values of the previous variables:

```
Get-AzNetworkInterface | Format-Table Name, ResourceGroupName, Location
```

3. Create a variable and set it to the existing NIC by typing the following command:

```
$MyNIC = Get-AzNetworkInterface -Name $NicName -ResourceGroupName $RgName
```

4. In the following commands, change *MyVNet* and *MySubnet* to the names of the VNet and subnet the NIC is connected to. Enter the commands to retrieve the VNet and subnet objects the NIC is connected to:

```
$MyVNet = Get-AzVirtualnetwork -Name MyVNet -ResourceGroupName $RgName
$Subnet = $MyVnet.Subnets | Where-Object { $_.Name -eq "MySubnet" }
```

If you don't know the VNet or subnet name the NIC is connected to, enter the following command:

```
$MyNIC.IpConfigurations
```

In the output, look for text similar to the following example output:

```
"Id":
"/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/virtualNetworks/MyVNet/subnets/MySubnet"
```

In this output, *MyVnet* is the VNet and *MySubnet* is the subnet the NIC is connected to.

5. Complete the steps in one of the following sections, based on your requirements:

#### Add a private IP address

To add a private IP address to a NIC, you must create an IP configuration. The following command creates a configuration with a static IP address of 10.0.0.7. When specifying a static IP address, it must be an unused address for the subnet. It's recommended that you first test the address to ensure it's available by entering the `Test-AzPrivateIPAddressAvailability -IPAddress 10.0.0.7 -VirtualNetwork $myVNet` command. If the IP address is available, the output returns *True*. If it's not available, the output returns *False*, and a list of addresses that are available.

```
Add-AzNetworkInterfaceIpConfig -Name IPConfig-4 -NetworkInterface
$MyNIC -Subnet $Subnet -PrivateIpAddress 10.0.0.7
```

Create as many configurations as you require, using unique configuration names and private IP addresses (for configurations with static IP addresses).

Add the private IP address to the VM operating system by completing the steps for your operating system in the [Add IP addresses to a VM operating system](#) section of this article.

#### Add a public IP address

A public IP address is added by associating a public IP address resource to either a new IP configuration or an existing IP configuration. Complete the steps in one of the sections that follow, as you require.

#### NOTE

Public IP addresses have a nominal fee. To learn more about IP address pricing, read the [IP address pricing](#) page. There is a limit to the number of public IP addresses that can be used in a subscription. To learn more about the limits, read the [Azure limits](#) article.

### Associate the public IP address resource to a new IP configuration

Whenever you add a public IP address in a new IP configuration, you must also add a private IP address, because all IP configurations must have a private IP address. You can either add an existing public IP address resource, or create a new one. To create a new one, enter the following command:

```
$myPublicIp3 = New-AzPublicIpAddress
-Name "myPublicIp3"
-ResourceGroupName $RgName
-Location $Location
-AllocationMethod Static
```

To create a new IP configuration with a static private IP address and the associated *myPublicIp3* public IP address resource, enter the following command:

```
Add-AzNetworkInterfaceIpConfig
-Name IPConfig-4
-NetworkInterface $myNIC
-Subnet $Subnet
-PrivateIpAddress 10.0.0.7
-PublicIpAddress $myPublicIp3
```

### Associate the public IP address resource to an existing IP configuration

A public IP address resource can only be associated to an IP configuration that doesn't already have one associated. You can determine whether an IP configuration has an associated public IP address by entering the following command:

```
$MyNIC.IpConfigurations | Format-Table Name, PrivateIPAddress, PublicIPAddress, Primary
```

You see output similar to the following:

Name	PrivateIpAddress	PublicIpAddress	Primary
IPConfig-1	10.0.0.4	Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress	True
IPConfig-2	10.0.0.5	Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress	False
IPConfig-3	10.0.0.6		False

Since the **PublicIpAddress** column for *IpConfig-3* is blank, no public IP address resource is currently associated to it. You can add an existing public IP address resource to *IpConfig-3*, or enter the following command to create one:

```
$MyPublicIp3 = New-AzPublicIpAddress
-Name "MyPublicIp3"
-ResourceGroupName $RgName
-Location $Location -AllocationMethod Static
```

Enter the following command to associate the public IP address resource to the existing IP configuration named *IpConfig-3*:

```
Set-AzNetworkInterfaceIpConfig
-Name IpConfig-3
-NetworkInterface $mynic
-Subnet $Subnet
-PublicIpAddress $myPublicIp3
```

6. Set the NIC with the new IP configuration by entering the following command:

```
Set-AzNetworkInterface -NetworkInterface $MyNIC
```

7. View the private IP addresses and the public IP address resources assigned to the NIC by entering the following command:

```
$MyNIC.IpConfigurations | Format-Table Name, PrivateIPAddress, PublicIPAddress, Primary
```

8. Add the private IP address to the VM operating system by completing the steps for your operating system in the [Add IP addresses to a VM operating system](#) section of this article. Do not add the public IP address to the operating system.

## Add IP addresses to a VM operating system

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses (including the primary) that you added to the VM. Complete the steps that follow for your VM operating system.

### **Windows Server**

► Expand

### **Linux (Ubuntu 14/16)**

► Expand

### **Linux (Ubuntu 18.04+)**

► Expand

### **Linux (Red Hat, CentOS, and others)**

► Expand

# Assign multiple IP addresses to virtual machines using the Azure CLI

2/18/2021 • 18 minutes to read • [Edit Online](#)

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it. Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any of the private IP addresses for any of the NICs to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary NIC could be added to a back-end pool. To learn more about how to load balance multiple IP configurations, read the [Load balancing multiple IP configurations](#) article.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. A public IP address resource has either a dynamic or static public IP address assigned to it. To learn more about IP addresses in Azure, read the [IP addresses in Azure](#) article.

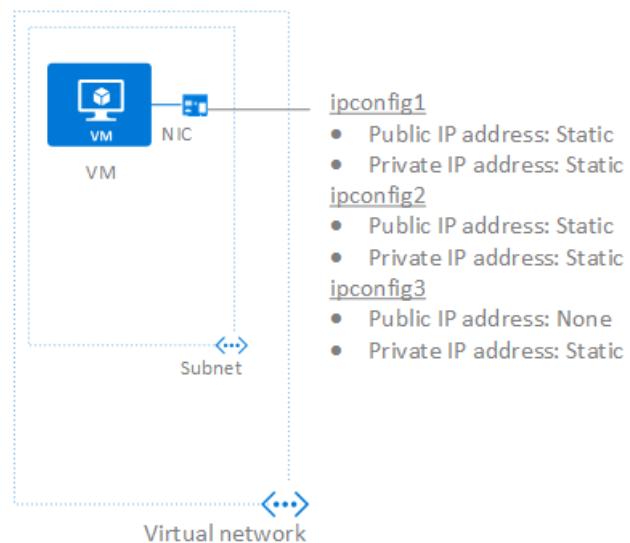
There is a limit to how many private IP addresses can be assigned to a NIC. There is also a limit to how many public IP addresses that can be used in an Azure subscription. See the [Azure limits](#) article for details.

This article explains how to create a virtual machine (VM) through the Azure Resource Manager deployment model using the Azure CLI. Multiple IP addresses cannot be assigned to resources created through the classic deployment model. To learn more about Azure deployment models, read the [Understand deployment models](#) article.

## Scenario

A VM with a single NIC is created and connected to a virtual network. The VM requires three different *private* IP addresses and two *public* IP addresses. The IP addresses are assigned to the following IP configurations:

- **IPConfig-1:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-2:** Assigns a *static* private IP address and a *static* public IP address.
- **IPConfig-3:** Assigns a *static* private IP address and no public IP address.



The IP configurations are associated to the NIC when the NIC is created and the NIC is attached to the VM when the VM is created. The types of IP addresses used for the scenario are for illustration. You can assign whatever IP address and assignment types you require.

#### NOTE

Though the steps in this article assigns all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, read the [Create a VM with multiple NICs](#) article.

## Create a VM with multiple IP addresses

The steps that follow explain how to create an example virtual machine with multiple IP addresses, as described in the scenario. Change variable values in "" and IP address types, as required, for your implementation.

1. Install the [Azure CLI](#) if you don't already have it installed.
2. Create an SSH public and private key pair for Linux VMs by completing the steps in the [Create an SSH public and private key pair for Linux VMs](#).
3. From a command shell, login with the command `az login` and select the subscription you're using.
4. Create the VM by executing the script that follows on a Linux or Mac computer. The script creates a resource group, one virtual network (VNet), one NIC with three IP configurations, and a VM with the two NICs attached to it. The NIC, public IP address, virtual network, and VM resources must all exist in the same location and subscription. Though the resources don't all have to exist in the same resource group, in the following script they do.

```
#!/bin/sh

RgName="myResourceGroup"
Location="westcentralus"
az group create --name $RgName --location $Location

Create a public IP address resource with a static IP address using the `--allocation-method Static` option. If you
do not specify this option, the address is allocated dynamically. The address is assigned to the resource
from a pool
of IP addresses unique to each Azure region. Download and view the file from
https://www.microsoft.com/en-us/download/details.aspx?id=41653 that lists the ranges for each region.

PipName="myPublicIP"
```

```

This name must be unique within an Azure location.
DnsName="myDNSName"

az network public-ip create \
--name $PipName \
--resource-group $RgName \
--location $Location \
--dns-name $DnsName\
--allocation-method Static

Create a virtual network with one subnet

VnetName="myVnet"
VnetPrefix="10.0.0.0/16"
VnetSubnetName="mySubnet"
VnetSubnetPrefix="10.0.0.0/24"

az network vnet create \
--name $VnetName \
--resource-group $RgName \
--location $Location \
--address-prefix $VnetPrefix \
--subnet-name $VnetSubnetName \
--subnet-prefix $VnetSubnetPrefix

Create a network interface connected to the subnet and associate the public IP address to it. Azure will
create the
first IP configuration with a static private IP address and will associate the public IP address resource
to it.

NicName="MyNic1"
az network nic create \
--name $NicName \
--resource-group $RgName \
--location $Location \
--subnet $VnetSubnet1Name \
--private-ip-address 10.0.0.4
--vnet-name $VnetName \
--public-ip-address $PipName

Create a second public IP address, a second IP configuration, and associate it to the NIC. This
configuration has a
static public IP address and a static private IP address.

az network public-ip create \
--resource-group $RgName \
--location $Location \
--name myPublicIP2 \
--dns-name mypublicdns2 \
--allocation-method Static

az network nic ip-config create \
--resource-group $RgName \
--nic-name $NicName \
--name IPConfig-2 \
--private-ip-address 10.0.0.5 \
--public-ip-name myPublicIP2

Create a third IP configuration, and associate it to the NIC. This configuration has static private IP
address and # no public IP address.

az network nic ip-config create \
--resource-group $RgName \
--nic-name $NicName \
--private-ip-address 10.0.0.6 \
--name IPConfig-3

Note: Though this article assigns all IP configurations to a single NIC, you can also assign multiple IP
configurations

```

```

to any NIC in a VM. To learn how to create a VM with multiple NICs
article: https://docs.microsoft.com/azure/virtual-network/virtual-network-deploy-multinic-arm-cli.

Create a VM and attach the NIC.

VmName="myVm"

Replace the value for the following **VmSize** variable with a value from the
https://docs.microsoft.com/azure/virtual-machines/sizes article. The script fails if the VM size
is not supported in the location you select. Run the `az vm sizes --location eastcentralus` command to
get a full list
of VMs in US West Central, for example.

VmSize="Standard_DS1"

Replace the value for the OsImage variable value with a value for *urn* from the output returned by
entering the
`az vm image list` command.

OsImage="credativ:Debian:8:latest"

Username="adminuser"

Replace the following value with the path to your public key file. If you're creating a Windows VM, remove
the following
line and you'll be prompted for the password you want to configure for the VM.

SshKeyValue="~/.ssh/id_rsa.pub"

az vm create \
--name $VmName \
--resource-group $RgName \
--image $OsImage \
--location $Location \
--size $VmSize \
--nics $NicName \
--admin-username $Username \
--ssh-key-value $SshKeyValue

```

In addition to creating a VM with a NIC with 3 IP configurations, the script creates:

- A single premium managed disk by default, but you have other options for the disk type you can create. Read the [Create a Linux VM using the Azure CLI](#) article for details.
- A virtual network with one subnet and two public IP addresses. Alternatively, you can use *existing* virtual network, subnet, NIC, or public IP address resources. To learn how to use existing network resources rather than creating additional resources, enter `az vm create -h`.

Public IP addresses have a nominal fee. To learn more about IP address pricing, read the [IP address pricing](#) page. There is a limit to the number of public IP addresses that can be used in a subscription. To learn more about the limits, read the [Azure limits](#) article.

After the VM is created, enter the `az network nic show --name MyNic1 --resource-group myResourceGroup` command to view the NIC configuration. Enter the

`az network nic ip-config list --nic-name MyNic1 --resource-group myResourceGroup --output table` to view a list of the IP configurations associated to the NIC.

Add the private IP addresses to the VM operating system by completing the steps for your operating system in the [Add IP addresses to a VM operating system](#) section of this article.

## Add IP addresses to a VM

You can add additional private and public IP addresses to an existing Azure network interface by completing the steps that follow. The examples build upon the [scenario](#) described in this article.

1. Open a command shell and complete the remaining steps in this section within a single session. If you don't already have Azure CLI installed and configured, complete the steps in the [Azure CLI installation](#) article and login to your Azure account with the `az-login` command.
2. Complete the steps in one of the following sections, based on your requirements:

### Add a private IP address

To add a private IP address to a NIC, you must create an IP configuration using the command that follows. The static IP address must be an unused address for the subnet.

```
az network nic ip-config create \
--resource-group myResourceGroup \
--nic-name myNic1 \
--private-ip-address 10.0.0.7 \
--name IPConfig-4
```

Create as many configurations as you require, using unique configuration names and private IP addresses (for configurations with static IP addresses).

### Add a public IP address

A public IP address is added by associating it to either a new IP configuration or an existing IP configuration. Complete the steps in one of the sections that follow, as you require.

Public IP addresses have a nominal fee. To learn more about IP address pricing, read the [IP address pricing](#) page. There is a limit to the number of public IP addresses that can be used in a subscription. To learn more about the limits, read the [Azure limits](#) article.

- **Associate the resource to a new IP configuration**

Whenever you add a public IP address in a new IP configuration, you must also add a private IP address, because all IP configurations must have a private IP address. You can either add an existing public IP address resource, or create a new one. To create a new one, enter the following command:

```
az network public-ip create \
--resource-group myResourceGroup \
--location westcentralus \
--name myPublicIP3 \
--dns-name mypublicdns
```

To create a new IP configuration with a static private IP address and the associated *myPublicIP3* public IP address resource, enter the following command:

```
az network nic ip-config create \
--resource-group myResourceGroup \
--nic-name myNic1 \
--name IPConfig-5 \
--private-ip-address 10.0.0.8 \
--public-ip-address myPublicIP3
```

- **Associate the resource to an existing IP configuration** A public IP address resource can only be associated to an IP configuration that doesn't already have one associated. You can determine whether an IP configuration has an associated public IP address by entering the following

command:

```
az network nic ip-config list \
--resource-group myResourceGroup \
--nic-name myNic1 \
--query "[?provisioningState=='Succeeded'].{ Name: name, PublicIpAddressId: publicIpAddress.id }" --output table
```

Returned output:

Name	PublicIpAddressId
ipconfig1	/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP1
IPConfig-2	/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP2
IPConfig-3	

Since the **PublicIpAddressId** column for *IpConfig-3* is blank in the output, no public IP address resource is currently associated to it. You can add an existing public IP address resource to *IPConfig-3*, or enter the following command to create one:

```
az network public-ip create \
--resource-group myResourceGroup
--location westcentralus \
--name myPublicIP3 \
--dns-name mypublicdns3 \
--allocation-method Static
```

Enter the following command to associate the public IP address resource to the existing IP configuration named *IPConfig-3*:

```
az network nic ip-config update \
--resource-group myResourceGroup \
--nic-name myNic1 \
--name IPConfig-3 \
--public-ip myPublicIP3
```

3. View the private IP addresses and the public IP address resource IDs assigned to the NIC by entering the following command:

```
az network nic ip-config list \
--resource-group myResourceGroup \
--nic-name myNic1 \
--query "[?provisioningState=='Succeeded'].{ Name: name, PrivateIpAddress: privateIpAddress, PrivateIpAllocationMethod: privateIpAllocationMethod, PublicIpAddressId: publicIpAddress.id }" --output table
```

Returned output:

Name	PrivateIpAddress	PrivateIpAllocationMethod	PublicIpAddressId
ipconfig1	10.0.0.4	Static	/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP1
IPConfig-2	10.0.0.5	Static	/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP2
IPConfig-3	10.0.0.6	Static	/subscriptions/[Id]/resourceGroups/myResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP3

- Add the private IP addresses you added to the NIC to the VM operating system by following the instructions in the [Add IP addresses to a VM operating system](#) section of this article. Do not add the public IP addresses to the operating system.

## Add IP addresses to a VM operating system

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses (including the primary) that you added to the VM. Complete the steps that following for your VM operating system.

### Windows Server

▶ Expand

### Linux (Ubuntu 14/16)

▶ Expand

### Linux (Ubuntu 18.04+)

▶ Expand

### Linux (Red Hat, CentOS, and others)

▶ Expand

# Add network interfaces to or remove network interfaces from virtual machines

2/18/2021 • 7 minutes to read • [Edit Online](#)

Learn how to add an existing network interface when you create an Azure virtual machine (VM). Also learn to add or remove network interfaces from an existing VM in the stopped (deallocated) state. A network interface enables an Azure VM to communicate with internet, Azure, and on-premises resources. A VM has one or more network interfaces.

If you need to add, change, or remove IP addresses for a network interface, see [Manage network interface IP addresses](#). To create, change, or delete network interfaces, see [Manage network interfaces](#).

## Before you begin

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

If you don't have one, set up an Azure account with an active subscription. [Create an account for free](#). Complete one of these tasks before starting the remainder of this article:

- **Portal users:** Sign in to the [Azure portal](#) with your Azure account.
- **PowerShell users:** Either run the commands in the [Azure Cloud Shell](#), or run PowerShell from your computer. 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. In the Azure Cloud Shell browser tab, find the **Select environment** dropdown list, then pick **PowerShell** if it isn't already selected.

If you're running PowerShell locally, use Azure PowerShell module version 1.0.0 or later. Run `Get-Module -ListAvailable Az.Network` to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). Run `Connect-AzAccount` to create a connection with Azure.

- **Azure Command-line interface (CLI) users:** Either run the commands in the [Azure Cloud Shell](#), or run the CLI from your computer. Use Azure CLI version 2.0.26 or later if you're running the Azure CLI locally. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). Run `az login` to create a connection with Azure.

## Add existing network interfaces to a new VM

When you create a virtual machine through the portal, the portal creates a network interface with default settings and attaches the network interface to the VM for you. You can't use the portal to add existing network interfaces to a new VM, or to create a VM with multiple network interfaces. You can do both by using the CLI or PowerShell. Be sure to familiarize yourself with the [constraints](#). If you create a VM with multiple network interfaces, you must also configure the operating system to use them properly after you create the VM. Learn how to configure [Linux](#) or [Windows](#) for multiple network interfaces.

### Commands

Before you create the VM, [Create a network interface](#).

TOOL	COMMAND
CLI	<code>az network nic create</code>
PowerShell	<code>New-AzNetworkInterface</code>

## Add a network interface to an existing VM

To add a network interface to your virtual machine:

1. Go to the [Azure portal](#) to find an existing virtual machine. Search for and select **Virtual machines**.
2. Select the name of your VM. The VM must support the number of network interfaces you want to add. To find out how many network interfaces each VM size supports, see the sizes in Azure for [Linux VMs](#) or [Windows VMs](#).
3. In the VM command bar, select **Stop**, and then **OK** in the confirmation dialog box. Then wait until the **Status** of the VM changes to **Stopped (deallocated)**.
4. From the VM menu bar, choose **Networking > Attach network interface**. Then in **Attach existing network interface**, choose the network interface you'd like to attach, and select **OK**.

### NOTE

The network interface you select can't have accelerated networking enabled, can't have an IPv6 address assigned to it, and must exist in the same virtual network with the network interface currently attached to the VM.

If you don't have an existing network interface, you must first create one. To do so, select **Create network interface**. To learn more about how to create a network interface, see [Create a network interface](#). To learn more about additional constraints when adding network interfaces to virtual machines, see [Constraints](#).

5. From the VM menu bar, choose **Overview > Start** to restart the virtual machine.

Now you can configure the VM operating system to use multiple network interfaces properly. Learn how to configure [Linux](#) or [Windows](#) for multiple network interfaces.

### Commands

TOOL	COMMAND
CLI	<code>az vm nic add</code> (reference); <a href="#">detailed steps</a>
PowerShell	<code>Add-AzVMNetworkInterface</code> (reference); <a href="#">detailed steps</a>

## View network interfaces for a VM

You can view the network interfaces currently attached to a VM to learn about each network interface's configuration, and the IP addresses assigned to each network interface.

1. Go to the [Azure portal](#) to find an existing virtual machine. Search for and select **Virtual machines**.

**NOTE**

Sign in using an account that is assigned the Owner, Contributor, or Network Contributor role for your subscription. To learn more about how to assign roles to accounts, see [Built-in roles for Azure role-based access control](#).

2. Select the name of the VM for which you want to view attached network interfaces.

3. In the VM menu bar, select **Networking**.

To learn about network interface settings and how to change them, see [Manage network interfaces](#). To learn about how to add, change, or remove IP addresses assigned to a network interface, see [Manage network interface IP addresses](#).

**Commands**

TOOL	COMMAND
CLI	<code>az vm nic list</code>
PowerShell	<code>Get-AzVM</code>

## Remove a network interface from a VM

1. Go to the [Azure portal](#) to find an existing virtual machine. Search for and select **Virtual machines**.
2. Select the name of the VM for which you want to view attached network interfaces.
3. In the VM toolbar, pick **Stop**.
4. Wait until the **Status** of the VM changes to **Stopped (deallocated)**.
5. From the VM menu bar, choose **Networking > Detach network interface**.
6. In the **Detach network interface** dialog box, select the network interface you'd like to detach. Then select **OK**.

**NOTE**

If only one network interface is listed, you can't detach it, because a virtual machine must always have at least one network interface attached to it.

**Commands**

TOOL	COMMAND
CLI	<code>az vm nic remove (reference); detailed steps</code>
PowerShell	<code>Remove-AzVMNetworkInterface (reference); detailed steps</code>

## Constraints

- A VM must have at least one network interface attached to it.
- A VM can only have as many network interfaces attached to it as the VM size supports. To learn more about how many network interfaces each VM size supports, see the sizes in Azure for [Linux VMs](#) or

[Windows VMs](#). All sizes support at least two network interfaces.

- The network interfaces you add to a VM can't currently be attached to another VM. To learn more about how to create network interfaces, see [Create a network interface](#).
- In the past, you could add network interfaces only to VMs that supported multiple network interfaces and were created with at least two network interfaces. You couldn't add a network interface to a VM that was created with one network interface, even if the VM size supported more than one network interface. Conversely, you could only remove network interfaces from a VM with at least three network interfaces, because VMs created with at least two network interfaces always had to have at least two network interfaces. These constraints no longer apply. You can now create a VM with any number of network interfaces (up to the number supported by the VM size).
- By default, the first network interface attached to a VM is the *primary* network interface. All other network interfaces in the VM are *secondary* network interfaces.
- You can control which network interface you send outbound traffic to. However, a VM by default sends all outbound traffic to the IP address that's assigned to the primary IP configuration of the primary network interface.
- In the past, all VMs within the same availability set were required to have a single, or multiple, network interfaces. VMs with any number of network interfaces can now exist in the same availability set, up to the number supported by the VM size. You can only add a VM to an availability set when it's created. To learn more about availability sets, see [Manage the availability of VMs in Azure](#).
- You can connect network interfaces in the same VM to different subnets within a virtual network. However, the network interfaces must all be connected to the same virtual network.
- You can add any IP address for any IP configuration of any primary or secondary network interface to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary network interface could be added to a back-end pool. To learn more about IP addresses and configurations, see [Add, change, or remove IP addresses](#).
- Deleting a VM doesn't delete the network interfaces that are attached to it. When you delete a VM, the network interfaces are detached from the VM. You can add those network interfaces to different VMs or delete them.
- Achieving the optimal performance documented requires Accelerated Networking. In some cases, you must explicitly enable Accelerated Networking for [Windows](#) or [Linux](#) virtual machines.

## Next steps

To create a VM with multiple network interfaces or IP addresses, see:

TASK	TOOL
Create a VM with multiple NICs	<a href="#">CLI</a> , <a href="#">PowerShell</a>
Create a single NIC VM with multiple IPv4 addresses	<a href="#">CLI</a> , <a href="#">PowerShell</a>
Create a single NIC VM with a private IPv6 address (behind an Azure Load Balancer)	<a href="#">CLI</a> , <a href="#">PowerShell</a> , <a href="#">Azure Resource Manager template</a>

# Create and manage a Windows virtual machine that has multiple NICs

11/2/2020 • 8 minutes to read • [Edit Online](#)

Virtual machines (VMs) in Azure can have multiple virtual network interface cards (NICs) attached to them. A common scenario is to have different subnets for front-end and back-end connectivity. You can associate multiple NICs on a VM to multiple subnets, but those subnets must all reside in the same virtual network (vNet). This article details how to create a VM that has multiple NICs attached to it. You also learn how to add or remove NICs from an existing VM. Different [VM sizes](#) support a varying number of NICs, so size your VM accordingly.

## Prerequisites

In the following examples, replace example parameter names with your own values. Example parameter names include *myResourceGroup*, *myVnet*, and *myVM*.

## Create a VM with multiple NICs

First, create a resource group. The following example creates a resource group named *myResourceGroup* in the *EastUS* location:

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

### Create virtual network and subnets

A common scenario is for a virtual network to have two or more subnets. One subnet may be for front-end traffic, the other for back-end traffic. To connect to both subnets, you then use multiple NICs on your VM.

1. Define two virtual network subnets with [New-AzVirtualNetworkSubnetConfig](#). The following example defines the subnets for *mySubnetFrontEnd* and *mySubnetBackEnd*.

```
$mySubnetFrontEnd = New-AzVirtualNetworkSubnetConfig -Name "mySubnetFrontEnd" `
 -AddressPrefix "192.168.1.0/24"
$mySubnetBackEnd = New-AzVirtualNetworkSubnetConfig -Name "mySubnetBackEnd" `
 -AddressPrefix "192.168.2.0/24"
```

2. Create your virtual network and subnets with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVnet*.

```
$myVnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
 -Location "EastUs" `
 -Name "myVnet" `
 -AddressPrefix "192.168.0.0/16" `
 -Subnet $mySubnetFrontEnd,$mySubnetBackEnd
```

### Create multiple NICs

Create two NICs with [New-AzNetworkInterface](#). Attach one NIC to the front-end subnet and one NIC to the back-end subnet. The following example creates NICs named *myNic1* and *myNic2*.

```

$frontEnd = $myVnet.Subnets | ?{$_ .Name -eq 'mySubnetFrontEnd'}
$myNic1 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic1" `
 -Location "EastUs" `
 -SubnetId $frontEnd.Id

$backEnd = $myVnet.Subnets | ?{$_ .Name -eq 'mySubnetBackEnd'}
$myNic2 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic2" `
 -Location "EastUs" `
 -SubnetId $backEnd.Id

```

Typically you also create a [network security group](#) to filter network traffic to the VM and a [load balancer](#) to distribute traffic across multiple VMs.

### Create the virtual machine

Now start to build your VM configuration. Each VM size has a limit for the total number of NICs that you can add to a VM. For more information, see [Windows VM sizes](#).

1. Set your VM credentials to the `$cred` variable as follows:

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

2. Define your VM with [New-AzVMConfig](#). The following example defines a VM named *myVM* and uses a VM size that supports more than two NICs (*Standard\_DS3\_v2*):

```
$vmConfig = New-AzVMConfig -VMName "myVM" -VMSize "Standard_DS3_v2"
```

3. Create the rest of your VM configuration with [Set-AzVMOperatingSystem](#) and [Set-AzVMSourceImage](#). The following example creates a Windows Server 2016 VM:

```

$vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
 -Windows `
 -ComputerName "myVM" `
 -Credential $cred `
 -ProvisionVMAgent `
 -EnableAutoUpdate
$vmConfig = Set-AzVMSourceImage -VM $vmConfig `
 -PublisherName "MicrosoftWindowsServer" `
 -Offer "WindowsServer" `
 -Skus "2016-Datacenter" `
 -Version "latest"

```

4. Attach the two NICs that you previously created with [Add-AzVMNetworkInterface](#):

```

$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic1.Id -Primary
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic2.Id

```

5. Create your VM with [New-AzVM](#):

```
New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location "EastUs"
```

6. Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

# Add a NIC to an existing VM

To add a virtual NIC to an existing VM, you deallocate the VM, add the virtual NIC, then start the VM. Different [VM sizes](#) support a varying number of NICs, so size your VM accordingly. If needed, you can [resize a VM](#).

1. Deallocation the VM with [Stop-AzVM](#). The following example deallocated the VM named *myVM* in *myResourceGroup*:

```
Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with [Get-AzVm](#). The following example gets information for the VM named *myVM* in *myResourceGroup*:

```
$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. The following example creates a virtual NIC with [New-AzNetworkInterface](#) named *myNic3* that is attached to *mySubnetBackend*. The virtual NIC is then attached to the VM named *myVM* in *myResourceGroup* with [Add-AzVMNetworkInterface](#):

```
Get info for the back end subnet
$myVnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName "myResourceGroup"
$backEnd = $myVnet.Subnets | ?{$_._Name -eq 'mySubnetBackend'}`

Create a virtual NIC
$myNic3 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic3" `
 -Location "EastUs" `
 -SubnetId $backEnd.Id

Get the ID of the new virtual NIC and add to VM
$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -Name "MyNic3").Id
Add-AzVMNetworkInterface -VM $vm -Id $nicId | Update-AzVm -ResourceGroupName "myResourceGroup"
```

## Primary virtual NICs

One of the NICs on a multi-NIC VM needs to be primary. If one of the existing virtual NICs on the VM is already set as primary, you can skip this step. The following example assumes that two virtual NICs are now present on a VM and you wish to add the first NIC ( [0] ) as the primary:

```
List existing NICs on the VM and find which one is primary
$vm.NetworkProfile.NetworkInterfaces

Set NIC 0 to be primary
$vm.NetworkProfile.NetworkInterfaces[0].Primary = $true
$vm.NetworkProfile.NetworkInterfaces[1].Primary = $false

Update the VM state in Azure
Update-AzVM -VM $vm -ResourceGroupName "myResourceGroup"
```

4. Start the VM with [Start-AzVm](#):

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

5. Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

## Remove a NIC from an existing VM

To remove a virtual NIC from an existing VM, you deallocate the VM, remove the virtual NIC, then start the VM.

1. Deallocate the VM with [Stop-AzVM](#). The following example deallocates the VM named *myVM* in *myResourceGroup*.

```
Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with [Get-AzVm](#). The following example gets information for the VM named *myVM* in *myResourceGroup*.

```
$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. Get information about the NIC remove with [Get-AzNetworkInterface](#). The following example gets information about *myNic3*:

```
List existing NICs on the VM if you need to determine NIC name
$vm.NetworkProfile.NetworkInterfaces

$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -Name "myNic3").Id
```

4. Remove the NIC with [Remove-AzVMNetworkInterface](#) and then update the VM with [Update-AzVm](#). The following example removes *myNic3* as obtained by `$nicId` in the preceding step:

```
Remove-AzVMNetworkInterface -VM $vm -NetworkInterfaceIDs $nicId | `
Update-AzVm -ResourceGroupName "myResourceGroup"
```

5. Start the VM with [Start-AzVm](#):

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

## Create multiple NICs with templates

Azure Resource Manager templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. Resource Manager templates use declarative JSON files to define your environment. For more information, see [overview of Azure Resource Manager](#). You can use `copy` to specify the number of instances to create:

```
"copy": {
 "name": "multiplenics",
 "count": "[parameters('count')]"
}
```

For more information, see [creating multiple instances by using copy](#).

You can also use `copyIndex()` to append a number to a resource name. You can then create *myNic1*, *MyNic2* and so on. The following code shows an example of appending the index value:

```
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of [creating multiple NICs by using Resource Manager templates](#).

Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

## Configure guest OS for multiple NICs

Azure assigns a default gateway to the first (primary) network interface attached to the virtual machine. Azure does not assign a default gateway to additional (secondary) network interfaces attached to a virtual machine. Therefore, you are unable to communicate with resources outside the subnet that a secondary network interface is in, by default. Secondary network interfaces can, however, communicate with resources outside their subnet, though the steps to enable communication are different for different operating systems.

1. From a Windows command prompt, run the `route print` command, which returns output similar to the following output for a virtual machine with two attached network interfaces:

```
=====
Interface List
3...00 0d 3a 10 92 ceMicrosoft Hyper-V Network Adapter #3
7...00 0d 3a 10 9b 2aMicrosoft Hyper-V Network Adapter #4
=====
```

In this example, **Microsoft Hyper-V Network Adapter #4** (interface 7) is the secondary network interface that doesn't have a default gateway assigned to it.

2. From a command prompt, run the `ipconfig` command to see which IP address is assigned to the secondary network interface. In this example, 192.168.2.4 is assigned to interface 7. No default gateway address is returned for the secondary network interface.
3. To route all traffic destined for addresses outside the subnet of the secondary network interface to the gateway for the subnet, run the following command:

```
route add -p 0.0.0.0 MASK 0.0.0.0 192.168.2.1 METRIC 5015 IF 7
```

The gateway address for the subnet is the first IP address (ending in .1) in the address range defined for the subnet. If you don't want to route all traffic outside the subnet, you could add individual routes to specific destinations, instead. For example, if you only wanted to route traffic from the secondary network interface to the 192.168.3.0 network, you enter the command:

```
route add -p 192.168.3.0 MASK 255.255.255.0 192.168.2.1 METRIC 5015 IF 7
```

4. To confirm successful communication with a resource on the 192.168.3.0 network, for example, enter the following command to ping 192.168.3.4 using interface 7 (192.168.2.4):

```
ping 192.168.3.4 -S 192.168.2.4
```

You may need to open ICMP through the Windows firewall of the device you're pinging with the following command:

```
netsh advfirewall firewall add rule name=Allow-ping protocol=icmpv4 dir=in action=allow
```

5. To confirm the added route is in the route table, enter the `route print` command, which returns output similar to the following text:

=====					
Active Routes:					
Network Destination	Netmask	Gateway	Interface	Metric	
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.4	15	
0.0.0.0	0.0.0.0	192.168.2.1	192.168.2.4	5015	

The route listed with **192.168.1.1** under **Gateway**, is the route that is there by default for the primary network interface. The route with **192.168.2.1** under **Gateway**, is the route you added.

## Next steps

Review [Windows VM sizes](#) when you're trying to create a VM that has multiple NICs. Pay attention to the maximum number of NICs that each VM size supports.

# How to create a Linux virtual machine in Azure with multiple network interface cards

11/2/2020 • 6 minutes to read • [Edit Online](#)

This article details how to create a VM with multiple NICs with the Azure CLI.

## Create supporting resources

Install the latest [Azure CLI](#) and log in to an Azure account using [az login](#).

In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *mystorageaccount*, and *myVM*.

First, create a resource group with [az group create](#). The following example creates a resource group named *myResourceGroup* in the *eastus* location:

```
az group create --name myResourceGroup --location eastus
```

Create the virtual network with [az network vnet create](#). The following example creates a virtual network named *myVnet* and subnet named *mySubnetFrontEnd*.

```
az network vnet create \
 --resource-group myResourceGroup \
 --name myVnet \
 --address-prefix 10.0.0.0/16 \
 --subnet-name mySubnetFrontEnd \
 --subnet-prefix 10.0.1.0/24
```

Create a subnet for the back-end traffic with [az network vnet subnet create](#). The following example creates a subnet named *mySubnetBackEnd*.

```
az network vnet subnet create \
 --resource-group myResourceGroup \
 --vnet-name myVnet \
 --name mySubnetBackEnd \
 --address-prefix 10.0.2.0/24
```

Create a network security group with [az network nsg create](#). The following example creates a network security group named *myNetworkSecurityGroup*.

```
az network nsg create \
 --resource-group myResourceGroup \
 --name myNetworkSecurityGroup
```

## Create and configure multiple NICs

Create two NICs with [az network nic create](#). The following example creates two NICs, named *myNic1* and *myNic2*, connected the network security group, with one NIC connecting to each subnet:

```
az network nic create \
--resource-group myResourceGroup \
--name myNic1 \
--vnet-name myVnet \
--subnet mySubnetFrontEnd \
--network-security-group myNetworkSecurityGroup
az network nic create \
--resource-group myResourceGroup \
--name myNic2 \
--vnet-name myVnet \
--subnet mySubnetBackEnd \
--network-security-group myNetworkSecurityGroup
```

## Create a VM and attach the NICs

When you create the VM, specify the NICs you created with `--nics`. You also need to take care when you select the VM size. There are limits for the total number of NICs that you can add to a VM. Read more about [Linux VM sizes](#).

Create a VM with [az vm create](#). The following example creates a VM named *myVM*.

```
az vm create \
--resource-group myResourceGroup \
--name myVM \
--image UbuntuLTS \
--size Standard_DS3_v2 \
--admin-username azureuser \
--generate-ssh-keys \
--nics myNic1 myNic2
```

Add routing tables to the guest OS by completing the steps in [Configure the guest OS for multiple NICs](#).

## Add a NIC to a VM

The previous steps created a VM with multiple NICs. You can also add NICs to an existing VM with the Azure CLI. Different [VM sizes](#) support a varying number of NICs, so size your VM accordingly. If needed, you can [resize a VM](#).

Create another NIC with [az network nic create](#). The following example creates a NIC named *myNic3* connected to the back-end subnet and network security group created in the previous steps:

```
az network nic create \
--resource-group myResourceGroup \
--name myNic3 \
--vnet-name myVnet \
--subnet mySubnetBackEnd \
--network-security-group myNetworkSecurityGroup
```

To add a NIC to an existing VM, first deallocate the VM with [az vm deallocate](#). The following example deallocates the VM named *myVM*.

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Add the NIC with [az vm nic add](#). The following example adds *myNic3* to *myVM*.

```
az vm nic add \
--resource-group myResourceGroup \
--vm-name myVM \
--nics myNic3
```

Start the VM with [az vm start](#):

```
az vm start --resource-group myResourceGroup --name myVM
```

Add routing tables to the guest OS by completing the steps in [Configure the guest OS for multiple NICs](#).

## Remove a NIC from a VM

To remove a NIC from an existing VM, first deallocate the VM with [az vm deallocate](#). The following example deallocates the VM named *myVM*.

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Remove the NIC with [az vm nic remove](#). The following example removes *myNic3* from *myVM*.

```
az vm nic remove \
--resource-group myResourceGroup \
--vm-name myVM \
--nics myNic3
```

Start the VM with [az vm start](#):

```
az vm start --resource-group myResourceGroup --name myVM
```

## Create multiple NICs using Resource Manager templates

Azure Resource Manager templates use declarative JSON files to define your environment. You can read an [overview of Azure Resource Manager](#). Resource Manager templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. You use *copy* to specify the number of instances to create:

```
"copy": {
 "name": "multiplenics"
 "count": "[parameters('count')]"
}
```

Read more about [creating multiple instances using copy](#).

You can also use a `copyIndex()` to then append a number to a resource name, which allows you to create `myNic1`, `myNic2`, etc. The following shows an example of appending the index value:

```
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of [creating multiple NICs using Resource Manager templates](#).

Add routing tables to the guest OS by completing the steps in [Configure the guest OS for multiple NICs](#).

## Configure guest OS for multiple NICs

The previous steps created a virtual network and subnet, attached NICs, then created a VM. A public IP address and network security group rules that allow SSH traffic were not created. To configure the guest OS for multiple NICs, you need to allow remote connections and run commands locally on the VM.

To allow SSH traffic, create a network security group rule with [az network nsg rule create](#) as follows:

```
az network nsg rule create \
--resource-group myResourceGroup \
--nsg-name myNetworkSecurityGroup \
--name allow_ssh \
--priority 101 \
--destination-port-ranges 22
```

Create a public IP address with [az network public-ip create](#) and assign it to the first NIC with [az network nic ip-config update](#):

```
az network public-ip create --resource-group myResourceGroup --name myPublicIP

az network nic ip-config update \
--resource-group myResourceGroup \
--nic-name myNic1 \
--name ipconfig1 \
--public-ip myPublicIP
```

To view the public IP address of the VM, use [az vm show](#) as follows::

```
az vm show --resource-group myResourceGroup --name myVM -d --query publicIps -o tsv
```

Now SSH to the public IP address of your VM. The default username provided in a previous step was *azureuser*. Provide your own username and public IP address:

```
ssh azureuser@137.117.58.232
```

To send to or from a secondary network interface, you have to manually add persistent routes to the operating system for each secondary network interface. In this article, *eth1* is the secondary interface. Instructions for adding persistent routes to the operating system vary by distro. See documentation for your distro for instructions.

When adding the route to the operating system, the gateway address is *.1* for whichever subnet the network interface is in. For example, if the network interface is assigned the address *10.0.2.4*, the gateway you specify for the route is *10.0.2.1*. You can define a specific network for the route's destination, or specify a destination of *0.0.0.0*, if you want all traffic for the interface to go through the specified gateway. The gateway for each subnet is managed by the virtual network.

Once you've added the route for a secondary interface, verify that the route is in your route table with [route -n](#). The following example output is for the route table that has the two network interfaces added to the VM in this article:

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.0.1.1	0.0.0.0	UG	0	0	0	eth0
0.0.0.0	10.0.2.1	0.0.0.0	UG	0	0	0	eth1
10.0.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
10.0.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
168.63.129.16	10.0.1.1	255.255.255.255	UGH	0	0	0	eth0
169.254.169.254	10.0.1.1	255.255.255.255	UGH	0	0	0	eth0

Confirm that the route you added persists across reboots by checking your route table again after a reboot. To test connectivity, you can enter the following command, for example, where *eth1* is the name of a secondary network interface:

```
ping bing.com -c 4 -I eth1
```

## Next steps

Review [Linux VM sizes](#) when trying to creating a VM with multiple NICs. Pay attention to the maximum number of NICs each VM size supports.

To further secure your VMs, use just in time VM access. This feature opens network security group rules for SSH traffic when needed, and for a defined period of time. For more information, see [Manage virtual machine access using just in time](#).

# Create a Windows VM with accelerated networking using Azure PowerShell

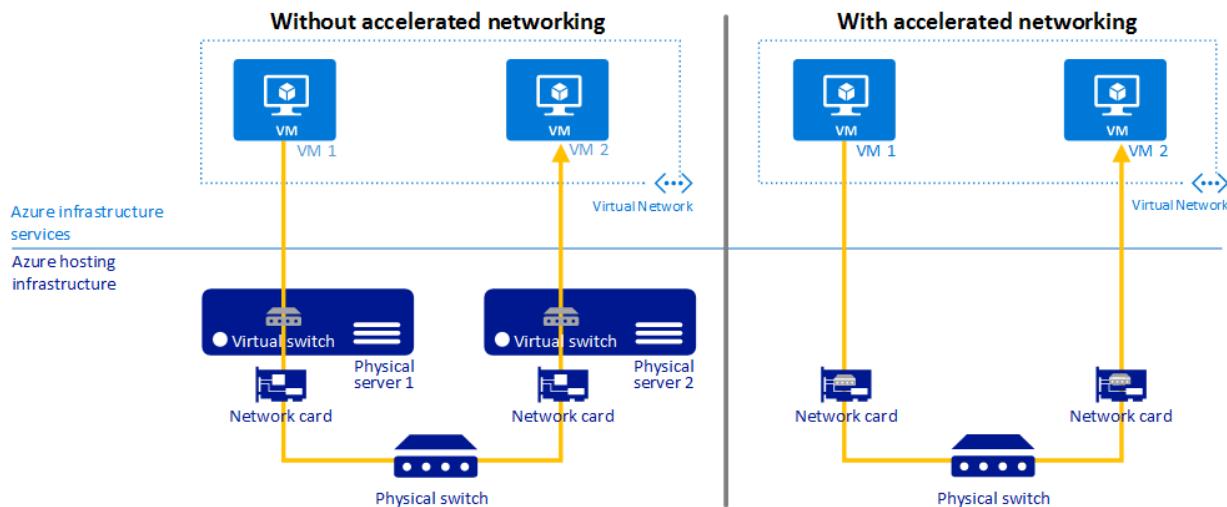
2/18/2021 • 10 minutes to read • [Edit Online](#)

In this tutorial, you learn how to create a Windows virtual machine (VM) with accelerated networking.

## NOTE

To use accelerated networking with a Linux virtual machine, see [Create a Linux VM with accelerated networking](#).

Accelerated networking enables single root I/O virtualization (SR-IOV) to a VM, greatly improving its networking performance. This high-performance path bypasses the host from the data path, which reduces latency, jitter, and CPU utilization for the most demanding network workloads on supported VM types. The following diagram illustrates how two VMs communicate with and without accelerated networking:



Without accelerated networking, all networking traffic in and out of the VM must traverse the host and the virtual switch. The virtual switch provides all policy enforcement, such as network security groups, access control lists, isolation, and other network virtualized services to network traffic.

## NOTE

To learn more about virtual switches, see [Hyper-V Virtual Switch](#).

With accelerated networking, network traffic arrives at the VM's network interface (NIC) and is then forwarded to the VM. All network policies that the virtual switch applies are now offloaded and applied in hardware. Because policy is applied in hardware, the NIC can forward network traffic directly to the VM. The NIC bypasses the host and the virtual switch, while it maintains all the policy it applied in the host.

The benefits of accelerated networking only apply to the VM that it's enabled on. For the best results, enable this feature on at least two VMs connected to the same Azure virtual network. When communicating across virtual networks or connecting on-premises, this feature has minimal impact to overall latency.

## Benefits

- Lower Latency / Higher packets per second (pps): Eliminating the virtual switch from the data path

removes the time packets spend in the host for policy processing. It also increases the number of packets that can be processed inside the VM.

- **Reduced jitter:** Virtual switch processing depends on the amount of policy that needs to be applied. It also depends on the workload of the CPU that's doing the processing. Offloading the policy enforcement to the hardware removes that variability by delivering packets directly to the VM. Offloading also removes the host-to-VM communication, all software interrupts, and all context switches.
- **Decreased CPU utilization:** Bypassing the virtual switch in the host leads to less CPU utilization for processing network traffic.

## Supported operating systems

The following distributions are supported directly from the Azure Gallery:

- Windows Server 2019 Datacenter
- Windows Server 2016 Datacenter
- Windows Server 2012 R2 Datacenter

## Limitations and constraints

### Supported VM instances

Accelerated Networking is supported on most general purpose and compute-optimized instance sizes with 2 or more vCPUs. On instances that support hyperthreading, Accelerated Networking is supported on VM instances with 4 or more vCPUs.

Support for Accelerated Networking can be found in the individual [virtual machine sizes](#) documentation.

### Custom images

If you're using a custom image and your image supports Accelerated Networking, be sure that you have the required drivers that work with Mellanox ConnectX-3 and ConnectX-4 Lx NICs on Azure.

### Regions

Accelerated networking is available in all global Azure regions and Azure Government Cloud.

### Enabling accelerated networking on a running VM

A supported VM size without accelerated networking enabled can only have the feature enabled when it's stopped and deallocated.

### Deployment through Azure Resource Manager

Virtual machines (classic) can't be deployed with accelerated networking.

## VM creation using the portal

Though this article provides steps to create a VM with accelerated networking using Azure PowerShell, you can also [use the Azure portal to create a virtual machine](#) that enables accelerated networking. When you create a VM in the portal, in the **Create a virtual machine** page, choose the **Networking** tab. This tab has an option for **Accelerated networking**. If you have chosen a [supported operating system](#) and [VM size](#), this option is automatically set to **On**. Otherwise, the option is set to **Off**, and Azure displays the reason why it can't be enabled.

#### NOTE

Only supported operating systems can be enabled through the portal. If you are using a custom image, and your image supports accelerated networking, please create your VM using CLI or PowerShell.

After you create the VM, you can confirm whether accelerated networking is enabled. Follow these instructions:

1. Go to the [Azure portal](#) to manage your VMs. Search for and select **Virtual machines**.
2. In the virtual machine list, choose your new VM.
3. In the VM menu bar, choose **Networking**.

In the network interface information, next to the **Accelerated networking** label, the portal displays either **Disabled** or **Enabled** for the accelerated networking status.

## VM creation using PowerShell

Before you proceed, install [Azure PowerShell](#) version 1.0.0 or later. To find your currently installed version, run `Get-Module -ListAvailable Az`. If you need to install or upgrade, install the latest version of the Az module from the [PowerShell Gallery](#). In a PowerShell session, sign in to an Azure account using [Connect-AzAccount](#).

In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *myNic*, and *myVM*.

### Create a virtual network

1. Create a resource group with [New-AzResourceGroup](#). The following command creates a resource group named *myResourceGroup* in the *centralus* location:

```
New-AzResourceGroup -Name "myResourceGroup" -Location "centralus"
```

2. Create a subnet configuration with [New-AzVirtualNetworkSubnetConfig](#). The following command creates a subnet named *mySubnet*:

```
$subnet = New-AzVirtualNetworkSubnetConfig `
-Name "mySubnet" `
-AddressPrefix "192.168.1.0/24"
```

3. Create a virtual network with [New-AzVirtualNetwork](#), with the *mySubnet* subnet.

```
$vnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
-Location "centralus" `
-Name "myVnet" `
-AddressPrefix "192.168.0.0/16" `
-Subnet $subnet
```

### Create a network security group

1. Create a network security group rule with [New-AzNetworkSecurityRuleConfig](#).

```
$rdp = New-AzNetworkSecurityRuleConfig `
 -Name 'Allow-RDP-All' `
 -Description 'Allow RDP' `
 -Access Allow `
 -Protocol Tcp `
 -Direction Inbound `
 -Priority 100 `
 -SourceAddressPrefix * `
 -SourcePortRange * `
 -DestinationAddressPrefix * `
 -DestinationPortRange 3389
```

2. Create a network security group with [New-AzNetworkSecurityGroup](#) and assign the *Allow-RDP-All* security rule to it. Aside from the *Allow-RDP-All* rule, the network security group contains several default rules. One default rule disables all inbound access from the internet. Once it's created, the *Allow-RDP-All* rule is assigned to the network security group so that you can remotely connect to the VM.

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName myResourceGroup `
 -Location centralus `
 -Name "myNsg" `
 -SecurityRules $rdp
```

3. Associate the network security group to the *mySubnet* subnet with [Set-AzVirtualNetworkSubnetConfig](#). The rule in the network security group is effective for all resources deployed in the subnet.

```
Set-AzVirtualNetworkSubnetConfig `
 -VirtualNetwork $vnet `
 -Name 'mySubnet' `
 -AddressPrefix "192.168.1.0/24" `
 -NetworkSecurityGroup $nsg
```

## Create a network interface with accelerated networking

1. Create a public IP address with [New-AzPublicIpAddress](#). A public IP address is unnecessary if you don't plan to access the VM from the internet. However, it's required to complete the steps in this article.

```
$publicIp = New-AzPublicIpAddress `
 -ResourceGroupName myResourceGroup `
 -Name 'myPublicIp' `
 -location centralus `
 -AllocationMethod Dynamic
```

2. Create a network interface with [New-AzNetworkInterface](#) with accelerated networking enabled, and assign the public IP address to the network interface. The following example creates a network interface named *myNic* in the *mySubnet* subnet of the *myVnet* virtual network, assigning the *myPublicIp* public IP address to it:

```
$nic = New-AzNetworkInterface `
 -ResourceGroupName "myResourceGroup" `
 -Name "myNic" `
 -Location "centralus" `
 -SubnetId $vnet.Subnets[0].Id `
 -PublicIpAddressId $publicIp.Id `
 -EnableAcceleratedNetworking
```

## Create a VM and attach the network interface

- Set your VM credentials to the `$cred` variable using [Get-Credential](#), which prompts you to sign in:

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

- Define your VM with [New-AzVMConfig](#). The following command defines a VM named *myVM* with a VM size that supports accelerated networking (*Standard\_DS4\_v2*):

```
$vmConfig = New-AzVMConfig -VMName "myVm" -VMSize "Standard_DS4_v2"
```

For a list of all VM sizes and characteristics, see [Windows VM sizes](#).

- Create the rest of your VM configuration with [Set-AzVMOperatingSystem](#) and [Set-AzVMSourceImage](#).

The following command creates a Windows Server 2016 VM:

```
$vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
 -Windows `
 -ComputerName "myVM" `
 -Credential $cred `
 -ProvisionVMAgent `
 -EnableAutoUpdate
$vmConfig = Set-AzVMSourceImage -VM $vmConfig `
 -PublisherName "MicrosoftWindowsServer" `
 -Offer "WindowsServer" `
 -Skus "2016-Datacenter" `
 -Version "latest"
```

- Attach the network interface that you previously created with [Add-AzVMNetworkInterface](#):

```
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id
```

- Create your VM with [New-AzVM](#).

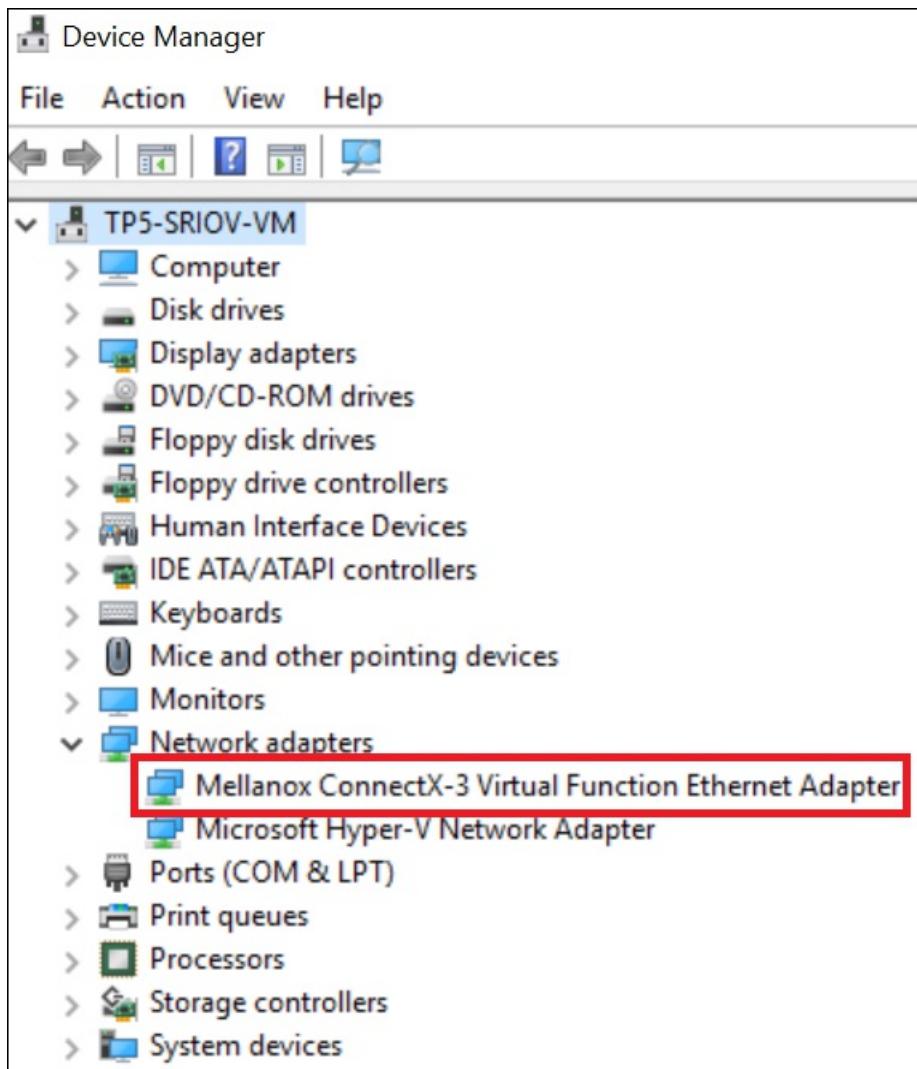
```
New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location "centralus"
```

## Confirm the Ethernet controller is installed in the Windows VM

Once you create the VM in Azure, connect to the VM and confirm that the Ethernet controller is installed in Windows.

- Go to the [Azure portal](#) to manage your VMs. Search for and select **Virtual machines**.
- In the virtual machine list, choose your new VM.
- In the VM overview page, if the **Status** of the VM is listed as **Creating**, wait until Azure finishes creating the VM. The **Status** will be changed to **Running** after VM creation is complete.
- From the VM overview toolbar, select **Connect > RDP > Download RDP File**.
- Open the .rdp file, and then sign in to the VM with the credentials you entered in the [Create a VM and attach the network interface](#) section. If you've never connected to a Windows VM in Azure, see [Connect to virtual machine](#).
- After the remote desktop session for your VM appears, right-click the Windows Start button and choose **Device Manager**.
- In the Device Manager window, expand the **Network adapters** node.

8. Confirm that the **Mellanox ConnectX-3 Virtual Function Ethernet Adapter** appears, as shown in the following image:



Accelerated networking is now enabled for your VM.

**NOTE**

If the Mellanox adapter fails to start, open an administrator prompt in the remote desktop session and enter the following command:

```
netsh int tcp set global rss = enabled
```

## Enable accelerated networking on existing VMs

If you've created a VM without accelerated networking, you may enable this feature on an existing VM. The VM must support accelerated networking by meeting the following prerequisites, which are also outlined above:

- The VM must be a supported size for accelerated networking.
- The VM must be a supported Azure Gallery image (and kernel version for Linux).
- All VMs in an availability set or a virtual machine scale set must be stopped or deallocated before you enable accelerated networking on any NIC.

### Individual VMs and VMs in an availability set

1. Stop or deallocate the VM or, if an availability set, all the VMs in the set:

```
Stop-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM"
```

#### NOTE

When you create a VM individually, without an availability set, you only need to stop or deallocate the individual VM to enable accelerated networking. If your VM was created with an availability set, you must stop or deallocate all VMs contained in the availability set before enabling accelerated networking on any of the NICs, so that the VMs end up on a cluster that supports accelerated networking. The stop or deallocate requirement is unnecessary if you disable accelerated networking, because clusters that support accelerated networking also work fine with NICs that don't use accelerated networking.

2. Enable accelerated networking on the NIC of your VM:

```
$nic = Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
-Name "myNic"

$nic.EnableAcceleratedNetworking = $true

$nic | Set-AzNetworkInterface
```

3. Restart your VM or, if in an availability set, all the VMs in the set, and confirm that accelerated networking is enabled:

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

## Virtual machine scale set

A virtual machine scale set is slightly different, but it follows the same workflow.

1. Stop the VMs:

```
Stop-AzVmss -ResourceGroupName "myResourceGroup" `
-VMScaleSetName "myScaleSet"
```

2. Update the accelerated networking property under the network interface:

```
$vmss = Get-AzVmss -ResourceGroupName "myResourceGroup" `
-VMScaleSetName "myScaleSet"

$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations[0].EnableAcceleratedNetworking = $true

Update-AzVmss -ResourceGroupName "myResourceGroup" `
-VMScaleSetName "myScaleSet" `
-VirtualMachineScaleSet $vmss
```

3. Set the applied updates to automatic so that the changes are immediately picked up:

```
$vmss.UpgradePolicy.Mode = "Automatic"

Update-AzVmss -ResourceGroupName "myResourceGroup" `
-VMScaleSetName "myScaleSet" `
-VirtualMachineScaleSet $vmss
```

**NOTE**

A scale set has VM upgrades that apply updates using three different settings: automatic, rolling, and manual. In these instructions, the policy is set to automatic, so the scale set picks up the changes immediately after it restarts.

4. Restart the scale set:

```
Start-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet"
```

Once you restart, wait for the upgrades to finish. After the upgrades are done, the virtual function (VF) appears inside the VM. Make sure you're using a supported OS and VM size.

**Resizing existing VMs with accelerated networking**

If a VM has accelerated networking enabled, you're only able to resize it to a VM that supports accelerated networking.

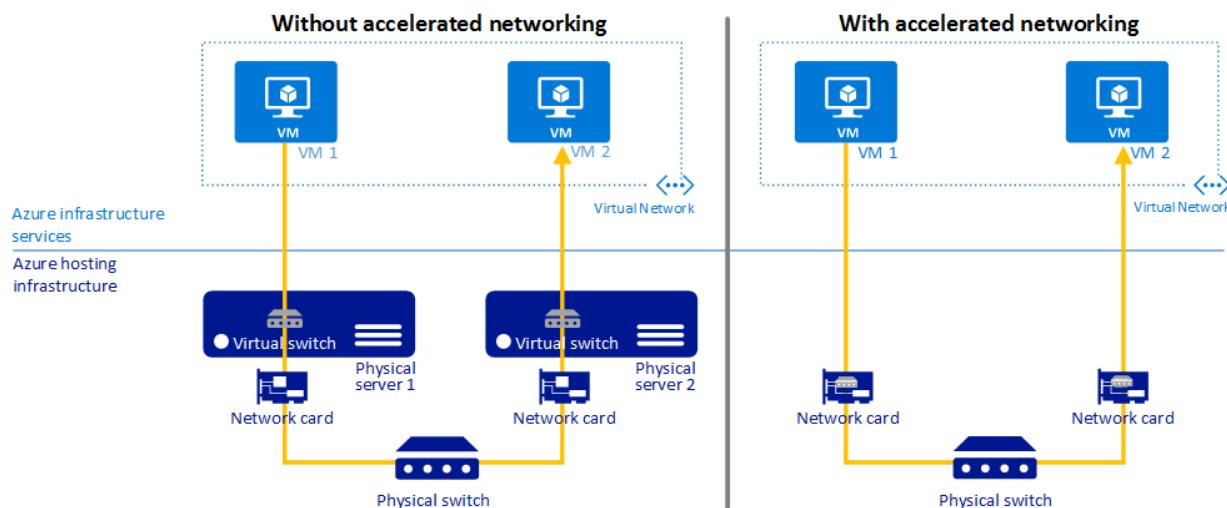
A VM with accelerated networking enabled can't be resized to a VM instance that doesn't support accelerated networking using the resize operation. Instead, to resize one of these VMs:

1. Stop or deallocate the VM. For an availability set or scale set, stop or deallocate all the VMs in the availability set or scale set.
2. Disable accelerated networking on the NIC of the VM. For an availability set or scale set, disable accelerated networking on the NICs of all VMs in the availability set or scale set.
3. After you disable accelerated networking, move the VM, availability set, or scale set to a new size that doesn't support accelerated networking, and then restart them.

# Create a Linux virtual machine with Accelerated Networking using Azure CLI

2/18/2021 • 10 minutes to read • [Edit Online](#)

In this tutorial, you learn how to create a Linux virtual machine (VM) with Accelerated Networking. To create a Windows VM with Accelerated Networking, see [Create a Windows VM with Accelerated Networking](#). Accelerated networking enables single root I/O virtualization (SR-IOV) to a VM, greatly improving its networking performance. This high-performance path bypasses the host from the datapath, reducing latency, jitter, and CPU utilization, for use with the most demanding network workloads on supported VM types. The following picture shows communication between two VMs with and without accelerated networking:



Without accelerated networking, all networking traffic in and out of the VM must traverse the host and the virtual switch. The virtual switch provides all policy enforcement, such as network security groups, access control lists, isolation, and other network virtualized services to network traffic. To learn more about virtual switches, read the [Hyper-V network virtualization and virtual switch](#) article.

With accelerated networking, network traffic arrives at the virtual machine's network interface (NIC), and is then forwarded to the VM. All network policies that the virtual switch applies are now offloaded and applied in hardware. Applying policy in hardware enables the NIC to forward network traffic directly to the VM, bypassing the host and the virtual switch, while maintaining all the policy it applied in the host.

The benefits of accelerated networking only apply to the VM that it is enabled on. For the best results, it is ideal to enable this feature on at least two VMs connected to the same Azure virtual network (VNet). When communicating across VNets or connecting on-premises, this feature has minimal impact to overall latency.

## Benefits

- **Lower Latency / Higher packets per second (pps):** Removing the virtual switch from the datapath removes the time packets spend in the host for policy processing and increases the number of packets that can be processed inside the VM.
- **Reduced jitter:** Virtual switch processing depends on the amount of policy that needs to be applied and the workload of the CPU that is doing the processing. Offloading the policy enforcement to the hardware removes that variability by delivering packets directly to the VM, removing the host to VM communication and all software interrupts and context switches.
- **Decreased CPU utilization:** Bypassing the virtual switch in the host leads to less CPU utilization for

processing network traffic.

## Supported operating systems

The following distributions are supported out of the box from the Azure Gallery:

- Ubuntu 14.04 with the linux-azure kernel
- Ubuntu 16.04 or later
- SLES12 SP3 or later
- RHEL 7.4 or later
- CentOS 7.4 or later
- CoreOS Linux
- Debian "Stretch" with backports kernel, Debian "Buster" or later
- Oracle Linux 7.4 and later with Red Hat Compatible Kernel (RHCK)
- Oracle Linux 7.5 and later with UEK version 5
- FreeBSD 10.4, 11.1 & 12.0 or later

## Limitations and Constraints

### Supported VM instances

Accelerated Networking is supported on most general purpose and compute-optimized instance sizes with 2 or more vCPUs. On instances that support hyperthreading, Accelerated Networking is supported on VM instances with 4 or more vCPUs.

Support for Accelerated Networking can be found in the individual [virtual machine sizes](#) documentation.

### Custom Images

If you are using a custom image, and your image supports Accelerated Networking, please make sure to have the required drivers to work with Mellanox ConnectX-3 and ConnectX-4 Lx NICs on Azure.

### Regions

Available in all public Azure regions as well as Azure Government Clouds.

### Enabling Accelerated Networking on a running VM

A supported VM size without accelerated networking enabled can only have the feature enabled when it is stopped and deallocated.

### Deployment through Azure Resource Manager

Virtual machines (classic) cannot be deployed with Accelerated Networking.

## Create a Linux VM with Azure Accelerated Networking

### Portal creation

Though this article provides steps to create a virtual machine with accelerated networking using the Azure CLI, you can also [create a virtual machine with accelerated networking using the Azure portal](#). When creating a virtual machine in the portal, in the **Create a virtual machine** blade, choose the **Networking** tab. In this tab, there is an option for **Accelerated networking**. If you have chosen a [supported operating system](#) and [VM size](#), this option will automatically populate to "On." If not, it will populate the "Off" option for Accelerated Networking and give the user a reason why it is not be enabled.

- *Note:* Only supported operating systems can be enabled through the portal. If you are using a custom image, and your image supports Accelerated Networking, please create your VM using CLI or PowerShell.

After the virtual machine is created, you can confirm Accelerated Networking is enabled by following the instructions in the [Confirm that accelerated networking is enabled](#).

## CLI creation

### Create a virtual network

Install the latest [Azure CLI](#) and log in to an Azure account using [az login](#). In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *myNic*, and *myVm*.

Create a resource group with [az group create](#). The following example creates a resource group named *myResourceGroup* in the *centralus* location:

```
az group create --name myResourceGroup --location centralus
```

Select a supported Linux region listed in [Linux accelerated networking](#).

Create a virtual network with [az network vnet create](#). The following example creates a virtual network named *myVnet* with one subnet:

```
az network vnet create \
 --resource-group myResourceGroup \
 --name myVnet \
 --address-prefix 192.168.0.0/16 \
 --subnet-name mySubnet \
 --subnet-prefix 192.168.1.0/24
```

### Create a network security group

Create a network security group with [az network nsg create](#). The following example creates a network security group named *myNetworkSecurityGroup*:

```
az network nsg create \
 --resource-group myResourceGroup \
 --name myNetworkSecurityGroup
```

The network security group contains several default rules, one of which disables all inbound access from the Internet. Open a port to allow SSH access to the virtual machine with [az network nsg rule create](#):

```
az network nsg rule create \
 --resource-group myResourceGroup \
 --nsg-name myNetworkSecurityGroup \
 --name Allow-SSH-Internet \
 --access Allow \
 --protocol Tcp \
 --direction Inbound \
 --priority 100 \
 --source-address-prefix Internet \
 --source-port-range "*" \
 --destination-address-prefix "*" \
 --destination-port-range 22
```

### Create a network interface with accelerated networking

Create a public IP address with [az network public-ip create](#). A public IP address isn't required if you don't plan to access the virtual machine from the Internet, but to complete the steps in this article, it is required.

```
az network public-ip create \
--name myPublicIp \
--resource-group myResourceGroup
```

Create a network interface with [az network nic create](#) with accelerated networking enabled. The following example creates a network interface named *myNic* in the *mySubnet* subnet of the *myVnet* virtual network and associates the *myNetworkSecurityGroup* network security group to the network interface:

```
az network nic create \
--resource-group myResourceGroup \
--name myNic \
--vnet-name myVnet \
--subnet mySubnet \
--accelerated-networking true \
--public-ip-address myPublicIp \
--network-security-group myNetworkSecurityGroup
```

## Create a VM and attach the NIC

When you create the VM, specify the NIC you created with `--nics`. Select a size and distribution listed in [Linux accelerated networking](#).

Create a VM with [az vm create](#). The following example creates a VM named *myVM* with the UbuntuLTS image and a size that supports Accelerated Networking (*Standard\_DS4\_v2*):

```
az vm create \
--resource-group myResourceGroup \
--name myVM \
--image UbuntuLTS \
--size Standard_DS4_v2 \
--admin-username azureuser \
--generate-ssh-keys \
--nics myNic
```

For a list of all VM sizes and characteristics, see [Linux VM sizes](#).

Once the VM is created, output similar to the following example output is returned. Take note of the `publicIpAddress`. This address is used to access the VM in subsequent steps.

```
{
 "fqdns": "",
 "id": "/subscriptions/<ID>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
 "location": "centralus",
 "macAddress": "00-0D-3A-23-9A-49",
 "powerState": "VM running",
 "privateIpAddress": "192.168.0.4",
 "publicIpAddress": "40.68.254.142",
 "resourceGroup": "myResourceGroup"
}
```

## Confirm that accelerated networking is enabled

Use the following command to create an SSH session with the VM. Replace `<your-public-ip-address>` with the public IP address assigned to the virtual machine you created, and replace `azureuser` if you used a different value for `--admin-username` when you created the VM.

```
ssh azureuser@<your-public-ip-address>
```

From the Bash shell, enter `uname -r` and confirm that the kernel version is one of the following versions, or greater:

- **Ubuntu 16.04:** 4.11.0-1013
- **SLES SP3:** 4.4.92-6.18
- **RHEL:** 7.4.2017120423
- **CentOS:** 7.4.20171206

Confirm the Mellanox VF device is exposed to the VM with the `lspci` command. The returned output is similar to the following output:

```
0000:00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (AGP disabled) (rev 03)
0000:00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 01)
0000:00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
0000:00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 02)
0000:00:08.0 VGA compatible controller: Microsoft Corporation Hyper-V virtual VGA
0001:00:02.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family [ConnectX-3/ConnectX-3 Pro
Virtual Function]
```

Check for activity on the VF (virtual function) with the `ethtool -S eth0 | grep vf_` command. If you receive output similar to the following sample output, accelerated networking is enabled and working.

```
vf_rx_packets: 992956
vf_rx_bytes: 2749784180
vf_tx_packets: 2656684
vf_tx_bytes: 1099443970
vf_tx_dropped: 0
```

Accelerated Networking is now enabled for your VM.

## Handle dynamic binding and revocation of virtual function

Applications must run over the synthetic NIC that is exposed in VM. If the application runs directly over the VF NIC, it doesn't receive **all** packets that are destined to the VM, since some packets show up over the synthetic interface. If you run an application over the synthetic NIC, it guarantees that the application receives **all** packets that are destined to it. It also makes sure that the application keeps running, even if the VF is revoked when the host is being serviced. Applications binding to the synthetic NIC is a **mandatory** requirement for all applications taking advantage of **Accelerated Networking**.

## Enable Accelerated Networking on existing VMs

If you have created a VM without Accelerated Networking, it is possible to enable this feature on an existing VM. The VM must support Accelerated Networking by meeting the following prerequisites that are also outlined above:

- The VM must be a supported size for Accelerated Networking
- The VM must be a supported Azure Gallery image (and kernel version for Linux)
- All VMs in an availability set or VMSS must be stopped/deallocated before enabling Accelerated Networking on any NIC

### Individual VMs & VMs in an availability set

First stop/deallocate the VM or, if an Availability Set, all the VMs in the Set:

```
az vm deallocate \
--resource-group myResourceGroup \
--name myVM
```

Important, please note, if your VM was created individually, without an availability set, you only need to stop/deallocate the individual VM to enable Accelerated Networking. If your VM was created with an availability set, all VMs contained in the availability set will need to be stopped/deallocated before enabling Accelerated Networking on any of the NICs.

Once stopped, enable Accelerated Networking on the NIC of your VM:

```
az network nic update \
--name myNic \
--resource-group myResourceGroup \
--accelerated-networking true
```

Restart your VM or, if in an Availability Set, all the VMs in the Set and confirm that Accelerated Networking is enabled:

```
az vm start --resource-group myResourceGroup \
--name myVM
```

## VMSS

VMSS is slightly different but follows the same workflow. First, stop the VMs:

```
az vmss deallocate \
--name myvmss \
--resource-group myrg
```

Once the VMs are stopped, update the Accelerated Networking property under the network interface:

```
az vmss update --name myvmss \
--resource-group myrg \
--set
virtualMachineProfile.networkProfile.networkInterfaceConfigurations[0].enableAcceleratedNetworking=true
```

Please note, a VMSS has VM upgrades that apply updates using three different settings, automatic, rolling and manual. In these instructions the policy is set to automatic so that the VMSS will pick up the changes immediately after restarting. To set it to automatic so that the changes are immediately picked up:

```
az vmss update \
--name myvmss \
--resource-group myrg \
--set upgradePolicy.mode="automatic"
```

Finally, restart the VMSS:

```
az vmss start \
--name myvmss \
--resource-group myrg
```

Once you restart, wait for the upgrades to finish but once completed, the VF will appear inside the VM. (Please make sure you are using a supported OS and VM size.)

## **Resizing existing VMs with Accelerated Networking**

VMs with Accelerated Networking enabled can only be resized to VMs that support Accelerated Networking.

A VM with Accelerated Networking enabled cannot be resized to a VM instance that does not support Accelerated Networking using the resize operation. Instead, to resize one of these VMs:

- Stop/Deallocate the VM or if in an availability set/VMSS, stop/deallocate all the VMs in the set/VMSS.
- Accelerated Networking must be disabled on the NIC of the VM or if in an availability set/VMSS, all VMs in the set/VMSS.
- Once Accelerated Networking is disabled, the VM/availability set/VMSS can be moved to a new size that does not support Accelerated Networking and restarted.

# Set up DPDK in a Linux virtual machine

2/18/2021 • 6 minutes to read • [Edit Online](#)

Data Plane Development Kit (DPDK) on Azure offers a faster user-space packet processing framework for performance-intensive applications. This framework bypasses the virtual machine's kernel network stack.

In typical packet processing that uses the kernel network stack, the process is interrupt-driven. When the network interface receives incoming packets, there is a kernel interrupt to process the packet and a context switch from the kernel space to the user space. DPDK eliminates context switching and the interrupt-driven method in favor of a user-space implementation that uses poll mode drivers for fast packet processing.

DPDK consists of sets of user-space libraries that provide access to lower-level resources. These resources can include hardware, logical cores, memory management, and poll mode drivers for network interface cards.

DPDK can run on Azure virtual machines that are supporting multiple operating system distributions. DPDK provides key performance differentiation in driving network function virtualization implementations. These implementations can take the form of network virtual appliances (NVAs), such as virtual routers, firewalls, VPNs, load balancers, evolved packet cores, and denial-of-service (DDoS) applications.

## Benefit

**Higher packets per second (PPS):** Bypassing the kernel and taking control of packets in the user space reduces the cycle count by eliminating context switches. It also improves the rate of packets that are processed per second in Azure Linux virtual machines.

## Supported operating systems

The following distributions from the Azure Marketplace are supported:

LINUX OS	KERNEL VERSION
Ubuntu 16.04	4.15.0-1014-azure+
Ubuntu 18.04	4.15.0-1014-azure+
SLES 15 SP1	4.12.14-8.19-azure+
RHEL 7.5	3.10.0-862.11.6.el7.x86_64+
CentOS 7.5	3.10.0-862.11.6.el7.x86_64+

## Custom kernel support

For any Linux kernel version that's not listed, see [Patches for building an Azure-tuned Linux kernel](#). For more information, you can also contact [aznetdpdk@microsoft.com](mailto:aznetdpdk@microsoft.com).

## Region support

All Azure regions support DPDK.

## Prerequisites

Accelerated networking must be enabled on a Linux virtual machine. The virtual machine should have at least two network interfaces, with one interface for management. Learn how to [create a Linux virtual machine with accelerated networking enabled](#).

## Install DPDK dependencies

### Ubuntu 16.04

```
sudo add-apt-repository ppa:canonical-server/dpdk-azure -y
sudo apt-get update
sudo apt-get install -y librddmacm-dev librddmacm1 build-essential libnuma-dev libmnl-dev
```

### Ubuntu 18.04

```
sudo add-apt-repository ppa:canonical-server/dpdk-azure -y
sudo apt-get update
sudo apt-get install -y librddmacm-dev librddmacm1 build-essential libnuma-dev libmnl-dev
```

### RHEL7.5/CentOS 7.5

```
yum -y groupinstall "Infiniband Support"
sudo dracut --add-drivers "mlx4_en mlx4_ib mlx5_ib" -f
yum install -y gcc kernel-devel-`uname -r` numactl-devel.x86_64 librddmacm-devel libmnl-devel
```

### SLES 15 SP1

#### Azure kernel

```
zypper \
--no-gpg-checks \
--non-interactive \
--gpg-auto-import-keys install kernel-azure kernel-devel-azure gcc make libnuma-devel numactl librddmacm1
rdma-core-devel
```

#### Default kernel

```
zypper \
--no-gpg-checks \
--non-interactive \
--gpg-auto-import-keys install kernel-default-devel gcc make libnuma-devel numactl librddmacm1 rdma-core-
devel
```

## Set up the virtual machine environment (once)

1. [Download the latest DPDK](#). Version 18.11 LTS or 19.11 LTS is required for Azure.
2. Build the default config with `make config T=x86_64-native-linuxapp-gcc`.
3. Enable Mellanox PMDs in the generated config with `sed -ri 's,(MLX._PMD=)n,\1y,' build/.config`.
4. Compile with `make`.
5. Install with `make install DESTDIR=<output folder>`.

## Configure the runtime environment

After restarting, run the following commands once:

## 1. Hugepages

- Configure hugepage by running the following command, once for each numa node:

```
echo 1024 | sudo tee /sys/devices/system/node/node*/hugepages/hugepages-2048kB/nr_hugepages
```

- Create a directory for mounting with `mkdir /mnt/huge`.
- Mount hugepages with `mount -t hugetlbfs nodev /mnt/huge`.
- Check that hugepages are reserved with `grep Huge /proc/meminfo`.

[NOTE] There is a way to modify the grub file so that hugepages are reserved on boot by following the [instructions](#) for the DPDK. The instructions are at the bottom of the page. When you're using an Azure Linux virtual machine, modify files under `/etc/config/grub.d` instead, to reserve hugepages across reboots.

## 2. MAC & IP addresses: Use `ifconfig -a` to view the MAC and IP address of the network interfaces. The *VF* network interface and *NETVSC* network interface have the same MAC address, but only the *NETVSC* network interface has an IP address. *VF* interfaces are running as subordinate interfaces of *NETVSC* interfaces.

## 3. PCI addresses

- Use `ethtool -i <vf interface name>` to find out which PCI address to use for *VF*.
- If *eth0* has accelerated networking enabled, make sure that testpmd doesn't accidentally take over the *VF* pci device for *eth0*. If the DPDK application accidentally takes over the management network interface and causes you to lose your SSH connection, use the serial console to stop the DPDK application. You can also use the serial console to stop or start the virtual machine.

## 4. Load *ibuverbs* on each reboot with `modprobe -a ib_uverbs`. For SLES 15 only, also load *mlx4\_ib* with `modprobe -a mlx4_ib`.

## Failsafe PMD

DPDK applications must run over the failsafe PMD that is exposed in Azure. If the application runs directly over the *VF* PMD, it doesn't receive all packets that are destined to the VM, since some packets show up over the synthetic interface.

If you run a DPDK application over the failsafe PMD, it guarantees that the application receives all packets that are destined to it. It also makes sure that the application keeps running in DPDK mode, even if the VF is revoked when the host is being serviced. For more information about failsafe PMD, see [Fail-safe poll mode driver library](#).

## Run testpmd

To run testpmd in root mode, use `sudo` before the *testpmd* command.

### Basic: Sanity check, failsafe adapter initialization

#### 1. Run the following commands to start a single port testpmd application:

```
testpmd -w <pci address from previous step> \
--vdev="net_vdev_netvsc0,iface=eth1" \
-- -i \
--port-topology=chained
```

2. Run the following commands to start a dual port testpmd application:

```
testpmd -w <pci address nic1> \
-w <pci address nic2> \
--vdev="net_vdev_netvsc0,iface=eth1" \
--vdev="net_vdev_netvsc1,iface=eth2" \
-- -i
```

If you're running testpmd with more than two NICs, the `--vdev` argument follows this pattern:

```
net_vdev_netvsc<id>,iface=<vf's pairing eth> .
```

3. After it's started, run `show port info all` to check port information. You should see one or two DPDK ports that are `net_failsafe` (not `net_ml4`).

4. Use `start <port> /stop <port>` to start traffic.

The previous commands start `testpmd` in interactive mode, which is recommended for trying out testpmd commands.

### Basic: Single sender/single receiver

The following commands periodically print the packets per second statistics:

1. On the TX side, run the following command:

```
testpmd \
-l <core-list> \
-n <num of mem channels> \
-w <pci address of the device you plan to use> \
--vdev="net_vdev_netvsc<id>,iface=<the iface to attach to>" \
-- --port-topology=chained \
--nb-cores <number of cores to use for test pmd> \
--forward-mode=txonly \
--eth-peer=<port id>,<receiver peer MAC address> \
--stats-period <display interval in seconds>
```

2. On the RX side, run the following command:

```
testpmd \
-l <core-list> \
-n <num of mem channels> \
-w <pci address of the device you plan to use> \
--vdev="net_vdev_netvsc<id>,iface=<the iface to attach to>" \
-- --port-topology=chained \
--nb-cores <number of cores to use for test pmd> \
--forward-mode=rxonly \
--eth-peer=<port id>,<sender peer MAC address> \
--stats-period <display interval in seconds>
```

When you're running the previous commands on a virtual machine, change `IP_SRC_ADDR` and `IP_DST_ADDR` in `app/test-pmd/txonly.c` to match the actual IP address of the virtual machines before you compile. Otherwise, the packets are dropped before reaching the receiver.

### Advanced: Single sender/single forwarder

The following commands periodically print the packets per second statistics:

1. On the TX side, run the following command:

```
testpmd \
-l <core-list> \
-n <num of mem channels> \
-w <pci address of the device you plan to use> \
--vdev="net_vdev_netvsc<id>,iface=<the iface to attach to>" \
-- --port-topology=chained \
--nb-cores <number of cores to use for test pmd> \
--forward-mode=txonly \
--eth-peer=<port id>,<receiver peer MAC address> \
--stats-period <display interval in seconds>
```

2. On the FWD side, run the following command:

```
testpmd \
-l <core-list> \
-n <num of mem channels> \
-w <pci address NIC1> \
-w <pci address NIC2> \
--vdev="net_vdev_netvsc<id>,iface=<the iface to attach to>" \
--vdev="net_vdev_netvsc<2nd id>,iface=<2nd iface to attach to>" (you need as many --vdev arguments as the number of devices used by testpmd, in this case) \
-- --nb-cores <number of cores to use for test pmd> \
--forward-mode=io \
--eth-peer=<recv port id>,<sender peer MAC address> \
--stats-period <display interval in seconds>
```

When you're running the previous commands on a virtual machine, change *IP\_SRC\_ADDR* and *IP\_DST\_ADDR* in `app/test-pmd/txonly.c` to match the actual IP address of the virtual machines before you compile. Otherwise, the packets are dropped before reaching the forwarder. You won't be able to have a third machine receive forwarded traffic, because the *testpmd* forwarder doesn't modify the layer-3 addresses, unless you make some code changes.

## References

- [EAL options](#)
- [Testpmd commands](#)

# TCP/IP performance tuning for Azure VMs

2/18/2021 • 23 minutes to read • [Edit Online](#)

This article discusses common TCP/IP performance tuning techniques and some things to consider when you use them for virtual machines running on Azure. It will provide a basic overview of the techniques and explore how they can be tuned.

## Common TCP/IP tuning techniques

### **MTU, fragmentation, and large send offload**

#### **MTU**

The maximum transmission unit (MTU) is the largest size frame (packet), specified in bytes, that can be sent over a network interface. The MTU is a configurable setting. The default MTU used on Azure VMs, and the default setting on most network devices globally, is 1,500 bytes.

#### **Fragmentation**

Fragmentation occurs when a packet is sent that exceeds the MTU of a network interface. The TCP/IP stack will break the packet into smaller pieces (fragments) that conform to the interface's MTU. Fragmentation occurs at the IP layer and is independent of the underlying protocol (such as TCP). When a 2,000-byte packet is sent over a network interface with an MTU of 1,500, the packet will be broken down into one 1,500-byte packet and one 500-byte packet.

Network devices in the path between a source and destination can either drop packets that exceed the MTU or fragment the packet into smaller pieces.

#### **The Don't Fragment bit in an IP packet**

The Don't Fragment (DF) bit is a flag in the IP protocol header. The DF bit indicates that network devices on the path between the sender and receiver must not fragment the packet. This bit could be set for many reasons. (See the "Path MTU Discovery" section of this article for one example.) When a network device receives a packet with the Don't Fragment bit set, and that packet exceeds the device's interface MTU, the standard behavior is for the device to drop the packet. The device sends an ICMP Fragmentation Needed message back to the original source of the packet.

#### **Performance implications of fragmentation**

Fragmentation can have negative performance implications. One of the main reasons for the effect on performance is the CPU/memory impact of the fragmentation and reassembly of packets. When a network device needs to fragment a packet, it will have to allocate CPU/memory resources to perform fragmentation.

The same thing happens when the packet is reassembled. The network device has to store all the fragments until they're received so it can reassemble them into the original packet. This process of fragmentation and reassembly can also cause latency.

The other possible negative performance implication of fragmentation is that fragmented packets might arrive out of order. When packets are received out of order, some types of network devices can drop them. When that happens, the whole packet has to be retransmitted.

Fragments are typically dropped by security devices like network firewalls or when a network device's receive buffers are exhausted. When a network device's receive buffers are exhausted, a network device is attempting to reassemble a fragmented packet but doesn't have the resources to store and reassume the packet.

Fragmentation can be seen as a negative operation, but support for fragmentation is necessary when you're connecting diverse networks over the internet.

## **Benefits and consequences of modifying the MTU**

Generally speaking, you can create a more efficient network by increasing the MTU. Every packet that's transmitted has header information that's added to the original packet. When fragmentation creates more packets, there's more header overhead, and that makes the network less efficient.

Here's an example. The Ethernet header size is 14 bytes plus a 4-byte frame check sequence to ensure frame consistency. If one 2,000-byte packet is sent, 18 bytes of Ethernet overhead is added on the network. If the packet is fragmented into a 1,500-byte packet and a 500-byte packet, each packet will have 18 bytes of Ethernet header, a total of 36 bytes.

Keep in mind that increasing the MTU won't necessarily create a more efficient network. If an application sends only 500-byte packets, the same header overhead will exist whether the MTU is 1,500 bytes or 9,000 bytes. The network will become more efficient only if it uses larger packet sizes that are affected by the MTU.

### **Azure and VM MTU**

The default MTU for Azure VMs is 1,500 bytes. The Azure Virtual Network stack will attempt to fragment a packet at 1,400 bytes.

Note that the Virtual Network stack isn't inherently inefficient because it fragments packets at 1,400 bytes even though VMs have an MTU of 1,500. A large percentage of network packets are much smaller than 1,400 or 1,500 bytes.

### **Azure and fragmentation**

Virtual Network stack is set up to drop "out of order fragments," that is, fragmented packets that don't arrive in their original fragmented order. These packets are dropped mainly because of a network security vulnerability announced in November 2018 called FragmentSmack.

FragmentSmack is a defect in the way the Linux kernel handled reassembly of fragmented IPv4 and IPv6 packets. A remote attacker could use this flaw to trigger expensive fragment reassembly operations, which could lead to increased CPU and a denial of service on the target system.

### **Tune the MTU**

You can configure an Azure VM MTU, as you can in any other operating system. But you should consider the fragmentation that occurs in Azure, described above, when you're configuring an MTU.

We don't encourage customers to increase VM MTUs. This discussion is meant to explain the details of how Azure implements MTU and performs fragmentation.

#### **IMPORTANT**

Increasing MTU isn't known to improve performance and could have a negative effect on application performance.

### **Large send offload**

Large send offload (LSO) can improve network performance by offloading the segmentation of packets to the Ethernet adapter. When LSO is enabled, the TCP/IP stack creates a large TCP packet and sends it to the Ethernet adapter for segmentation before forwarding it. The benefit of LSO is that it can free the CPU from segmenting packets into sizes that conform to the MTU and offload that processing to the Ethernet interface where it's performed in hardware. To learn more about the benefits of LSO, see [Supporting large send offload](#).

When LSO is enabled, Azure customers might see large frame sizes when they perform packet captures. These large frame sizes might lead some customers to think fragmentation is occurring or that a large MTU is being used when it's not. With LSO, the Ethernet adapter can advertise a larger maximum segment size (MSS) to the TCP/IP stack to create a larger TCP packet. This entire non-segmented frame is then forwarded to the Ethernet adapter and would be visible in a packet capture performed on the VM. But the packet will be broken down into many smaller frames by the Ethernet adapter, according to the Ethernet adapter's MTU.

### **TCP MSS window scaling and PMTUD**

## TCP maximum segment size

TCP maximum segment size (MSS) is a setting that limits the size of TCP segments, which avoids fragmentation of TCP packets. Operating systems will typically use this formula to set MSS:

$$\text{MSS} = \text{MTU} - (\text{IP header size} + \text{TCP header size})$$

The IP header and the TCP header are 20 bytes each, or 40 bytes total. So an interface with an MTU of 1,500 will have an MSS of 1,460. But the MSS is configurable.

This setting is agreed to in the TCP three-way handshake when a TCP session is set up between a source and a destination. Both sides send an MSS value, and the lower of the two is used for the TCP connection.

Keep in mind that the MTUs of the source and destination aren't the only factors that determine the MSS value. Intermediary network devices, like VPN gateways, including Azure VPN Gateway, can adjust the MTU independently of the source and destination to ensure optimal network performance.

### Path MTU Discovery

MSS is negotiated, but it might not indicate the actual MSS that can be used. This is because other network devices in the path between the source and the destination might have a lower MTU value than the source and destination. In this case, the device whose MTU is smaller than the packet will drop the packet. The device will send back an ICMP Fragmentation Needed (Type 3, Code 4) message that contains its MTU. This ICMP message allows the source host to reduce its Path MTU appropriately. The process is called Path MTU Discovery (PMTUD).

The PMTUD process is inefficient and affects network performance. When packets are sent that exceed a network path's MTU, the packets need to be retransmitted with a lower MSS. If the sender doesn't receive the ICMP Fragmentation Needed message, maybe because of a network firewall in the path (commonly referred to as a *PMTUD blackhole*), the sender doesn't know it needs to lower the MSS and will continuously retransmit the packet. This is why we don't recommend increasing the Azure VM MTU.

### VPN and MTU

If you use VMs that perform encapsulation (like IPsec VPNs), there are some additional considerations regarding packet size and MTU. VPNs add more headers to packets, which increases the packet size and requires a smaller MSS.

For Azure, we recommend that you set TCP MSS clamping to 1,350 bytes and tunnel interface MTU to 1,400. For more information, see the [VPN devices and IPsec/IKE parameters page](#).

## Latency, round-trip time, and TCP window scaling

### Latency and round-trip time

Network latency is governed by the speed of light over a fiber optic network. Network throughput of TCP is also effectively governed by the round-trip time (RTT) between two network devices.

ROUTE	DISTANCE	ONE-WAY TIME	RTT
New York to San Francisco	4,148 km	21 ms	42 ms
New York to London	5,585 km	28 ms	56 ms
New York to Sydney	15,993 km	80 ms	160 ms

This table shows the straight-line distance between two locations. In networks, the distance is typically longer than the straight-line distance. Here's a simple formula to calculate minimum RTT as governed by the speed of light:

$$\text{minimum RTT} = 2 * (\text{Distance in kilometers} / \text{Speed of propagation})$$

You can use 200 for the speed of propagation. This is the distance, in kilometers, that light travels in 1

millisecond.

Let's take New York to San Francisco as an example. The straight-line distance is 4,148 km. Plugging that value into the equation, we get the following:

$$\text{Minimum RTT} = 2 * (4,148 / 200)$$

The output of the equation is in milliseconds.

If you want to get the best network performance, the logical option is to select destinations with the shortest distance between them. You should also design your virtual network to optimize the path of traffic and reduce latency. For more information, see the "Network design considerations" section of this article.

#### Latency and round-trip time effects on TCP

Round-trip time has a direct effect on maximum TCP throughput. In TCP protocol, *window size* is the maximum amount of traffic that can be sent over a TCP connection before the sender needs to receive acknowledgement from the receiver. If the TCP MSS is set to 1,460 and the TCP window size is set to 65,535, the sender can send 45 packets before it has to receive acknowledgement from the receiver. If the sender doesn't get acknowledgement, it will retransmit the data. Here's the formula:

$$\text{TCP window size} / \text{TCP MSS} = \text{packets sent}$$

In this example, 65,535 / 1,460 is rounded up to 45.

This "waiting for acknowledgement" state, a mechanism to ensure reliable delivery of data, is what causes RTT to affect TCP throughput. The longer the sender waits for acknowledgement, the longer it needs to wait before sending more data.

Here's the formula for calculating the maximum throughput of a single TCP connection:

$$\text{Window size} / (\text{RTT latency in milliseconds} / 1,000) = \text{maximum bytes/second}$$

This table shows the maximum megabytes/second throughput of a single TCP connection. (For readability, megabytes is used for the unit of measure.)

TCP WINDOW SIZE (BYTES)	RTT LATENCY (MS)	MAXIMUM MEGABYTE/SECOND THROUGHPUT	MAXIMUM MEGABIT/SECOND THROUGHPUT
65,535	1	65.54	524.29
65,535	30	2.18	17.48
65,535	60	1.09	8.74
65,535	90	.73	5.83
65,535	120	.55	4.37

If packets are lost, the maximum throughput of a TCP connection will be reduced while the sender retransmits data it has already sent.

#### TCP window scaling

TCP window scaling is a technique that dynamically increases the TCP window size to allow more data to be sent before an acknowledgement is required. In the previous example, 45 packets would be sent before an acknowledgement was required. If you increase the number of packets that can be sent before an acknowledgement is needed, you're reducing the number of times a sender is waiting for acknowledgement, which increases the TCP maximum throughput.

This table illustrates those relationships:

TCP WINDOW SIZE (BYTES)	RTT LATENCY (MS)	MAXIMUM MEGABYTE/SECOND THROUGHPUT	MAXIMUM MEGABIT/SECOND THROUGHPUT
65,535	30	2.18	17.48
131,070	30	4.37	34.95
262,140	30	8.74	69.91
524,280	30	17.48	139.81

But the TCP header value for TCP window size is only 2 bytes long, which means the maximum value for a receive window is 65,535. To increase the maximum window size, a TCP window scale factor was introduced.

The scale factor is also a setting that you can configure in an operating system. Here's the formula for calculating the TCP window size by using scale factors:

$$\text{TCP window size} = \text{TCP window size in bytes} \times (2^{\text{scale factor}})$$

Here's the calculation for a window scale factor of 3 and a window size of 65,535:

$$65,535 \times (2^3) = 262,140 \text{ bytes}$$

A scale factor of 14 results in a TCP window size of 14 (the maximum offset allowed). The TCP window size will be 1,073,725,440 bytes (8.5 gigabits).

#### Support for TCP window scaling

Windows can set different scaling factors for different connection types. (Classes of connections include datacenter, internet, and so on.) You use the `Get-NetTCPConnection` PowerShell command to view the window scaling connection type:

```
Get-NetTCPConnection
```

You can use the `Get-NetTCPSetting` PowerShell command to view the values of each class:

```
Get-NetTCPSetting
```

You can set the initial TCP window size and TCP scaling factor in Windows by using the `Set-NetTCPSetting` PowerShell command. For more information, see [Set-NetTCPSetting](#).

```
Set-NetTCPSetting
```

These are the effective TCP settings for `AutoTuningLevel`:

AUTOTUNINGLEVEL	SCALING FACTOR	SCALING MULTIPLIER	FORMULA TO CALCULATE MAXIMUM WINDOW SIZE
Disabled	None	None	Window size
Restricted	4	$2^4$	Window size * $(2^4)$

AUTOTUNINGLEVEL	SCALING FACTOR	SCALING MULTIPLIER	FORMULA TO CALCULATE MAXIMUM WINDOW SIZE
Highly restricted	2	$2^2$	Window size * $(2^2)$
Normal	8	$2^8$	Window size * $(2^8)$
Experimental	14	$2^{14}$	Window size * $(2^{14})$

These settings are the most likely to affect TCP performance, but keep in mind that many other factors across the internet, outside the control of Azure, can also affect TCP performance.

#### Increase MTU size

Because a larger MTU means a larger MSS, you might wonder whether increasing the MTU can increase TCP performance. Probably not. There are pros and cons to packet size beyond just TCP traffic. As discussed earlier, the most important factors affecting TCP throughput performance are TCP window size, packet loss, and RTT.

#### IMPORTANT

We don't recommend that Azure customers change the default MTU value on virtual machines.

### Accelerated networking and receive side scaling

#### Accelerated networking

Virtual machine network functions have historically been CPU intensive on both the guest VM and the hypervisor/host. Every packet that transits through the host is processed in software by the host CPU, including all virtual network encapsulation and decapsulation. So the more traffic that goes through the host, the higher the CPU load. And if the host CPU is busy with other operations, that will also affect network throughput and latency. Azure addresses this issue with accelerated networking.

Accelerated networking provides consistent ultralow network latency via the in-house programmable hardware of Azure and technologies like SR-IOV. Accelerated networking moves much of the Azure software-defined networking stack off the CPUs and into FPGA-based SmartNICs. This change enables end-user applications to reclaim compute cycles, which puts less load on the VM, decreasing jitter and inconsistency in latency. In other words, performance can be more deterministic.

Accelerated networking improves performance by allowing the guest VM to bypass the host and establish a datapath directly with a host's SmartNIC. Here are some benefits of accelerated networking:

- **Lower latency / higher packets per second (pps):** Removing the virtual switch from the datapath eliminates the time packets spend in the host for policy processing and increases the number of packets that can be processed in the VM.
- **Reduced jitter:** Virtual switch processing depends on the amount of policy that needs to be applied and the workload of the CPU that's doing the processing. Offloading the policy enforcement to the hardware removes that variability by delivering packets directly to the VM, eliminating the host-to-VM communication and all software interrupts and context switches.
- **Decreased CPU utilization:** Bypassing the virtual switch in the host leads to less CPU utilization for processing network traffic.

To use accelerated networking, you need to explicitly enable it on each applicable VM. See [Create a Linux virtual machine with Accelerated Networking](#) for instructions.

#### Receive side scaling

Receive side scaling (RSS) is a network driver technology that distributes the receiving of network traffic more

efficiently by distributing receive processing across multiple CPUs in a multiprocessor system. In simple terms, RSS allows a system to process more received traffic because it uses all available CPUs instead of just one. For a more technical discussion of RSS, see [Introduction to receive side scaling](#).

To get the best performance when accelerated networking is enabled on a VM, you need to enable RSS. RSS can also provide benefits on VMs that don't use accelerated networking. For an overview of how to determine if RSS is enabled and how to enable it, see [Optimize network throughput for Azure virtual machines](#).

### TCP TIME\_WAIT and TIME\_WAIT assassination

TCP TIME\_WAIT is another common setting that affects network and application performance. On busy VMs that are opening and closing many sockets, either as clients or as servers (Source IP:Source Port + Destination IP:Destination Port), during the normal operation of TCP, a given socket can end up in a TIME\_WAIT state for a long time. The TIME\_WAIT state is meant to allow any additional data to be delivered on a socket before closing it. So TCP/IP stacks generally prevent the reuse of a socket by silently dropping the client's TCP SYN packet.

The amount of time a socket is in TIME\_WAIT is configurable. It could range from 30 seconds to 240 seconds. Sockets are a finite resource, and the number of sockets that can be used at any given time is configurable. (The number of available sockets is typically about 30,000.) If the available sockets are consumed, or if clients and servers have mismatched TIME\_WAIT settings, and a VM tries to reuse a socket in a TIME\_WAIT state, new connections will fail as TCP SYN packets are silently dropped.

The value for port range for outbound sockets is usually configurable within the TCP/IP stack of an operating system. The same thing is true for TCP TIME\_WAIT settings and socket reuse. Changing these numbers can potentially improve scalability. But, depending on the situation, these changes could cause interoperability issues. You should be careful if you change these values.

You can use TIME\_WAIT assassination to address this scaling limitation. TIME\_WAIT assassination allows a socket to be reused in certain situations, like when the sequence number in the IP packet of the new connection exceeds the sequence number of the last packet from the previous connection. In this case, the operating system will allow the new connection to be established (it will accept the new SYN/ACK) and force close the previous connection that was in a TIME\_WAIT state. This capability is supported on Windows VMs in Azure. To learn about support in other VMs, check with the OS vendor.

To learn about configuring TCP TIME\_WAIT settings and source port range, see [Settings that can be modified to improve network performance](#).

## Virtual network factors that can affect performance

### VM maximum outbound throughput

Azure provides a variety of VM sizes and types, each with a different mix of performance capabilities. One of these capabilities is network throughput (or bandwidth), which is measured in megabits per second (Mbps). Because virtual machines are hosted on shared hardware, the network capacity needs to be shared fairly among the virtual machines using the same hardware. Larger virtual machines are allocated more bandwidth than smaller virtual machines.

The network bandwidth allocated to each virtual machine is metered on egress (outbound) traffic from the virtual machine. All network traffic leaving the virtual machine is counted toward the allocated limit, regardless of destination. For example, if a virtual machine has a 1,000-Mbps limit, that limit applies whether the outbound traffic is destined for another virtual machine in the same virtual network or one outside of Azure.

Ingress is not metered or limited directly. But there are other factors, like CPU and storage limits, that can affect a virtual machine's ability to process incoming data.

Accelerated networking is designed to improve network performance, including latency, throughput, and CPU utilization. Accelerated networking can improve a virtual machine's throughput, but it can do that only up to the virtual machine's allocated bandwidth.

Azure virtual machines have at least one network interface attached to them. They might have several. The bandwidth allocated to a virtual machine is the sum of all outbound traffic across all network interfaces attached to the machine. In other words, the bandwidth is allocated on a per-virtual machine basis, regardless of how many network interfaces are attached to the machine.

Expected outbound throughput and the number of network interfaces supported by each VM size are detailed in [Sizes for Windows virtual machines in Azure](#). To see maximum throughput, select a type, like **General purpose**, and then find the section about the size series on the resulting page (for example, "Dv2-series"). For each series, there's a table that provides networking specifications in the last column, which is titled "Max NICs / Expected network bandwidth (Mbps)."

The throughput limit applies to the virtual machine. Throughput is not affected by these factors:

- **Number of network interfaces:** The bandwidth limit applies to the sum of all outbound traffic from the virtual machine.
- **Accelerated networking:** Though this feature can be helpful in achieving the published limit, it doesn't change the limit.
- **Traffic destination:** All destinations count toward the outbound limit.
- **Protocol:** All outbound traffic over all protocols counts towards the limit.

For more information, see [Virtual machine network bandwidth](#).

## Internet performance considerations

As discussed throughout this article, factors on the internet and outside the control of Azure can affect network performance. Here are some of those factors:

- **Latency:** The round-trip time between two destinations can be affected by issues on intermediate networks, by traffic that doesn't take the "shortest" distance path, and by suboptimal peering paths.
- **Packet loss:** Packet loss can be caused by network congestion, physical path issues, and underperforming network devices.
- **MTU size/Fragmentation:** Fragmentation along the path can lead to delays in data arrival or in packets arriving out of order, which can affect the delivery of packets.

Traceroute is a good tool for measuring network performance characteristics (like packet loss and latency) along every network path between a source device and a destination device.

## Network design considerations

Along with the considerations discussed earlier in this article, the topology of a virtual network can affect the network's performance. For example, a hub-and-spoke design that backhauls traffic globally to a single-hub virtual network will introduce network latency, which will affect overall network performance.

The number of network devices that network traffic passes through can also affect overall latency. For example, in a hub-and-spoke design, if traffic passes through a spoke network virtual appliance and a hub virtual appliance before transiting to the internet, the network virtual appliances can introduce latency.

## Azure regions, virtual networks, and latency

Azure regions are made up of multiple datacenters that exist within a general geographic area. These datacenters might not be physically next to each other. In some cases they're separated by as much as 10 kilometers. The virtual network is a logical overlay on top of the Azure physical datacenter network. A virtual network doesn't imply any specific network topology within the datacenter.

For example, two VMs that are in the same virtual network and subnet might be in different racks, rows, or even datacenters. They could be separated by feet of fiber optic cable or by kilometers of fiber optic cable. This

variation could introduce variable latency (a few milliseconds difference) between different VMs.

The geographic placement of VMs, and the potential resulting latency between two VMs, can be influenced by the configuration of availability sets and Availability Zones. But the distance between datacenters in a region is region-specific and primarily influenced by datacenter topology in the region.

### Source NAT port exhaustion

A deployment in Azure can communicate with endpoints outside of Azure on the public internet and/or in the public IP space. When an instance initiates an outbound connection, Azure dynamically maps the private IP address to a public IP address. After Azure creates this mapping, return traffic for the outbound originated flow can also reach the private IP address where the flow originated.

For every outbound connection, the Azure Load Balancer needs to maintain this mapping for some period of time. With the multitenant nature of Azure, maintaining this mapping for every outbound flow for every VM can be resource intensive. So there are limits that are set and based on the configuration of the Azure Virtual Network. Or, to say that more precisely, an Azure VM can only make a certain number of outbound connections at a given time. When these limits are reached, the VM won't be able to make more outbound connections.

But this behavior is configurable. For more information about SNAT and SNAT port exhaustion, see [this article](#).

## Measure network performance on Azure

A number of the performance maximums in this article are related to the network latency / round-trip time (RTT) between two VMs. This section provides some suggestions for how to test latency/RTT and how to test TCP performance and VM network performance. You can tune and performance test the TCP/IP and network values discussed earlier by using the techniques described in this section. You can plug latency, MTU, MSS, and window size values into the calculations provided earlier and compare theoretical maximums to actual values that you observe during testing.

### Measure round-trip time and packet loss

TCP performance relies heavily on RTT and packet Loss. The PING utility available in Windows and Linux provides the easiest way to measure RTT and packet loss. The output of PING will show the minimum/maximum/average latency between a source and destination. It will also show packet loss. PING uses the ICMP protocol by default. You can use PsPing to test TCP RTT. For more information, see [PsPing](#).

### Measure actual throughput of a TCP connection

NTttcp is a tool for testing the TCP performance of a Linux or Windows VM. You can change various TCP settings and then test the benefits by using NTttcp. For more information, see these resources:

- [Bandwidth/Throughput testing \(NTttcp\)](#)
- [NTttcp Utility](#)

### Measure actual bandwidth of a virtual machine

You can test the performance of different VM types, accelerated networking, and so on, by using a tool called iPerf. iPerf is also available on Linux and Windows. iPerf can use TCP or UDP to test overall network throughput. iPerf TCP throughput tests are influenced by the factors discussed in this article (like latency and RTT). So UDP might yield better results if you just want to test maximum throughput.

For more information, see these articles:

- [Troubleshooting Expressroute network performance](#)
- [How to validate VPN throughput to a virtual network](#)

### Detect inefficient TCP behaviors

In packet captures, Azure customers might see TCP packets with TCP flags (SACK, DUP ACK, RETRANSMIT, and

FAST RETRANSMIT) that could indicate network performance problems. These packets specifically indicate network inefficiencies that result from packet loss. But packet loss isn't necessarily caused by Azure performance problems. Performance problems could be the result of application problems, operating system problems, or other problems that might not be directly related to the Azure platform.

Also, keep in mind that some retransmission and duplicate ACKs are normal on a network. TCP protocols were built to be reliable. Evidence of these TCP packets in a packet capture doesn't necessarily indicate a systemic network problem, unless they're excessive.

Still, these packet types are indications that TCP throughput isn't achieving its maximum performance, for reasons discussed in other sections of this article.

## Next steps

Now that you've learned about TCP/IP performance tuning for Azure VMs, you might want to read about other considerations for [planning virtual networks](#) or [learn more about connecting and configuring virtual networks](#).

# Virtual machine network bandwidth

2/18/2021 • 3 minutes to read • [Edit Online](#)

Azure offers a variety of VM sizes and types, each with a different mix of performance capabilities. One capability is network throughput (or bandwidth), measured in megabits per second (Mbps). Because virtual machines are hosted on shared hardware, the network capacity must be shared fairly among the virtual machines sharing the same hardware. Larger virtual machines are allocated relatively more bandwidth than smaller virtual machines.

The network bandwidth allocated to each virtual machine is metered on egress (outbound) traffic from the virtual machine. All network traffic leaving the virtual machine is counted toward the allocated limit, regardless of destination. For example, if a virtual machine has a 1,000 Mbps limit, that limit applies whether the outbound traffic is destined for another virtual machine in the same virtual network, or outside of Azure.

Ingress is not metered or limited directly. However, there are other factors, such as CPU and storage limits, which can impact a virtual machine's ability to process incoming data.

Accelerated networking is a feature designed to improve network performance, including latency, throughput, and CPU utilization. While accelerated networking can improve a virtual machine's throughput, it can do so only up to the virtual machine's allocated bandwidth. To learn more about Accelerated networking, see Accelerated networking for [Windows](#) or [Linux](#) virtual machines.

Azure virtual machines must have one, but may have several, network interfaces attached to them. Bandwidth allocated to a virtual machine is the sum of all outbound traffic across all network interfaces attached to a virtual machine. In other words, the allocated bandwidth is per virtual machine, regardless of how many network interfaces are attached to the virtual machine. To learn how many network interfaces different Azure VM sizes support, see Azure [Windows](#) and [Linux](#) VM sizes.

## Expected network throughput

Expected outbound throughput and the number of network interfaces supported by each VM size is detailed in Azure [Windows](#) and [Linux](#) VM sizes. Select a type, such as General purpose, then select a size-series on the resulting page, such as the Dv2-series. Each series has a table with networking specifications in the last column titled, **Max NICs / Expected network performance (Mbps)**.

The throughput limit applies to the virtual machine. Throughput is unaffected by the following factors:

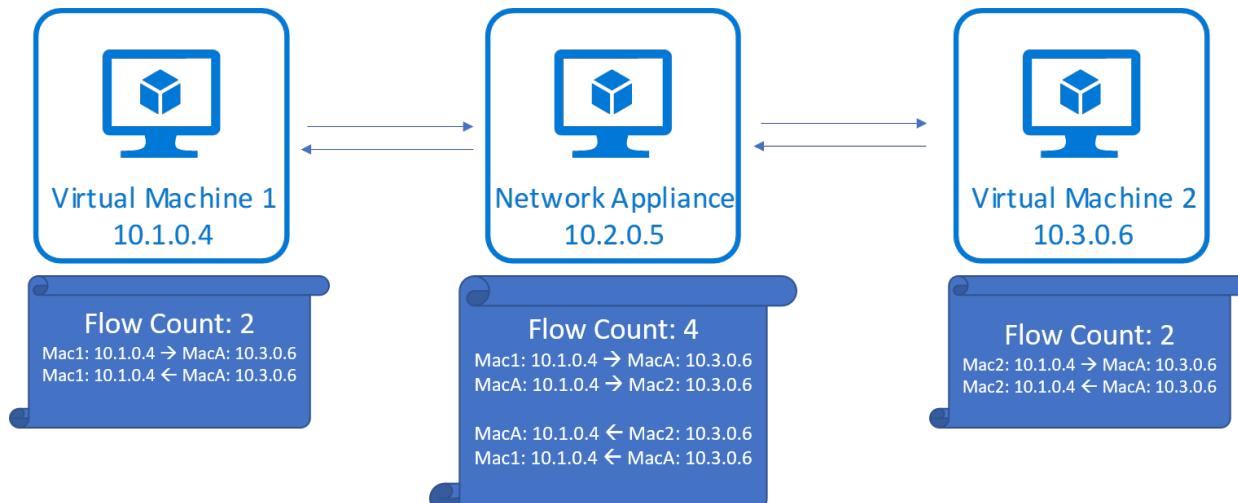
- **Number of network interfaces:** The bandwidth limit is cumulative of all outbound traffic from the virtual machine.
- **Accelerated networking:** Though the feature can be helpful in achieving the published limit, it does not change the limit.
- **Traffic destination:** All destinations count toward the outbound limit.
- **Protocol:** All outbound traffic over all protocols counts towards the limit.

## Network Flow Limits

In addition to bandwidth, the number of network connections present on a VM at any given time can affect its network performance. The Azure networking stack maintains state for each direction of a TCP/UDP connection in data structures called 'flows'. A typical TCP/UDP connection will have 2 flows created, one for the inbound and another for the outbound direction.

Data transfer between endpoints requires creation of several flows in addition to those that perform the data

transfer. Some examples are flows created for DNS resolution and flows created for load balancer health probes. Also note that network virtual appliances (NVAs) such as gateways, proxies, firewalls, will see flows being created for connections terminated at the appliance and originated by the appliance.



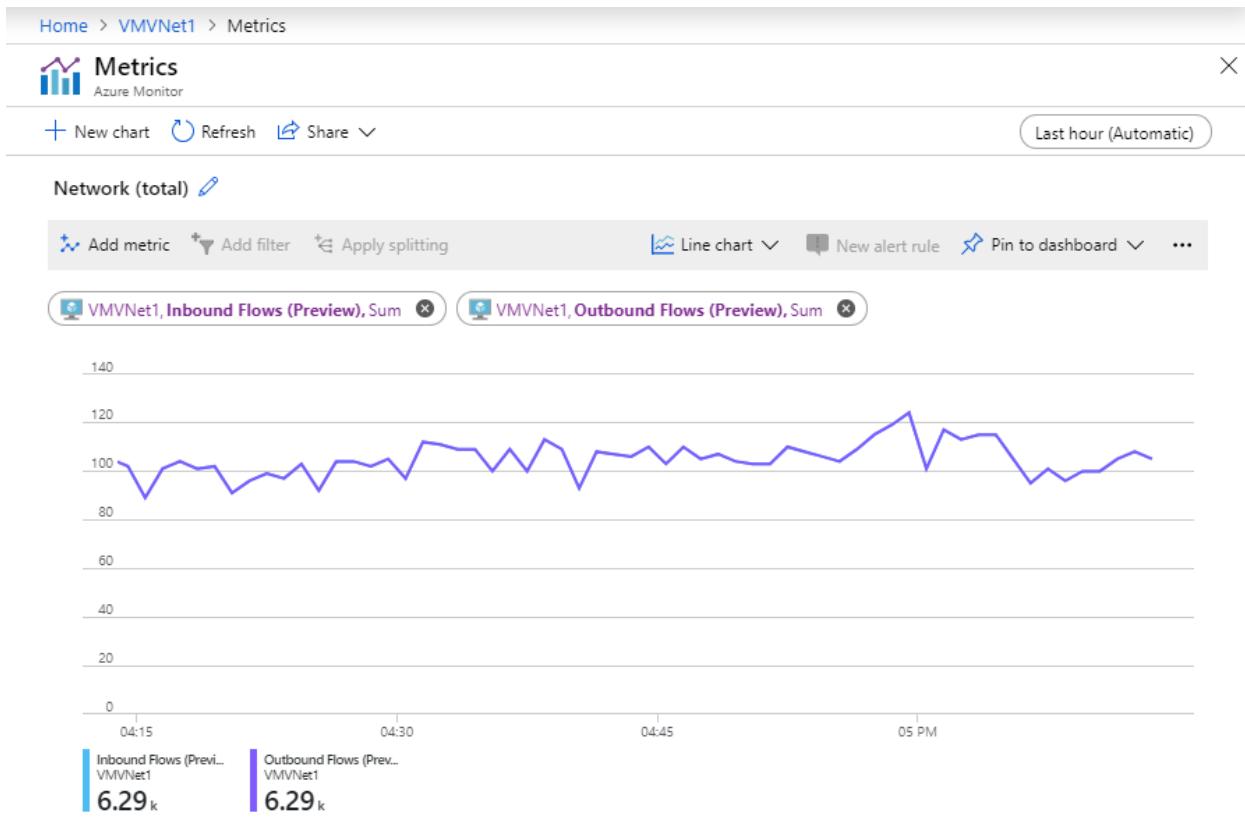
## Flow Limits and Active Connections Recommendations

Today, the Azure networking stack supports 1M total flows (500k inbound and 500k outbound) for a VM. Total active connections that can be handled by a VM in different scenarios are as follows.

- VMs that belongs to VNET can handle 500k ***active connections*** for all VM sizes with 500k ***active flows in each direction***.
- VMs with network virtual appliances (NVAs) such as gateway, proxy, firewall can handle 250k ***active connections*** with 500k ***active flows in each direction*** due to the forwarding and additional new flow creation on new connection setup to the next hop as shown in the above diagram.

Once this limit is hit, additional connections are dropped. Connection establishment and termination rates can also affect network performance as connection establishment and termination shares CPU with packet processing routines. We recommend that you benchmark workloads against expected traffic patterns and scale out workloads appropriately to match your performance needs.

Metrics are available in [Azure Monitor](#) to track the number of network flows and the flow creation rate on your VM or VMSS instances.



Connection establishment and termination rates can also affect network performance as connection establishment and termination shares CPU with packet processing routines. We recommend that you benchmark workloads against expected traffic patterns and scale out workloads appropriately to match your performance needs.

## Next steps

- [Optimize network throughput for a virtual machine operating system](#)
- [Test network throughput for a virtual machine.](#)

# Tutorial: Move Azure VMs across regions

2/18/2021 • 8 minutes to read • [Edit Online](#)

In this article, learn how to move Azure VMs, and related network/storage resources, to a different Azure region, using [Azure Resource Mover](#)..

In this tutorial, you learn how to:

- Check prerequisites and requirements.
- Select the resources you want to move.
- Resolve resource dependencies.
- Prepare and move the source resource group.
- Prepare and move the other resources.
- Decide whether you want to discard or commit the move.
- Optionally remove resources in the source region after the move.

## NOTE

Tutorials show the quickest path for trying out a scenario, and use default options.

If you don't have an Azure subscription, create a [free account](#) before you begin. Then sign in to the [Azure portal](#).

## Prerequisites

REQUIREMENT	DESCRIPTION
<b>Subscription permissions</b>	<p>Check you have <i>Owner</i> access on the subscription containing the resources that you want to move</p> <p><b>Why do I need Owner access?</b> The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a <a href="#">system-assigned managed identity</a> (formerly known as Managed Service Identity (MSI)) that's trusted by the subscription. To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs <i>Owner</i> permissions on the subscription. <a href="#">Learn more</a> about Azure roles.</p>
<b>VM support</b>	<p>Check that the VMs you want to move are supported.</p> <ul style="list-style-type: none"><li>- <a href="#">Verify</a> supported Windows VMs.</li><li>- <a href="#">Verify</a> supported Linux VMs and kernel versions.</li><li>- Check supported <a href="#">compute</a>, <a href="#">storage</a>, and <a href="#">networking</a> settings.</li></ul>
<b>Destination subscription</b>	The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have quota, <a href="#">request additional limits</a> .

Requirement	Description
Destination region charges	Verify pricing and charges associated with the target region to which you're moving VMs. Use the <a href="#">pricing calculator</a> to help you.

## Prepare VMs

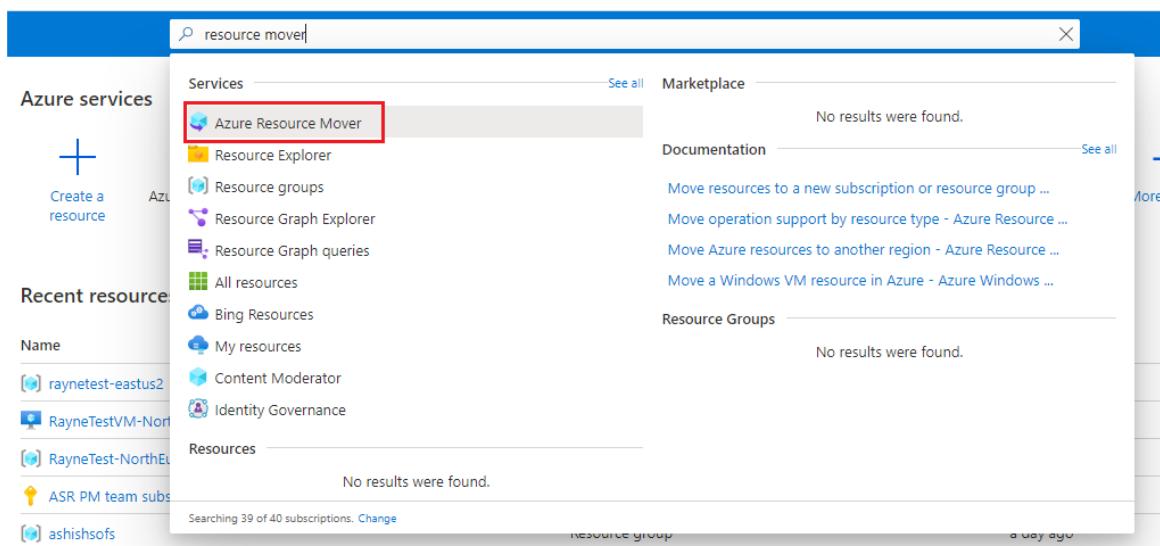
- After checking that VMs meet requirements, make sure that VMs you want to move are turned on. All VMs disks that you want to be available in the destination region must be attached and initialized in the VM.
- Make sure VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do this:
  - On Windows VMs, install the latest Windows updates.
  - On Linux VMs, follow distributor guidance so that machines have the latest certificates and CRL.
- Allow outbound connectivity from VMs:
  - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#)
  - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

## Select resources

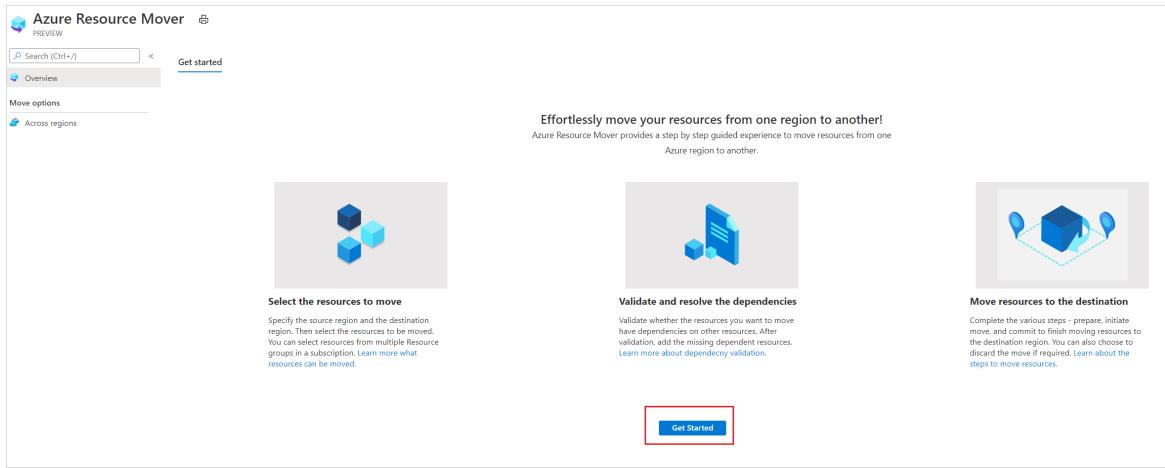
Select resources you want to move.

- All supported resource types in resource groups within the selected source region are displayed.
- Resources that have already been added for moving across regions aren't shown.
- You move resources to a target region in the same subscription as the source region. If you want to change the subscription, you can do that after the resources are moved.

- In the Azure portal, search for *resource mover*. Then, under **Services**, select **Azure Resource Mover**.



- In **Overview**, click **Get started**.



3. In **Move resources > Source + destination**, select the source subscription and region.

4. In **Destination**, select the region to which you want to move the VMs. Then click **Next**.

### Move resources

Move across regions | PREVIEW

**1 Source + destination**    **2 Resources to move**    **3 Review + Add**

Select the source subscription and the source region of the resources you want to move, and the destination region you want to move your resources to. [View support matrix](#)

**Source**

**Subscription \*** [?](#)

**Region \*** [?](#)

**Destination**

**Subscription** [?](#)

**Region \*** [?](#)

**Info** You can change the subscription after moving resources to the destination region. Use the 'Move across subscriptions' feature for this. [Learn more](#)

5. In **Resources to move**, click **Select resources**.

6. In **Select resources**, select the VM. You can only add **resources supported for move**. Then click **Done**.

## Select resources

**Tip** You can only select resources that are supported for moving across regions. Click [here](#) to learn more about the support matrix.

The screenshot shows a search bar with 'rayne' and filters for 'Resource group == (all)' and 'Type == (all)'. Below is a table with columns: Name, Type, Resource group, and Location. The table lists the following resources:

Name	Type	Resource group	Location
RayneTest-vnet	Virtual network	raynetest	East US
RayneTestVM-1	Virtual machine	raynetest	East US
RayneTestVM-1-ip	Public IP address	raynetest	East US
RayneTestVM-1-nsg	Network security group	raynetest	East US
raynetestvm-1286	Network interface	raynetest	East US

7. In **Resources to move**, click **Next**.

8. In **Review**, check the source and destination settings.

The screenshot shows the 'Move resources' review page. The top navigation bar includes 'Home > Azure Resource Mover > Move resources' and a 'Move across regions | PREVIEW' link. Below are tabs: 'Source + destination' (checked), 'Resources to move' (checked), and 'Review + Add' (selected). The 'Selection summary' section details the move configuration:

Source subscription	subscription-id
Source region	East US
Destination subscription	subscription-id
Destination region	East US 2
Number of resources to move	1

A note below states: 'You have now chosen the resources you want to move to the destination region.'

**The Azure Resource Mover will help you navigate through the following steps in the upcoming screens.**

- Validate dependencies:** Validate whether resources you want to move have dependencies on other resources in the source region. After validation, add the dependent resources to the move.
- Prepare:** The preparation process depends on the resource being moved, but might typically include exporting an ARM template, or initiating data replication. This doesn't have any impact on the availability of the resources in the source region.
- Initiate move:** Bring up the resources in the destination region. The process depends on the resource you're moving, but might typically include recreating the resource in the target region, or bringing up a replica copy.

**Finish up the move**

- Discard (optional):** After verifying the resources in the destination region, you can optionally roll back the move.
- Commit:** If everything's running as expected in the destination region, you can commit the move. This step may result in downtime depending on the source resource type.
- Delete source:** Finally, after everything's up and running in the new region, delete the resources in the source region. This is important to avoid double billing, and to ensure a stable infrastructure.

[Learn more](#)

9. Click **Proceed**, to begin adding the resources.

10. After the add process finishes successfully, click **Adding resources for move** in the notification icon.

11. After clicking the notification, review the resources on the **Across regions** page.

#### NOTE

- Added resources are in a *Prepare pending* state.
- The resource group for the VMs is added automatically.
- If you want to remove a resource from a move collection, the method for doing that depends on where you are in the move process. [Learn more](#).

## Resolve dependencies

- If resources show a *Validate dependencies* message in the **Issues** column, click the **Validate dependencies** button. The validation process begins.
- If dependencies are found, click **Add dependencies**.
- In **Add dependencies**, leave the default **Show all dependencies** option.
  - Show all dependencies iterates through all of the direct and indirect dependencies for a resource. For example, for a VM it shows the NIC, virtual network, network security groups (NSGs) etc.
  - Show first level dependencies only shows only direct dependencies. For example, for a VM it shows the NIC, but not the virtual network.
- Select the dependent resources you want to add > **Add dependencies**. Monitor progress in the notifications.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 2

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1286	Network interface	raynetest
<input checked="" type="checkbox"/> raynetest	Resource group	-

- Validate dependencies again.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 3

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest
<input checked="" type="checkbox"/> raynetest-vnet	Virtual network	raynetest
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest

## Move the source resource group

Before you can prepare and move VMs, the VM resource group must be present in the target region.

## Prepare to move the source resource group

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

Prepare as follows:

1. In **Across regions**, select the source resource group > **Prepare**.
2. In **Prepare resources**, click **Prepare**.

Name	Type
raynetest	Resource group

### NOTE

After preparing the resource group, it's in the *Initiate move pending* state.

## Move the source resource group

Initiate the move as follows:

1. In **Across regions**, select the resource group > **Initiate Move**
2. In **Move Resources**, click **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

<input type="checkbox"/> RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest	Resource group	-	(new) raynetest-eastus2	<span>'Commit move' pending</span>	-
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	<span>Prepare' pending</span>	-

To commit and finish the move process:

1. In **Across regions**, select the resource group > **Commit move**.
2. In **Move Resources**, click **Commit**.

### NOTE

After committing the move, the source resource group is in a *Delete source pending* state.

## Prepare resources to move

Now that the source resource group is moved, you can prepare to move other resources that are in the *Prepare pending* state.

1. In **Across regions**, verify that resources are now in a *Prepare pending* state, with no issues. If they're not, validate again and resolve any outstanding issues.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Prepare' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Prepare' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Prepare' pending

- If you want to edit target settings before beginning the move, select the link in the **Destination configuration** column for the resource, and edit the settings. If you edit the target VM settings, the target VM size shouldn't be smaller than the source VM size.

Now that the source resource group is moved, you can prepare to move the other resources.

- Select the resources you want to prepare.

**Prepare resources**

Azure Resource Mover - Across regions | PREVIEW

Want to assign an existing resource in the destination region? [Edit the target properties before you prepare.](#)

Learn more about how different resources are prepared.

Name	Type	Resource group
RayneTestVM-1	Virtual machine	RayneTest
raynetestvm-1286	Network interface	raynetest
raynetestvm-1-nsg	Network security group	raynetest
raynetest-vnet	Virtual network	raynetest
raynetestvm-1-ip	Public IP address	raynetest

- Select **Prepare**.

#### NOTE

- During the prepare process, the Azure Site Recovery Mobility agent is installed on VMs, to replicate them.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'initiate move' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'initiate move' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'initiate move' pending

## Initiate the move

With resources prepared, you can now initiate the move.

- In **Across regions**, select resources with state *Initiate move pending*. Then click **Initiate move**.
- In **Move resources**, click **Initiate move**.

Name	Type
RayneTestVM-1	Virtual machine
raynetestvm-1286	Network interface
raynetestvm-1-nsg	Network security group
raynetest-vnet	Virtual network
raynetestvm-1-ip	Public IP address

Initiate move  
 Cancel

3. Track move progress in the notifications bar.

**NOTE**

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- After moving resources, they're in an *Commit move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Discard or commit?

After the initial move, you can decide whether you want to commit the move, or to discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

## Discard the move

You can discard the move as follows:

1. In **Across regions**, select resources with state *Commit move pending*, and click **Discard move**.
2. In **Discard move**, click **Discard**.
3. Track move progress in the notifications bar.

### NOTE

After discarding resources, VMs are in an *Initiate move pending* state.

## Commit the move

If you want to complete the move process, commit the move.

1. In **Across regions**, select resources with state *Commit move pending*, and click **Commit move**.
2. In **Commit resources**, click **Commit**.

## Commit resources

PREVIEW

Name	Type
 RayneTestVM-1	Virtual machine
 raynetestvm-1286	Network interface
 raynetestvm-1-nsg	Network security group
 raynetest-vnet	Virtual network
 raynetestvm-1-ip	Public IP address

**Commit**

**Cancel**

3. Track the commit progress in the notifications bar.

#### NOTE

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Configure settings after the move

- The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.
- Modify Azure role-based access control (Azure RBAC) rules after the move.

## Delete source resources after commit

After the move, you can optionally delete resources in the source region.

#### NOTE

A few resources, for example key vaults and SQL Server servers, can't be deleted from the portal, and must be deleted from the resource property page.

1. In **Across Regions**, click the name of the source resource that you want to delete.
2. Select **Delete source**.

## Delete additional resources created for move

After the move, you can manually delete the move collection, and Site Recovery resources that were created.

- The move collection is hidden by default. To see it you need to turn on hidden resources.
- The cache storage has a lock that must be deleted, before it can be deleted.

Delete as follows:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`.
2. Check that all the VM and other source resources in the source region have been moved or deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
  - The move collection name is `movecollection-<sourcerregion>-<target-region>`.
  - The cache storage account name is `resmovecache<guid>`
  - The vault name is `ResourceMove-<sourcerregion>-<target-region>-GUID`.

## Next steps

In this tutorial, you:

- Moved Azure VMs to another Azure region.

- Moved resources associated with VMs to another region.

Now, trying moving Azure SQL databases and elastic pools to another region.

[Move Azure SQL resources](#)

# Tutorial: Move Azure VMs across regions

2/18/2021 • 8 minutes to read • [Edit Online](#)

In this article, learn how to move Azure VMs, and related network/storage resources, to a different Azure region, using [Azure Resource Mover](#)..

In this tutorial, you learn how to:

- Check prerequisites and requirements.
- Select the resources you want to move.
- Resolve resource dependencies.
- Prepare and move the source resource group.
- Prepare and move the other resources.
- Decide whether you want to discard or commit the move.
- Optionally remove resources in the source region after the move.

## NOTE

Tutorials show the quickest path for trying out a scenario, and use default options.

If you don't have an Azure subscription, create a [free account](#) before you begin. Then sign in to the [Azure portal](#).

## Prerequisites

REQUIREMENT	DESCRIPTION
<b>Subscription permissions</b>	<p>Check you have <i>Owner</i> access on the subscription containing the resources that you want to move</p> <p><b>Why do I need Owner access?</b> The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a <a href="#">system-assigned managed identity</a> (formerly known as Managed Service Identity (MSI)) that's trusted by the subscription. To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs <i>Owner</i> permissions on the subscription. <a href="#">Learn more</a> about Azure roles.</p>
<b>VM support</b>	<p>Check that the VMs you want to move are supported.</p> <ul style="list-style-type: none"><li>- <a href="#">Verify</a> supported Windows VMs.</li><li>- <a href="#">Verify</a> supported Linux VMs and kernel versions.</li><li>- Check supported <a href="#">compute</a>, <a href="#">storage</a>, and <a href="#">networking</a> settings.</li></ul>
<b>Destination subscription</b>	The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have quota, <a href="#">request additional limits</a> .

Requirement	Description
Destination region charges	Verify pricing and charges associated with the target region to which you're moving VMs. Use the <a href="#">pricing calculator</a> to help you.

## Prepare VMs

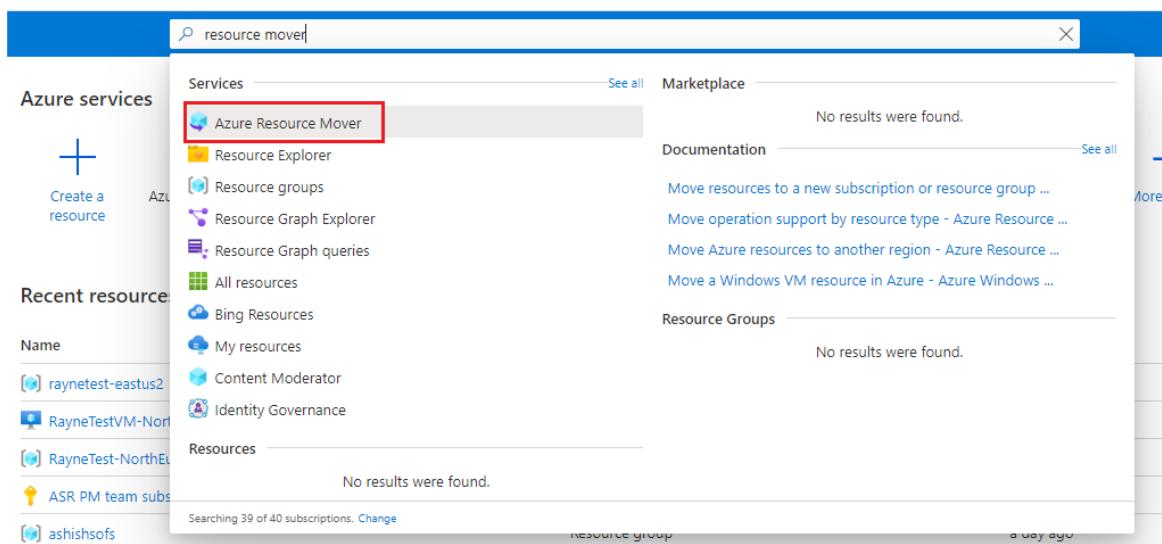
1. After checking that VMs meet requirements, make sure that VMs you want to move are turned on. All VMs disks that you want to be available in the destination region must be attached and initialized in the VM.
2. Make sure VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do this:
  - On Windows VMs, install the latest Windows updates.
  - On Linux VMs, follow distributor guidance so that machines have the latest certificates and CRL.
3. Allow outbound connectivity from VMs:
  - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#)
  - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

## Select resources

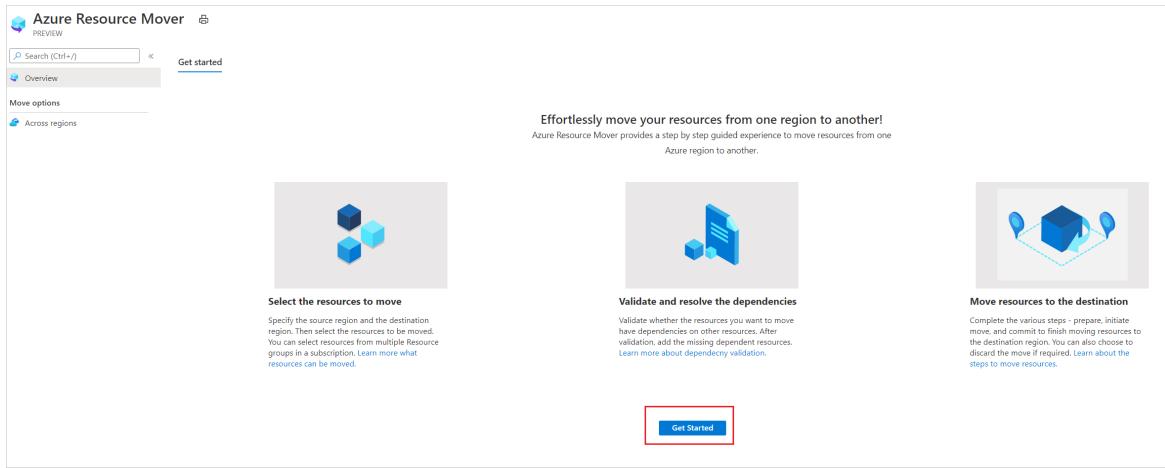
Select resources you want to move.

- All supported resource types in resource groups within the selected source region are displayed.
- Resources that have already been added for moving across regions aren't shown.
- You move resources to a target region in the same subscription as the source region. If you want to change the subscription, you can do that after the resources are moved.

1. In the Azure portal, search for *resource mover*. Then, under **Services**, select **Azure Resource Mover**.



2. In **Overview**, click **Get started**.



3. In **Move resources > Source + destination**, select the source subscription and region.

4. In **Destination**, select the region to which you want to move the VMs. Then click **Next**.

### Move resources

Move across regions | PREVIEW

**1 Source + destination**    **2 Resources to move**    **3 Review + Add**

Select the source subscription and the source region of the resources you want to move, and the destination region you want to move your resources to. [View support matrix](#)

**Source**

Subscription \* [?](#)

Region \* [?](#)

**Destination**

Subscription [?](#)

**Info** You can change the subscription after moving resources to the destination region. Use the 'Move across subscriptions' feature for this. [Learn more](#)

Region \* [?](#)

5. In **Resources to move**, click **Select resources**.

6. In **Select resources**, select the VM. You can only add **resources supported for move**. Then click **Done**.

## Select resources

**Tip** You can only select resources that are supported for moving across regions. Click [here](#) to learn more about the support matrix.

The screenshot shows a search bar with 'rayne' and filters for 'Resource group == (all)' and 'Type == (all)'. Below is a table with columns: Name, Type, Resource group, and Location. The table lists several resources under the 'rayne' search result:

Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓
RayneTest-vnet	Virtual network	raynetest	East US
RayneTestVM-1	Virtual machine	raynetest	East US
RayneTestVM-1-ip	Public IP address	raynetest	East US
RayneTestVM-1-nsg	Network security group	raynetest	East US
raynetestvm-1286	Network interface	raynetest	East US

7. In **Resources to move**, click **Next**.

8. In **Review**, check the source and destination settings.

The screenshot shows the 'Move resources' review page. The top navigation bar includes 'Home > Azure Resource Mover > Move resources' and a 'Move across regions | PREVIEW' link. Below are tabs: 'Source + destination' (checked), 'Resources to move' (checked), and 'Review + Add' (selected). The 'Selection summary' section details the move configuration:

Source subscription	subscription-id
Source region	East US
Destination subscription	subscription-id
Destination region	East US 2
Number of resources to move	1

A note below states: 'You have now chosen the resources you want to move to the destination region.'

**The Azure Resource Mover will help you navigate through the following steps in the upcoming screens.**

- Validate dependencies:** Validate whether resources you want to move have dependencies on other resources in the source region. After validation, add the dependent resources to the move.
- Prepare:** The preparation process depends on the resource being moved, but might typically include exporting an ARM template, or initiating data replication. This doesn't have any impact on the availability of the resources in the source region.
- Initiate move:** Bring up the resources in the destination region. The process depends on the resource you're moving, but might typically include recreating the resource in the target region, or bringing up a replica copy.

**Finish up the move**

- Discard (optional):** After verifying the resources in the destination region, you can optionally roll back the move.
- Commit:** If everything's running as expected in the destination region, you can commit the move. This step may result in downtime depending on the source resource type.
- Delete source:** Finally, after everything's up and running in the new region, delete the resources in the source region. This is important to avoid double billing, and to ensure a stable infrastructure.

[Learn more](#)

9. Click **Proceed**, to begin adding the resources.

10. After the add process finishes successfully, click **Adding resources for move** in the notification icon.

11. After clicking the notification, review the resources on the **Across regions** page.

#### NOTE

- Added resources are in a *Prepare pending* state.
- The resource group for the VMs is added automatically.
- If you want to remove a resource from a move collection, the method for doing that depends on where you are in the move process. [Learn more](#).

## Resolve dependencies

- If resources show a *Validate dependencies* message in the **Issues** column, click the **Validate dependencies** button. The validation process begins.
- If dependencies are found, click **Add dependencies**.
- In **Add dependencies**, leave the default **Show all dependencies** option.
  - Show all dependencies iterates through all of the direct and indirect dependencies for a resource. For example, for a VM it shows the NIC, virtual network, network security groups (NSGs) etc.
  - Show first level dependencies only shows only direct dependencies. For example, for a VM it shows the NIC, but not the virtual network.
- Select the dependent resources you want to add > **Add dependencies**. Monitor progress in the notifications.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 2

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1286	Network interface	raynetest
<input checked="" type="checkbox"/> raynetest	Resource group	-

- Validate dependencies again.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 3

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest
<input checked="" type="checkbox"/> raynetest-vnet	Virtual network	raynetest
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest

## Move the source resource group

Before you can prepare and move VMs, the VM resource group must be present in the target region.

## Prepare to move the source resource group

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

Prepare as follows:

1. In **Across regions**, select the source resource group > **Prepare**.
2. In **Prepare resources**, click **Prepare**.

Name	Type
raynetest	Resource group

### NOTE

After preparing the resource group, it's in the *Initiate move pending* state.

## Move the source resource group

Initiate the move as follows:

1. In **Across regions**, select the resource group > **Initiate Move**
2. In **Move Resources**, click **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

<input type="checkbox"/> RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest	Resource group	-	(new) raynetest-eastus2	<span>'Commit move' pending</span>	-
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	<span>Prepare' pending</span>	-

To commit and finish the move process:

1. In **Across regions**, select the resource group > **Commit move**.
2. In **Move Resources**, click **Commit**.

### NOTE

After committing the move, the source resource group is in a *Delete source pending* state.

## Prepare resources to move

Now that the source resource group is moved, you can prepare to move other resources that are in the *Prepare pending* state.

1. In **Across regions**, verify that resources are now in a *Prepare pending* state, with no issues. If they're not, validate again and resolve any outstanding issues.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Prepare' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Prepare' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Prepare' pending

- If you want to edit target settings before beginning the move, select the link in the **Destination configuration** column for the resource, and edit the settings. If you edit the target VM settings, the target VM size shouldn't be smaller than the source VM size.

Now that the source resource group is moved, you can prepare to move the other resources.

- Select the resources you want to prepare.

**Prepare resources**

Azure Resource Mover - Across regions | PREVIEW

Want to assign an existing resource in the destination region? [Edit the target properties before you prepare.](#)

Learn more about how different resources are prepared.

Name	Type	Resource group
RayneTestVM-1	Virtual machine	RayneTest
raynetestvm-1286	Network interface	raynetest
raynetestvm-1-nsg	Network security group	raynetest
raynetest-vnet	Virtual network	raynetest
raynetestvm-1-ip	Public IP address	raynetest

- Select **Prepare**.

#### NOTE

- During the prepare process, the Azure Site Recovery Mobility agent is installed on VMs, to replicate them.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'initiate move' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'initiate move' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'initiate move' pending

## Initiate the move

With resources prepared, you can now initiate the move.

- In **Across regions**, select resources with state *Initiate move pending*. Then click **Initiate move**.
- In **Move resources**, click **Initiate move**.

Name	Type
RayneTestVM-1	Virtual machine
raynetestvm-1286	Network interface
raynetestvm-1-nsg	Network security group
raynetest-vnet	Virtual network
raynetestvm-1-ip	Public IP address

Initiate move  
 Cancel

3. Track move progress in the notifications bar.

**NOTE**

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- After moving resources, they're in an *Commit move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Discard or commit?

After the initial move, you can decide whether you want to commit the move, or to discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

## Discard the move

You can discard the move as follows:

1. In **Across regions**, select resources with state *Commit move pending*, and click **Discard move**.
2. In **Discard move**, click **Discard**.
3. Track move progress in the notifications bar.

### NOTE

After discarding resources, VMs are in an *Initiate move pending* state.

## Commit the move

If you want to complete the move process, commit the move.

1. In **Across regions**, select resources with state *Commit move pending*, and click **Commit move**.
2. In **Commit resources**, click **Commit**.

## Commit resources

PREVIEW

Name	Type
 RayneTestVM-1	Virtual machine
 raynetestvm-1286	Network interface
 raynetestvm-1-nsg	Network security group
 raynetest-vnet	Virtual network
 raynetestvm-1-ip	Public IP address

**Commit**

**Cancel**

3. Track the commit progress in the notifications bar.

#### NOTE

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Configure settings after the move

- The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.
- Modify Azure role-based access control (Azure RBAC) rules after the move.

## Delete source resources after commit

After the move, you can optionally delete resources in the source region.

#### NOTE

A few resources, for example key vaults and SQL Server servers, can't be deleted from the portal, and must be deleted from the resource property page.

1. In **Across Regions**, click the name of the source resource that you want to delete.
2. Select **Delete source**.

## Delete additional resources created for move

After the move, you can manually delete the move collection, and Site Recovery resources that were created.

- The move collection is hidden by default. To see it you need to turn on hidden resources.
- The cache storage has a lock that must be deleted, before it can be deleted.

Delete as follows:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`.
2. Check that all the VM and other source resources in the source region have been moved or deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
  - The move collection name is `movecollection-<sourcerregion>-<target-region>`.
  - The cache storage account name is `resmovecache<guid>`
  - The vault name is `ResourceMove-<sourcerregion>-<target-region>-GUID`.

## Next steps

In this tutorial, you:

- Moved Azure VMs to another Azure region.

- Moved resources associated with VMs to another region.

Now, trying moving Azure SQL databases and elastic pools to another region.

[Move Azure SQL resources](#)

# Tutorial: Move Azure VMs across regions

2/18/2021 • 8 minutes to read • [Edit Online](#)

In this article, learn how to move Azure VMs, and related network/storage resources, to a different Azure region, using [Azure Resource Mover](#)..

In this tutorial, you learn how to:

- Check prerequisites and requirements.
- Select the resources you want to move.
- Resolve resource dependencies.
- Prepare and move the source resource group.
- Prepare and move the other resources.
- Decide whether you want to discard or commit the move.
- Optionally remove resources in the source region after the move.

## NOTE

Tutorials show the quickest path for trying out a scenario, and use default options.

If you don't have an Azure subscription, create a [free account](#) before you begin. Then sign in to the [Azure portal](#).

## Prerequisites

REQUIREMENT	DESCRIPTION
<b>Subscription permissions</b>	<p>Check you have <i>Owner</i> access on the subscription containing the resources that you want to move</p> <p><b>Why do I need Owner access?</b> The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a <a href="#">system-assigned managed identity</a> (formerly known as Managed Service Identity (MSI)) that's trusted by the subscription. To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs <i>Owner</i> permissions on the subscription. <a href="#">Learn more</a> about Azure roles.</p>
<b>VM support</b>	<p>Check that the VMs you want to move are supported.</p> <ul style="list-style-type: none"><li>- <a href="#">Verify</a> supported Windows VMs.</li><li>- <a href="#">Verify</a> supported Linux VMs and kernel versions.</li><li>- Check supported <a href="#">compute</a>, <a href="#">storage</a>, and <a href="#">networking</a> settings.</li></ul>
<b>Destination subscription</b>	The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have quota, <a href="#">request additional limits</a> .

Requirement	Description
Destination region charges	Verify pricing and charges associated with the target region to which you're moving VMs. Use the <a href="#">pricing calculator</a> to help you.

## Prepare VMs

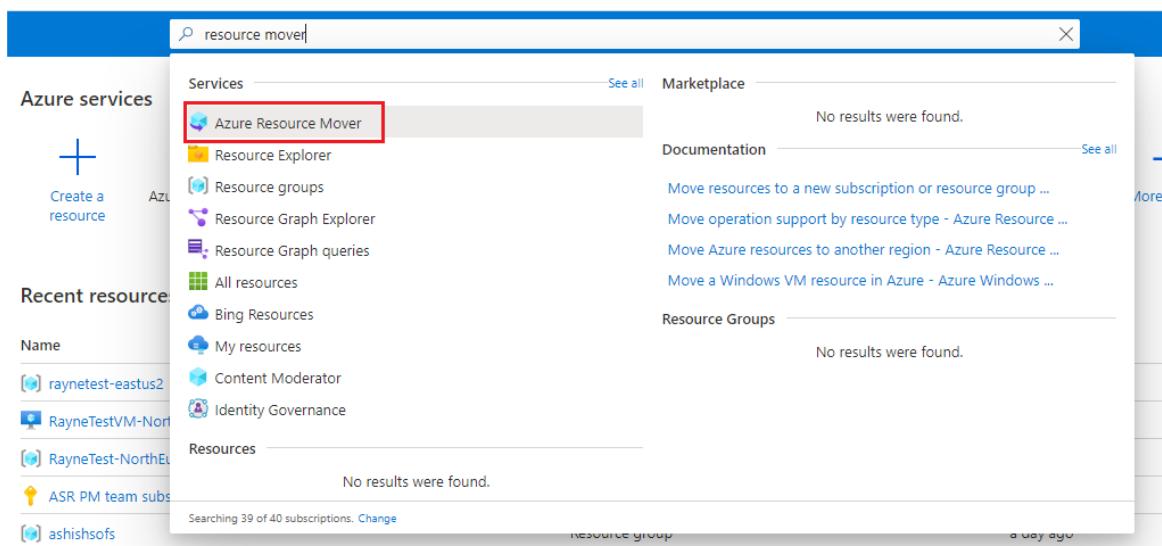
- After checking that VMs meet requirements, make sure that VMs you want to move are turned on. All VMs disks that you want to be available in the destination region must be attached and initialized in the VM.
- Make sure VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do this:
  - On Windows VMs, install the latest Windows updates.
  - On Linux VMs, follow distributor guidance so that machines have the latest certificates and CRL.
- Allow outbound connectivity from VMs:
  - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#)
  - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

## Select resources

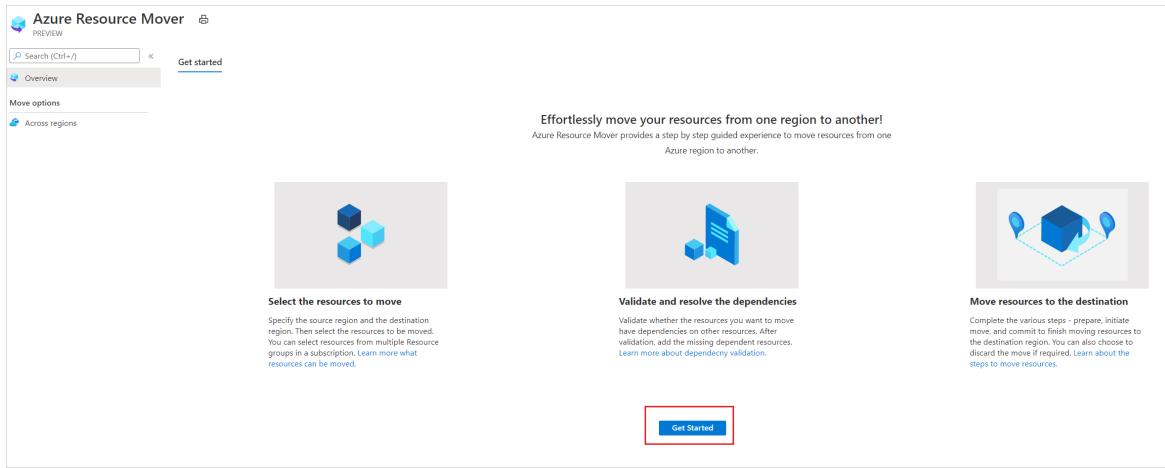
Select resources you want to move.

- All supported resource types in resource groups within the selected source region are displayed.
- Resources that have already been added for moving across regions aren't shown.
- You move resources to a target region in the same subscription as the source region. If you want to change the subscription, you can do that after the resources are moved.

- In the Azure portal, search for *resource mover*. Then, under **Services**, select **Azure Resource Mover**.



- In **Overview**, click **Get started**.



3. In **Move resources > Source + destination**, select the source subscription and region.

4. In **Destination**, select the region to which you want to move the VMs. Then click **Next**.

### Move resources

Move across regions | PREVIEW

**1 Source + destination**    **2 Resources to move**    **3 Review + Add**

Select the source subscription and the source region of the resources you want to move, and the destination region you want to move your resources to. [View support matrix](#)

**Source**

Subscription \* [?](#)

Region \* [?](#)

**Destination**

Subscription [?](#)

**Info** You can change the subscription after moving resources to the destination region. Use the 'Move across subscriptions' feature for this. [Learn more](#)

Region \* [?](#)

5. In **Resources to move**, click **Select resources**.

6. In **Select resources**, select the VM. You can only add **resources supported for move**. Then click **Done**.

## Select resources

**Tip** You can only select resources that are supported for moving across regions. Click [here](#) to learn more about the support matrix.

The screenshot shows a search bar with 'rayne' and filters for 'Resource group == (all)' and 'Type == (all)'. Below is a table with columns: Name, Type, Resource group, and Location. The table lists the following resources:

Name	Type	Resource group	Location
RayneTest-vnet	Virtual network	raynetest	East US
RayneTestVM-1	Virtual machine	raynetest	East US
RayneTestVM-1-ip	Public IP address	raynetest	East US
RayneTestVM-1-nsg	Network security group	raynetest	East US
raynetestvm-1286	Network interface	raynetest	East US

7. In **Resources to move**, click **Next**.

8. In **Review**, check the source and destination settings.

The screenshot shows the 'Move resources' review page. The top navigation bar includes 'Home > Azure Resource Mover > Move resources' and a 'Move across regions | PREVIEW' link. Below are tabs: 'Source + destination' (checked), 'Resources to move' (checked), and 'Review + Add' (selected). The 'Selection summary' section details the move configuration:

Source subscription	subscription-id
Source region	East US
Destination subscription	subscription-id
Destination region	East US 2
Number of resources to move	1

A note below states: 'You have now chosen the resources you want to move to the destination region.'

**The Azure Resource Mover will help you navigate through the following steps in the upcoming screens.**

- Validate dependencies:** Validate whether resources you want to move have dependencies on other resources in the source region. After validation, add the dependent resources to the move.
- Prepare:** The preparation process depends on the resource being moved, but might typically include exporting an ARM template, or initiating data replication. This doesn't have any impact on the availability of the resources in the source region.
- Initiate move:** Bring up the resources in the destination region. The process depends on the resource you're moving, but might typically include recreating the resource in the target region, or bringing up a replica copy.

**Finish up the move**

- Discard (optional):** After verifying the resources in the destination region, you can optionally roll back the move.
- Commit:** If everything's running as expected in the destination region, you can commit the move. This step may result in downtime depending on the source resource type.
- Delete source:** Finally, after everything's up and running in the new region, delete the resources in the source region. This is important to avoid double billing, and to ensure a stable infrastructure.

[Learn more](#)

9. Click **Proceed**, to begin adding the resources.

10. After the add process finishes successfully, click **Adding resources for move** in the notification icon.

11. After clicking the notification, review the resources on the **Across regions** page.

#### NOTE

- Added resources are in a *Prepare pending* state.
- The resource group for the VMs is added automatically.
- If you want to remove a resource from a move collection, the method for doing that depends on where you are in the move process. [Learn more](#).

## Resolve dependencies

- If resources show a *Validate dependencies* message in the **Issues** column, click the **Validate dependencies** button. The validation process begins.
- If dependencies are found, click **Add dependencies**.
- In **Add dependencies**, leave the default **Show all dependencies** option.
  - Show all dependencies iterates through all of the direct and indirect dependencies for a resource. For example, for a VM it shows the NIC, virtual network, network security groups (NSGs) etc.
  - Show first level dependencies only shows only direct dependencies. For example, for a VM it shows the NIC, but not the virtual network.
- Select the dependent resources you want to add > **Add dependencies**. Monitor progress in the notifications.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 2

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1286	Network interface	raynetest
<input checked="" type="checkbox"/> raynetest	Resource group	-

- Validate dependencies again.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Info** Some resources in your list may require other resources to be moved along with them. Review and add the dependencies

**Note** Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties. [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 3

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest
<input checked="" type="checkbox"/> raynetest-vnet	Virtual network	raynetest
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest

## Move the source resource group

Before you can prepare and move VMs, the VM resource group must be present in the target region.

## Prepare to move the source resource group

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

Prepare as follows:

1. In **Across regions**, select the source resource group > **Prepare**.
2. In **Prepare resources**, click **Prepare**.

Name	Type
raynetest	Resource group

### NOTE

After preparing the resource group, it's in the *Initiate move pending* state.

## Move the source resource group

Initiate the move as follows:

1. In **Across regions**, select the resource group > **Initiate Move**
2. In **Move Resources**, click **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

<input type="checkbox"/> RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest	Resource group	-	(new) raynetest-eastus2	<span>'Commit move' pending</span>	-
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	<span>Prepare' pending</span>	-
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	<span>Prepare' pending</span>	-

To commit and finish the move process:

1. In **Across regions**, select the resource group > **Commit move**.
2. In **Move Resources**, click **Commit**.

### NOTE

After committing the move, the source resource group is in a *Delete source pending* state.

## Prepare resources to move

Now that the source resource group is moved, you can prepare to move other resources that are in the *Prepare pending* state.

1. In **Across regions**, verify that resources are now in a *Prepare pending* state, with no issues. If they're not, validate again and resolve any outstanding issues.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Prepare' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Prepare' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Prepare' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Prepare' pending

- If you want to edit target settings before beginning the move, select the link in the **Destination configuration** column for the resource, and edit the settings. If you edit the target VM settings, the target VM size shouldn't be smaller than the source VM size.

Now that the source resource group is moved, you can prepare to move the other resources.

- Select the resources you want to prepare.

**Prepare resources**

Azure Resource Mover - Across regions | PREVIEW

Want to assign an existing resource in the destination region? [Edit the target properties before you prepare.](#)

Learn more about how different resources are prepared.

Name	Type	Resource group
RayneTestVM-1	Virtual machine	RayneTest
raynetestvm-1286	Network interface	raynetest
raynetestvm-1-nsg	Network security group	raynetest
raynetest-vnet	Virtual network	raynetest
raynetestvm-1-ip	Public IP address	raynetest

- Select **Prepare**.

#### NOTE

- During the prepare process, the Azure Site Recovery Mobility agent is installed on VMs, to replicate them.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'initiate move' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'initiate move' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'initiate move' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'initiate move' pending

## Initiate the move

With resources prepared, you can now initiate the move.

- In **Across regions**, select resources with state *Initiate move pending*. Then click **Initiate move**.
- In **Move resources**, click **Initiate move**.

Name	Type
RayneTestVM-1	Virtual machine
raynetestvm-1286	Network interface
raynetestvm-1-nsg	Network security group
raynetest-vnet	Virtual network
raynetestvm-1-ip	Public IP address

Initiate move  
 Cancel

3. Track move progress in the notifications bar.

**NOTE**

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- After moving resources, they're in an *Commit move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Discard or commit?

After the initial move, you can decide whether you want to commit the move, or to discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

## Discard the move

You can discard the move as follows:

1. In **Across regions**, select resources with state *Commit move pending*, and click **Discard move**.
2. In **Discard move**, click **Discard**.
3. Track move progress in the notifications bar.

### NOTE

After discarding resources, VMs are in an *Initiate move pending* state.

## Commit the move

If you want to complete the move process, commit the move.

1. In **Across regions**, select resources with state *Commit move pending*, and click **Commit move**.
2. In **Commit resources**, click **Commit**.

## Commit resources

PREVIEW

Name	Type
 RayneTestVM-1	Virtual machine
 raynetestvm-1286	Network interface
 raynetestvm-1-nsg	Network security group
 raynetest-vnet	Virtual network
 raynetestvm-1-ip	Public IP address

**Commit**

**Cancel**

3. Track the commit progress in the notifications bar.

#### NOTE

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Configure settings after the move

- The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.
- Modify Azure role-based access control (Azure RBAC) rules after the move.

## Delete source resources after commit

After the move, you can optionally delete resources in the source region.

#### NOTE

A few resources, for example key vaults and SQL Server servers, can't be deleted from the portal, and must be deleted from the resource property page.

1. In **Across Regions**, click the name of the source resource that you want to delete.
2. Select **Delete source**.

## Delete additional resources created for move

After the move, you can manually delete the move collection, and Site Recovery resources that were created.

- The move collection is hidden by default. To see it you need to turn on hidden resources.
- The cache storage has a lock that must be deleted, before it can be deleted.

Delete as follows:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`.
2. Check that all the VM and other source resources in the source region have been moved or deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
  - The move collection name is `movecollection-<sourcerregion>-<target-region>`.
  - The cache storage account name is `resmovecache<guid>`
  - The vault name is `ResourceMove-<sourcerregion>-<target-region>-GUID`.

## Next steps

In this tutorial, you:

- Moved Azure VMs to another Azure region.

- Moved resources associated with VMs to another region.

Now, trying moving Azure SQL databases and elastic pools to another region.

[Move Azure SQL resources](#)

# Tutorial: Move Azure VMs across regions

2/18/2021 • 8 minutes to read • [Edit Online](#)

In this article, learn how to move Azure VMs, and related network/storage resources, to a different Azure region, using [Azure Resource Mover](#).

In this tutorial, you learn how to:

- Check prerequisites and requirements.
- Select the resources you want to move.
- Resolve resource dependencies.
- Prepare and move the source resource group.
- Prepare and move the other resources.
- Decide whether you want to discard or commit the move.
- Optionally remove resources in the source region after the move.

## NOTE

Tutorials show the quickest path for trying out a scenario, and use default options.

If you don't have an Azure subscription, create a [free account](#) before you begin. Then sign in to the [Azure portal](#).

## Prerequisites

REQUIREMENT	DESCRIPTION
Subscription permissions	<p>Check you have <i>Owner</i> access on the subscription containing the resources that you want to move</p> <p><b>Why do I need Owner access?</b> The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a <a href="#">system-assigned managed identity</a> (formerly known as Managed Service Identity (MSI)) that's trusted by the subscription. To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs <i>Owner</i> permissions on the subscription. <a href="#">Learn more</a> about Azure roles.</p>
VM support	<p>Check that the VMs you want to move are supported.</p> <ul style="list-style-type: none"><li>- <a href="#">Verify</a> supported Windows VMs.</li><li>- <a href="#">Verify</a> supported Linux VMs and kernel versions.</li><li>- Check supported <a href="#">compute</a>, <a href="#">storage</a>, and <a href="#">networking</a> settings.</li></ul>
Destination subscription	The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have quota, <a href="#">request additional limits</a> .

Requirement	Description
Destination region charges	Verify pricing and charges associated with the target region to which you're moving VMs. Use the <a href="#">pricing calculator</a> to help you.

## Prepare VMs

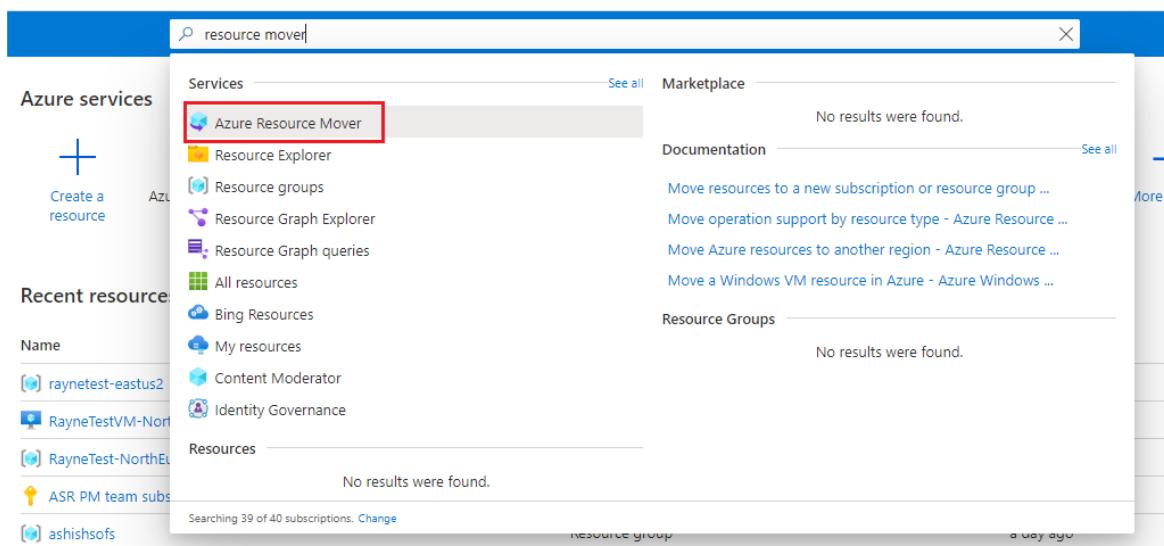
- After checking that VMs meet requirements, make sure that VMs you want to move are turned on. All VM disks that you want to be available in the destination region must be attached and initialized in the VM.
- Make sure VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do this:
  - On Windows VMs, install the latest Windows updates.
  - On Linux VMs, follow distributor guidance so that machines have the latest certificates and CRL.
- Allow outbound connectivity from VMs:
  - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#)
  - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

## Select resources

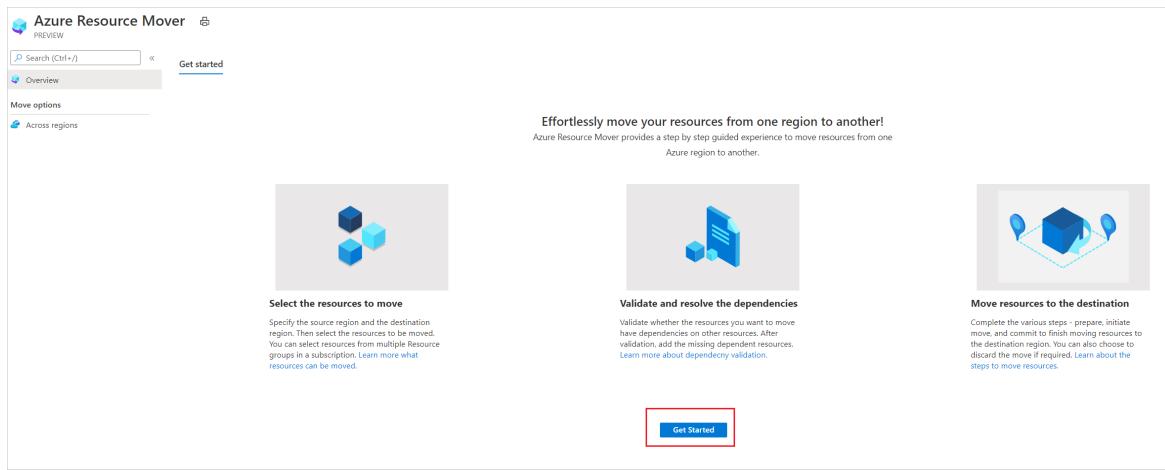
Select resources you want to move.

- All supported resource types in resource groups within the selected source region are displayed.
- Resources that have already been added for moving across regions aren't shown.
- You move resources to a target region in the same subscription as the source region. If you want to change the subscription, you can do that after the resources are moved.

- In the Azure portal, search for *resource mover*. Then, under **Services**, select **Azure Resource Mover**.



- In **Overview**, click **Get started**.



3. In **Move resources > Source + destination**, select the source subscription and region.

4. In **Destination**, select the region to which you want to move the VMs. Then click **Next**.

## Move resources

Move across regions | PREVIEW

**1 Source + destination**   **2 Resources to move**   **3 Review + Add**

Select the source subscription and the source region of the resources you want to move, and the destination region you want to move your resources to. [View support matrix](#)

**Source**

Subscription \*  ▼

Region \*  ▼

**Destination**

Subscription

ⓘ You can change the subscription after moving resources to the destination region. Use the 'Move across subscriptions' feature for this. [Learn more](#)

Region \*  ▼

5. In **Resources to move**, click **Select resources**.

6. In **Select resources**, select the VM. You can only add **resources supported for move**. Then click **Done**.

## Select resources

ⓘ You can only select resources that are supported for moving across regions. Click [here](#) to learn more about the support matrix.

rayne Resource group == (all) × Type == (all) × + Add filter

Show hidden types  ⓘ

Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓
<input type="checkbox"/> RayneTest-vnet	Virtual network	raynetest	East US
<input checked="" type="checkbox"/> RayneTestVM-1	Virtual machine	raynetest	East US
<input type="checkbox"/> RayneTestVM-1-ip	Public IP address	raynetest	East US
<input type="checkbox"/> RayneTestVM-1-nsg	Network security group	raynetest	East US
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	East US

7. In **Resources to move**, click **Next**.

8. In **Review**, check the source and destination settings.

Home > Azure Resource Mover >

## Move resources ⓘ

Move across regions | PREVIEW

✓ Source + destination ✓ Resources to move 3 Review + Add

**Selection summary**

Source subscription	subscription-id
Source region	East US
Destination subscription	subscription-id
Destination region	East US 2
Number of resources to move	1

ⓘ You have now chosen the resources you want to move to the destination region.

**The Azure Resource Mover will help you navigate through the following steps in the upcoming screens.**

- Validate dependencies:** Validate whether resources you want to move have dependencies on other resources in the source region. After validation, add the dependent resources to the move.
- Prepare:** The preparation process depends on the resource being moved, but might typically include exporting an ARM template, or initiating data replication. This doesn't have any impact on the availability of the resources in the source region.
- Initiate move:** Bring up the resources in the destination region. The process depends on the resource you're moving, but might typically include recreating the resource in the target region, or bringing up a replica copy.

**Finish up the move**

- Discard (optional):** After verifying the resources in the destination region, you can optionally roll back the move.
- Commit:** If everything's running as expected in the destination region, you can commit the move. This step may result in downtime depending on the source resource type.
- Delete source:** Finally, after everything's up and running in the new region, delete the resources in the source region. This is important to avoid double billing, and to ensure a stable infrastructure.

[Learn more](#)

9. Click **Proceed**, to begin adding the resources.

10. After the add process finishes successfully, click **Adding resources for move** in the notification icon.

11. After clicking the notification, review the resources on the **Across regions** page.

#### NOTE

- Added resources are in a *Prepare pending* state.
- The resource group for the VMs is added automatically.
- If you want to remove a resource from a move collection, the method for doing that depends on where you are in the move process. [Learn more](#).

## Resolve dependencies

- If resources show a *Validate dependencies* message in the **Issues** column, click the **Validate dependencies** button. The validation process begins.
- If dependencies are found, click **Add dependencies**.
- In **Add dependencies**, leave the default **Show all dependencies** option.
  - Show all dependencies iterates through all of the direct and indirect dependencies for a resource. For example, for a VM it shows the NIC, virtual network, network security groups (NSGs) etc.
  - Show first level dependencies only shows only direct dependencies. For example, for a VM it shows the NIC, but not the virtual network.
- Select the dependent resources you want to add > **Add dependencies**. Monitor progress in the notifications.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Some resources in your list may require other resources to be moved along with them. Review and add the dependencies**

**Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties.** [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 2

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1286	Network interface	raynetest
<input checked="" type="checkbox"/> raynetest	Resource group	-

- Validate dependencies again.

**Add dependencies**

Azure Resource Mover - Across regions | PREVIEW

**Some resources in your list may require other resources to be moved along with them. Review and add the dependencies**

**Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties.** [Learn more](#).

Select all Clear selection | Selected items: 2 | Total items: 3

Name	Type	Resource group
<input checked="" type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest
<input checked="" type="checkbox"/> raynetest-vnet	Virtual network	raynetest
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest

## Move the source resource group

Before you can prepare and move VMs, the VM resource group must be present in the target region.

## Prepare to move the source resource group

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

Prepare as follows:

1. In **Across regions**, select the source resource group > **Prepare**.
2. In **Prepare resources**, click **Prepare**.

Name	Type
raynetest	Resource group

### NOTE

After preparing the resource group, it's in the *Initiate move pending* state.

## Move the source resource group

Initiate the move as follows:

1. In **Across regions**, select the resource group > **Initiate Move**
2. In **Move Resources**, click **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

<input type="checkbox"/> RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	<span>ⓘ 'Prepare' pending</span>
<input type="checkbox"/> raynetest	Resource group	-	(new) raynetest-eastus2	<span>ⓘ 'Commit move' pending</span>
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	<span>ⓘ 'Prepare' pending</span>
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	<span>ⓘ 'Prepare' pending</span>
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	<span>ⓘ 'Prepare' pending</span>
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	<span>ⓘ 'Prepare' pending</span>

To commit and finish the move process:

1. In **Across regions**, select the resource group > **Commit move**.
2. In **Move Resources**, click **Commit**.

### NOTE

After committing the move, the source resource group is in a *Delete source pending* state.

## Prepare resources to move

Now that the source resource group is moved, you can prepare to move other resources that are in the *Prepare pending* state.

1. In **Across regions**, verify that resources are now in a *Prepare pending* state, with no issues. If they're not, validate again and resolve any outstanding issues.

<input type="checkbox"/>  RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	 'Prepare' pending
<input type="checkbox"/>  raynetest	Resource group	-	(new) raynetest-eastus2	 'Prepare' pending
<input type="checkbox"/>  raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	 'Prepare' pending
<input type="checkbox"/>  raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	 'Prepare' pending
<input type="checkbox"/>  raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	 'Prepare' pending
<input type="checkbox"/>  raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	 'Prepare' pending

- If you want to edit target settings before beginning the move, select the link in the **Destination configuration** column for the resource, and edit the settings. If you edit the target VM settings, the target VM size shouldn't be smaller than the source VM size.

Now that the source resource group is moved, you can prepare to move the other resources.

- Select the resources you want to prepare.

**Prepare resources** 

Azure Resource Mover - Across regions | PREVIEW

 Want to assign an existing resource in the destination region? [Edit the target properties before you prepare.](#)

 Learn more about how different resources are prepared.

Name	Type	Resource group
 RayneTestVM-1	Virtual machine	RayneTest
 raynetestvm-1286	Network interface	raynetest
 raynetestvm-1-nsg	Network security group	raynetest
 raynetest-vnet	Virtual network	raynetest
 raynetestvm-1-ip	Public IP address	raynetest

- Select **Prepare**.

#### NOTE

- During the prepare process, the Azure Site Recovery Mobility agent is installed on VMs, to replicate them.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

<input type="checkbox"/>  RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	 'Initiate move' pending
<input type="checkbox"/>  raynetest	Resource group	-	(new) raynetest-eastus2	 'Delete source' pending
<input type="checkbox"/>  raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	 'Initiate move' pending
<input type="checkbox"/>  raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	 'Initiate move' pending
<input type="checkbox"/>  raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	 'Initiate move' pending
<input type="checkbox"/>  raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	 'Initiate move' pending

## Initiate the move

With resources prepared, you can now initiate the move.

- In **Across regions**, select resources with state *Initiate move pending*. Then click **Initiate move**.
- In **Move resources**, click **Initiate move**.

Name	Type
RayneTestVM-1	Virtual machine
raynetestvm-1286	Network interface
raynetestvm-1-nsg	Network security group
raynetest-vnet	Virtual network
raynetestvm-1-ip	Public IP address

Initiate move  
 Cancel

3. Track move progress in the notifications bar.

**NOTE**

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- After moving resources, they're in an *Commit move pending* state.

<input type="checkbox"/>	 RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	 'Delete source' pending
<input type="checkbox"/>	 raynetest	Resource group	-	(new) raynetest-eastus2	 'Delete source' pending
<input type="checkbox"/>	 raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	 'Delete source' pending
<input type="checkbox"/>	 raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	 'Delete source' pending
<input type="checkbox"/>	 raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	 'Delete source' pending
<input type="checkbox"/>	 raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	 'Delete source' pending

## Discard or commit?

After the initial move, you can decide whether you want to commit the move, or to discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

## Discard the move

You can discard the move as follows:

1. In **Across regions**, select resources with state *Commit move pending*, and click **Discard move**.
2. In **Discard move**, click **Discard**.
3. Track move progress in the notifications bar.

### NOTE

After discarding resources, VMs are in an *Initiate move pending* state.

## Commit the move

If you want to complete the move process, commit the move.

1. In **Across regions**, select resources with state *Commit move pending*, and click **Commit move**.
2. In **Commit resources**, click **Commit**.

## Commit resources

PREVIEW

Name	Type
 RayneTestVM-1	Virtual machine
 raynetestvm-1286	Network interface
 raynetestvm-1-nsg	Network security group
 raynetest-vnet	Virtual network
 raynetestvm-1-ip	Public IP address

**Commit**

**Cancel**

3. Track the commit progress in the notifications bar.

#### NOTE

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	'Delete source' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	'Delete source' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	'Delete source' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	'Delete source' pending

## Configure settings after the move

- The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.
- Modify Azure role-based access control (Azure RBAC) rules after the move.

## Delete source resources after commit

After the move, you can optionally delete resources in the source region.

#### NOTE

A few resources, for example key vaults and SQL Server servers, can't be deleted from the portal, and must be deleted from the resource property page.

1. In **Across Regions**, click the name of the source resource that you want to delete.
2. Select **Delete source**.

## Delete additional resources created for move

After the move, you can manually delete the move collection, and Site Recovery resources that were created.

- The move collection is hidden by default. To see it you need to turn on hidden resources.
- The cache storage has a lock that must be deleted, before it can be deleted.

Delete as follows:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`.
2. Check that all the VM and other source resources in the source region have been moved or deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
  - The move collection name is `movecollection-<sourcerregion>-<target-region>`.
  - The cache storage account name is `resmovecache<guid>`
  - The vault name is `ResourceMove-<sourcerregion>-<target-region>-GUID`.

## Next steps

In this tutorial, you:

- Moved Azure VMs to another Azure region.

- Moved resources associated with VMs to another region.

Now, trying moving Azure SQL databases and elastic pools to another region.

[Move Azure SQL resources](#)

# Move Azure network security group (NSG) to another region using the Azure portal

2/18/2021 • 4 minutes to read • [Edit Online](#)

There are various scenarios in which you'd want to move your existing NSGs from one region to another. For example, you may want to create an NSG with the same configuration and security rules for testing. You may also want to move an NSG to another region as part of disaster recovery planning.

Azure security groups can't be moved from one region to another. You can however, use an Azure Resource Manager template to export the existing configuration and security rules of an NSG. You can then stage the resource in another region by exporting the NSG to a template, modifying the parameters to match the destination region, and then deploy the template to the new region. For more information on Resource Manager and templates, see [Quickstart: Create and deploy Azure Resource Manager templates by using the Azure portal](#).

## Prerequisites

- Make sure that the Azure network security group is in the Azure region from which you want to move.
- Azure network security groups can't be moved between regions. You'll have to associate the new NSG to resources in the target region.
- To export an NSG configuration and deploy a template to create an NSG in another region, you'll need the Network Contributor role or higher.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, public IPs, and virtual networks.
- Verify that your Azure subscription allows you to create NSGs in the target region that's used. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support the addition of NSGs for this process. See [Azure subscription and service limits, quotas, and constraints](#).

## Prepare and move

The following steps show how to prepare the network security group for the configuration and security rule move using a Resource Manager template, and move the NSG configuration and security rules to the target region using the portal.

### Export the template and deploy from the portal

1. Login to the [Azure portal](#) > **Resource Groups**.
2. Locate the Resource Group that contains the source NSG and click on it.
3. Select > **Settings** > **Export template**.
4. Choose **Deploy** in the **Export template** blade.
5. Click **TEMPLATE** > **Edit parameters** to open the **parameters.json** file in the online editor.
6. To edit the parameter of the NSG name, change the **value** property under **parameters**:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "networkSecurityGroups_myVM1_nsg_name": {
 "value": "<target-nsg-name>"
 }
 }
}
```

7. Change the source NSG value in the editor to a name of your choice for the target NSG. Ensure you enclose the name in quotes.
8. Click **Save** in the editor.
9. Click **TEMPLATE > Edit template** to open the **template.json** file in the online editor.
10. To edit the target region where the NSG configuration and security rules will be moved, change the **location** property under **resources** in the online editor:

```
"resources": [
{
 "type": "Microsoft.Network/networkSecurityGroups",
 "apiVersion": "2019-06-01",
 "name": "[parameters('networkSecurityGroups_myVM1_nsg_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "2c846acf-58c8-416d-be97-ccd00a4ccd78",
 }
}
]
```

11. To obtain region location codes, see [Azure Locations](#). The code for a region is the region name with no spaces, **Central US = centralus**.
12. You can also change other parameters in the template if you choose, and are optional depending on your requirements:
  - **Security rules** - You can edit which rules are deployed into the target NSG by adding or removing rules to the **securityRules** section in the **template.json** file:

```

"resources": [
 {
 "type": "Microsoft.Network/networkSecurityGroups",
 "apiVersion": "2019-06-01",
 "name": "[parameters('networkSecurityGroups_myVM1_nsg_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "2c846acf-58c8-416d-be97-ccd00a4cccd78",
 "securityRules": [
 {
 "name": "RDP",
 "etag": "W/\\"c630c458-6b52-4202-8fd7-172b7ab49cf5\\"",
 "properties": {
 "provisioningState": "Succeeded",
 "protocol": "TCP",
 "sourcePortRange": "*",
 "destinationPortRange": "3389",
 "sourceAddressPrefix": "*",
 "destinationAddressPrefix": "*",
 "access": "Allow",
 "priority": 300,
 "direction": "Inbound",
 "sourcePortRanges": [],
 "destinationPortRanges": [],
 "sourceAddressPrefixes": [],
 "destinationAddressPrefixes": []
 }
 },
],
 }
 }
]
}

```

To complete the addition or the removal of the rules in the target NSG, you must also edit the custom rule types at the end of the **template.json** file in the format of the example below:

```

{
 "type": "Microsoft.Network/networkSecurityGroups/securityRules",
 "apiVersion": "2019-06-01",
 "name": "[concat(parameters('networkSecurityGroups_myVM1_nsg_name'), '/Port_80')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/networkSecurityGroups',
parameters('networkSecurityGroups_myVM1_nsg_name'))]"
],
 "properties": {
 "provisioningState": "Succeeded",
 "protocol": "*",
 "sourcePortRange": "*",
 "destinationPortRange": "80",
 "sourceAddressPrefix": "*",
 "destinationAddressPrefix": "*",
 "access": "Allow",
 "priority": 310,
 "direction": "Inbound",
 "sourcePortRanges": [],
 "destinationPortRanges": [],
 "sourceAddressPrefixes": [],
 "destinationAddressPrefixes": []
 }
}

```

13. Click **Save** in the online editor.
14. Click **BASICS > Subscription** to choose the subscription where the target NSG will be deployed.
15. Click **BASICS > Resource group** to choose the resource group where the target NSG will be deployed.

You can click **Create new** to create a new resource group for the target NSG. Ensure the name isn't the same as the source resource group of the existing NSG.

16. Verify **BASICS > Location** is set to the target location where you wish for the NSG to be deployed.
17. Verify under **SETTINGS** that the name matches the name that you entered in the parameters editor above.
18. Check the box under **TERMS AND CONDITIONS**.
19. Click the **Purchase** button to deploy the target network security group.

## Discard

If you wish to discard the target NSG, delete the resource group that contains the target NSG. To do so, select the resource group from your dashboard in the portal and select **Delete** at the top of the overview page.

## Clean up

To commit the changes and complete the move of the NSG, delete the source NSG or resource group. To do so, select the network security group or resource group from your dashboard in the portal and select **Delete** at the top of each page.

## Next steps

In this tutorial, you moved an Azure network security group from one region to another and cleaned up the source resources. To learn more about moving resources between regions and disaster recovery in Azure, refer to:

- [Move resources to a new resource group or subscription](#)
- [Move Azure VMs to another region](#)

# Move Azure network security group (NSG) to another region using Azure PowerShell

2/18/2021 • 4 minutes to read • [Edit Online](#)

There are various scenarios in which you'd want to move your existing NSGs from one region to another. For example, you may want to create an NSG with the same configuration and security rules for testing. You may also want to move an NSG to another region as part of disaster recovery planning.

Azure security groups can't be moved from one region to another. You can however, use an Azure Resource Manager template to export the existing configuration and security rules of an NSG. You can then stage the resource in another region by exporting the NSG to a template, modifying the parameters to match the destination region, and then deploy the template to the new region. For more information on Resource Manager and templates, see [Export resource groups to templates](#).

## Prerequisites

- Make sure that the Azure network security group is in the Azure region from which you want to move.
- Azure network security groups can't be moved between regions. You'll have to associate the new NSG to resources in the target region.
- To export an NSG configuration and deploy a template to create an NSG in another region, you'll need the Network Contributor role or higher.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, public IPs, and virtual networks.
- Verify that your Azure subscription allows you to create NSGs in the target region that's used. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support the addition of NSGs for this process. See [Azure subscription and service limits, quotas, and constraints](#).

## Prepare and move

The following steps show how to prepare the network security group for the configuration and security rule move using a Resource Manager template, and move the NSG configuration and security rules to the target region using Azure PowerShell.

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

### Export the template and deploy from a script

1. Sign in to your Azure subscription with the `Connect-AzAccount` command and follow the on-screen directions:

```
Connect-AzAccount
```

2. Obtain the resource ID of the NSG you want to move to the target region and place it in a variable using [Get-AzNetworkSecurityGroup](#):

```
$sourceNSGID = (Get-AzNetworkSecurityGroup -Name <source-nsg-name> -ResourceGroupName <source-resource-group-name>).Id
```

3. Export the source NSG to a .json file into the directory where you execute the command [Export-AzResourceGroup](#):

```
Export-AzResourceGroup -ResourceGroupName <source-resource-group-name> -Resource $sourceNSGID -IncludeParameterDefaultValue
```

4. The file downloaded will be named after the resource group the resource was exported from. Locate the file that was exported from the command named <resource-group-name>.json and open it in an editor of your choice:

```
notepad <source-resource-group-name>.json
```

5. To edit the parameter of the NSG name, change the property **defaultValue** of the source NSG name to the name of your target NSG, ensure the name is in quotes:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "networkSecurityGroups_myVM1_nsg_name": {
 "defaultValue": "<target-nsg-name>",
 "type": "String"
 }
 }
}
```

6. To edit the target region where the NSG configuration and security rules will be moved, change the **location** property under **resources**:

```
"resources": [
{
 "type": "Microsoft.Network/networkSecurityGroups",
 "apiVersion": "2019-06-01",
 "name": "[parameters('networkSecurityGroups_myVM1_nsg_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "2c846acf-58c8-416d-be97-ccd00a4ccd78",
 }
}]
```

7. To obtain region location codes, you can use the Azure PowerShell cmdlet [Get-AzLocation](#) by running the following command:

```
Get-AzLocation | format-table
```

8. You can also change other parameters in the <resource-group-name>.json if you choose, and are optional depending on your requirements:

- **Security rules** - You can edit which rules are deployed into the target NSG by adding or removing rules to the **securityRules** section in the <resource-group-name>.json file:

```
"resources": [
 {
 "type": "Microsoft.Network/networkSecurityGroups",
 "apiVersion": "2019-06-01",
 "name": "[parameters('networkSecurityGroups_myVM1_nsg_name')]",
 "location": "TARGET REGION",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "2c846acf-58c8-416d-be97-ccd00a4ccd78",
 "securityRules": [
 {
 "name": "RDP",
 "etag": "W/\\"c630c458-6b52-4202-8fd7-172b7ab49cf5\\\"",
 "properties": {
 "provisioningState": "Succeeded",
 "protocol": "TCP",
 "sourcePortRange": "*",
 "destinationPortRange": "3389",
 "sourceAddressPrefix": "*",
 "destinationAddressPrefix": "*",
 "access": "Allow",
 "priority": 300,
 "direction": "Inbound",
 "sourcePortRanges": [],
 "destinationPortRanges": [],
 "sourceAddressPrefixes": [],
 "destinationAddressPrefixes": []
 }
 }
]
 }
 }
]
```

To complete the addition or the removal of the rules in the target NSG, you must also edit the custom rule types at the end of the <resource-group-name>.json file in the format of the example below:

```
{
 "type": "Microsoft.Network/networkSecurityGroups/securityRules",
 "apiVersion": "2019-06-01",
 "name": "[concat(parameters('networkSecurityGroups_myVM1_nsg_name'), '/Port_80')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/networkSecurityGroups',
parameters('networkSecurityGroups_myVM1_nsg_name'))]"
],
 "properties": {
 "provisioningState": "Succeeded",
 "protocol": "*",
 "sourcePortRange": "*",
 "destinationPortRange": "80",
 "sourceAddressPrefix": "*",
 "destinationAddressPrefix": "*",
 "access": "Allow",
 "priority": 310,
 "direction": "Inbound",
 "sourcePortRanges": [],
 "destinationPortRanges": [],
 "sourceAddressPrefixes": [],
 "destinationAddressPrefixes": []
 }
}
```

9. Save the <resource-group-name>.json file.

10. Create a resource group in the target region for the target NSG to be deployed using [New-AzResourceGroup](#):

```
New-AzResourceGroup -Name <target-resource-group-name> -location <target-region>
```

11. Deploy the edited <resource-group-name>.json file to the resource group created in the previous step using [New-AzResourceGroupDeployment](#):

```
New-AzResourceGroupDeployment -ResourceGroupName <target-resource-group-name> -TemplateFile <source-resource-group-name>.json
```

12. To verify the resources were created in the target region, use [Get-AzResourceGroup](#) and [Get-AzNetworkSecurityGroup](#):

```
Get-AzResourceGroup -Name <target-resource-group-name>
```

```
Get-AzNetworkSecurityGroup -Name <target-nsg-name> -ResourceGroupName <target-resource-group-name>
```

## Discard

After the deployment, if you wish to start over or discard the NSG in the target, delete the resource group that was created in the target and the moved NSG will be deleted. To remove the resource group, use [Remove-AzResourceGroup](#):

```
Remove-AzResourceGroup -Name <target-resource-group-name>
```

## Clean up

To commit the changes and complete the move of the NSG, delete the source NSG or resource group, use [Remove-AzResourceGroup](#) or [Remove-AzNetworkSecurityGroup](#):

```
Remove-AzResourceGroup -Name <source-resource-group-name>
```

```
Remove-AzNetworkSecurityGroup -Name <source-nsg-name> -ResourceGroupName <source-resource-group-name>
```

## Next steps

In this tutorial, you moved an Azure network security group from one region to another and cleaned up the source resources. To learn more about moving resources between regions and disaster recovery in Azure, refer to:

- [Move resources to a new resource group or subscription](#)
- [Move Azure VMs to another region](#)

# Move an Azure virtual network to another region by using the Azure portal

2/18/2021 • 5 minutes to read • [Edit Online](#)

There are various scenarios for moving an existing Azure virtual network from one region to another. For example, you might want to create a virtual network with the same configuration for testing and availability as your existing virtual network. Or you might want to move a production virtual network to another region as part of your disaster recovery planning.

You can use an Azure Resource Manager template to complete the move of the virtual network to another region. You do this by exporting the virtual network to a template, modifying the parameters to match the destination region, and then deploying the template to the new region. For more information about Resource Manager templates, see [Quickstart: Create and deploy Azure Resource Manager templates by using the Azure portal](#).

## Prerequisites

- Make sure that your virtual network is in the Azure region that you want to move from.
- To export a virtual network and deploy a template to create a virtual network in another region, you need to have the Network Contributor role or higher.
- Virtual network peerings won't be re-created, and they'll fail if they're still present in the template. Before you export the template, you have to remove any virtual network peers. You can then reestablish them after the virtual network move.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, network security groups (NSGs), and public IPs.
- Verify that your Azure subscription allows you to create virtual networks in the target region. To enable the required quota, contact support.
- Make sure that your subscription has enough resources to support the addition of virtual networks for this process. For more information, see [Azure subscription and service limits, quotas, and constraints](#).

## Prepare for the move

In this section, you prepare the virtual network for the move by using a Resource Manager template. You then move the virtual network to the target region by using the Azure portal.

To export the virtual network and deploy the target virtual network by using the Azure portal, do the following:

1. Sign in to the [Azure portal](#), and then select **Resource Groups**.
2. Locate the resource group that contains the source virtual network, and then select it.
3. Select **Settings > Export template**.
4. In the **Export template** pane, select **Deploy**.
5. To open the *parameters.json* file in your online editor, select **Template > Edit parameters**.
6. To edit the parameter of the virtual network name, change the **value** property under **parameters**:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "virtualNetworks_myVNET1_name": {
 "value": "<target-virtual-network-name>"
 }
 }
}
```

7. In the editor, change the source virtual network name value in the editor to a name that you want for the target virtual network. Be sure to enclose the name in quotation marks.
8. Select **Save** in the editor.
9. To open the *template.json* file in the online editor, select **Template > Edit template**.
10. In the online editor, to edit the target region where the virtual network will be moved, change the **location** property under **resources**:

```
"resources": [
 {
 "type": "Microsoft.Network/virtualNetworks",
 "apiVersion": "2019-06-01",
 "name": "[parameters('virtualNetworks_myVNET1_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
 "addressSpace": {
 "addressPrefixes": [
 "10.0.0.0/16"
]
 },
 }
 }
],
```

11. To obtain region location codes, see [Azure Locations](#). The code for a region is the region name, without spaces (for example, **Central US** = **centralus**).
12. (Optional) You can also change other parameters in the template, depending on your requirements:

- **Address Space:** Before you save the file, you can alter the address space of the virtual network by modifying the **resources > addressSpace** section and changing the **addressPrefixes** property:

```
"resources": [
 {
 "type": "Microsoft.Network/virtualNetworks",
 "apiVersion": "2019-06-01",
 "name": "[parameters('virtualNetworks_myVNET1_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
 "addressSpace": {
 "addressPrefixes": [
 "10.0.0.0/16"
]
 },
 }
 }
],
```

- **Subnet:** You can change or add to the subnet name and the subnet address space by changing the

template's **subnets** section. You can change the name of the subnet by changing the **name** property. And you can change the subnet address space by changing the **addressPrefix** property:

```
"subnets": [
 {
 "name": "subnet-1",
 "etag": "W/\"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea\"",
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.0.0/24",
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
 },
 {
 "name": "GatewaySubnet",
 "etag": "W/\"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea\"",
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.1.0/29",
 "serviceEndpoints": [],
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
 }
]
```

To change the address prefix in the *template.json* file, edit it in two places: in the code in the preceding section and in the **type** section of the following code. Change the **addressPrefix** property in the following code to match the **addressPrefix** property in the code in the preceding section.

```

"type": "Microsoft.Network/virtualNetworks/subnets",
"apiVersion": "2019-06-01",
"name": "[concat(parameters('virtualNetworks_myVNET1_name'), '/GatewaySubnet')]",
"dependsOn": [
 "[resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworks_myVNET1_name'))]"
],
"properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.1.0/29",
 "serviceEndpoints": [],
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
}
},
{
 "type": "Microsoft.Network/virtualNetworks/subnets",
 "apiVersion": "2019-06-01",
 "name": "[concat(parameters('virtualNetworks_myVNET1_name'), '/subnet-1')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworks_myVNET1_name'))]"
],
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.0.0/24",
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
}
]

```

13. In the online editor, select **Save**.
14. To choose the subscription where the target virtual network will be deployed, select **Basics > Subscription**.
15. To choose the resource group where the target virtual network will be deployed, select **Basics > Resource group**.  
If you need to create a new resource group for the target virtual network, select **Create new**. Make sure that the name isn't the same as the source resource group name in the existing virtual network.
16. Verify that **Basics > Location** is set to the target location where you want the virtual network to be deployed.
17. Under **Settings**, verify that the name matches the name that you entered previously in the parameters editor.
18. Select the **Terms and Conditions** check box.
19. To deploy the target virtual network, select **Purchase**.

## Delete the target virtual network

To discard the target virtual network, you delete the resource group that contains the target virtual network. To do so:

1. On the Azure portal dashboard, select the resource group.
2. At the top of the **Overview** pane, select **Delete**.

## Clean up

To commit the changes and complete the virtual network move, you delete the source virtual network or resource group. To do so:

1. On the Azure portal dashboard, select the virtual network or resource group.
2. At the top of each pane, select **Delete**.

## Next steps

In this tutorial, you moved an Azure virtual network from one region to another by using the Azure portal and then cleaned up the unneeded source resources. To learn more about moving resources between regions and disaster recovery in Azure, see:

- [Move resources to a new resource group or subscription](#)
- [Move Azure virtual machines to another region](#)

# Move an Azure virtual network to another region by using Azure PowerShell

2/18/2021 • 5 minutes to read • [Edit Online](#)

There are various scenarios for moving an existing Azure virtual network from one region to another. For example, you might want to create a virtual network with the same configuration for testing and availability as your existing virtual network. Or you might want to move a production virtual network to another region as part of your disaster recovery planning.

You can use an Azure Resource Manager template to complete the move of the virtual network to another region. You do this by exporting the virtual network to a template, modifying the parameters to match the destination region, and then deploying the template to the new region. For more information about Resource Manager templates, see [Export resource groups to templates](#).

## Prerequisites

- Make sure that your virtual network is in the Azure region that you want to move from.
- To export a virtual network and deploy a template to create a virtual network in another region, you need to have the Network Contributor role or higher.
- Virtual network peerings won't be re-created, and they'll fail if they're still present in the template. Before you export the template, you have to remove any virtual network peers. You can then reestablish them after the virtual network move.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, network security groups (NSGs), and public IPs.
- Verify that your Azure subscription allows you to create virtual networks in the target region. To enable the required quota, contact support.
- Make sure that your subscription has enough resources to support the addition of virtual networks for this process. For more information, see [Azure subscription and service limits, quotas, and constraints](#).

## Prepare for the move

In this section, you prepare the virtual network for the move by using a Resource Manager template. You then move the virtual network to the target region by using Azure PowerShell commands.

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

To export the virtual network and deploy the target virtual network by using PowerShell, do the following:

1. Sign in to your Azure subscription with the `Connect-AzAccount` command, and then follow the on-screen directions:

```
Connect-AzAccount
```

2. Obtain the resource ID of the virtual network that you want to move to the target region, and then place it in a variable by using [Get-AzVirtualNetwork](#):

```
$sourceVNETID = (Get-AzVirtualNetwork -Name <source-virtual-network-name> -ResourceGroupName <source-resource-group-name>).Id
```

3. Export the source virtual network to a json file in the directory where you execute the command [Export-AzResourceGroup](#):

```
Export-AzResourceGroup -ResourceGroupName <source-resource-group-name> -Resource $sourceVNETID -
IncludeParameterDefaultValue
```

4. The downloaded file has the same name as the resource group that the resource was exported from. Locate the *<resource-group-name>.json* file, which you exported with the command, and then open it in your editor:

```
notepad <source-resource-group-name>.json
```

5. To edit the parameter of the virtual network name, change the **defaultValue** property of the source virtual network name to the name of your target virtual network. Be sure to enclose the name in quotation marks.

```
"$schema": "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplateResourceGroupVNET.json#",
"contentVersion": "1.0.0.0",
"parameters": {
 "virtualNetworks_myVNET1_name": {
 "defaultValue": "<target-virtual-network-name>",
 "type": "String"
 }
}
```

6. To edit the target region where the virtual network will be moved, change the **location** property under resources:

```
"resources": [
 {
 "type": "Microsoft.Network/virtualNetworks",
 "apiVersion": "2019-06-01",
 "name": "[parameters('virtualNetworks_myVNET1_name')]",
 "location": "<target-region>",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
 "addressSpace": {
 "addressPrefixes": [
 "10.0.0.0/16"
]
 },
 },
 },
```

7. To obtain region location codes, you can use the Azure PowerShell cmdlet [Get-AzLocation](#) by running the following command:

```
Get-AzLocation | format-table
```

8. (Optional) You can also change other parameters in the *<resource-group-name>.json* file, depending on your requirements:

- **Address Space:** Before you save the file, you can alter the address space of the virtual network by modifying the **resources > addressSpace** section and changing the **addressPrefixes** property:

```
"resources": [
 {
 "type": "Microsoft.Network/virtualNetworks",
 "apiVersion": "2019-06-01",
 "name": "[parameters('virtualNetworks_myVNET1_name')]",
 "location": "<target-region",
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
 "addressSpace": {
 "addressPrefixes": [
 "10.0.0.0/16"
]
 },
 }
},
```

- **Subnet:** You can change or add to the subnet name and the subnet address space by changing the file's **subnets** section. You can change the name of the subnet by changing the **name** property. And you can change the subnet address space by changing the **addressPrefix** property:

```
"subnets": [
 {
 "name": "subnet-1",
 "etag": "W/"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea"",
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.0.0/24",
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
 },
 {
 "name": "GatewaySubnet",
 "etag": "W/"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea"",
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.1.0/29",
 "serviceEndpoints": [],
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
 }
]
```

To change the address prefix, edit the file in two places: in the code in the preceding section and in the **type** section of the following code. Change the **addressPrefix** property in the following code to match the **addressPrefix** property in the code in the preceding section.

```

"type": "Microsoft.Network/virtualNetworks/subnets",
"apiVersion": "2019-06-01",
"name": "[concat(parameters('virtualNetworks_myVNET1_name'), '/GatewaySubnet')]",
"dependsOn": [
 "[resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworks_myVNET1_name'))]"
],
"properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.1.0/29",
 "serviceEndpoints": [],
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
}
},
{
 "type": "Microsoft.Network/virtualNetworks/subnets",
 "apiVersion": "2019-06-01",
 "name": "[concat(parameters('virtualNetworks_myVNET1_name'), '/subnet-1')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworks_myVNET1_name'))]"
],
 "properties": {
 "provisioningState": "Succeeded",
 "addressPrefix": "10.0.0.0/24",
 "delegations": [],
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
}
]

```

9. Save the *<resource-group-name>.json* file.

10. Create a resource group in the target region for the target virtual network to be deployed by using [New-AzResourceGroup](#):

```
New-AzResourceGroup -Name <target-resource-group-name> -location <target-region>
```

11. Deploy the edited *<resource-group-name>.json* file to the resource group that you created in the previous step by using [New-AzResourceGroupDeployment](#):

```
New-AzResourceGroupDeployment -ResourceGroupName <target-resource-group-name> -TemplateFile <source-resource-group-name>.json
```

12. To verify that the resources were created in the target region, use [Get-AzResourceGroup](#) and [Get-AzVirtualNetwork](#):

```
Get-AzResourceGroup -Name <target-resource-group-name>
```

```
Get-AzVirtualNetwork -Name <target-virtual-network-name> -ResourceGroupName <target-resource-group-name>
```

## Delete the virtual network or resource group

After you've deployed the virtual network, to start over or discard the virtual network in the target region, delete the resource group that you created in the target region, and the moved virtual network will be deleted.

To remove the resource group, use [Remove-AzResourceGroup](#):

```
Remove-AzResourceGroup -Name <target-resource-group-name>
```

## Clean up

To commit your changes and complete the virtual network move, do either of the following:

- Delete the resource group by using [Remove-AzResourceGroup](#):

```
Remove-AzResourceGroup -Name <source-resource-group-name>
```

- Delete the source virtual network by using [Remove-AzVirtualNetwork](#):

```
Remove-AzVirtualNetwork -Name <source-virtual-network-name> -ResourceGroupName <source-resource-group-name>
```

## Next steps

In this tutorial, you moved a virtual network from one region to another by using PowerShell and then cleaned up the unneeded source resources. To learn more about moving resources between regions and disaster recovery in Azure, see:

- [Move resources to a new resource group or subscription](#)
- [Move Azure virtual machines to another region](#)

# Move Azure Public IP configuration to another region using the Azure portal

2/18/2021 • 4 minutes to read • [Edit Online](#)

There are various scenarios in which you'd want to move your existing Azure Public IP configurations from one region to another. For example, you may want to create a public IP with the same configuration and sku for testing. You may also want to move a public IP configuration to another region as part of disaster recovery planning.

**Azure Public IPs are region specific and can't be moved from one region to another.** You can however, use an Azure Resource Manager template to export the existing configuration of a public IP. You can then stage the resource in another region by exporting the public IP to a template, modifying the parameters to match the destination region, and then deploy the template to the new region. For more information on Resource Manager and templates, see [Quickstart: Create and deploy Azure Resource Manager templates by using the Azure portal](#).

## Prerequisites

- Make sure that the Azure Public IP is in the Azure region from which you want to move.
- Azure Public IPs can't be moved between regions. You'll have to associate the new public ip to resources in the target region.
- To export a public IP configuration and deploy a template to create a public IP in another region, you'll need the Network Contributor role or higher.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, network security groups (NSGs), and virtual networks.
- Verify that your Azure subscription allows you to create public IPs in the target region that's used. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support the addition of public IPs for this process. See [Azure subscription and service limits, quotas, and constraints](#).

## Prepare and move

The following steps show how to prepare the public IP for the configuration move using a Resource Manager template, and move the public IP configuration to the target region using the Azure portal.

### Export the template and deploy from a script

1. Login to the [Azure portal](#) > **Resource Groups**.
2. Locate the Resource Group that contains the source public IP and click on it.
3. Select > **Settings** > **Export template**.
4. Choose **Deploy** in the **Export template** blade.
5. Click **TEMPLATE** > **Edit parameters** to open the **parameters.json** file in the online editor.
6. To edit the parameter of the public IP name, change the property under **parameters** > **value** from the source public IP name to the name of your target public IP, ensure the name is in quotes:

```
{
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
"contentVersion": "1.0.0.0",
"parameters": {
 "publicIPAddresses_myVM1pubIP_name": {
 "value": "<target-publicip-name>"
 }
}
}
```

7. Click **Save** in the editor.
8. Click **TEMPLATE > Edit template** to open the **template.json** file in the online editor.
9. To edit the target region where the public IP will be moved, change the **location** property under **resources**:

```
"resources": [
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
 "ipAddress": "52.177.6.204",
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Dynamic",
 "idleTimeoutInMinutes": 4,
 "ipTags": []
 }
}
]
```

10. To obtain region location codes, see [Azure Locations](#). The code for a region is the region name with no spaces, **Central US = centralus**.
11. You can also change other parameters in the template if you choose, and are optional depending on your requirements:
  - **Sku** - You can change the sku of the public IP in the configuration from standard to basic or basic to standard by altering the **sku > name** property in the **template.json** file:

```
"resources": [
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
}
```

For more information on the differences between basic and standard sku public ips, see [Create](#),

[change, or delete a public IP address](#):

- **Public IP allocation method and Idle timeout** - You can change both of these options in the template by altering the **publicIPAllocationMethod** property from **Dynamic** to **Static** or **Static** to **Dynamic**. The idle timeout can be changed by altering the **idleTimeoutInMinutes** property to your desired amount. The default is 4:

```
"resources": [
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
 "ipAddress": "52.177.6.204",
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Dynamic",
 "idleTimeoutInMinutes": 4,
 "ipTags": []
 }
}
```

For more information on the allocation methods and the idle timeout values, see [Create, change, or delete a public IP address](#).

12. Click **Save** in the online editor.
13. Click **BASICS > Subscription** to choose the subscription where the target public IP will be deployed.
14. Click **BASICS > Resource group** to choose the resource group where the target public IP will be deployed. You can click **Create new** to create a new resource group for the target public IP. Ensure the name isn't the same as the source resource group of the existing source public IP.
15. Verify **BASICS > Location** is set to the target location where you wish for the public IP to be deployed.
16. Verify under **SETTINGS** that the name matches the name that you entered in the parameters editor above.
17. Check the box under **TERMS AND CONDITIONS**.
18. Click the **Purchase** button to deploy the target public IP.

## Discard

If you wish to discard the target public IP, delete the resource group that contains the target public IP. To do so, select the resource group from your dashboard in the portal and select **Delete** at the top of the overview page.

## Clean up

To commit the changes and complete the move of the public IP, delete the source public IP or resource group. To do so, select the public IP or resource group from your dashboard in the portal and select **Delete** at the top of each page.

## Next steps

In this tutorial, you moved an Azure Public IP from one region to another and cleaned up the source resources. To learn more about moving resources between regions and disaster recovery in Azure, refer to:

- [Move resources to a new resource group or subscription](#)
- [Move Azure VMs to another region](#)

# Move Azure Public IP configuration to another region using Azure PowerShell

2/18/2021 • 5 minutes to read • [Edit Online](#)

There are various scenarios in which you'd want to move your existing Azure Public IP configurations from one region to another. For example, you may want to create a public IP with the same configuration and sku for testing. You may also want to move a public IP configuration to another region as part of disaster recovery planning.

**Azure Public IPs are region specific and can't be moved from one region to another.** You can however, use an Azure Resource Manager template to export the existing configuration of a public IP. You can then stage the resource in another region by exporting the public IP to a template, modifying the parameters to match the destination region, and then deploy the template to the new region. For more information on Resource Manager and templates, see [Export resource groups to templates](#)

## Prerequisites

- Make sure that the Azure Public IP is in the Azure region from which you want to move.
- Azure Public IPs cannot be moved between regions. You'll have to associate the new public ip to resources in the target region.
- To export a public IP configuration and deploy a template to create a public IP in another region, you'll need the Network Contributor role or higher.
- Identify the source networking layout and all the resources that you're currently using. This layout includes but isn't limited to load balancers, network security groups (NSGs), and virtual networks.
- Verify that your Azure subscription allows you to create public IPs in the target region that's used. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support the addition of public IPs for this process. See [Azure subscription and service limits, quotas, and constraints](#).

## Prepare and move

The following steps show how to prepare the public IP for the configuration move using a Resource Manager template, and move the public IP configuration to the target region using Azure PowerShell.

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

### Export the template and deploy from a script

1. Sign in to your Azure subscription with the [Connect-AzAccount](#) command and follow the on-screen directions:

```
Connect-AzAccount
```

2. Obtain the resource ID of the public IP you want to move to the target region and place it in a variable using [Get-AzPublicIPAddress](#):

```
$sourcePubIPID = (Get-AzPublicIPAddress -Name <source-public-ip-name> -ResourceGroupName <source-resource-group-name>).Id
```

3. Export the source virtual network to a json file into the directory where you execute the command [Export-AzResourceGroup](#):

```
Export-AzResourceGroup -ResourceGroupName <source-resource-group-name> -Resource $sourceVNETID -
IncludeParameterDefaultValue
```

4. The file downloaded will be named after the resource group the resource was exported from. Locate the file that was exported from the command named <resource-group-name>.json and open it in an editor of your choice:

```
notepad <source-resource-group-name>.json
```

5. To edit the parameter of the public IP name, change the property **defaultValue** of the source public IP name to the name of your target public IP, ensure the name is in quotes:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "publicIPAddresses_myVM1pubIP_name": {
 "defaultValue": "<target-publicip-name>",
 "type": "String"
 }
 }
}
```

6. To edit the target region where the public IP will be moved, change the **location** property under resources:

```
"resources": [
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
 "ipAddress": "52.177.6.204",
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Dynamic",
 "idleTimeoutInMinutes": 4,
 "ipTags": []
 }
}
]
```

7. To obtain region location codes, you can use the Azure PowerShell cmdlet [Get-AzLocation](#) by running the following command:

```
Get-AzLocation | format-table
```

8. You can also change other parameters in the template if you choose, and are optional depending on your requirements:

- **Sku** - You can change the sku of the public IP in the configuration from standard to basic or basic to standard by altering the `sku > name` property in the `<resource-group-name>.json` file:

```
"resources": [
 {
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
],
}
```

For more information on the differences between basic and standard sku public ips, see [Create, change, or delete a public IP address](#).

- **Public IP allocation method and Idle timeout** - You can change both of these options in the template by altering the `publicIPAllocationMethod` property from **Dynamic** to **Static** or **Static** to **Dynamic**. The idle timeout can be changed by altering the `idleTimeoutInMinutes` property to your desired amount. The default is 4:

```
"resources": [
 {
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2019-06-01",
 "name": "[parameters('publicIPAddresses_myPubIP_name')]",
 "location": "<target-region>",
 "sku": {
 "name": "Basic",
 "tier": "Regional"
 },
 "properties": {
 "provisioningState": "Succeeded",
 "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
 "ipAddress": "52.177.6.204",
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Dynamic",
 "idleTimeoutInMinutes": 4,
 "ipTags": []
 }
 }
]
```

For more information on the allocation methods and the idle timeout values, see [Create, change, or delete a public IP address](#).

9. Save the `<resource-group-name>.json` file.

10. Create a resource group in the target region for the target public IP to be deployed using [New-AzResourceGroup](#).

```
New-AzResourceGroup -Name <target-resource-group-name> -location <target-region>
```

11. Deploy the edited <resource-group-name>.json file to the resource group created in the previous step using [New-AzResourceGroupDeployment](#):

```
New-AzResourceGroupDeployment -ResourceGroupName <target-resource-group-name> -TemplateFile <source-resource-group-name>.json
```

12. To verify the resources were created in the target region, use [Get-AzResourceGroup](#) and [Get-AzPublicIPAddress](#):

```
Get-AzResourceGroup -Name <target-resource-group-name>
```

```
Get-AzPublicIPAddress -Name <target-publicip-name> -ResourceGroupName <target-resource-group-name>
```

## Discard

After the deployment, if you wish to start over or discard the public ip in the target, delete the resource group that was created in the target and the moved public IP will be deleted. To remove the resource group, use [Remove-AzResourceGroup](#):

```
Remove-AzResourceGroup -Name <target-resource-group-name>
```

## Clean up

To commit the changes and complete the move of the virtual network, delete the source virtual network or resource group, use [Remove-AzResourceGroup](#) or [Remove-AzPublicIPAddress](#):

```
Remove-AzResourceGroup -Name <source-resource-group-name>
```

```
Remove-AzPublicIpAddress -Name <source-publicip-name> -ResourceGroupName <resource-group-name>
```

## Next steps

In this tutorial, you moved an Azure Public IP from one region to another and cleaned up the source resources. To learn more about moving resources between regions and disaster recovery in Azure, refer to:

- [Move resources to a new resource group or subscription](#)

- Move Azure VMs to another region

# Tutorial: Create a NAT gateway using Azure PowerShell

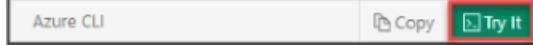
2/18/2021 • 8 minutes to read • [Edit Online](#)

This tutorial shows you how to use Azure Virtual Network NAT service. You'll create a NAT gateway to provide outbound connectivity for a virtual machine in Azure.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

You can complete this tutorial using Azure Cloud Shell or run the commands locally. If you haven't used Azure Cloud Shell, [sign in now](#).

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Create a resource group

Create a resource group with [New-AzResourceGroup](#). An Azure resource group is a logical container into which

Azure resources are deployed and managed.

The following example creates a resource group named **myResourceGroupNAT** in the **eastus2** location:

```
$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'

New-AzResourceGroup -Name $rsg -Location $loc
```

## Create the NAT gateway

Public IP options for NAT gateway are:

- Public IP addresses
- Public IP prefixes

Both can be used with NAT gateway.

We'll add a public IP address and a public IP prefix to this scenario to demonstrate.

### Create a public IP address

To access the Internet, you need one or more public IP addresses for the NAT gateway. Use [New-AzPublicIpAddress](#) to create a public IP address resource named **myPublicIP** in **myResourceGroupNAT**. The result of this command will be stored in a variable **\$publicIP** for later use.

```
$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'
$sku = 'Standard'
$pbnm = 'myPublicIP'

$publicIP =
New-AzPublicIpAddress -Name $pbnm -ResourceGroupName $rsg -AllocationMethod Static -Location $loc -Sku $sku
```

### Create a public IP prefix

Use [New-AzPublicIpPrefix](#) to create a public IP prefix resource named **myPublicIPprefix** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$publicIPPrefix** for later use.

```
$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'
$pxnm = 'myPublicIPprefix'

$publicIPPrefix =
New-AzPublicIpPrefix -Name $pxnm -ResourceGroupName $rsg -Location $loc -PrefixLength 31
```

### Create a NAT gateway resource

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a global Azure NAT gateway with [New-AzNatGateway](#). The result of this command will create a gateway resource named **myNATgateway** that uses the public IP address **myPublicIP** and the public IP prefix **myPublicIPprefix**. The idle timeout is set to 10 minutes. The result of this command will be stored in a variable named **\$natGateway** for later use.

```

$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'
$sku = 'Standard'
$gnm = 'myNATgateway'

$natGateway =
New-AzNatGateway -Name $gnm -ResourceGroupName $rsg -PublicIpAddress $publicIP -PublicIpPrefix
$publicIPPrefix -Location $loc -Sku $sku -IdleTimeoutInMinutes 10

```

At this point, the NAT gateway is functional and all that is missing is to configure which subnets of a virtual network should use it.

## Configure virtual network

Create the virtual network and associate the subnet to the gateway.

Create a virtual network named **myVnet** with a subnet named **mySubnet** using [New-AzVirtualNetworkSubnetConfig](#) in the **myResourceGroup** using [New-AzVirtualNetwork](#). The IP address space for the virtual network is **192.168.0.0/16**. The subnet within the virtual network is **192.168.0.0/24**. The result of the commands will be stored in variables named **\$subnet** and **\$vnet** for later use.

```

$sbnm = 'mySubnet'
$vnnm = 'myVnet'
$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'
$pfxsub = '192.168.0.0/24'
$pfxvn = '192.168.0.0/16'

$subnet =
New-AzVirtualNetworkSubnetConfig -Name $sbnm -AddressPrefix $pfxsub -NatGateway $natGateway

$vnet =
New-AzVirtualNetwork -Name $vnnm -ResourceGroupName $rsg -Location $loc -AddressPrefix $pfxvn -Subnet
$subnet

```

All outbound traffic to Internet destinations is now using the NAT service. It isn't necessary to configure a UDR.

## Create a VM to use the NAT service

We'll now create a VM to use the NAT service. This VM has a public IP to use as an instance-level Public IP to allow you to access the VM. NAT service is flow direction aware and will replace the default Internet destination in your subnet. The VM's public IP address won't be used for outbound connections.

### Create public IP for source VM

We create a public IP to be used to access the VM. Use [New-AzPublicIpAddress](#) to create a public IP address resource named **myPublicIPVM** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$publicIpVM** for later use.

```

$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'
$ipnm = 'myPublicIPVM'
$sku = 'Standard'

$publicIpVM =
New-AzPublicIpAddress -Name $ipnm -ResourceGroupName $rsg -AllocationMethod Static -Location $loc -Sku $sku

```

### Create an NSG and expose SSH endpoint for VM

Standard public IP addresses are 'secure by default', we need to create an NSG to allow inbound access for ssh. Use [New-AzNetworkSecurityGroup](#) to create an NSG resource named **myNSG**. Use [New-AzNetworkSecurityRuleConfig](#) to create an NSG rule for SSH access named **ssh** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$nsg** for later use.

```
$rnm = 'ssh'
$rdesc = 'SSH access'
$acc = 'Allow'
$pro = 'Tcp'
$dir = 'Inbound'
$pri = '100'
$prt = '22'
$rsg = 'myResourceGroupNAT'
$rnm = 'myNSG'
$loc = 'eastus2'

$sshrule =
New-AzNetworkSecurityRuleConfig -Name $rnm -Description $rdesc -Access $acc -Protocol $pro -Direction $dir -Priority $pri -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange $prt

$nsg =
New-AzNetworkSecurityGroup -ResourceGroupName $rsg -Name $rnm -Location $loc -SecurityRules $sshrule
```

## Create NIC for VM

Create a network interface with [New-AzNetworkInterface](#) named **myNic**. This command associates the Public IP address and the network security group. The result of this command will be stored in a variable named **\$nic** for later use.

```
$rsg = 'myResourceGroupNAT'
$nmn = 'myNic'
$loc = 'eastus2'

$nic =
New-AzNetworkInterface -ResourceGroupName $rsg -Name $nmn -NetworkSecurityGroupID $nsg.Id -PublicIPAddressID $publicIPVM.Id -SubnetID $vnet.Subnets[0].Id -Location $loc
```

## Create VM

### Create SSH key pair

You need an SSH key pair to complete this quickstart. If you already have an SSH key pair, you can skip this step.

Use ssh-keygen to create an SSH key pair.

```
ssh-keygen -t rsa -b 2048
```

For more detailed information on how to create SSH key pairs, including the use of PuTTy, see [How to use SSH keys with Windows](#).

If you create the SSH key pair using the Cloud Shell, the key pair is stored in a container image. This [storage account is automatically created](#). Don't delete the storage account, or the file share within, until after you've retrieved your keys.

### Create VM Configuration

To create a VM in PowerShell, you create a configuration that has settings for the image to use, size, and authentication options. The configuration is used to build the VM.

Define the SSH credentials, OS information, and VM size. In this example, the SSH key is stored in `~/ssh/id_rsa.pub`.

```

#Define a credential object

$securePassword =
ConvertTo-SecureString ' ' -AsPlainText -Force

$cred =
New-Object System.Management.Automation.PSCredential ("azureuser", $securePassword)

Create a virtual machine configuration

$vnm = 'myVM'
$vsz = 'Standard_D1'
$pub = 'Canonical'
$off = 'UbuntuServer'
$sku = '18.04-LTS'
$ver = 'latest'

$vmConfig =
New-AzVMConfig -VMName $vnm -VMSize $vsz

Set-AzVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vnm -Credential $cred -
DisablePasswordAuthentication

Set-AzVMSourceImage -VM $vmConfig -PublisherName $pub -Offer $off -Skus $sku -Version $ver

Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id

Configure the SSH key

$sshPublicKey = cat ~/.ssh/id_rsa.pub

Add-AzVMSshPublicKey -VM $vmconfig -KeyData $sshPublicKey -Path "/home/azureuser/.ssh/authorized_keys"

```

Combine the configuration definitions to create a VM named **myVM** with [New-AzVM](#) in **myResourceGroupNAT**.

```

$rsg = 'myResourceGroupNAT'
$loc = 'eastus2'

New-AzVM -ResourceGroupName $rsg -Location $loc -VM $vmconfig

```

Wait for the VM to prepare to deploy then continue with the rest of the steps.

## Discover the IP address of the VM

First we need to discover the IP address of the VM you've created. To get the public IP address of the VM, use [Get-AzPublicIpAddress](#).

```

$rsg = 'myResourceGroupNAT'
$nmn = 'myPublicIPVM'

Get-AzPublicIpAddress -ResourceGroupName $rsg -Name $nmn | select IPAddress

```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it to access the VM.

## Sign in to VM

The SSH credentials should be stored in your Cloud Shell from the previous operation. Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh azureuser@<ip-address-destination>
```

You're now ready to use the NAT service.

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group and all resources contained within.

```
Remove-AzResourceGroup -Name myResourceGroupNAT
```

## Next steps

In this tutorial, you created a NAT gateway and a VM to use it.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is addressed by adding additional public IP address resources or public IP prefix resources or both.

- Learn about [Azure Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).
- Quickstart for deploying [NAT gateway resource using Azure CLI](#).
- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

# Create a NAT gateway using Azure CLI

2/18/2021 • 6 minutes to read • [Edit Online](#)

This tutorial shows you how to use Azure Virtual Network NAT service. You'll create a NAT gateway to provide outbound connectivity for a virtual machine in Azure.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).

 [Launch Cloud Shell](#)

- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- This article requires version 2.0.71 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

Create a resource group with [az group create](#). An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named `myResourceGroupNAT` in the `eastus2` location:

```
az group create \
--name myResourceGroupNAT \
--location eastus2
```

## Create the NAT Gateway

### Create a public IP address

To access the public Internet, you need one or more public IP addresses for the NAT gateway. Use [az network public-ip create](#) to create a public IP address resource named `myPublicIP` in `myResourceGroupNAT`.

```
az network public-ip create \
--resource-group myResourceGroupNAT \
--name myPublicIP \
--sku standard
```

### Create a public IP prefix

You can use one or more public IP address resources, public IP prefixes, or both with NAT gateway. We'll add a public IP prefix resource to this scenario to demonstrate. Use [az network public-ip prefix create](#) to create a public IP prefix resource named **myPublicIPprefix** in **myResourceGroupNAT**.

```
az network public-ip prefix create \
--resource-group myResourceGroupNAT \
--name myPublicIPprefix \
--length 31
```

## Create a NAT gateway resource

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a global Azure NAT gateway with [az network nat gateway create](#) named **myNATgateway**. The command uses both the public IP address **myPublicIP** and the public IP prefix **myPublicIPprefix**. The command changes the idle timeout to **10** minutes.

```
az network nat gateway create \
--resource-group myResourceGroupNAT \
--name myNATgateway \
--public-ip-addresses myPublicIP \
--public-ip-prefixes myPublicIPprefix \
--idle-timeout 10
```

At this point, the NAT gateway is functional and all that is missing is to configure which subnets of a virtual network should use it.

## Configure virtual network

Before you deploy a VM and can use your NAT gateway, we need to create the virtual network.

Create a virtual network named **myVnet** with a subnet named **mySubnet** in the **myResourceGroupNAT** using [az network vnet create](#). The IP address space for the virtual network is **192.168.0.0/16**. The subnet within the virtual network is **192.168.0.0/24**.

```
az network vnet create \
--resource-group myResourceGroupNAT \
--location eastus2 \
--name myVnet \
--address-prefix 192.168.0.0/16 \
--subnet-name mySubnet \
--subnet-prefix 192.168.0.0/24
```

## Configure NAT service for source subnet

We'll configure the source subnet **mySubnet** in virtual network **myVnet** to use a specific NAT gateway resource **myNATgateway** with [az network vnet subnet update](#). This command will activate the NAT service on the specified subnet.

```
az network vnet subnet update \
--resource-group myResourceGroupNAT \
--vnet-name myVnet \
--name mySubnet \
--nat-gateway myNATgateway
```

All outbound traffic to Internet destinations is now using the NAT gateway. It's not necessary to configure a UDR.

## Create a VM to use the NAT service

We'll now create a VM to use the NAT service. This VM has a public IP to use as an instance-level Public IP to allow you to access the VM. NAT service is flow direction aware and will replace the default Internet destination in your subnet. The VM's public IP address won't be used for outbound connections.

### Create public IP for source VM

We create a public IP to be used to access the VM. Use [az network public-ip create](#) to create a public IP address resource named **myPublicIPVM** in **myResourceGroupNAT**.

```
az network public-ip create \
--resource-group myResourceGroupNAT \
--name myPublicIPVM \
--sku standard
```

### Create an NSG for VM

Because Standard Public IP addresses are 'secure by default', we need to create an NSG to allow inbound access for ssh access. Use [az network nsg create](#) to create an NSG resource named **myNSG** in **myResourceGroupNAT**.

```
az network nsg create \
--resource-group myResourceGroupNAT \
--name myNSG
```

### Expose SSH endpoint on source VM

We create a rule in the NSG for SSH access to the source vm. Use [az network nsg rule create](#) to create an NSG rule named **ssh** in the NSG named **myNSG** in **myResourceGroupNAT**.

```
az network nsg rule create \
--resource-group myResourceGroupNAT \
--nsg-name myNSG \
--priority 100 \
--name ssh \
--description "SSH access" \
--access allow \
--protocol tcp \
--direction inbound \
--destination-port-ranges 22
```

### Create NIC for VM

Create a network interface with [az network nic create](#) and associate with the Public IP address and the network security group.

```
az network nic create \
--resource-group myResourceGroupNAT \
--name myNic \
--vnet-name myVnet \
--subnet mySubnet \
--public-ip-address myPublicIPVM \
--network-security-group myNSG
```

## Create VM

Create the virtual machine with [az vm create](#). We generate ssh keys for this VM and store the private key to use later.

```
az vm create \
--resource-group myResourceGroupNAT \
--name myVM \
--nics myNic \
--image UbuntuLTS \
--generate-ssh-keys
```

Wait for the VM to deploy then continue with the rest of the steps.

## Discover the IP address of the VM

First we need to discover the IP address of the VM you've created. To retrieve the public IP address of the VM, use [az network public-ip show](#).

```
az network public-ip show \
--resource-group myResourceGroupNAT \
--name myPublicIPVM \
--query [ipAddress] \
--output tsv
```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it to access the VM.

## Sign in to VM

The SSH credentials should be stored in your Cloud Shell from the previous operation. Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <ip-address-destination>
```

You're now ready to use the NAT service.

## Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group and all resources contained within.

```
az group delete \
--name myResourceGroupNAT
```

## Next steps

In this tutorial, you created a NAT gateway and a VM to use it.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is addressed by adding additional public IP address resources or public IP prefix resources or both.

- Learn about [Azure Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).
- Quickstart for deploying [NAT gateway resource using Azure CLI](#).
- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

## Quickstart: Create a NAT gateway - ARM template

2/18/2021 • 5 minutes to read • [Edit Online](#)

Get started with Virtual Network NAT by using an Azure Resource Manager template (ARM template). This template deploys a virtual network, a NAT gateway resource, and Ubuntu virtual machine. The Ubuntu virtual machine is deployed to a subnet that is associated with the NAT gateway resource.

An [ARM template](#) is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. In declarative syntax, you describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template will open in the Azure portal.



## Prerequisites

If you don't have an Azure subscription, create a [free account](#) before you begin.

## Review the template

The template used in this quickstart is from [Azure Quickstart Templates](#).

This template is configured to create a:

- Virtual network
  - NAT gateway resource
  - Ubuntu virtual machine

The Ubuntu VM is deployed to a subnet that's associated with the NAT gateway resource.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "vmname": {
 "defaultValue": "myVM",
 "type": "String",
 "metadata": {
 "description": "Name of the virtual machine"
 }
 },
 "vmsize": {
 "defaultValue": "Standard_D2s_v3",
 "type": "String",
 "metadata": {
 "description": "Size of the virtual machine"
 }
 },
 "vnetname": {
 "defaultValue": "myVnet",
 "type": "String",
 "metadata": {
 "description": "Name of the virtual network"
 }
 }
 }
}
```

```
},
"subnetname": {
 "defaultValue": "mySubnet",
 "type": "String",
 "metadata": {
 "description": "Name of the subnet for virtual network"
 }
},
"vnetaddressspace": {
 "defaultValue": "192.168.0.0/16",
 "type": "String",
 "metadata": {
 "description": "Address space for virtual network"
 }
},
"vnetsubnetprefix": {
 "defaultValue": "192.168.0.0/24",
 "type": "String",
 "metadata": {
 "description": "Subnet prefix for virtual network"
 }
},
"natgatewayname": {
 "defaultValue": "myNATgateway",
 "type": "String",
 "metadata": {
 "description": "Name of the NAT gateway"
 }
},
"networkinterfacename": {
 "defaultValue": "myvmNIC",
 "type": "String",
 "metadata": {
 "description": "Name of the virtual machine nic"
 }
},
"publicipname": {
 "defaultValue": "myPublicIP",
 "type": "String",
 "metadata": {
 "description": "Name of the NAT gateway public IP"
 }
},
"nsgname": {
 "defaultValue": "myVMnsg",
 "type": "String",
 "metadata": {
 "description": "Name of the virtual machine NSG"
 }
},
"publicipvmname": {
 "defaultValue": "myPublicIPVM",
 "type": "String",
 "metadata": {
 "description": "Name of the virtual machine public IP"
 }
},
"publicipprefixname": {
 "defaultValue": "myPublicIPPPrefix",
 "type": "String",
 "metadata": {
 "description": "Name of the NAT gateway public IP"
 }
},
"adminusername": {
 "type": "String",
 "metadata": {
 "description": "Administrator username for virtual machine"
 }
}
```

```
},
"adminpassword": {
 "type": "secureString",
 "metadata": {
 "description": "Administrator password for virtual machine"
 }
},
"location": {
 "defaultValue": "[resourceGroup().location]",
 "type": "String",
 "metadata": {
 "description": "Name of resource group"
 }
}
},
"variables": {},
"resources": [
{
 "type": "Microsoft.Network/networkSecurityGroups",
 "apiVersion": "2020-06-01",
 "name": "[parameters('nsgname')]",
 "location": "[parameters('location')]",
 "properties": {
 "securityRules": [
 {
 "name": "SSH",
 "properties": {
 "protocol": "TCP",
 "sourcePortRange": "*",
 "destinationPortRange": "22",
 "sourceAddressPrefix": "*",
 "destinationAddressPrefix": "*",
 "access": "Allow",
 "priority": 300,
 "direction": "Inbound"
 }
 }
]
 }
},
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2020-06-01",
 "name": "[parameters('publicipname')]",
 "location": "[parameters('location')]",
 "sku": {
 "name": "Standard"
 },
 "properties": {
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Static",
 "idleTimeoutInMinutes": 4
 }
},
{
 "type": "Microsoft.Network/publicIPAddresses",
 "apiVersion": "2020-06-01",
 "name": "[parameters('publicipvmname')]",
 "location": "[parameters('location')]",
 "sku": {
 "name": "Standard"
 },
 "properties": {
 "publicIPAddressVersion": "IPv4",
 "publicIPAllocationMethod": "Static",
 "idleTimeoutInMinutes": 4
 }
},
```

```

 "type": "Microsoft.Network/publicIPPrefixes",
 "apiVersion": "2020-06-01",
 "name": "[parameters('publicipprefixname')]",
 "location": "[parameters('location')]",
 "sku": {
 "name": "Standard"
 },
 "properties": {
 "prefixLength": 31,
 "publicIPAddressVersion": "IPv4"
 }
 },
 {
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2020-06-01",
 "name": "[parameters('vmname')]",
 "location": "[parameters('location')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/networkInterfaces', parameters('networkinterfacename'))]"
],
 "properties": {
 "hardwareProfile": {
 "vmSize": "[parameters('vmsize')]"
 },
 "storageProfile": {
 "imageReference": {
 "publisher": "Canonical",
 "offer": "UbuntuServer",
 "sku": "18.04-LTS",
 "version": "latest"
 },
 "osDisk": {
 "osType": "Linux",
 "name": "[concat(parameters('vmname'), '_disk1')]",
 "createOption": "FromImage",
 "caching": "ReadWrite",
 "managedDisk": {
 "storageAccountType": "Premium_LRS"
 },
 "diskSizeGB": 30
 }
 },
 "osProfile": {
 "computerName": "[parameters('vmname')]",
 "adminUsername": "[parameters('adminusername')]",
 "adminPassword": "[parameters('adminpassword')]",
 "linuxConfiguration": {
 "disablePasswordAuthentication": false,
 "provisionVMAgent": true
 },
 "allowExtensionOperations": true
 },
 "networkProfile": {
 "networkInterfaces": [
 {
 "id": "[resourceId('Microsoft.Network/networkInterfaces',
parameters('networkinterfacename'))]"
 }
]
 }
 }
 },
 {
 "type": "Microsoft.Network/virtualNetworks",
 "apiVersion": "2020-06-01",
 "name": "[parameters('vnetname')]",
 "location": "[parameters('location')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
]
 }
]

```

```

],
"properties": {
 "addressSpace": {
 "addressPrefixes": [
 "[parameters('vnetaddressspace')]"
]
 },
 "subnets": [
 {
 "name": "[parameters('subnetname')]",
 "properties": {
 "addressPrefix": "[parameters('vnetsubnetprefix')]",
 "natGateway": {
 "id": "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
 },
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
 }
],
 "enableDdosProtection": false,
 "enableVmProtection": false
}
},
{
 "type": "Microsoft.Network/natGateways",
 "apiVersion": "2020-06-01",
 "name": "[parameters('natgatewayname')]",
 "location": "[parameters('location')]",
 "sku": {
 "name": "Standard"
 },
 "dependsOn": [
 "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicipname'))]",
 "[resourceId('Microsoft.Network/publicIPPrefixes', parameters('publicipprefixname'))]"
],
 "properties": {
 "idleTimeoutInMinutes": 4,
 "publicIpAddresses": [
 {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicipname'))]"
 }
],
 "publicIpPrefixes": [
 {
 "id": "[resourceId('Microsoft.Network/publicIPPrefixes', parameters('publicipprefixname'))]"
 }
]
 }
},
{
 "type": "Microsoft.Network/virtualNetworks/subnets",
 "apiVersion": "2020-05-01",
 "name": "[concat(parameters('vnetname'), '/mySubnet')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/virtualNetworks', parameters('vnetname'))]",
 "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
],
 "properties": {
 "addressPrefix": "[parameters('vnetsubnetprefix')]",
 "natGateway": {
 "id": "[resourceId('Microsoft.Network/natGateways', parameters('natgatewayname'))]"
 },
 "privateEndpointNetworkPolicies": "Enabled",
 "privateLinkServiceNetworkPolicies": "Enabled"
 }
},
{
 "type": "Microsoft.Network/networkInterfaces",

```

```

 "apiVersion": "2020-06-01",
 "name": "[parameters('networkinterfacename')]",
 "location": "[parameters('location')]",
 "dependsOn": [
 "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicipvmname'))]",
 "[resourceId('Microsoft.Network/virtualNetworks/subnets', parameters('vnetname'), 'mySubnet')]",
 "[resourceId('Microsoft.Network/networkSecurityGroups', parameters('nsgname'))]"
],
 "properties": {
 "ipConfigurations": [
 {
 "name": "ipconfig1",
 "properties": {
 "privateIPAddress": "192.168.0.4",
 "privateIPAllocationMethod": "Dynamic",
 "publicIPAddress": {
 "id": "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicipvmname'))]"
 },
 "subnet": {
 "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets', parameters('vnetname'), 'mySubnet')]"
 },
 "primary": true,
 "privateIPAddressVersion": "IPv4"
 }
 }
],
 "enableAcceleratedNetworking": false,
 "enableIPForwarding": false,
 "networkSecurityGroup": {
 "id": "[resourceId('Microsoft.Network/networkSecurityGroups', parameters('nsgname'))]"
 }
 }
}
]
}

```

Nine Azure resources are defined in the template:

- **Microsoft.Network/networkSecurityGroups**: Creates a network security group.
- **Microsoft.Network/networkSecurityGroups/securityRules**: Creates a security rule.
- **Microsoft.Network/publicIPAddresses**: Creates a public IP address.
- **Microsoft.Network/publicIPPrefixes**: Creates a public IP prefix.
- **Microsoft.Compute/virtualMachines**: Creates a virtual machine.
- **Microsoft.Network/virtualNetworks**: Creates a virtual network.
- **Microsoft.Network/natGateways**: Creates a NAT gateway resource.
- **Microsoft.Network/virtualNetworks/subnets**: Creates a virtual network subnet.
- **Microsoft.Network/networkInterfaces**: Creates a network interface.

## Deploy the template

Azure CLI

```
read -p "Enter the location (i.e. westcentralus): " location
resourceGroupName="myResourceGroupNAT"
templateUri="https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-nat-gateway-1-vm/azuredeploy.json"

az group create \
--name $resourceGroupName \
--location $location

az deployment group create \
--resource-group $resourceGroupName \
--template-uri $templateUri
```

## Azure PowerShell

```
$location = Read-Host -Prompt "Enter the location (i.e. westcentralus)"
$templateUri = "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-nat-gateway-1-vm/azuredeploy.json"

$resourceGroupName = "myResourceGroupNAT"

New-AzResourceGroup -Name $resourceGroupName -Location $location
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateUri $templateUri
```

## Azure portal



## Review deployed resources

1. Sign in to the [Azure portal](#).
2. Select **Resource groups** from the left pane.
3. Select the resource group that you created in the previous section. The default resource group name is **myResourceGroupNAT**
4. Verify the following resources were created in the resource group:

The screenshot shows the Azure portal interface for the resource group 'myResourceGroupNAT'. On the left, there's a navigation sidebar with sections like Overview, Activity log, Access control (IAM), Tags, Events, Settings, Cost Management, Monitoring, and more. The main area displays the following details:

- Subscription (change):** Contoso Subscription, Subscription ID: c4bccf53-b501-4179-983c-cab9884d32ac
- Deployments:** 1 Succeeded
- Tags (change):** Click here to add tags

Below these, there's a search bar and filter options (Type == all, Location == all, Add filter). A table lists 9 records:

Name	Type	Location
myNATgateway	NAT gateway	East US 2
myPublicIP	Public IP address	East US 2
myPublicIPPrefix	Public IP Prefix	East US 2
myPublicIPVPM	Public IP address	East US 2
myVM	Virtual machine	East US 2
myVM_disk1	Disk	East US 2
myvmNIC	Network interface	East US 2
myVMsg	Network security group	East US 2
myVNet	Virtual network	East US 2

Pagination at the bottom shows Page 1 of 1.

## Clean up resources

### Azure CLI

When no longer needed, you can use the [az group delete](#) command to remove the resource group and all resources contained within.

```
az group delete \
--name myResourceGroupNAT
```

### Azure PowerShell

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group and all resources contained within.

```
Remove-AzResourceGroup -Name myResourceGroupNAT
```

### Azure portal

When no longer needed, delete the resource group, NAT gateway, and all related resources. Select the resource group **myResourceGroupNAT** that contains the NAT gateway, and then select **Delete**.

## Next steps

In this quickstart, you created a:

- NAT gateway resource
- Virtual network
- Ubuntu virtual machine

The virtual machine is deployed to a virtual network subnet associated with the NAT gateway.

To learn more about Virtual Network NAT and Azure Resource Manager, continue to the articles below.

- Read an [Overview of Virtual Network NAT](#)
- Read about the [NAT Gateway resource](#)
- Learn more about [Azure Resource Manager](#)

# Tutorial: Create a NAT gateway using Azure PowerShell and test the NAT service

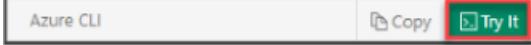
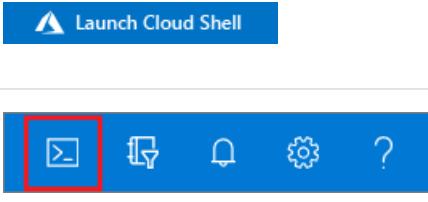
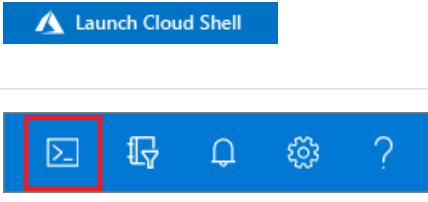
2/18/2021 • 13 minutes to read • [Edit Online](#)

In this tutorial, you'll create a NAT gateway to provide outbound connectivity for virtual machines in Azure. To test the NAT gateway, you deploy a source and destination virtual machine. You'll test the NAT gateway by making outbound connections to a public IP address. These connections will come from the source to the destination virtual machine. This tutorial deploys source and destination in two different virtual networks in the same resource group for simplicity.

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the <a href="#">Azure portal</a> .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

You can complete this tutorial using Azure Cloud Shell or run the respective commands locally. If you have never used Azure Cloud Shell, you should [sign in now](#).

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Create a resource group

Create a resource group with [az group create](#). An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named **myResourceGroupNAT** in the **eastus2** location:

```
$rgname = 'myResourceGroupNAT'
$loc = 'eastus2'

$rg = New-AzResourceGroup -Name $rgname -Location $loc
```

## Create the NAT gateway

### Create a public IP address

To access the Internet, you need one or more public IP addresses for the NAT gateway. Use [New-AzPublicIpAddress](#) to create a public IP address resource named **myPublicIPsource** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$publicIPsource** for later use.

```
$pipname = 'myPublicIPsource'
$alm = 'Static'
$sku = 'Standard'

$publicIPsource =
New-AzPublicIpAddress -Name $pipname -ResourceGroupName $rg.ResourceGroupName -AllocationMethod $alm -Sku
$sku -Location $rg.Location
```

### Create a public IP prefix

Use [New-AzPublicIpPrefix](#) to create a public IP prefix resource named **myPublicIPprefixsource** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$publicIPPrefixsource** for later use.

```
$prefixname = 'mypublicIPprefixsource'

$publicIPPrefixsource =
New-AzPublicIpPrefix -Name $prefixname -ResourceGroupName $rg.ResourceGroupName -PrefixLength 31 -Location
$rg.Location
```

### Create a NAT gateway resource

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a global Azure NAT gateway with [New-AzNatGateway](#). The result of this command will create a gateway resource named **myNatGateway** that uses the public IP address **myPublicIPsource** and the public IP prefix **myPublicIPprefixsource**. The idle timeout is set to 10 minutes. The result of this command will be stored in a variable named **\$natGateway** for later use.

```
$sku = 'Standard'
$natname = 'myNATgateway'

$natGateway =
New-AzNatGateway -Name $natname -ResourceGroupName $rg.ResourceGroupName -PublicIpAddress $publicIPsource -
PublicIpPrefix $publicIPPrefixsource -Sku $sku -IdleTimeoutInMinutes 10 -Location $rg.Location
```

At this point, the NAT gateway is functional and all that is missing is to configure which subnets of a virtual network should use it.

## Prepare the source for outbound traffic

We'll guide you through setup of a full test environment. You'll set up a test using open-source tools to verify the NAT gateway. We'll start with the source, which will use the NAT gateway we created previously.

### Configure virtual network for source

Create the virtual network and associate the subnet to the gateway.

Create a virtual network named `myVnetsource` with a subnet named `mySubnetsource` using [New-AzVirtualNetworkSubnetConfig](#) in the `myResourceGroupNAT` using [New-AzVirtualNetwork](#). The IP address space for the virtual network is `192.168.0.0/16`. The subnet within the virtual network is `192.168.0.0/24`. The result of the commands will be stored in variables named `$subnetsource` and `$vnetsource` for later use.

```
$subnetname = 'mySubnetsource'
$subnetprefix = '192.168.0.0/24'
$vnetname = 'myVnetsource'
$vnetprefix = '192.168.0.0/16'

$subnetsource =
New-AzVirtualNetworkSubnetConfig -Name $subnetname -AddressPrefix $subnetprefix -NatGateway $natGateway

$vnetsource =
New-AzVirtualNetwork -Name $vnetname -ResourceGroupName $rg.ResourceGroupName -AddressPrefix $vnetprefix -
Subnet $subnetsource -Location $rg.Location
```

All outbound traffic to Internet destinations is now using the NAT service. It isn't necessary to configure a UDR.

Before we can test the NAT gateway, we need to create a source VM. We'll assign a public IP address resource as an instance-level Public IP to access this VM from the outside. This address is only used to access it for the test. We'll demonstrate how the NAT service takes precedence over other outbound options.

You could also create this VM without a public IP and create another VM to use as a jumpbox without a public IP as an exercise.

### Create public IP for source VM

We create a public IP to be used to access the VM. Use [New-AzPublicIpAddress](#) to create a public IP address resource named `myPublicIPVM` in `myResourceGroupNAT`. The result of this command will be stored in a variable named `$publicIPsourceVM` for later use.

```

$sku = 'Standard'
$ipvmname = 'myPublicIpSourceVM'
$alm = 'Static'

$publicIpSourceVM =
New-AzPublicIpAddress -Name $ipvmname -ResourceGroupName $rg.ResourceGroupName -AllocationMethod $alm -sku
$sku -Location $rg.Location

```

## Create an NSG and expose SSH endpoint for VM

Because Standard Public IP addresses are 'secure by default', we create an NSG to allow inbound access for ssh. NAT service is flow direction aware. This NSG won't be used for outbound once NAT gateway is configured on the same subnet. Use [New-AzNetworkSecurityGroup](#) to create an NSG resource named **myNSGsource**. Use [New-AzNetworkSecurityRuleConfig](#) to create an NSG rule for SSH access named **ssh** in **myResourceGroupNAT**. The result of this command will be stored in variable named **\$nsgsource** for later use.

```

$rnm = 'ssh'
$rdsc = 'SSH access'
$acc = 'Allow'
$prt = 'Tcp'
$dir = 'Inbound'
$nsnm = 'myNSGsource'

$sshrule =
New-AzNetworkSecurityRuleConfig -Name $rnm -Description $rdsc -Access $acc -Protocol $prt -Direction $dir -
Priority 100 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 22

$nsgsource =
New-AzNetworkSecurityGroup -ResourceGroupName $rg.ResourceGroupName -Name $nsnm -SecurityRules $sshrule -
Location $rg.Location

```

## Create NIC for source VM

Create a network interface with [New-AzNetworkInterface](#) named **myNicsource**. This command will associate the Public IP address and the network security group. The result of this command will be stored in a variable named **\$nicsource** for later use.

```

$nin = 'myNicsource'

$nicsource =
New-AzNetworkInterface -ResourceGroupName $rg.ResourceGroupName -Name $nin -NetworkSecurityGroupID
$nsgsource.Id -PublicIPAddressID $publicIPVMsource.Id -SubnetID $vnetsource.Subnets[0].Id -Location
$rg.Location

```

## Create a source VM

### Create SSH key pair

You need an SSH key pair to complete this quickstart. If you already have an SSH key pair, you can skip this step.

Use ssh-keygen to create an SSH key pair.

```
ssh-keygen -t rsa -b 2048
```

For more detailed information on how to create SSH key pairs, including the use of PuTTY, see [How to use SSH](#)

keys with Windows.

If you create the SSH key pair using the Cloud Shell, the key pair is stored in a container image. This [storage account is automatically created](#). Don't delete the storage account, or the file share within, until after you've retrieved your keys.

#### Create VM Configuration

To create a VM in PowerShell, you create a configuration that has settings like the image to use, size, and authentication options. Then the configuration is used to build the VM.

Define the SSH credentials, OS information, and VM size. In this example, the SSH key is stored in `~/ssh/id_rsa.pub`.

```
Define a credential object

$securePassword =
ConvertTo-SecureString ' ' -AsPlainText -Force
$cred =
New-Object System.Management.Automation.PSCredential ("azureuser", $securePassword)

Create a virtual machine configuration
$vnn = 'myVMsource'
$vms = 'Standard_DS1_v2'
$pub = 'Canonical'
$off = 'UbuntuServer'
$skus = '18.04-LTS'
$ver = 'latest'

$vmConfigsource =
New-AzVMConfig -VMName $vnn -VMSize $vms

Set-AzVMOperatingSystem -VM $vmConfigsource -Linux -ComputerName $vnn -Credential $cred -
DisablePasswordAuthentication

Set-AzVMSourceImage -VM $vmConfigsource -PublisherName $pub -Offer $off -Skus $skus -Version $ver

Add-AzVMNetworkInterface -VM $vmConfigsource -Id $nicsource.Id

Configure the SSH key

$sshPublicKey = cat ~/ssh/id_rsa.pub

Add-AzVMSshPublicKey -VM $vmConfigsource -KeyData $sshPublicKey -Path "/home/azureuser/.ssh/authorized_keys"
```

Combine the configuration definitions to create a VM named `myVMsource` with [New-AzVM](#) in `myResourceGroupNAT`.

```
New-AzVM -ResourceGroupName $rg.ResourceGroupName -VM $vmConfigsource -Location $rg.Location
```

While the command will return immediately, it may take a few minutes for the VM to get deployed.

## Prepare destination for outbound traffic

We'll now create a destination for the outbound traffic translated by the NAT service to allow you to test it.

#### Configure virtual network for destination

We need to create a virtual network where the destination virtual machine will be. These commands are the same steps as for the source VM. Small changes have been added to expose the destination endpoint.

Create a virtual network named **myVnetdestination** with a subnet named **mySubnetdestination** using [New-AzVirtualNetworkSubnetConfig](#) in the **myResourceGroupNAT** using [New-AzVirtualNetwork](#). The IP address space for the virtual network is **192.168.0.0/16**. The subnet within the virtual network is **192.168.0.0/24**. The result of the commands will be stored in variables named **\$subnetdestination** and **\$vnetdestination** for later use.

```
$sbdn = 'mySubnetdestination'
$spfx = '192.168.0.0/24'
$vdn = 'myVnetdestination'
$vpfx = '192.168.0.0/16'

$subnetdestination =
New-AzVirtualNetworkSubnetConfig -Name $sbdn -AddressPrefix $spfx

$vnetdestination =
New-AzVirtualNetwork -Name $vdn -ResourceGroupName $rg.ResourceGroupName -AddressPrefix $vpfx -Subnet
$subnetdestination -Location $rg.Location
```

### Create public IP for destination VM

We create a public IP to be used to access the destination VM. Use [New-AzPublicIpAddress](#) to create a public IP address resource named **myPublicIPdestinationVM** in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$publicIpdestinationVM** for later use.

```
$sku = 'Standard'
$all = 'Static'
$pipd = 'myPublicIPdestinationVM'

$publicIpdestinationVM =
New-AzPublicIpAddress -Name $pipd -ResourceGroupName $rg.ResourceGroupName -AllocationMethod $all -Sku $sku
-Location $rg.Location
```

### Create an NSG and expose SSH and HTTP endpoint for VM

Standard Public IP addresses are 'secure by default', we create an NSG to allow inbound access for ssh. Use [New-AzNetworkSecurityGroup](#) to create an NSG resource named **myNSGdestination**. Use [New-AzNetworkSecurityRuleConfig](#) to create an NSG rule for SSH access named **ssh**. Use [New-AzNetworkSecurityRuleConfig](#) to create an NSG rule for HTTP access named **http**. Both rules will be created in **myResourceGroupNAT**. The result of this command will be stored in a variable named **\$nsgdestination** for later use.

```

$snm = 'ssh'
$sdsc = 'SSH access'
$acc = 'Allow'
$pprt = 'Tcp'
$dir = 'Inbound'
$hnm = 'http'
$hdsc = 'HTTP access'
$nsnm = 'myNSGdestination'

$sshrule =
New-AzNetworkSecurityRuleConfig -Name $snm -Description $sdsc -Access $acc -Protocol $prt -Direction $dir -
Priority 100 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 22

$httprule =
New-AzNetworkSecurityRuleConfig -Name $hnm -Description $hdsc -Access $acc -Protocol $prt -Direction $dir -
Priority 101 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 80

$nsgdestination =
New-AzNetworkSecurityGroup -ResourceGroupName $rg.ResourceGroupName -Name $nsnm -SecurityRules
$sshrule,$httprule -Location $rg.Location

```

## Create NIC for destination VM

Create a network interface with [New-AzNetworkInterface](#) named `myNicdestination`. This command will associate with the Public IP address and the network security group. The result of this command will be stored in a variable named `$nicdestination` for later use.

```

$nnm = 'myNicdestination'

$nicdestination =
New-AzNetworkInterface -ResourceGroupName $rg.ResourceGroupName -Name $nnm -NetworkSecurityGroupID
$nsgdestination.Id -PublicIPAddressID $publicIPdestinationVM.Id -SubnetID $vnetdestination.Subnets[0].Id -
Location $rg.Location

```

## Create a destination VM

### Create VM Configuration

To create a VM in PowerShell, you create a configuration that has settings like the image to use, size, and authentication options. Then the configuration is used to build the VM.

Define the SSH credentials, OS information, and VM size. In this example, the SSH key is stored in `~/.ssh/id_rsa.pub`.

```

Define a credential object

$securePassword =
ConvertTo-SecureString ' ' -AsPlainText -Force
$cred =
New-Object System.Management.Automation.PSCredential ("azureuser", $securePassword)

Create a virtual machine configuration

$vmd = 'myVMdestination'
$vms = 'Standard_DS1_v2'
$pub = 'Canonical'
$off = 'UbuntuServer'
$skus = '18.04-LTS'
$ver = 'latest'

$vmConfigdestination = New-AzVMConfig -VMName $vmd -VMSize $vms

Set-AzVMOperatingSystem -VM $vmConfigdestination -Linux -ComputerName $vmd -Credential $cred -
DisablePasswordAuthentication

Set-AzVMSourceImage -VM $vmConfigdestination -PublisherName $pub -Offer $off -Skus $skus -Version $ver

Add-AzVMNetworkInterface -VM $vmConfigdestination -Id $nicdestination.Id

Configure the SSH key

$sshPublicKey = cat ~/.ssh/id_rsa.pub

Add-AzVMSshPublicKey -VM $vmConfigdestination -KeyData $sshPublicKey -Path
"/home/azureuser/.ssh/authorized_keys"

```

Combine the configuration definitions to create a VM named **myVMdestination** with [New-AzVM](#) in **myResourceGroupNAT**.

```
New-AzVM -ResourceGroupName $rg.ResourceGroupName -VM $vmConfigdestination -Location $rg.Location
```

While the command will return immediately, it may take a few minutes for the VM to get deployed.

## Prepare a web server and test payload on destination VM

First we need to discover the IP address of the destination VM. To get the public IP address of the VM, use [Get-AzPublicIpAddress](#).

```

$pipname = 'myPublicIPdestinationVM'

$destip = Get-AzPublicIpAddress -ResourceGroupName $rg.ResourceGroupName -Name $pipname | selectIpAddress

$destip

```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the destination virtual machine.

## Sign in to destination VM

The SSH credentials should be stored in your Cloud Shell from the previous operation. Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh azureuser@$destip
```

Copy and paste the following commands once you've logged in.

```
sudo apt -y update && \
sudo apt -y upgrade && \
sudo apt -y install nginx && \
sudo ln -sf /dev/null /var/log/nginx/access.log && \
sudo touch /var/www/html/index.html && \
sudo rm /var/www/html/index.nginx-debian.html && \
sudo dd if=/dev/zero of=/var/www/html/100k bs=1024 count=100
```

These commands will update your virtual machine, install nginx, and create a 100-KBytes file. This file will be retrieved from the source VM using the NAT service.

Close the SSH session with the destination VM.

## Prepare test on source VM

First we need to discover the IP address of the source VM. To get the public IP address of the VM, use [Get-AzPublicIpAddress](#).

```
$pipname = 'myPublicIPsourceVM'

$srcip = Get-AzPublicIpAddress -ResourceGroupName $rg.ResourceGroupName -Name $pipname | select IPAddress

$srcip
```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the source virtual machine.

## Log into source VM

Again, the SSH credentials are stored in Cloud Shell. Open a new tab for [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh azureuser@$srcip
```

Copy and paste the following commands to prepare for testing the NAT service.

```
sudo apt -y update && \
sudo apt -y upgrade && \
sudo apt install -y nload golang && \
echo 'export GOPATH=${HOME}/go' >> .bashrc && \
echo 'export PATH=${PATH}: ${GOPATH}/bin' >> .bashrc && \
. ~/.bashrc &&
go get -u github.com/rakyll/hey
```

These commands will update your virtual machine, install go, install [hey](#) from GitHub, and update your shell environment.

You're now ready to test NAT service.

## Validate NAT service

While logged into the source VM, you can use [curl](#) and [hey](#) to generate requests to the destination IP address.

Use curl to retrieve the 100-KBytes file. Replace <ip-address-destination> in the example below with the destination IP address you have previously copied. The --output parameter indicates that the retrieved file will be discarded.

```
curl http://<ip-address-destination>/100k --output /dev/null
```

You can also generate a series of requests using [hey](#). Again, replace <ip-address-destination> with the destination IP address you have previously copied.

```
hey -n 100 -c 10 -t 30 --disable-keepalive http://<ip-address-destination>/100k
```

This command will generate 100 requests, 10 concurrently, with a timeout of 30 seconds. The TCP connection won't be reused. Each request will retrieve 100 Kbytes. At the end of the run, [hey](#) will report some statistics about how well the NAT service did.

## Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group and all resources contained within.

```
Remove-AzResourceGroup -Name $rg.ResourceGroupName
```

## Next steps

In this tutorial, you created a NAT gateway, created a source and destination VM, and then tested the NAT gateway.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is easily addressed by adding additional public IP address resources or public IP prefix resources or both.

- Learn about [Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).

- Quickstart for deploying [NAT gateway resource using Azure CLI](#).
- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

# Tutorial: Create a NAT gateway using Azure CLI and test the NAT service

2/18/2021 • 11 minutes to read • [Edit Online](#)

In this tutorial, you'll create a NAT gateway to provide outbound connectivity for virtual machines in Azure. To test the NAT gateway, you deploy a source and destination virtual machine. You'll test the NAT gateway by making outbound connections to a public IP address. These connections will come from the source to the destination virtual machine. This tutorial deploys source and destination in two different virtual networks in the same resource group for simplicity only.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

## Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#).  
[Launch Cloud Shell](#)
- If you prefer, [install](#) the Azure CLI to run CLI reference commands.
  - If you're using a local installation, sign in to the Azure CLI by using the `az login` command. To finish the authentication process, follow the steps displayed in your terminal. For additional sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install Azure CLI extensions on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run `az version` to find the version and dependent libraries that are installed. To upgrade to the latest version, run `az upgrade`.
- This article requires version 2.0.71 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

## Create a resource group

Create a resource group with `az group create`. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named `myResourceGroupNAT` in the `eastus2` location:

```
az group create \
--name myResourceGroupNAT \
--location eastus2
```

## Create the NAT Gateway

### Create a public IP address

To access the public Internet, you need one or more public IP addresses for the NAT gateway. Use `az network public-ip create` to create a public IP address resource named `myPublicIPsource` in `myResourceGroupNAT`.

```
az network public-ip create \
--resource-group myResourceGroupNAT \
--name myPublicIPsource \
--sku standard
```

## Create a public IP prefix

You can use one or more public IP address resources, public IP prefixes or both with NAT gateway. We'll add a public IP prefix resource to this scenario to demonstrate. Use [az network public-ip prefix create](#) to create a public IP prefix resource named **myPublicIPprefixsource** in **myResourceGroupNAT**.

```
az network public-ip prefix create \
--resource-group myResourceGroupNAT \
--name myPublicIPprefixsource \
--length 31
```

## Create a NAT gateway resource

This section details how you can create and configure the following components of the NAT service using the NAT gateway resource:

- A public IP pool and public IP prefix to use for outbound flows translated by the NAT gateway resource.
- Change the idle timeout from the default of 4 minutes to 10 minutes.

Create a global Azure NAT gateway with [az network nat gateway create](#) named **myNATgateway**. The command uses both the public IP address **myPublicIP** and the public IP prefix **myPublicIPprefix**. The command also changes the idle timeout to 10 minutes.

```
az network nat gateway create \
--resource-group myResourceGroupNAT \
--name myNATgateway \
--public-ip-addresses myPublicIPsource \
--public-ip-prefixes myPublicIPprefixsource \
--idle-timeout 10
```

At this point, the NAT gateway is functional and all that is missing is to configure which subnets of a virtual network should use it.

## Prepare the source for outbound traffic

We'll guide you through setup of a full test environment. You'll set up a test using open-source tools to verify the NAT gateway. We'll start with the source, which will use the NAT gateway we created previously.

### Configure virtual network for source

Before you deploy a VM and can test your NAT gateway, we need to create the virtual network.

Create a virtual network named **myVnetsource** with a subnet named **mySubnetsource** in the **myResourceGroupNAT** using [az network Microsoft Azure Virtual Network create](#). The IP address space for the virtual network is **192.168.0.0/16**. The subnet within the virtual network is **192.168.0.0/24**.

```
az network vnet create \
--resource-group myResourceGroupNAT \
--name myVnetsource \
--address-prefix 192.168.0.0/16 \
--subnet-name mySubnetsource \
--subnet-prefix 192.168.0.0/24
```

## Configure NAT service for source subnet

Configure the source subnet **mySubnetsource** in virtual network **myVnetsource** to use a specific NAT gateway resource **myNATgateway** with [az network Microsoft Azure Virtual Network subnet update](#). This command will activate the NAT service on the specified subnet.

```
az network vnet subnet update \
--resource-group myResourceGroupNAT \
--vnet-name myVnetsource \
--name mySubnetsource \
--nat-gateway myNATgateway
```

All outbound traffic to Internet destinations is now using the NAT service. It's not necessary to configure a UDR.

Before we can test the NAT gateway, we need to create a source VM. We'll assign a public IP address resource as an instance-level public IP to access this VM from the outside. This address is only used to access it for the test. We'll demonstrate how the NAT service takes precedence over other outbound options.

You could also create this VM without a public IP and create another VM to use as a jumpbox without a public IP as an exercise.

## Create public IP for source VM

We create a public IP to be used to access the source VM. Use [az network public-ip create](#) to create a public IP address resource named **myPublicIPsourceVM** in **myResourceGroupNAT**.

```
az network public-ip create \
--resource-group myResourceGroupNAT \
--name myPublicIPsourceVM \
--sku standard
```

## Create an NSG for source VM

Because Standard public IP addresses are 'secure by default', we need to create an NSG to allow inbound access for ssh access. Azure NAT service is flow direction aware. This NSG won't be used for outbound once the NAT gateway is configured on the same subnet. Use [az network nsg create](#) to create an NSG resource named **myNSGsource** in **myResourceGroupNAT**.

```
az network nsg create \
--resource-group myResourceGroupNAT \
--name myNSGsource
```

## Expose SSH endpoint on source VM

We create a rule in the NSG for SSH access to the source vm. Use [az network nsg rule create](#) to create an NSG rule named **ssh**. This rule will be created in the NSG named **myNSGsource** in the resource group **myResourceGroupNAT**.

```
az network nsg rule create \
--resource-group myResourceGroupNAT \
--nsg-name myNSGsource \
--priority 100 \
--name ssh \
--description "SSH access" \
--access allow \
--protocol tcp \
--direction inbound \
--destination-port-ranges 22
```

## Create NIC for source VM

Create a network interface with [az network nic create](#) and associate with the public IP address and the network security group.

```
az network nic create \
--resource-group myResourceGroupNAT \
--name myNicsource \
--vnet-name myVnetsource \
--subnet mySubnetsource \
--public-ip-address myPublicIPSourceVM \
--network-security-group myNSGsource
```

## Create a source VM

Create the virtual machine with [az vm create](#). We generate ssh keys for this VM and store the private key to use later.

```
az vm create \
--resource-group myResourceGroupNAT \
--name myVMsource \
--nics myNicsource \
--image UbuntuLTS \
--generate-ssh-keys \
--no-wait
```

While the command will return immediately, it may take a few minutes for the VM to get deployed.

## Prepare destination for outbound traffic

We'll now create a destination for the outbound traffic translated by the NAT service to allow you to test it.

### Configure virtual network for destination

We need to create a virtual network where the destination virtual machine will be. These commands are the same steps as for the source VM with small changes to expose the destination endpoint.

Create a virtual network named **myVnetdestination** with a subnet named **mySubnetdestination** in the **myResourceGroupNAT** using [az network Microsoft Azure Virtual Network create](#). The IP address space for the virtual network is **192.168.0.0/16**. The subnet within the virtual network is **192.168.0.0/24**.

```
az network vnet create \
--resource-group myResourceGroupNAT \
--name myVnetdestination \
--address-prefix 192.168.0.0/16 \
--subnet-name mySubnetdestination \
--subnet-prefix 192.168.0.0/24
```

## Create public IP for destination VM

We create a public IP to be used to access the source VM. Use [az network public-ip create](#) to create a public IP address resource named **myPublicIPdestinationVM** in **myResourceGroupNAT**.

```
az network public-ip create \
--resource-group myResourceGroupNAT \
--name myPublicIPdestinationVM \
--sku standard
```

## Create an NSG for destination VM

Standard Public IP addresses are 'secure by default', you'll need to create an NSG to allow inbound access for ssh. The Azure NAT service is flow direction aware. This NSG won't be used for outbound once the NAT gateway is configured on the same subnet. Use [az network nsg create](#) to create an NSG resource named **myNSGdestination** in **myResourceGroupNAT**.

```
az network nsg create \
--resource-group myResourceGroupNAT \
--name myNSGdestination
```

## Expose SSH endpoint on destination VM

We create a rule in the NSG for SSH access to the destination vm. Use [az network nsg rule create](#) to create an NSG rule named **ssh**. This rule will be created in the NSG named **myNSGdestination** in the resource group **myResourceGroupNAT**.

```
az network nsg rule create \
--resource-group myResourceGroupNAT \
--nsg-name myNSGdestination \
--priority 100 \
--name ssh \
--description "SSH access" \
--access allow \
--protocol tcp \
--direction inbound \
--destination-port-ranges 22
```

## Expose HTTP endpoint on destination VM

We create a rule in the NSG for HTTP access to the destination vm. Use [az network nsg rule create](#) to create an NSG rule named **http** in the NSG named **myNSGdestination** in **myResourceGroupNAT**.

```
az network nsg rule create \
--resource-group myResourceGroupNAT \
--nsg-name myNSGdestination \
--priority 101 \
--name http \
--description "HTTP access" \
--access allow \
--protocol tcp \
--direction inbound \
--destination-port-ranges 80
```

## Create NIC for destination VM

Create a network interface with [az network nic create](#) and associate with the public IP address **myPublicIPdestinationVM** and the network security group **myNSGdestination**.

```
az network nic create \
--resource-group myResourceGroupNAT \
--name myNicdestination \
--vnet-name myVnetdestination \
--subnet mySubnetdestination \
--public-ip-address myPublicIPdestinationVM \
--network-security-group myNSGdestination
```

## Create a destination VM

Create the virtual machine with [az vm create](#). We generate ssh keys for this VM and store the private key to use later.

```
az vm create \
--resource-group myResourceGroupNAT \
--name myVMdestination \
--nics myNicdestination \
--image UbuntuLTS \
--generate-ssh-keys \
--no-wait
```

While the command will return immediately, it may take a few minutes for the VM to get deployed.

## Prepare a web server and test payload on destination VM

First we need to discover the IP address of the destination VM. To get the public IP address of the destination VM, use [az network public-ip show](#).

```
az network public-ip show \
--resource-group myResourceGroupNAT \
--name myPublicIPdestinationVM \
--query [ipAddress] \
--output tsv
```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the destination virtual machine.

## Sign in to destination VM

The SSH credentials should be stored in your Cloud Shell from the previous operation. Open an [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <ip-address-destination>
```

Copy and paste the following commands once you've signed in.

```
sudo apt -y update && \
sudo apt -y upgrade && \
sudo apt -y install nginx && \
sudo ln -sf /dev/null /var/log/nginx/access.log && \
sudo touch /var/www/html/index.html && \
sudo rm /var/www/html/index.nginx-debian.html && \
sudo dd if=/dev/zero of=/var/www/html/100k bs=1024 count=100
```

These commands will update your virtual machine, install nginx, and create a 100-KBytes file. This file will be retrieved from the source VM using the NAT service.

Close the SSH session with the destination VM.

## Prepare test on source VM

First we need to discover the IP address of the source VM. To get the public IP address of the source VM, use [az network public-ip show](#).

```
az network public-ip show \
--resource-group myResourceGroupNAT \
--name myPublicIPsourceVM \
--query [ipAddress] \
--output tsv
```

### IMPORTANT

Copy the public IP address, and then paste it into a notepad so you can use it in subsequent steps. Indicate this is the source virtual machine.

## Sign in to source VM

Again, the SSH credentials are stored in Cloud Shell. Open a new tab for [Azure Cloud Shell](#) in your browser. Use the IP address retrieved in the previous step to SSH to the virtual machine.

```
ssh <ip-address-source>
```

Copy and paste the following commands to prepare for testing the NAT service.

```
sudo apt -y update && \
sudo apt -y upgrade && \
sudo apt install -y nload golang && \
echo 'export GOPATH=${HOME}/go' >> .bashrc && \
echo 'export PATH=$PATH:${GOPATH}/bin' >> .bashrc && \
. ~/.bashrc &&
go get -u github.com/rakyll/hey
```

This command will update your virtual machine, install go, install [hey](#) from GitHub, and update your shell environment.

You're now ready to test the NAT service.

## Validate NAT service

While logged into the source VM, you can use `curl` and `hey` to generate requests to the destination IP address.

Use `curl` to retrieve the 100-KBytes file. Replace `<ip-address-destination>` in the example below with the destination IP address you have previously copied. The `--output` parameter indicates that the retrieved file will be discarded.

```
curl http://<ip-address-destination>/100k --output /dev/null
```

You can also generate a series of requests using `hey`. Again, replace `<ip-address-destination>` with the destination IP address you have previously copied.

```
hey -n 100 -c 10 -t 30 --disable-keepalive http://<ip-address-destination>/100k
```

This command will generate 100 requests, 10 concurrently, with a timeout of 30 seconds. The TCP connection won't be reused. Each request will retrieve 100 Kbytes. At the end of the run, `hey` will report some statistics about how well the NAT service did.

## Clean up resources

When no longer needed, you can use the `az group delete` command to remove the resource group and all resources contained within.

```
az group delete --name myResourceGroupNAT
```

## Next steps

In this tutorial, you created a NAT gateway, created a source and destination VM, and then tested the NAT gateway.

Review metrics in Azure Monitor to see your NAT service operating. Diagnose issues such as resource exhaustion of available SNAT ports. Resource exhaustion of SNAT ports is easily addressed by adding additional public IP address resources or public IP prefix resources or both.

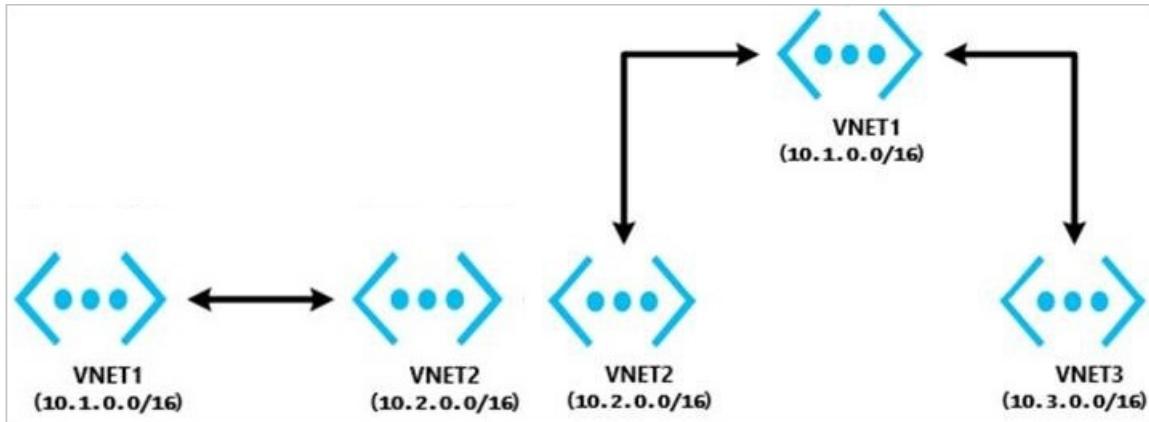
- Learn about [Virtual Network NAT](#)
- Learn about [NAT gateway resource](#).
- Quickstart for deploying [NAT gateway resource using Azure CLI](#).

- Quickstart for deploying [NAT gateway resource using Azure PowerShell](#).
- Quickstart for deploying [NAT gateway resource using Azure portal](#).

# Troubleshoot virtual network peering issues

2/18/2021 • 7 minutes to read • [Edit Online](#)

This troubleshooting guide provides steps to help you resolve most [virtual network peering](#) issues.



## Configure virtual network peering between two virtual networks

Are the virtual networks in the same subscription or in different subscriptions?

### The virtual networks are in the same subscription

To configure virtual network peering for the virtual networks that are in the same subscription, use the methods in the following articles:

- If the virtual networks are in the *same region*, see [Create a peering](#).
- If the virtual networks are in the *different regions*, see [Virtual network peering](#).

#### NOTE

Connectivity doesn't work over global virtual network peering for the following resources:

- Virtual machines (VMs) behind Basic internal load balancer (ILB) SKU
- Redis cache (uses Basic ILB SKU)
- Application gateway (uses Basic ILB SKU)
- Virtual machine scale sets (uses Basic ILB SKU)
- Azure Service Fabric clusters (uses Basic ILB SKU)
- SQL Server Always On (uses Basic ILB SKU)
- Azure App Service Environment for PowerApps (uses Basic ILB SKU)
- Azure API Management (uses Basic ILB SKU)
- Azure Active Directory Domain Services (Azure AD DS) (uses Basic ILB SKU)

For more information, see the [requirements and constraints](#) of global peering.

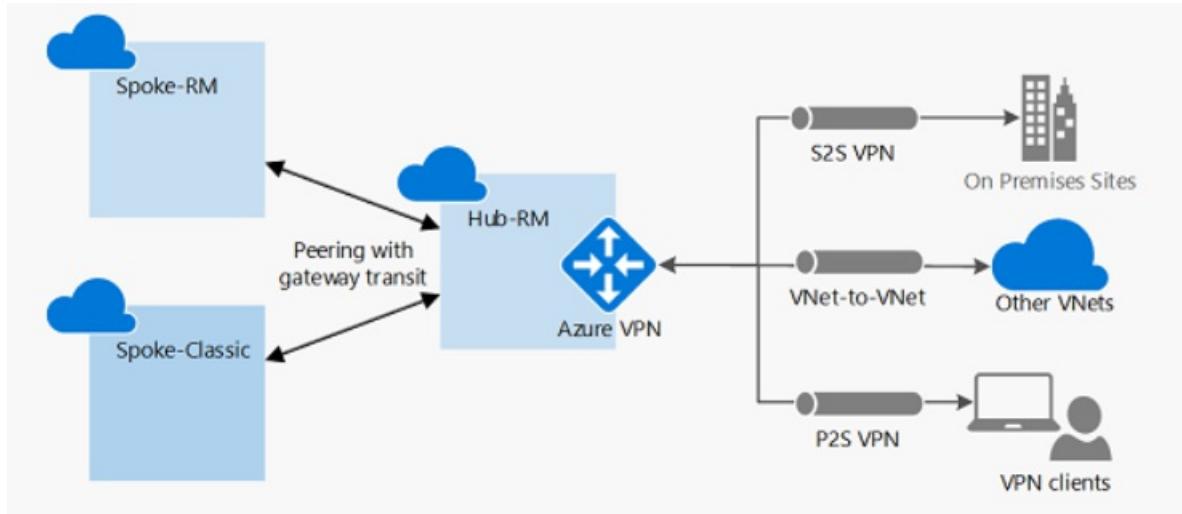
### The virtual networks are in different subscriptions or Active Directory tenants

To configure virtual network peering for virtual networks in different subscriptions or Active Directory tenants, see [Create peering in different subscriptions for Azure CLI](#).

#### NOTE

To configure network peering, you must have **Network Contributor** permissions in both subscriptions. For more information, see [Peering permissions](#).

## Configure virtual network peering with hub-spoke topology that uses on-premises resources



On-Premises Spoke

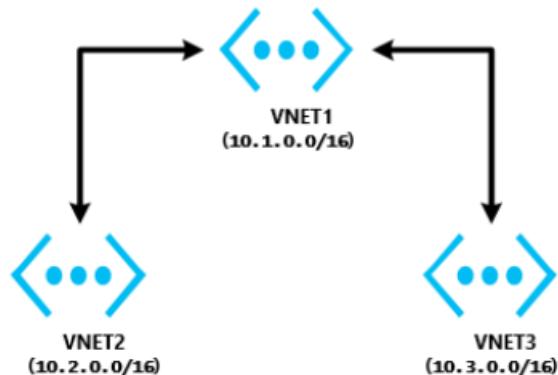
### For a site-to-site connection or an ExpressRoute connection

Follow the steps in: [Configure VPN gateway transit for virtual network peering](#).

### For point-to-site connections

1. Follow the steps in: [Configure VPN gateway transit for virtual network peering](#).
2. After virtual network peering is established or changed, download and reinstall the point-to-site package so that the point-to-site clients get the updated routes to the spoke virtual network.

## Configure virtual network peering with hub-spoke topology virtual network



Virtual Network Spoke

## The virtual networks are in the same region

1. In the hub virtual network, configure a network virtual appliance (NVA).
2. In the spoke virtual networks, have user-defined routes with the next hop type "network virtual appliance" applied.

For more information, see [Service chaining](#).

### NOTE

If you require help to set up an NVA, [contact the NVA vendor](#).

For help with troubleshooting the NVA device setup and routing, see [Network virtual appliance issues in Azure](#).

## The virtual networks are in different regions

Transit over global virtual network peering is now supported. Connectivity does not work over global virtual network peering for the following resources:

- VMs behind Basic ILB SKU
- Redis cache (uses Basic ILB SKU)
- Application gateway (uses Basic ILB SKU)
- Scale sets (uses Basic ILB SKU)
- Service Fabric clusters (uses Basic ILB SKU)
- SQL Server Always On (uses Basic ILB SKU)
- App Service Environment (uses Basic ILB SKU)
- API Management (uses Basic ILB SKU)
- Azure AD DS (uses Basic ILB SKU)

To learn more about global peering requirements and restraints, see [Virtual network peering](#).

## Troubleshoot a connectivity issue between two peered virtual networks

Sign in to the [Azure portal](#) with an account that has the necessary [roles and permissions](#). Select the virtual network, select **Peering**, and then check the **Status** field. What is the status?

### The peering status is "Connected"

To troubleshoot this issue:

1. Check the network traffic flows:

Use [Connection Troubleshoot](#) and [IP flow verify](#) from the source VM to the destination VM to determine whether there is an NSG or UDR that is causing interference in traffic flows.

If you're using a firewall or NVA:

- a. Document the UDR parameters so that you can restore them after this step is complete.
- b. Remove the UDR from the source VM subnet or NIC that points to the NVA as the next hop. Verify connectivity from the source VM directly to the destination that is bypassing the NVA. If this step doesn't work, see the [NVA troubleshooter](#).

2. Take a network trace:

- a. Start a network trace on the destination VM. For Windows, you can use **Netsh**. For Linux, use **TCPDump**.

- b. Run **TcpPing** or **PsPing** from the source to the destination IP.

This is an example of a **TcpPing** command: `tcping64.exe -t <destination VM address> 3389`

- c. After the **TcpPing** is complete, stop the network trace on the destination.
- d. If packets arrive from the source, there is no networking issue. Examine both the VM firewall and the application listening on that port to locate the configuration issue.

#### NOTE

You can't connect to the following resource types over global virtual network peering (virtual networks in different regions):

- VMs behind Basic ILB SKU
- Redis cache (uses Basic ILB SKU)
- Application gateway (uses Basic ILB SKU)
- Scale sets (uses Basic ILB SKU)
- Service Fabric clusters (uses Basic ILB SKU)
- SQL Server Always On (uses Basic ILB SKU)
- App Service Environment (uses Basic ILB SKU)
- API Management (uses Basic ILB SKU)
- Azure AD DS (uses Basic ILB SKU)

For more information, see the [requirements and constraints](#) of global peering.

#### The peering status is "Disconnected"

To resolve this issue, delete the peering from both virtual networks, and then re-create them.

## Troubleshoot a connectivity issue between a hub-spoke virtual network and an on-premises resource

Does your network use a third-party NVA or VPN gateway?

#### My network uses a third-party NVA or VPN gateway

To troubleshoot connectivity issues that affect a third-party NVA or VPN gateway, see the following articles:

- [NVA troubleshooter](#)
- [Service chaining](#)

#### My network does not use a third-party NVA or VPN gateway

Do the hub virtual network and the spoke virtual network have a VPN gateway?

#### Both the hub virtual network and the spoke virtual network have a VPN gateway

Using a remote gateway isn't supported.

If the spoke virtual network already has a VPN gateway, the **Use remote gateway** option isn't supported on the spoke virtual network. This is because of a virtual network peering limitation.

#### Both the hub virtual network and the spoke virtual network do not have a VPN gateway

For site-to-site or Azure ExpressRoute connections, check the following primary causes of connectivity issues to the remote virtual network from on-premises:

- On the virtual network that has a gateway, verify that the **Allow forwarded traffic** check box is selected.
- On the virtual network that doesn't have a gateway, verify that the **Use remote gateway** check box is selected.

- Have your network administrator check your on-premises devices to verify that they all have the remote virtual network address space added.

For point-to-site connections:

- On the virtual network that has a gateway, verify that the **Allow forwarded traffic** check box is selected.
- On the virtual network that doesn't have a gateway, verify that the **Use remote gateway** check box is selected.
- Download and reinstall the point-to-site client package. Virtual network routes that are newly peered don't automatically add routes to point-to-site clients.

## Troubleshoot a hub-spoke network connectivity issue between spoke virtual networks in the same region

A hub network must include an NVA. Configure UDRs in spokes that have an NVA set as the next hop, and enable **Allow forwarded traffic** in the hub virtual network.

For more information, see [Service chaining](#), and discuss these requirements with the [NVA vendor](#) of your choice.

## Troubleshoot a hub-spoke network connectivity issue between spoke virtual networks in different regions

Transit over global virtual network peering is now supported. Connectivity doesn't work over global virtual network peering for the following resources:

- VMs behind Basic ILB SKU
- Redis cache (uses Basic ILB SKU)
- Application gateway (uses Basic ILB SKU)
- Scale sets (uses Basic ILB SKU)
- Service Fabric clusters (uses Basic ILB SKU)
- SQL Server Always On (uses Basic ILB SKU)
- App Service Environment (uses Basic ILB SKU)
- API Management (uses Basic ILB SKU)
- Azure AD DS (uses Basic ILB SKU)

For more information, see the [requirements and constraints](#) of global peering and [Different VPN Topologies](#).

## Troubleshoot a hub-spoke network connectivity issue between a web app and the spoke virtual network

To troubleshoot this issue:

1. Sign in to the Azure portal.
2. In the web app, select **networking**, and then select **VNet Integration**.
3. Check whether you can see the remote virtual network. Manually enter the remote virtual network address space (**Sync Network** and **Add Routes**).

For more information, see the following articles:

- [Integrate your app with an Azure virtual network](#)
- [About Point-to-Site VPN routing](#)

# Troubleshoot a virtual network peering configuration error message

**Current tenant <TENANT ID> isn't authorized to access linked subscription**

To resolve this issue, see [Create peering - Azure CLI](#).

## Not connected

To resolve this issue, delete the peering from both virtual networks, and then recreate them.

## Failed to peer a Databricks virtual network

To resolve this issue, configure the virtual network peering under **Azure Databricks**, and then specify the target virtual network by using **Resource ID**. For more information, see [Peer a Databricks virtual network to a remote virtual network](#).

## The remote virtual network lacks a gateway

This issue occurs when you peer virtual networks from different tenants and later want to configure [Use Remote Gateways](#). A limitation of the Azure portal is that it can't validate the presence of a virtual network gateway in another tenant's virtual network.

There are two ways to resolve the issue:

- Delete the peerings and activate the [Use Remote Gateways](#) option when you create a new peering.
- Use PowerShell or CLI, instead of the Azure portal, to enable [Use Remote Gateways](#).

## Next steps

- [Troubleshooting connectivity problems between Azure VMs](#)

# Configure and validate virtual network or VPN connections

2/18/2021 • 15 minutes to read • [Edit Online](#)

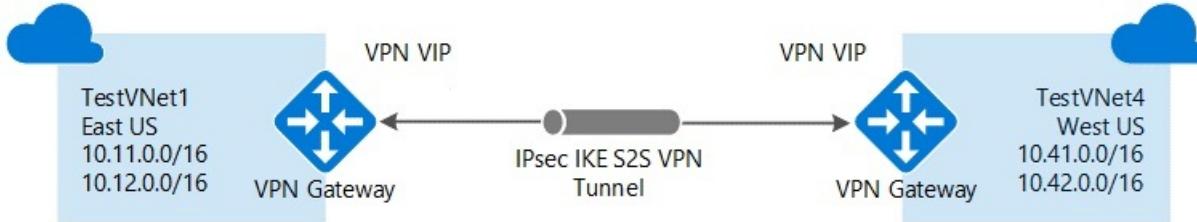
This walkthrough provides step-by-step guidance to configure and validate various Azure VPN and virtual network deployments. Scenarios include transit routing, network-to-network connections, Border Gateway Protocol (BGP), multisite connections, and point-to-site connections.

Azure VPN gateways enable flexibility in arranging almost any kind of connected virtual network topology in Azure. For example, you can connect virtual networks:

- Across regions.
- Between virtual network types (Azure Resource Manager versus classic).
- Within Azure or within an on-premises hybrid environment.
- In different subscriptions.

## Network-to-network VPN connection

Connecting a virtual network to another virtual network (network-to-network) via VPN is similar to connecting a virtual network to an on-premises site location. Both connectivity types use a VPN gateway to provide a secure tunnel through IPsec and IKE. The virtual networks can be in the same or different regions, and from the same or different subscriptions.



If your virtual networks are in the same region, you might want to consider connecting them by using virtual network peering. Virtual network peering doesn't use a VPN gateway. It increases throughput and decreases latency. To configure a virtual network peering connection, select **Configure and validate VNet Peering**.

If your virtual networks were created through the Azure Resource Manager deployment model, select **Configure and validate a Resource Manager VNet to a Resource Manager VNet connection** to configure a VPN connection.

If one of the virtual networks was created through the Azure classic deployment model, and the other was created through Resource Manager, select **Configure and validate a classic VNet to a Resource Manager VNet connection** to configure a VPN connection.

### Configure virtual network peering for two virtual networks in the same region

Before you start to implement and configure Azure virtual network peering, make sure that you meet the following prerequisites:

- The peered virtual networks must exist in the same Azure region.
- The peered virtual networks must have IP address spaces that don't overlap.
- Virtual network peering is between two virtual networks. There's no derived transitive relationship across peerings. For example, if VNetA is peered with VNetB, and VNetB is peered with VNetC, VNetA is *not* peered with VNetC.

When you meet the requirements, you can follow [Tutorial: Connect virtual networks with virtual network peering by using the Azure portal](#) to create and configure the peering.

To check the peering configuration, use the following method:

1. Sign in to the [Azure portal](#) by using an account that has the necessary [roles and permissions](#).
2. In the box that contains the text **Search resources** at the top of the portal, type **virtual networks**. When **Virtual networks** appears in the search results, select it.
3. In the **Virtual networks** blade that appears, select the virtual network for which you want to create a peering.
4. In the pane that appears for the virtual network, select **Peerings** in the **Settings** section.
5. Select a peering and view the configuration results.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons like App Service, Functions, Logic Apps, etc. The main area has a breadcrumb navigation bar: Microsoft Azure > Virtual networks > VNET10-01 - Peerings. The left pane is titled "Virtual networks" and lists several virtual networks: HubVnet, VNET10-01, VNET10-02, and VNET10-03. The right pane is titled "VNET10-01 - Peerings" and shows a table of peerings. One entry is visible: "LinkToVNET10-02" with status "Connected", peer "VNET10-02", and gateway transit "Disabled". The "Peerings" option in the left sidebar is highlighted.

For Azure PowerShell, run the command [Get-AzureRmVirtualNetworkPeering](#) to get the virtual network peering. Here's an example:

```
PS C:\Users\User1> Get-AzureRmVirtualNetworkPeering -VirtualNetworkName vnet10-01 -ResourceGroupName dev-vnets
Name : LinkToVNET10-02
Id : /subscriptions/GUID/resourceGroups/dev-vnets/providers/Microsoft.Network/virtualNetworks/VNET10-01/virtualNetworkPeerings/LinkToVNET10-02
2
Etag : W/"GUID"
ResourceGroupName : dev-vnets
VirtualNetworkName : vnet10-01
ProvisioningState : Succeeded
RemoteVirtualNetwork :
{
 "Id": "/subscriptions/GUID/resourceGroups/DEV-VNET-s/providers/Microsoft.Network/virtualNetworks/VNET10-02"
}
AllowVirtualNetworkAccess : True
AllowForwardedTraffic : False
AllowGatewayTransit : False
UseRemoteGateways : False
RemoteGateways : null
RemoteVirtualNetworkAddressSpace : null
```

## Connect a Resource Manager virtual network to another Resource Manager virtual network

You can configure a connection from one Resource Manager virtual network to another Resource Manager virtual network directly. Or you can configure the connection by using IPsec.

## Configure a VPN connection between Resource Manager virtual networks

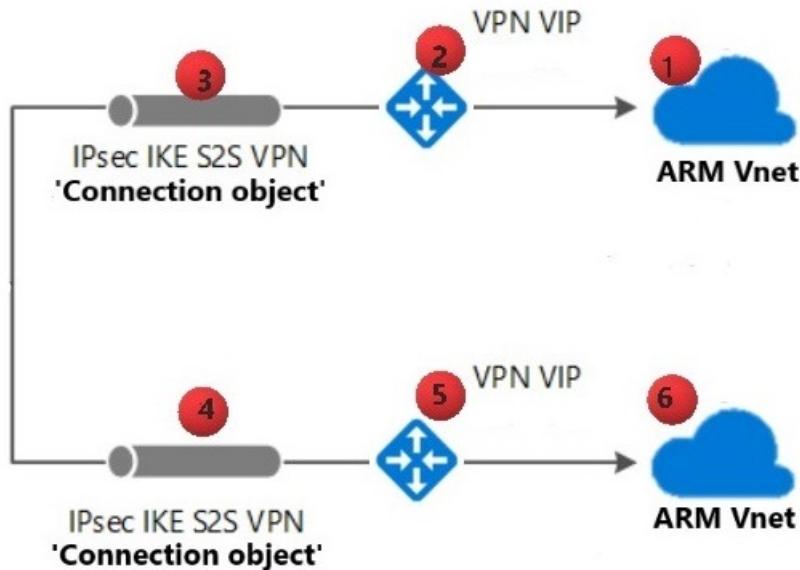
To configure a connection between Resource Manager virtual networks without IPsec, see [Configure a network-to-network VPN gateway connection by using the Azure portal](#).

To configure a connection with IPsec between two Resource Manager virtual networks, follow steps 1 to 5 in [Create a site-to-site connection in the Azure portal](#) for each virtual network.

### NOTE

These steps work only for virtual networks in the same subscription. If your virtual networks are in different subscriptions, you must use PowerShell to make the connection. See the [PowerShell](#) article.

## Validate the VPN connection between Resource Manager virtual networks



To check that your VPN connection is configured correctly, follow these instructions.

### NOTE

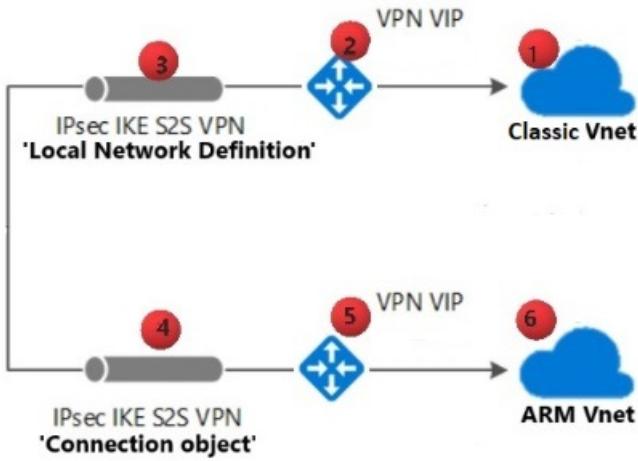
The numbers after virtual network components in these steps correspond to the numbers in the preceding diagram.

1. Make sure there are no overlapping address spaces in the connected virtual networks.
2. Verify that the address range for the Azure Resource Manager virtual network (1) is defined accurately in the **Connection object** instance (4).
3. Verify that the address range for the Azure Resource Manager virtual network (6) is defined accurately in the **Connection object** instance (3).
4. Verify that the pre-shared keys are matching on the connection objects.
5. Verify that the Azure Resource Manager virtual network gateway VIP (2) is defined accurately in the **Connection object** instance (4).
6. Verify that the Azure Resource Manager virtual network gateway VIP (5) is defined accurately in the **Connection object** instance (3).

## Connect a classic virtual network to a Resource Manager virtual network

You can create a connection between virtual networks that are in different subscriptions and in different regions. You can also connect virtual networks that already have connections to on-premises networks, as long as you have configured the gateway type as route-based.

To configure a connection between a classic virtual network and a Resource Manager virtual network, see [Connect virtual networks from different deployment models by using the Azure portal](#).



To check the configuration when you connect a classic virtual network to an Azure Resource Manager virtual network, follow these instructions.

**NOTE**

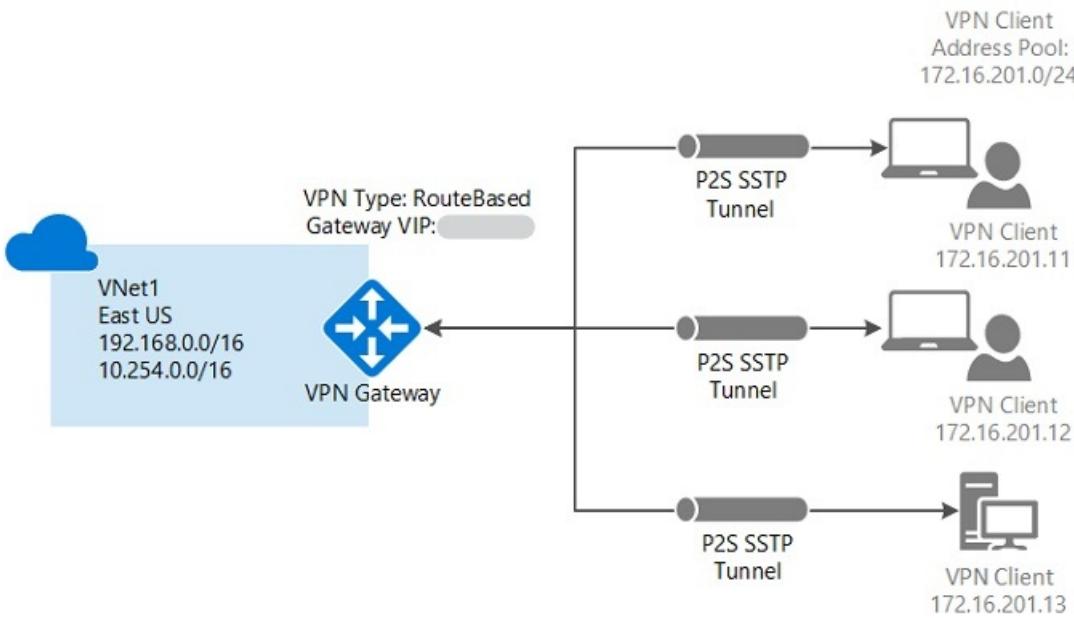
The numbers after virtual network components in these steps correspond to the numbers in the preceding diagram.

1. Make sure there are no overlapping address spaces in the connected virtual networks.
2. Verify that the address range for the Azure Resource Manager virtual network (6) is defined accurately in the classic local network definition (3).
3. Verify that the address range for the classic virtual network (1) is defined accurately in the Azure Resource Manager **Connection object** instance (4).
4. Verify that the classic virtual network gateway VIP (2) is defined accurately in the Azure Resource Manager **Connection object** instance (4).
5. Verify that the Azure Resource Manager virtual network gateway (5) is defined accurately in the classic **Local Network Definition** instance (3).
6. Verify that the pre-shared keys are matching on both connected virtual networks:
  - Classic virtual network: **Local Network Definition** (3)
  - Azure Resource Manager virtual network: **Connection object** (4)

## Create a point-to-site VPN connection

A point-to-site (*P2S* in the following diagram) configuration lets you create a secure connection from an individual client computer to a virtual network. Point-to-site connections are useful when you want to connect to your virtual network from a remote location, such as from home or a conference. They're also useful when you have only a few clients that need to connect to a virtual network.

The point-to-site VPN connection is initiated from the client computer through the native Windows VPN client. Connecting clients use certificates to authenticate.



Point-to-site connections don't require a VPN device. They create the VPN connection over Secure Socket Tunneling Protocol (SSTP). You can connect a point-to-site connection to a virtual network by using various deployment tools and deployment models:

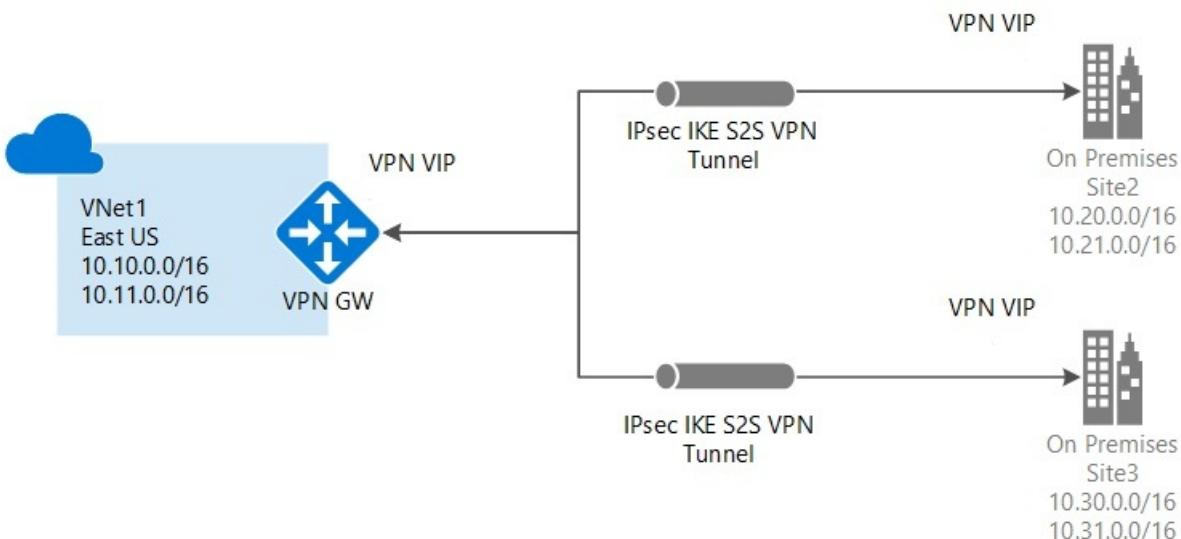
- [Configure a point-to-site connection to a virtual network by using the Azure portal](#)
- [Configure a point-to-site connection to a virtual network by using the Azure portal \(classic\)](#)
- [Configure a point-to-site connection to a virtual network by using PowerShell](#)

#### Validate your point-to-site connection

The article [Troubleshooting: Azure point-to-site connection problems](#) walks through common issues with point-to-site connections.

## Create a multisite VPN connection

You can add a site-to-site (S2S in the following diagram) connection to a virtual network that already has a site-to-site connection, point-to-site connection, or network-to-network connection. This kind of connection is often called a *multisite* configuration.



Azure currently works with two deployment models: Resource Manager and classic. The two models aren't completely compatible with each other. To configure a multisite connection with different models, see the following articles:

- Add a site-to-site connection to a virtual network with an existing VPN gateway connection
- Add a site-to-site connection to a virtual network with an existing VPN gateway connection (classic)

**NOTE**

The steps in those articles don't apply to Azure ExpressRoute and site-to-site coexisting connection configurations. For more information, see [ExpressRoute and site-to-site coexisting connections](#).

## Configure transit routing

Transit routing is a specific routing scenario where you connect multiple networks in a daisy-chain topology. This routing enables resources in virtual networks at either end of the chain to communicate with one another through virtual networks in between. Without transit routing, networks or devices peered through a hub can't reach one another.

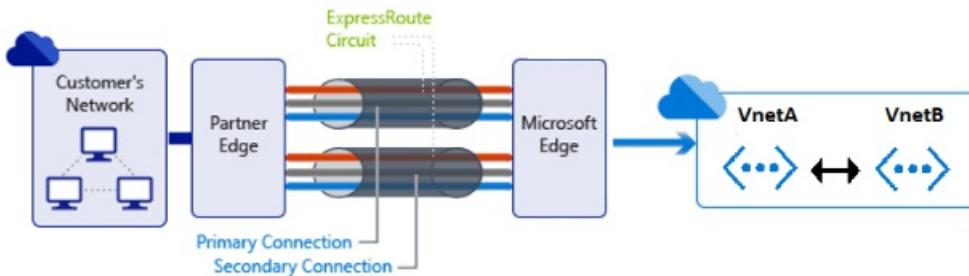
### Configure transit routing in a point-to-site connection

Imagine a scenario in which you want to configure a site-to-site VPN connection between VNetA and VNetB. You also want to configure a point-to-site VPN for the client to connect to the gateway of VNetA. Then, you want to enable transit routing for the point-to-site clients to connect to VNetB, which passes through VNetA.

This scenario is supported when BGP is enabled on the site-to-site VPN between VNetA and VNetB. For more information, see [About point-to-site VPN routing](#).

### Configure transit routing in an ExpressRoute connection

Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Microsoft 365, and Dynamics 365. For more information, see [ExpressRoute overview](#).



**NOTE**

We recommend that if VNetA and VNetB are in the same geopolitical region, you [link both virtual networks to the ExpressRoute circuit](#) instead of configuring transit routing. If your virtual networks are in different geopolitical regions, you can also link them to your circuit directly if you have [ExpressRoute Premium](#).

If you have ExpressRoute and site-to-site coexistence, transit routing isn't supported. For more information, see [Configure ExpressRoute and site-to-site by using PowerShell](#).

If you have enabled ExpressRoute to connect your local networks to an Azure virtual network, you can enable peering between the virtual networks where you want to have transit routing. To allow your local networks to connect to the remote virtual network, you must configure [virtual network peering](#).

#### NOTE

Virtual network peering is available only for virtual networks in the same region.

To check whether you have configured transit routing for virtual network peering, follow these instructions:

1. Sign in to the [Azure portal](#) by using an account that has the necessary [roles and permissions](#).
2. [Create a peering between VNetA and VNetB](#) as shown in the earlier diagram.
3. In the pane that appears for the virtual network, select **Peerings** in the **Settings** section.
4. Select the peering you want to view. Then, select **Configuration** to validate that you have enabled **Allow Gateway Transit** on the VNetA network connected to the ExpressRoute circuit and **Use Remote Gateway** on the remote VNetB network not connected to the ExpressRoute circuit.

#### Configure transit routing in a virtual network peering connection

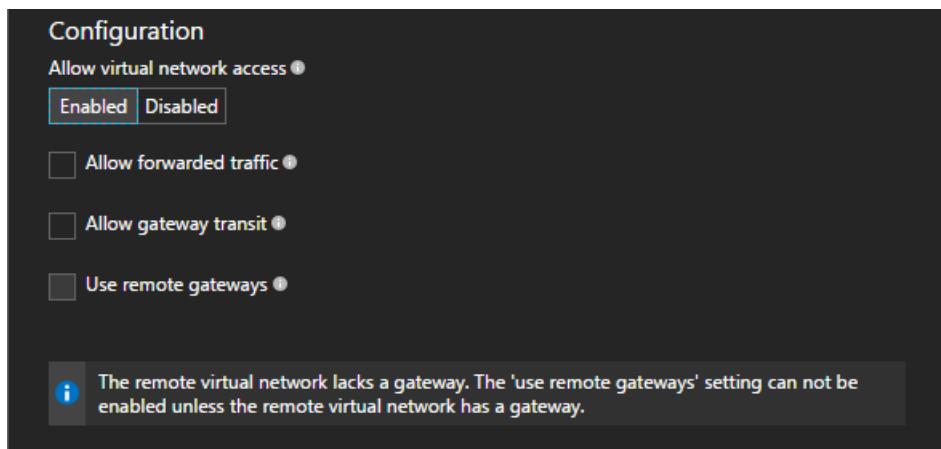
When virtual networks are peered, you can also configure the gateway in the peered virtual network as a transit point to an on-premises network. To configure a transit route in virtual network peering, see [Network-to-network connections](#).

#### NOTE

Gateway transit isn't supported in the peering relationship between virtual networks created through different deployment models. Both virtual networks in the peering relationship must have been created through Resource Manager for gateway transit to work.

To check whether you have configured a transit route for virtual network peering, follow these instructions:

1. Sign in to the [Azure portal](#) by using an account that has the necessary [roles and permissions](#).
2. In the box that contains the text **Search resources** at the top of the portal, type **virtual networks**. When **Virtual networks** appears in the search results, select it.
3. In the **Virtual networks** blade that appears, select the virtual network for which you want to check the peering setting.
4. In the pane that appears for the virtual network that you selected, select **Peerings** in the **Settings** section.
5. Select the peering that you want to view. Validate that you have enabled **Allow gateway transit** and **Use remote gateways** under **Configuration**.



#### Configure transit routing in a network-to-network connection

To configure transit routing between virtual networks, you must enable BGP on all intermediate network-to-network connections by using the Resource Manager deployment model and PowerShell. For instructions, see [How to configure BGP on Azure VPN gateways by using PowerShell](#).

Transit traffic through Azure VPN gateways is possible through the classic deployment model, but that relies on statically defined address spaces in the network configuration file. BGP isn't yet supported with Azure virtual networks and VPN gateways through the classic deployment model. Without BGP, manually defining transit address spaces is error prone, and we don't recommend it.

**NOTE**

You configure classic network-to-network connections by using the Azure classic portal, or by using a network configuration file in the classic portal. You can't create or modify a classic virtual network through the Azure Resource Manager deployment model or the Azure portal. For more information on transit routing for classic virtual networks, see the [Microsoft Developer blog](#).

### Configure transit routing in a site-to-site connection

To configure transit routing between your on-premises network and a virtual network with a site-to-site connection, you must enable BGP on all intermediate site-to-site connections by using the Resource Manager deployment model and PowerShell. See [How to configure BGP on Azure VPN gateways by using PowerShell](#) for instructions.

Transit traffic through Azure VPN gateways is possible through the classic deployment model, but that relies on statically defined address spaces in the network configuration file. BGP isn't yet supported with Azure virtual networks and VPN gateways through the classic deployment model. Without BGP, manually defining transit address spaces is error prone, and we don't recommend it.

**NOTE**

You configure classic site-to-site connections by using the Azure classic portal, or by using a network configuration file in the classic portal. You can't create or modify a classic virtual network through the Azure Resource Manager deployment model or the Azure portal. For more information on transit routing for classic virtual networks, see the [Microsoft Developer blog](#).

## Configure BGP for a VPN gateway

BGP is the standard routing protocol used on the internet to exchange routing and reachability information between two or more networks. When BGP is used in the context of Azure virtual networks, it enables the Azure VPN gateways and your on-premises VPN devices, known as BGP peers or neighbors. They exchange "routes" that will inform both gateways on the availability and reachability for those prefixes to go through the gateways or routers involved.

BGP can also enable transit routing among multiple networks by propagating routes that a BGP gateway learns from one BGP peer to all other BGP peers. For more information, see [Overview of BGP with Azure VPN Gateway](#).

### Configure BGP for a VPN connection

To configure a VPN connection that uses BGP, see [How to configure BGP on Azure VPN gateways by using PowerShell](#).

Enable BGP on the virtual network gateway by creating an autonomous system (AS) number for it. Basic gateways don't support BGP. To check the SKU of the gateway, go to the [Overview](#) section of the [VPN Gateway](#) blade in the Azure portal. If your SKU is **Basic**, you have to change the SKU (see [Resizing the gateway](#)) to **VpnGw1**.

Checking the SKU will cause 20 to 30 minutes of downtime. As soon as the gateway has the correct SKU, you can add the AS number by using the `Set-AzureRmVirtualNetworkGateway` PowerShell commandlet. After you configure the AS number, a BGP peer IP for the gateway will be provided automatically.

You must manually provide `LocalNetworkGateway` with an AS number and a BGP peer address. You can set the `ASN` and `-BgpPeeringAddress` values by using either the [New-AzureRmLocalNetworkGateway](#) or the [Set-AzureRmLocalNetworkGateway](#) PowerShell commandlet. Some AS numbers are reserved for Azure, and you can't use them as described in [About BGP with Azure VPN Gateway](#).

The connection object must have BGP enabled. You can set the `-EnableBGP` value to `$True` through [New-AzureRmVirtualNetworkGatewayConnection](#) or [Set-AzureRmVirtualNetworkGatewayConnection](#).

### Validate the BGP configuration

To check whether BGP is configured correctly, you can run the `get-AzureRmVirtualNetworkGateway` and `get-AzureRmLocalNetworkGateway` commandlets. Then you'll notice BGP-related output in the `BgpSettingsText` part. For example:

```
{
 "Asn": AsnNumber,
 "BgpPeeringAddress": "IP address",
 "PeerWeight": 0
}
```

## Create a highly available active/active VPN connection

The key differences between the active/active and active/standby gateways are:

- You must create two gateway IP configurations with two public IP addresses.
- You must set the `EnableActiveActiveFeature` flag.
- The gateway SKU must be `VpnGw1`, `VpnGw2`, or `VpnGw3`.

To achieve high availability for cross-premises and network-to-network connectivity, you should deploy multiple VPN gateways and establish multiple parallel connections between your networks and Azure. For an overview of connectivity options and topology, see [Highly available cross-premises and network-to-network connectivity](#).

To create active/active cross-premises and network-to-network connections, follow the instructions in [Configure active/active S2S VPN connections with Azure VPN gateways](#) to configure an Azure VPN gateway in active/active mode.

#### NOTE

- When you add addresses to the local network gateway for BGP-enabled active/active mode, *add only the /32 addresses of the BGP peers*. If you add more addresses, they will be considered static routes and take precedence over BGP routes.
- You must use different BGP AS numbers for your on-premises networks that are connecting to Azure. (If they're the same, you have to change your virtual network AS number if your on-premises VPN device already uses the ASN to peer with other BGP neighbors.)

## Change an Azure VPN gateway type after deployment

You can't change an Azure virtual network gateway type from policy-based to route-based or the other way directly. You must first delete the gateway. After that, the IP address and the pre-shared key won't be preserved. Then you can create a new gateway of the desired type.

To delete and create a gateway, follow these steps:

1. Delete any connections associated with the original gateway.
2. Delete the gateway by using the Azure portal, PowerShell, or classic PowerShell:
  - [Delete a virtual network gateway by using the Azure portal](#)
  - [Delete a virtual network gateway by using PowerShell](#)
  - [Delete a virtual network gateway by using PowerShell \(classic\)](#)
3. Follow the steps in [Create the VPN gateway](#) to create the new gateway of the desired type and complete the VPN setup.

**NOTE**

This process will take around 60 minutes.

## Next steps

- [Troubleshooting connectivity problems between Azure VMs](#)

# Diagnose a virtual machine network traffic filter problem

2/18/2021 • 12 minutes to read • [Edit Online](#)

In this article, you learn how to diagnose a network traffic filter problem by viewing the network security group (NSG) security rules that are effective for a virtual machine (VM).

NSGs enable you to control the types of traffic that flow in and out of a VM. You can associate an NSG to a subnet in an Azure virtual network, a network interface attached to a VM, or both. The effective security rules applied to a network interface are an aggregation of the rules that exist in the NSG associated to a network interface, and the subnet the network interface is in. Rules in different NSGs can sometimes conflict with each other and impact a VM's network connectivity. You can view all the effective security rules from NSGs that are applied on your VM's network interfaces. If you're not familiar with virtual network, network interface, or NSG concepts, see [Virtual network overview](#), [Network interface](#), and [Network security groups overview](#).

## Scenario

You attempt to connect to a VM over port 80 from the internet, but the connection fails. To determine why you can't access port 80 from the Internet, you can view the effective security rules for a network interface using the [Azure portal](#), [PowerShell](#), or the [Azure CLI](#).

The steps that follow assume you have an existing VM to view the effective security rules for. If you don't have an existing VM, first deploy a [Linux](#) or [Windows](#) VM to complete the tasks in this article with. The examples in this article are for a VM named *myVM* with a network interface named *myVMVMNic*. The VM and network interface are in a resource group named *myResourceGroup*, and are in the *East US* region. Change the values in the steps, as appropriate, for the VM you are diagnosing the problem for.

## Diagnose using Azure portal

1. Log into the Azure [portal](#) with an Azure account that has the [necessary permissions](#).
2. At the top of the Azure portal, enter the name of the VM in the search box. When the name of the VM appears in the search results, select it.
3. Under **SETTINGS**, select **Networking**, as shown in the following picture:

The screenshot shows the Azure portal interface for managing a virtual machine named 'myVM - Networking'. In the left sidebar, under 'SETTINGS', the 'Networking' option is highlighted with a red box. The main content area displays the configuration for a network interface named 'myVMVMNic'. It shows two sets of security rules:

- INBOUND PORT RULES** (Associated with Network Security Group mySubnetNSG):
 

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
1000	rdp	3389	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
- OUTBOUND PORT RULES** (Associated with Network Security Group myVMNSG):
 

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
1000	rdp	3389	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

The rules you see listed in the previous picture are for a network interface named **myVMVMNic**. You see that there are **INBOUND PORT RULES** for the network interface from two different network security groups:

- **mySubnetNSG**: Associated to the subnet that the network interface is in.
- **myVMNSG**: Associated to the network interface in the VM named **myVMVMNic**.

The rule named **DenyAllInBound** is what's preventing inbound communication to the VM over port 80, from the internet, as described in the [scenario](#). The rule lists **0.0.0.0/0** for **SOURCE**, which includes the internet. No other rule with a higher priority (lower number) allows port 80 inbound. To allow port 80 inbound to the VM from the internet, see [Resolve a problem](#). To learn more about security rules and how Azure applies them, see [Network security groups](#).

At the bottom of the picture, you also see **OUTBOUND PORT RULES**. Under that are the outbound port rules for the network interface. Though the picture only shows four inbound rules for each NSG, your NSGs may have many more than four rules. In the picture, you see **VirtualNetwork** under **SOURCE** and **DESTINATION** and **AzureLoadBalancer** under **SOURCE**. **VirtualNetwork** and **AzureLoadBalancer** are [service tags](#). Service tags represent a group of IP address prefixes to help minimize complexity for security rule creation.

4. Ensure that the VM is in the running state, and then select **Effective security rules**, as shown in the previous picture, to see the effective security rules, shown in the following picture:

Effective security rules

[Download](#) [Refresh](#)

Showing only top 50 security rules in each grid, click Download above to see all.

Select a network interface below to see the effective security rules and network security groups associated with it.

Scope: Network interface (myVMVMNic)

Associated NSGs: myVMNSG (Network interface), mySubnetNSG (Subnet)

Click on a rule row to see the expanded list of prefixes.

NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
rdp	1000	0.0.0.0/0	0-65535	0.0.0.0/0	3389-3389	TCP	Allow
AllowVnetInBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All	Allow
AllowAzureLoadBalancerInbound	65001	Azure load balancer (1 prefixes)	0-65535	0.0.0.0/0	0-65535	All	Allow
DenyAllInBound	65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	All	Deny

NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
AllowVnetOutBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All	Allow
AllowInternetOutBound	65001	0.0.0.0/0	0-65535	Internet (82 prefixes)	0-65535	All	Allow
DenyAllOutBound	65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	All	Deny

The rules listed are the same as you saw in step 3, though there are different tabs for the NSG associated to the network interface and the subnet. As you can see in the picture, only the first 50 rules are shown. To download a .csv file that contains all of the rules, select **Download**.

To see which prefixes each service tag represents, select a rule, such as the rule named **AllowAzureLoadBalancerInbound**. The following picture shows the prefixes for the **AzureLoadBalancer** service tag:

Home > Resource groups > myResourceGroup > myVM - Networking > Effective security rules

Address prefixes  
AllowAzureLoadBalancerInbound

Source	Destination
168.63.129.16/32	

Network Interface: myVMVMNic Effective security rules Topology

Virtual network/subnet: myVMVNET/myVMSubnet Public IP: 104.45.155.116 Private IP: 10.0.0.4 networking: Disabled

INBOUND PORT RULES

Network security group mySubnetNSG (attached to subnet: myVMSubnet)  
Impacts 1 subnets, 0 network interfaces

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION
1000	rdp	3389	Any	Any	Any
65000	AllowVnetInBound	Any	Any	VirtualNetwork	Virtua
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalanc	Any

Though the **AzureLoadBalancer** service tag only represents one prefix, other service tags represent several prefixes.

- The previous steps showed the security rules for a network interface named **myVMVMNic**, but you've also seen a network interface named **myVMVMNic2** in some of the previous pictures. The VM in this

example has two network interfaces attached to it. The effective security rules can be different for each network interface.

To see the rules for the **myVMVMNic2** network interface, select it. As shown in the picture that follows, the network interface has the same rules associated to its subnet as the **myVMVMNic** network interface, because both network interfaces are in the same subnet. When you associate an NSG to a subnet, its rules are applied to all network interfaces in the subnet.

**INBOUND PORT RULES**

Network security group **mySubnetNSG** (attached to subnet: **myVMSubnet**)  
Impacts 1 subnets, 0 network interfaces

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	...
1000	⚠️ rdp	3389	Any	Any	Any	Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowAzureLoadBalancerInBou...	Any	Any	AzureLoadBala...	Any	Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	Deny	...

**OUTBOUND PORT RULES**

Network security group **mySubnetNSG** (attached to subnet: **myVMSubnet**)  
Impacts 1 subnets, 0 network interfaces

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	...
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow	...
65500	DenyAllOutBound	Any	Any	Any	Any	Deny	...

Unlike the **myVMVMNic** network interface, the **myVMVMNic2** network interface does not have a network security group associated to it. Each network interface and subnet can have zero, or one, NSG associated to it. The NSG associated to each network interface or subnet can be the same, or different. You can associate the same network security group to as many network interfaces and subnets as you choose.

Though effective security rules were viewed through the VM, you can also view effective security rules through an individual:

- **Network interface:** Learn how to [view a network interface](#).
- **NSG:** Learn how to [view an NSG](#).

## Diagnose using PowerShell

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

You can run the commands that follow in the [Azure Cloud Shell](#), or by running PowerShell from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use

with your account. If you run PowerShell from your computer, you need the Azure PowerShell module, version 1.0.0 or later. Run `Get-Module -ListAvailable Az` on your computer, to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to log into Azure with an account that has the [necessary permissions](#).

Get the effective security rules for a network interface with `Get-AzEffectiveNetworkSecurityGroup`. The following example gets the effective security rules for a network interface named *myVMVMNic*, that is in a resource group named *myResourceGroup*.

```
Get-AzEffectiveNetworkSecurityGroup `
 -NetworkInterfaceName myVMVMNic `
 -ResourceGroupName myResourceGroup
```

Output is returned in json format. To understand the output, see [interpret command output](#). Output is only returned if an NSG is associated with the network interface, the subnet the network interface is in, or both. The VM must be in the running state. A VM may have multiple network interfaces with different NSGs applied. When troubleshooting, run the command for each network interface.

If you're still having a connectivity problem, see [additional diagnosis and considerations](#).

If you don't know the name of a network interface, but do know the name of the VM the network interface is attached to, the following commands return the IDs of all network interfaces attached to a VM:

```
$VM = Get-AzVM -Name myVM -ResourceGroupName myResourceGroup
$VM.NetworkProfile
```

You receive output similar to the following example:

```
NetworkInterfaces

{/subscriptions/<ID>/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic}
```

In the previous output, the network interface name is *myVMVMNic*.

## Diagnose using Azure CLI

If using Azure Command-line interface (CLI) commands to complete tasks in this article, either run the commands in the [Azure Cloud Shell](#), or by running the CLI from your computer. This article requires the Azure CLI version 2.0.32 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` and log into Azure with an account that has the [necessary permissions](#).

Get the effective security rules for a network interface with `az network nic list-effective-nsg`. The following example gets the effective security rules for a network interface named *myVMVMNic* that is in a resource group named *myResourceGroup*.

```
az network nic list-effective-nsg \
 --name myVMVMNic \
 --resource-group myResourceGroup
```

Output is returned in json format. To understand the output, see [interpret command output](#). Output is only returned if an NSG is associated with the network interface, the subnet the network interface is in, or both. The VM must be in the running state. A VM may have multiple network interfaces with different NSGs applied. When troubleshooting, run the command for each network interface.

If you're still having a connectivity problem, see [additional diagnosis and considerations](#).

If you don't know the name of a network interface, but do know the name of the VM the network interface is attached to, the following commands return the IDs of all network interfaces attached to a VM:

```
az vm show \
--name myVM \
--resource-group myResourceGroup
```

Within the returned output, you see information similar to the following example:

```
"networkProfile": {
 "additionalProperties": {},
 "networkInterfaces": [
 {
 "additionalProperties": {},
 "id": "/subscriptions/<ID>/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMVMNic",
 "primary": true,
 "resourceGroup": "myResourceGroup"
 },
],
}
```

In the previous output, the network interface name is *myVMVMNic interface*.

## Interpret command output

Regardless of whether you used the [PowerShell](#), or the [Azure CLI](#) to diagnose the problem, you receive output that contains the following information:

- **NetworkSecurityGroup**: The ID of the network security group.
- **Association**: Whether the network security group is associated to a *NetworkInterface* or *Subnet*. If an NSG is associated to both, output is returned with **NetworkSecurityGroup**, **Association**, and **EffectiveSecurityRules**, for each NSG. If the NSG is associated or disassociated immediately before running the command to view the effective security rules, you may need to wait a few seconds for the change to reflect in the command output.
- **EffectiveSecurityRules**: An explanation of each property is detailed in [Create a security rule](#). Rule names prefaced with *defaultSecurityRules*/are default security rules that exist in every NSG. Rule names prefaced with *securityRules*/are rules that you've created. Rules that specify a [service tag](#), such as **Internet**, **VirtualNetwork**, and **AzureLoadBalancer** for the **destinationAddressPrefix** or **sourceAddressPrefix** properties, also have values for the **expandedDestinationAddressPrefix** property. The **expandedDestinationAddressPrefix** property lists all address prefixes represented by the service tag.

If you see duplicate rules listed in the output, it's because an NSG is associated to both the network interface and the subnet. Both NSGs have the same default rules, and may have additional duplicate rules, if you've created your own rules that are the same in both NSGs.

The rule named **defaultSecurityRules/DenyAllInBound** is what's preventing inbound communication to the VM over port 80, from the internet, as described in the [scenario](#). No other rule with a higher priority (lower number) allows port 80 inbound from the internet.

## Resolve a problem

Whether you use the Azure [portal](#), [PowerShell](#), or the [Azure CLI](#) to diagnose the problem presented in the [scenario](#) in this article, the solution is to create a network security rule with the following properties:

PROPERTY	VALUE
Source	Any
Source port ranges	Any
Destination	The IP address of the VM, a range of IP addresses, or all addresses in the subnet.
Destination port ranges	80
Protocol	TCP
Action	Allow
Priority	100
Name	Allow-HTTP-All

After you create the rule, port 80 is allowed inbound from the internet, because the priority of the rule is higher than the default security rule named *DenyAllInBound*, that denies the traffic. Learn how to [create a security rule](#). If different NSGs are associated to both the network interface, and the subnet, you must create the same rule in both NSGs.

When Azure processes inbound traffic, it processes rules in the NSG associated to the subnet (if there is an associated NSG), and then it processes the rules in the NSG associated to the network interface. If there is an NSG associated to the network interface and the subnet, the port must be open in both NSGs, for the traffic to reach the VM. To ease administration and communication problems, we recommend that you associate an NSG to a subnet, rather than individual network interfaces. If VMs within a subnet need different security rules, you can make the network interfaces members of an application security group (ASG), and specify an ASG as the source and destination of a security rule. Learn more about [application security groups](#).

If you're still having communication problems, see [Considerations](#) and [Additional diagnosis](#).

## Considerations

Consider the following points when troubleshooting connectivity problems:

- Default security rules block inbound access from the internet, and only permit inbound traffic from the virtual network. To allow inbound traffic from the Internet, add security rules with a higher priority than default rules. Learn more about [default security rules](#), or how to [add a security rule](#).
- If you have peered virtual networks, by default, the **VIRTUAL\_NETWORK** service tag automatically expands to include prefixes for peered virtual networks. To troubleshoot any issues related to virtual network peering, you can view the prefixes in the **ExpandedAddressPrefix** list. Learn more about [virtual network peering](#) and [service tags](#).
- Effective security rules are only shown for a network interface if there is an NSG associated with the VM's network interface and, or, subnet, and if the VM is in the running state.
- If there are no NSGs associated with the network interface or subnet, and you have a [public IP address](#) assigned to a VM, all ports are open for inbound access from and outbound access to anywhere. If the VM has a public IP address, we recommend applying an NSG to the subnet the network interface.

## Additional diagnosis

- To run a quick test to determine if traffic is allowed to or from a VM, use the [IP flow verify](#) capability of Azure

Network Watcher. IP flow verify tells you if traffic is allowed or denied. If denied, IP flow verify tells you which security rule is denying the traffic.

- If there are no security rules causing a VM's network connectivity to fail, the problem may be due to:
  - Firewall software running within the VM's operating system
  - Routes configured for virtual appliances or on-premises traffic. Internet traffic can be redirected to your on-premises network via [forced-tunneling](#). If you force tunnel internet traffic to a virtual appliance, or on-premises, you may not be able to connect to the VM from the internet. To learn how to diagnose route problems that may impede the flow of traffic out of the VM, see [Diagnose a virtual machine network traffic routing problem](#).

## Next steps

- Learn about all tasks, properties, and settings for a [network security group](#) and [security rules](#).
- Learn about [default security rules](#), [service tags](#), and [how Azure processes security rules for inbound and outbound traffic](#) for a VM.

# Diagnose a virtual machine routing problem

11/2/2020 • 9 minutes to read • [Edit Online](#)

In this article, you learn how to diagnose a routing problem by viewing the routes that are effective for a network interface in a virtual machine (VM). Azure creates several default routes for each virtual network subnet. You can override Azure's default routes by defining routes in a route table, and then associating the route table to a subnet. The combination of routes you create, Azure's default routes, and any routes propagated from your on-premises network through an Azure VPN gateway (if your virtual network is connected to your on-premises network) via the border gateway protocol (BGP), are the effective routes for all network interfaces in a subnet. If you're not familiar with virtual network, network interface, or routing concepts, see [Virtual network overview](#), [Network interface](#), and [Routing overview](#).

## Scenario

You attempt to connect to a VM, but the connection fails. To determine why you can't connect to the VM, you can view the effective routes for a network interface using the [Azure portal](#), [PowerShell](#), or the [Azure CLI](#).

The steps that follow assume you have an existing VM to view the effective routes for. If you don't have an existing VM, first deploy a [Linux](#) or [Windows](#) VM to complete the tasks in this article with. The examples in this article are for a VM named *myVM* with a network interface named *myVMNic1*. The VM and network interface are in a resource group named *myResourceGroup*, and are in the *East US* region. Change the values in the steps, as appropriate, for the VM you are diagnosing the problem for.

## Diagnose using Azure portal

1. Log into the [Azure portal](#) with an Azure account that has the [necessary permissions](#).
2. At the top of the Azure portal, enter the name of a VM that is in the running state, in the search box. When the name of the VM appears in the search results, select it.
3. Under **Settings** on the left, select **Networking**, and navigate to the network interface resource by selecting its name.

The screenshot shows the Azure portal interface for a VM named 'myVM'. The left sidebar has 'Networking' selected under 'Settings'. The main content area shows the 'myVM - Networking' blade. It displays the 'Network Interface' 'myVMNic1' with its details: Virtual network/subnet: 'myResourceGroup-vnet/default', NIC Public IP: '40.122.43.96', and NIC Private IP: '10.1.1.4'. It also shows that 'Accelerated networking: Disabled'. Below this, there are tabs for 'Inbound port rules', 'Outbound port rules', 'Application security groups', and 'Load balancing'. The 'Inbound port rules' tab is active, showing three rules:

Priority	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBo...	Any	Any	AzureLoadBal...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

4. On the left, select **Effective routes**. The effective routes for a network interface named *myVMNic1* are

shown, in the following picture:

Source	State	Address Prefixes	Next Hop Type	User Defined Route Name
Default	Active	10.1.1.0/24	Virtual network	-
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-

If there are multiple network interfaces attached to the VM, you can view the effective routes for any network interface by selecting it. Since each network interface can be in a different subnet, each network interface can have different effective routes.

In the example shown in the previous picture, the listed routes are default routes that Azure creates for each subnet. Your list has at least these routes, but may have additional routes, depending on capabilities you may have enabled for your virtual network such as it being peered with another virtual network or connected to your on-premises network through an Azure VPN gateway. To learn more about each of the routes, and other routes you may see for your network interface, see [Virtual network traffic routing](#). If your list has a large number of routes, you may find it easier to select **Download**, to download a .csv file with the list of routes.

Though effective routes were viewed through the VM in the previous steps, you can also view effective routes through an:

- **Individual network interface:** Learn how to [view a network interface](#).
- **Individual route table:** Learn how to [view a route table](#).

## Diagnose using PowerShell

### NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

You can run the commands that follow in the [Azure Cloud Shell](#), or by running PowerShell from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use with your account. If you run PowerShell from your computer, you need the Azure PowerShell module, version 1.0.0 or later. Run `Get-Module -ListAvailable Az` on your computer, to find the installed version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzAccount` to log into Azure with an account that has the [necessary permissions](#).

Get the effective routes for a network interface with `Get-AzEffectiveRouteTable`. The following example gets the effective routes for a network interface named *myVMNic1*, that is in a resource group named *myResourceGroup*:

```
Get-AzEffectiveRouteTable `
 -NetworkInterfaceName myVMNic1 `
 -ResourceGroupName myResourceGroup `
 | Format-Table
```

To understand the information returned in the output, see [Routing overview](#). Output is only returned if the VM is in the running state. If there are multiple network interfaces attached to the VM, you can review the effective routes for each network interface. Since each network interface can be in a different subnet, each network interface can have different effective routes. If you're still having a communication problem, see [additional diagnosis and considerations](#).

If you don't know the name of a network interface, but do know the name of the VM the network interface is attached to, the following commands return the IDs of all network interfaces attached to a VM:

```
$VM = Get-AzVM -Name myVM `
 -ResourceGroupName myResourceGroup
$VM.NetworkProfile
```

You receive output similar to the following example:

```
NetworkInterfaces

{/subscriptions/<ID>/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/myVMNic1}
```

In the previous output, the network interface name is *myVMNic1*.

## Diagnose using Azure CLI

You can run the commands that follow in the [Azure Cloud Shell](#), or by running the CLI from your computer. This article requires the Azure CLI version 2.0.32 or later. Run `az --version` to find the installed version. If you need to install or upgrade, see [Install Azure CLI](#). If you are running the Azure CLI locally, you also need to run `az login` and log into Azure with an account that has the [necessary permissions](#).

Get the effective routes for a network interface with [az network nic show-effective-route-table](#). The following example gets the effective routes for a network interface named *myVMNic1* that is in a resource group named *myResourceGroup*.

```
az network nic show-effective-route-table \
 --name myVMNic1 \
 --resource-group myResourceGroup
```

To understand the information returned in the output, see [Routing overview](#). Output is only returned if the VM is in the running state. If there are multiple network interfaces attached to the VM, you can review the effective routes for each network interface. Since each network interface can be in a different subnet, each network interface can have different effective routes. If you're still having a communication problem, see [additional diagnosis and considerations](#).

If you don't know the name of a network interface, but do know the name of the VM the network interface is attached to, the following commands return the IDs of all network interfaces attached to a VM:

```
az vm show \
 --name myVM \
 --resource-group myResourceGroup
```

# Resolve a problem

Resolving routing problems typically consists of:

- Adding a custom route to override one of Azure's default routes. Learn how to [add a custom route](#).
- Change or remove a custom route that may cause routing to an undesired location. Learn how to [change](#) or [delete](#) a custom route.
- Ensuring that the route table that contains any custom routes you've defined is associated to the subnet the network interface is in. Learn how to [associate a route table to a subnet](#).
- Ensuring that devices such as Azure VPN gateway or network virtual appliances you've deployed are operable. Use the [VPN diagnostics](#) capability of Network Watcher to determine any problems with an Azure VPN gateway.

If you're still having communication problems, see [Considerations](#) and [Additional diagnosis](#).

## Considerations

Consider the following points when troubleshooting communication problems:

- Routing is based on longest prefix match (LPM) among routes that you've defined, border gateway protocol (BGP), and system routes. If there is more than one route with the same LPM match, then a route is selected based on its origin in the order listed in [Routing overview](#). With effective routes, you can only see effective routes that are an LPM match, based on all the available routes. Seeing how the routes are evaluated for a network interface makes it a lot easier to troubleshoot specific routes that may be impacting communication from your VM.
- If you've defined custom routes to a network virtual appliance (NVA), with *Virtual Appliance* as the next hop type, ensure that IP forwarding is enabled on the NVA receiving the traffic, or packets are dropped. Learn more about [enabling IP forwarding for a network interface](#). Additionally, the operating system, or application within the NVA must also be able to forward network traffic and be configured to do so.
- If you've created a route to 0.0.0.0/0, all outbound internet traffic is routed to the next hop you specified, such as to an NVA or VPN gateway. Creating such a route is often referred to as forced tunneling. Remote connections using the RDP or SSH protocols from the internet to your VM may not work with this route, depending on how the next hop handles the traffic. Forced-tunneling can be enabled:
  - When using site-to-site VPN, by creating a route with a next hop type of *VPN Gateway*. Learn more about [configuring forced tunneling](#).
  - If a 0.0.0.0/0 (default route) is advertised over BGP through a virtual network gateway when using a site-to-site VPN, or ExpressRoute circuit. Learn more about using BGP with a [site-to-site VPN](#) or [ExpressRoute](#).
- For virtual network peering traffic to work correctly, a system route with a next hop type of *VNet Peering* must exist for the peered virtual network's prefix range. If such a route doesn't exist, and the virtual network peering link is **Connected**:
  - Wait a few seconds, and retry. If it's a newly established peering link, it occasionally takes longer to propagate routes to all the network interfaces in a subnet. To learn more about virtual network peering, see [Virtual network peering overview](#) and [manage virtual network peering](#).
  - Network security group rules may be impacting communication. For more information, see [Diagnose a virtual machine network traffic filter problem](#).
- Though Azure assigns default routes to each Azure network interface, if you have multiple network interfaces attached to the VM, only the primary network interface is assigned a default route (0.0.0.0/0), or gateway, within the VM's operating system. Learn how to create a default route for secondary network interfaces attached to a [Windows](#) or [Linux](#) VM. Learn more about [primary and secondary network interfaces](#).

## Additional diagnosis

- To run a quick test to determine the next hop type for traffic destined to a location, use the [Next hop](#) capability of Azure Network Watcher. Next hop tells you what the next hop type is for traffic destined to a specified location.
- If there are no routes causing a VM's network communication to fail, the problem may be due to firewall software running within the VM's operating system
- If you are [force tunneling](#) traffic to an on-premises device through a VPN gateway, or NVA, you may not be able to connect to a VM from the internet, depending on how you've configured routing for the devices. Confirm that the routing you have configured for the device routes traffic to either a public or private IP address for the VM.
- Use the [connection troubleshoot](#) capability of Network Watcher to determine routing, filtering, and in-OS causes of outbound communication problems.

## Next steps

- Learn about all tasks, properties, and settings for a [route table and routes](#).
- Learn about all [next hop types, system routes, and how Azure selects a route](#).

# Bandwidth/Throughput testing (NTTCP)

2/18/2021 • 3 minutes to read • [Edit Online](#)

When testing network throughput performance in Azure, it's best to use a tool that targets the network for testing and minimizes the use of other resources that could impact performance. NTTCP is recommended.

Copy the tool to two Azure VMs of the same size. One VM functions as SENDER and the other as RECEIVER.

## Deploying VMs for testing

For the purposes of this test, the two VMs should be in either the same [Proximity Placement Group](#) or the same Availability Set so that we can use their internal IPs and exclude the Load Balancers from the test. It is possible to test with the VIP but this kind of testing is outside the scope of this document.

Make a note of the RECEIVER's IP address. Let's call that IP "a.b.c.r"

Make a note of the number of cores on the VM. Let's call this "#num\_cores"

Run the NTTCP test for 300 seconds (or 5 minutes) on the sender VM and receiver VM.

Tip: When setting up this test for the first time, you might try a shorter test period to get feedback sooner. Once the tool is working as expected, extend the test period to 300 seconds for the most accurate results.

### NOTE

The sender **and** receiver must specify the **same** test duration parameter (-t).

To test a single TCP stream for 10 seconds:

Receiver parameters: nttcp -r -t 10 -P 1

Sender parameters: nttcp -s10.27.33.7 -t 10 -n 1 -P 1

### NOTE

The preceding sample should only be used to confirm your configuration. Valid examples of testing are covered later in this document.

## Testing VMs running WINDOWS:

### Get NTTCP onto the VMs.

Download the latest version: <https://github.com/microsoft/nttcp/releases/download/v5.35/NTtcp.exe>

Or view the top-level GitHub Page: [https://github.com/microsoft/nttcp\](https://github.com/microsoft/nttcp)

Consider putting NTTCP in separate folder, like c:\tools

### Allow NTTCP through the Windows firewall

On the RECEIVER, create an Allow rule on the Windows Firewall to allow the NTTCP traffic to arrive. It's easiest to allow the entire NTTCP program by name rather than to allow specific TCP ports inbound.

Allow nttcp through the Windows Firewall like this:

```
netsh advfirewall firewall add rule program=<PATH>\nttcp.exe name="nttcp" protocol=any dir=in
action=allow enable=yes profile=ANY
```

For example, if you copied ntttcp.exe to the "c:\tools" folder, this would be the command:

```
netsh advfirewall firewall add rule program=c:\tools\ntttcp.exe name="ntttcp" protocol=any dir=in
action=allow enable=yes profile=ANY
```

#### Running NTTTCP tests

Start NTTTCP on the RECEIVER (run from CMD, not from PowerShell):

```
ntttcp -r -m [2*#num_cores],*,a.b.c.r -t 300
```

If the VM has four cores and an IP address of 10.0.0.4, it would look like this:

```
ntttcp -r -m 8,*,10.0.0.4 -t 300
```

Start NTTTCP on the SENDER (run from CMD, not from PowerShell):

```
ntttcp -s -m 8,*,10.0.0.4 -t 300
```

Wait for the results.

## Testing VMs running LINUX:

Use nttcp-for-linux. It is available from <https://github.com/Microsoft/ntttcp-for-linux>

On the Linux VMs (both SENDER and RECEIVER), run these commands to prepare nttcp-for-linux on your VMs:

CentOS - Install Git:

```
yum install gcc -y
yum install git -y
```

Ubuntu - Install Git:

```
apt-get -y install build-essential
apt-get -y install git
```

Make and Install on both:

```
git clone https://github.com/Microsoft/ntttcp-for-linux
cd nttcp-for-linux/src
make && make install
```

As in the Windows example, we assume the Linux RECEIVER's IP is 10.0.0.4

Start NTTCP-for-Linux on the RECEIVER:

```
ntttcp -r -t 300
```

And on the SENDER, run:

```
ntttcp -s10.0.0.4 -t 300
```

Test length defaults to 60 seconds if no time parameter is given

## Testing between VMs running Windows and LINUX:

On this scenarios we should enable the no-sync mode so the test can run. This is done by using the **-N flag** for Linux, and **-ns flag** for Windows.

#### **From Linux to Windows:**

Receiver <Windows>:

```
nttcp -r -m <2 x nr cores>,*,<Windows server IP>
```

Sender <Linux> :

```
nttcp -s -m <2 x nr cores>,*,<Windows server IP> -N -t 300
```

#### **From Windows to Linux:**

Receiver <Linux>:

```
nttcp -r -m <2 x nr cores>,*,<Linux server IP>
```

Sender <Windows>:

```
nttcp -s -m <2 x nr cores>,*,<Linux server IP> -ns -t 300
```

## Testing Cloud Service Instances:

You need to add following section into your ServiceDefinition.csdef

```
<Endpoints>
 <InternalEndpoint name="Endpoint3" protocol="any" />
</Endpoints>
```

## Next steps

- Depending on results, there may be room to [Optimize network throughput machines](#) for your scenario.
- Read about how [bandwidth is allocated to virtual machines](#)
- Learn more with [Azure Virtual Network frequently asked questions \(FAQ\)](#)

# Test VM network latency

2/18/2021 • 5 minutes to read • [Edit Online](#)

To achieve the most accurate results, measure your Azure virtual machine (VM) network latency with a tool that's designed for the task. Publicly available tools such as SockPerf (for Linux) and latte.exe (for Windows) can isolate and measure network latency while excluding other types of latency, such as application latency. These tools focus on the kind of network traffic that affects application performance (namely, Transmission Control Protocol [TCP] and User Datagram Protocol [UDP] traffic).

Other common connectivity tools, such as Ping, might measure latency, but their results might not represent the network traffic that's used in real workloads. That's because most of these tools employ the Internet Control Message Protocol (ICMP), which can be treated differently from application traffic and whose results might not apply to workloads that use TCP and UDP.

For accurate network latency testing of the protocols used by most applications, SockPerf (for Linux) and latte.exe (for Windows) produce the most relevant results. This article covers both of these tools.

## Overview

By using two VMs, one as sender and one as receiver, you create a two-way communications channel. With this approach, you can send and receive packets in both directions and measure the round-trip time (RTT).

You can use this approach to measure network latency between two VMs or even between two physical computers. Latency measurements can be useful for the following scenarios:

- Establish a benchmark for network latency between the deployed VMs.
- Compare the effects of changes in network latency after related changes are made to:
  - Operating system (OS) or network stack software, including configuration changes.
  - A VM deployment method, such as deploying to an availability zone or proximity placement group (PPG).
  - VM properties, such as Accelerated Networking or size changes.
  - A virtual network, such as routing or filtering changes.

## Tools for testing

To measure latency, you have two different tool options:

- For Windows-based systems: [latte.exe \(Windows\)](#)
- For Linux-based systems: [SockPerf \(Linux\)](#)

By using these tools, you help ensure that only TCP or UDP payload delivery times are measured and not ICMP (Ping) or other packet types that aren't used by applications and don't affect their performance.

## Tips for creating an optimal VM configuration

When you create your VM configuration, keep in mind the following recommendations:

- Use the latest version of Windows or Linux.
- Enable Accelerated Networking for best results.
- Deploy VMs with an [Azure proximity placement group](#).
- Larger VMs generally perform better than smaller VMs.

## Tips for analysis

As you're analyzing test results, keep in mind the following recommendations:

- Establish a baseline early, as soon as deployment, configuration, and optimizations are complete.
- Always compare new results to a baseline or, otherwise, from one test to another with controlled changes.
- Repeat tests whenever changes are observed or planned.

## Test VMs that are running Windows

### Get latte.exe onto the VMs

Download the [latest version of latte.exe](#).

Consider putting latte.exe in separate folder, such as `c:\tools`.

### Allow latte.exe through Windows Defender Firewall

On the *receiver*, create an Allow rule on Windows Defender Firewall to allow the latte.exe traffic to arrive. It's easiest to allow the entire latte.exe program by name rather than to allow specific TCP ports inbound.

Allow latte.exe through Windows Defender Firewall by running the following command:

```
netsh advfirewall firewall add rule program=<path>\latte.exe name="Latte" protocol=any dir=in action=allow enable=yes profile=ANY
```

For example, if you copied latte.exe to the `c:\tools` folder, this would be the command:

```
netsh advfirewall firewall add rule program=c:\tools\latte.exe name="Latte" protocol=any dir=in action=allow enable=yes profile=ANY
```

### Run latency tests

- On the *receiver*, start latte.exe (run it from the CMD window, not from PowerShell):

```
latte -a <Receiver IP address>:<port> -i <iterations>
```

Around 65,000 iterations is long enough to return representative results.

Any available port number is fine.

If the VM has an IP address of 10.0.0.4, the command would look like this:

```
latte -a 10.0.0.4:5005 -i 65100
```

- On the *sender*, start latte.exe (run it from the CMD window, not from PowerShell):

```
latte -c -a <Receiver IP address>:<port> -i <iterations>
```

The resulting command is the same as on the receiver, except with the addition of `-c` to indicate that this is the *client*, or *sender*:

```
latte -c -a 10.0.0.4:5005 -i 65100
```

Wait for the results. Depending on how far apart the VMs are, the test could take a few minutes to finish.

Consider starting with fewer iterations to test for success before running longer tests.

## Test VMs that are running Linux

To test VMs that are running Linux, use [SockPerf](#).

## Install SockPerf on the VMs

On the Linux VMs, both *sender* and *receiver*, run the following commands to prepare SockPerf on the VMs.

Commands are provided for the major distros.

### For Red Hat Enterprise Linux (RHEL)/CentOS

Run the following commands:

```
#RHEL/CentOS - Install Git and other helpful tools
sudo yum install gcc -y -q
sudo yum install git -y -q
sudo yum install gcc-c++ -y
sudo yum install ncurses-devel -y
sudo yum install -y automake
sudo yum install -y autoconf
```

### For Ubuntu

Run the following commands:

```
#Ubuntu - Install Git and other helpful tools
sudo apt-get install build-essential -y
sudo apt-get install git -y -q
sudo apt-get install -y autotools-dev
sudo apt-get install -y automake
sudo apt-get install -y autoconf
```

### For all distros

Copy, compile, and install SockPerf according to the following steps:

```
#Bash - all distros

#From bash command line (assumes Git is installed)
git clone https://github.com/mellanox/sockperf
cd sockperf/
./autogen.sh
./configure --prefix=

#make is slower, may take several minutes
make

#make install is fast
sudo make install
```

## Run SockPerf on the VMs

After the SockPerf installation is complete, the VMs are ready to run the latency tests.

First, start SockPerf on the *receiver*.

Any available port number is fine. In this example, we use port 12345:

```
#Server/Receiver - assumes server's IP is 10.0.0.4:
sudo sockperf sr --tcp -i 10.0.0.4 -p 12345
```

Now that the server is listening, the client can begin sending packets to the server on the port on which it is listening (in this case, 12345).

About 100 seconds is long enough to return representative results, as shown in the following example:

```
#Client/Sender - assumes server's IP is 10.0.0.4:
sockperf ping-pong -i 10.0.0.4 --tcp -m 350 -t 101 -p 12345 --full-rtt
```

Wait for the results. Depending on how far apart the VMs are, the number of iterations will vary. To test for success before you run longer tests, consider starting with shorter tests of about 5 seconds.

This SockPerf example uses a 350-byte message size, which is typical for an average packet. You can adjust the size higher or lower to achieve results that more accurately represent the workload that's running on your VMs.

## Next steps

- Improve latency with an [Azure proximity placement group](#).
- Learn how to [Optimize networking for VMs](#) for your scenario.
- Read about [how bandwidth is allocated to virtual machines](#).
- For more information, see [Azure Virtual Network FAQ](#).

# Troubleshooting: Failed to delete a virtual network in Azure

2/18/2021 • 2 minutes to read • [Edit Online](#)

You might receive errors when you try to delete a virtual network in Microsoft Azure. This article provides troubleshooting steps to help you resolve this problem.

If your Azure issue is not addressed in this article, visit the Azure forums on [MSDN and Stack Overflow](#). You can post your issue in these forums, or post to [@AzureSupport on Twitter](#). You also can submit an Azure support request. To submit a support request, on the [Azure support](#) page, select **Get support**.

## Troubleshooting guidance

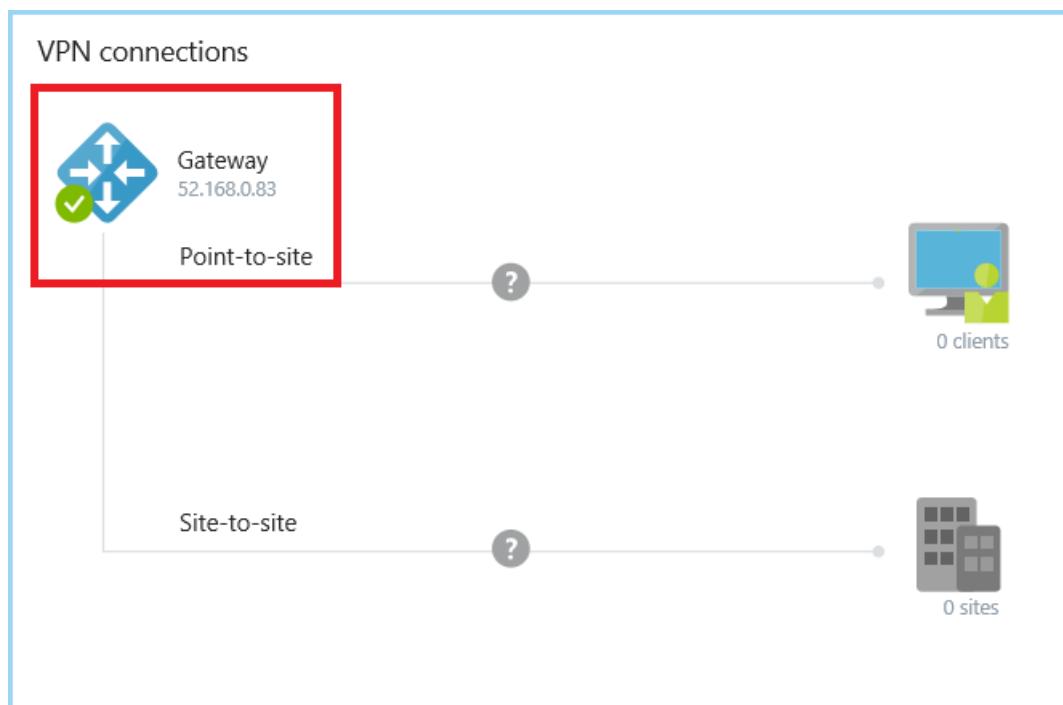
1. [Check whether a virtual network gateway is running in the virtual network.](#)
2. [Check whether an application gateway is running in the virtual network.](#)
3. [Check whether Azure container instances still exist in the virtual network.](#)
4. [Check whether Azure Active Directory Domain Service is enabled in the virtual network.](#)
5. [Check whether the virtual network is connected to other resource.](#)
6. [Check whether a virtual machine is still running in the virtual network.](#)
7. [Check whether the virtual network is stuck in migration.](#)

## Troubleshooting steps

### **Check whether a virtual network gateway is running in the virtual network**

To remove the virtual network, you must first remove the virtual network gateway.

For classic virtual networks, go to the [Overview](#) page of the classic virtual network in the Azure portal. In the **VPN connections** section, if the gateway is running in the virtual network, you will see the IP address of the gateway.



For virtual networks, go to the **Overview** page of the virtual network. Check **Connected devices** for the virtual network gateway.

Connected devices				
DEVICE	TYPE	IP ADDRESS	SUBNET	
testrbac914	Network interface	10.0.0.4	default	
thomasvm195	Network interface	10.0.0.5	default	
Gateway01	Virtual network gateway	-	GatewaySubnet	

Before you can remove the gateway, first remove any **Connection** objects in the gateway.

### Check whether an application gateway is running in the virtual network

Go to the **Overview** page of the virtual network. Check the **Connected devices** for the application gateway.

Connected devices				
DEVICE	TYPE	IP ADDRESS	SUBNET	
testrbac914	Network interface	10.0.0.4	default	
thomasvm195	Network interface	10.0.0.5	default	
AppGateway-Test	Application gateway		gateway	

If there is an application gateway, you must remove it before you can delete the virtual network.

### Check whether Azure container instances still exist in the virtual network

1. In the Azure portal, go to the resource group's **Overview** page.
2. In the header for the list of the resource group's resources, select **Show hidden types**. The network profile type is hidden in the Azure portal by default.
3. Select the network profile related to the container groups.
4. Select **Delete**.

5. Delete the subnet or virtual network again.

If these steps don't resolve the issue, use these [Azure CLI commands](#) to clean up resources.

### Check whether Azure Active Directory Domain Service is enabled in the virtual network

If the Active Directory Domain Service is enabled and connected to the virtual network, you cannot delete this virtual network.

The screenshot shows the Azure portal interface for managing Azure AD Domain Services. On the left, there's a navigation bar with 'Home > Azure AD Domain Services >'. Below it is a search bar and a sidebar with links for 'Overview', 'Activity log', 'Access control (IAM)', and 'Properties' (which is highlighted with a dashed blue border). The main content area displays details for a specific domain: 'DNS DOMAIN NAME: <Domain name>', 'LOCATION: East Asia', and a section titled 'AVAILABLE IN VIRTUAL NETWORK/SUBNET' containing the placeholder '<VNet name/subnet>'. The entire 'AVAILABLE IN VIRTUAL NETWORK/SUBNET' section is enclosed in a red box.

To disable the service, see [Disable Azure Active Directory Domain Services using the Azure portal](#).

### Check whether the virtual network is connected to other resource

Check for Circuit Links, connections, and virtual network peerings. Any of these can cause a virtual network deletion to fail.

The recommended deletion order is as follows:

1. Gateway connections
2. Gateways
3. IPs
4. Virtual network peerings
5. App Service Environment (ASE)

### Check whether a virtual machine is still running in the virtual network

Make sure that no virtual machine is in the virtual network.

### Check whether the virtual network is stuck in migration

If the virtual network is stuck in a migration state, it cannot be deleted. Run the following command to abort the migration, and then delete the virtual network.

```
Move-AzureVirtualNetwork -VirtualNetworkName "Name" -Abort
```

## Next steps

- [Azure Virtual Network](#)
- [Azure Virtual Network frequently asked questions \(FAQ\)](#)

# Troubleshooting connectivity problems between Azure VMs

2/18/2021 • 3 minutes to read • [Edit Online](#)

You might experience connectivity problems between Azure virtual machines (VMs). This article provides troubleshooting steps to help you resolve this problem.

If your Azure issue is not addressed in this article, visit the Azure forums on [MSDN and Stack Overflow](#). You can post your issue in these forums, or post to [@AzureSupport on Twitter](#). You also can submit an Azure support request. To submit a support request, on the [Azure support](#) page, select **Get support**.

## Symptom

One Azure VM cannot connect to another Azure VM.

## Troubleshooting guidance

1. [Check whether NIC is misconfigured](#)
2. [Check whether network traffic is blocked by NSG or UDR](#)
3. [Check whether network traffic is blocked by VM firewall](#)
4. [Check whether VM app or service is listening on the port](#)
5. [Check whether the problem is caused by SNAT](#)
6. [Check whether traffic is blocked by ACLs for the classic VM](#)
7. [Check whether the endpoint is created for the classic VM](#)
8. [Try to connect to a VM network share](#)
9. [Check Inter-Vnet connectivity](#)

## Troubleshooting steps

Follow these steps to troubleshoot the problem. After you complete each step, check whether the problem is resolved.

### Step 1: Check whether NIC is misconfigured

Follow the steps in [How to reset network interface for Azure Windows VM](#).

If the problem occurs after you modify the network interface (NIC), follow these steps:

#### Multi-NIC VMs

1. Add a NIC.
2. Fix the problems in the bad NIC or remove the bad NIC. Then add the NIC again.

For more information, see [Add network interfaces to or remove from virtual machines](#).

#### Single-NIC VM

- [Redeploy Windows VM](#)
- [Redeploy Linux VM](#)

### Step 2: Check whether network traffic is blocked by NSG or UDR

Use [Network Watcher IP Flow Verify](#) and [NSG Flow Logging](#) to determine whether there is a Network Security Group (NSG) or User-Defined Route (UDR) that is interfering with traffic flow.

### **Step 3: Check whether network traffic is blocked by VM firewall**

Disable the firewall, and then test the result. If the problem is resolved, verify the firewall settings, and then re-enable the firewall.

### **Step 4: Check whether VM app or service is listening on the port**

You can use one of the following methods to check whether the VM app or service is listening on the port.

- Run the following commands to check whether the server is listening on that port.

#### **Windows VM**

```
netstat -ano
```

#### **Linux VM**

```
netstat -l
```

- Run the **telnet** command on the virtual machine itself to test the port. If the test fails, the application or service is not configured to listen on that port.

### **Step 5: Check whether the problem is caused by SNAT**

In some scenarios, the VM is placed behind a load balance solution that has a dependency on resources outside of Azure. In these scenarios, if you experience intermittent connection problems, the problem may be caused by [SNAT port exhaustion](#). To resolve the issue, create a VIP (or ILPIP for classic) for each VM that is behind the load balancer and secure with NSG or ACL.

### **Step 6: Check whether traffic is blocked by ACLs for the classic VM**

An access control list (ACL) provides the ability to selectively permit or deny traffic for a virtual machine endpoint. For more information, see [Manage the ACL on an endpoint](#).

### **Step 7: Check whether the endpoint is created for the classic VM**

All VMs that you create in Azure by using the classic deployment model can automatically communicate over a private network channel with other virtual machines in the same cloud service or virtual network. However, computers on other virtual networks require endpoints to direct the inbound network traffic to a virtual machine. For more information, see [How to set up endpoints](#).

### **Step 8: Try to connect to a VM network share**

If you cannot connect to a VM network share, the problem may be caused by unavailable NICs in the VM. To delete the unavailable NICs, see [How to delete the unavailable NICs](#).

### **Step 9: Check Inter-Vnet connectivity**

Use [Network Watcher IP Flow Verify](#) and [NSG Flow Logging](#) to determine whether there is a NSG or UDR that is interfering with traffic flow. You can also verify your Inter-Vnet configuration [here](#).

### **Need help? Contact support.**

If you still need help, [contact support](#) to get your issue resolved quickly.

# Configure reverse lookup zones for an SMTP banner check

11/2/2020 • 2 minutes to read • [Edit Online](#)

This article describes how to use a reverse zone in Azure DNS and create a Reverse DNS (PTR) record for SMTP Banner Check.

## Symptom

If you host an SMTP server in Microsoft Azure, you may receive the following error message when send or receive a message from remote mail servers:

**554: No PTR Record**

## Solution

For a virtual IP address in Azure, the reverse records are created in Microsoft owned domain zones, not custom domain zones.

To configure PTR records in Microsoft owned zones, use the `-ReverseFqdn` property on the `PublicIpAddress` resource. For more information, see [Configure reverse DNS for services hosted in Azure](#).

When you configure the PTR records, make sure that the IP address and the reverse FQDN are owned by the subscription. If you try to set a reverse FQDN that does not belong to the subscription, you receive the following error message:

```
Set-AzPublicIpAddress : ReverseFqdn mail.contoso.com that PublicIPAddress ip01 is trying to use does not belong to subscription <Subscription ID>. One of the following conditions need to be met to establish ownership:
```

- 1) ReverseFqdn matches fqdn of any public ip resource under the subscription;
- 2) ReverseFqdn resolves to the fqdn (through CName records chain) of any public ip resource under the subscription;
- 3) It resolves to the ip address (through CName and A records chain) of a static public ip resource under the subscription.

If you manually change your SMTP banner to match our default reverse FQDN, the remote mail server can still fail because it may expect the SMTP banner host to match the MX record for the domain.

# Network virtual appliance issues in Azure

2/18/2021 • 5 minutes to read • [Edit Online](#)

## NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

You may experience VM or VPN connectivity issues and errors when using a third party Network Virtual Appliance (NVA) in Microsoft Azure. This article provides basic steps to help you validate basic Azure Platform requirements for NVA configurations.

Technical support for third-party NVAs and their integration with the Azure platform is provided by the NVA vendor.

## NOTE

If you have a connectivity or routing problem that involves an NVA, you should [contact the vendor of the NVA](#) directly.

If your Azure issue is not addressed in this article, visit the Azure forums on [MSDN and Stack Overflow](#). You can post your issue in these forums, or post to [@AzureSupport on Twitter](#). You also can submit an Azure support request. To submit a support request, on the [Azure support](#) page, select **Get support**.

## Checklist for troubleshooting with NVA vendor

- Software updates for NVA VM software
- Service Account setup and functionality
- User-defined routes (UDRs) on virtual network subnets that direct traffic to NVA
- UDRs on virtual network subnets that direct traffic from NVA
- Routing tables and rules within the NVA (for example, from NIC1 to NIC2)
- Tracing on NVA NICs to verify receiving and sending network traffic
- When using a Standard SKU and Public IPs, there must be an NSG created and an explicit rule to allow the traffic to be routed to the NVA.

## Basic troubleshooting steps

- Check the basic configuration
- Check NVA performance
- Advanced network troubleshooting

## Check the minimum configuration requirements for NVAs on Azure

Each NVA has basic configuration requirements to function correctly on Azure. The following section provides the steps to verify these basic configurations. For more information, [contact the vendor of the NVA](#).

### Check whether IP forwarding is enabled on NVA

Use Azure portal

1. Locate the NVA resource in the [Azure portal](#), select Networking, and then select the Network interface.
2. On the Network interface page, select IP configuration.
3. Make sure that IP forwarding is enabled.

#### Use PowerShell

1. Open PowerShell and then sign in to your Azure account.
2. Run the following command (replace the bracketed values with your information):

```
Get-AzNetworkInterface -ResourceGroupName <ResourceGroupName> -Name <NicName>
```

3. Check the **EnableIPForwarding** property.
4. If IP forwarding is not enabled, run the following commands to enable it:

```
$nic2 = Get-AzNetworkInterface -ResourceGroupName <ResourceGroupName> -Name <NicName>
$nic2.EnableIPForwarding = 1
Set-AzNetworkInterface -NetworkInterface $nic2
Execute: $nic2 #and check for an expected output:
EnableIPForwarding : True
NetworkSecurityGroup : null
```

**Check for NSG when using Standard SKU Public IP** When using a Standard SKU and Public IPs, there must be an NSG created and an explicit rule to allow the traffic to the NVA.

#### Check whether the traffic can be routed to the NVA

1. On [Azure portal](#), open **Network Watcher**, select **Next Hop**.
2. Specify a VM that is configured to redirect the traffic to the NVA, and a destination IP address at which to view the next hop.
3. If the NVA is not listed as the **next hop**, check and update the Azure route tables.

#### Check whether the traffic can reach the NVA

1. In [Azure portal](#), open **Network Watcher**, and then select **IP Flow Verify**.
2. Specify the VM and the IP address of the NVA, and then check whether the traffic is blocked by any Network security groups (NSG).
3. If there is an NSG rule that blocks the traffic, locate the NSG in **effective security** rules and then update it to allow traffic to pass. Then run **IP Flow Verify** again and use **Connection troubleshoot** to test TCP communications from VM to your internal or external IP address.

#### Check whether NVA and VMs are listening for expected traffic

1. Connect to the NVA by using RDP or SSH, and then run following command:

For Windows:

```
netstat -an
```

For Linux:

```
netstat -an | grep -i listen
```

2. If you don't see the TCP port that's used by the NVA software that's listed in the results you must

configure the application on the NVA and VM to listen and respond to traffic that reaches those ports. [Contact the NVA vendor for assistance as needed.](#)

## Check NVA Performance

### Validate VM CPU

If CPU usage gets close to 100 percent, you may experience issues that affect network packet drops. Your VM reports average CPU for a specific time span in the Azure portal. During a CPU spike, investigate which process on the guest VM is causing the high CPU, and mitigate it, if possible. You may also have to resize the VM to a larger SKU size or, for virtual machine scale set, increase the instance count or set to auto-scale on CPU usage. For either of these issues, [contact the NVA vendor for assistance](#), as needed.

### Validate VM Network statistics

If the VM network use spikes or shows periods of high usage, you may also have to increase the SKU size of the VM to obtain higher throughput capabilities. You can also redeploy the VM by having Accelerated Networking enabled. To verify whether the NVA supports Accelerated Networking feature, [contact the NVA vendor for assistance](#), as needed.

## Advanced network administrator troubleshooting

### Capture network trace

Capture a simultaneous network trace on the source VM, the NVA, and the destination VM while you run [PsPing](#) or [Nmap](#), and then stop the trace.

1. To capture a simultaneous network trace, run the following command:

#### For Windows

```
netsh trace start capture=yes tracefile=c:\server_IPetl scenario=netconnection
```

#### For Linux

```
sudo tcpdump -s0 -i eth0 -X -w vmtrace.cap
```

2. Use [PsPing](#) or [Nmap](#) from the source VM to the destination VM (for example: `PsPing 10.0.0.4:80` or `Nmap -p 80 10.0.0.4` ).
3. Open the network trace from the destination VM by using [Network Monitor](#) or tcpdump. Apply a display filter for the IP of the Source VM you ran [PsPing](#) or [Nmap](#) from, such as `IPv4.address==10.0.0.4 (Windows netmon)` or `tcpdump -nn -r vmtrace.cap src or dst host 10.0.0.4` (Linux).

### Analyze traces

If you do not see the packets incoming to the backend VM trace, there is likely an NSG or UDR interfering or the NVA routing tables are incorrect.

If you do see the packets coming in but no response, then you may need to address a VM application or a firewall issue. For either of these issues, [contact the NVA vendor for assistance as needed](#).

# Troubleshoot outbound SMTP connectivity problems in Azure

2/18/2021 • 4 minutes to read • [Edit Online](#)

Starting on November 15, 2017, outbound email messages that are sent directly to external domains (like outlook.com and gmail.com) from a virtual machine (VM) are made available only to certain subscription types in Azure. Outbound SMTP connections that use TCP port 25 were blocked. (Port 25 is used mainly for unauthenticated email delivery.)

This change in behavior applies only to subscriptions and deployments that were created after November 15, 2017.

## Recommended method of sending email

We recommend you use authenticated SMTP relay services to send email from Azure VMs or from Azure App Service. (These relay services typically connect through TCP port 587, but they support other ports.) These services are used to maintain IP or domain reputation to minimize the possibility that third-party email providers will reject messages. [SendGrid](#) is one such SMTP relay service, but there are others. You might also have a secure SMTP relay service running on-premises that you can use.

Using these email delivery services isn't restricted in Azure, regardless of the subscription type.

## Enterprise Agreement

For Enterprise Agreement Azure users, there's no change in the technical ability to send email without using an authenticated relay. Both new and existing Enterprise Agreement users can try outbound email delivery from Azure VMs directly to external email providers without any restrictions from the Azure platform. There's no guarantee that email providers will accept incoming email from any given user. But the Azure platform won't block delivery attempts for VMs within Enterprise Agreement subscriptions. You'll have to work directly with email providers to fix any message delivery or SPAM filtering problems that involve specific providers.

## Pay-as-you-go

If you signed up before November 15, 2017, for a pay-as-you-go subscription, there will be no change in your technical ability to try outbound email delivery. You'll still be able to try outbound email delivery from Azure VMs within these subscriptions directly to external email providers without any restrictions from the Azure platform. Again, there's no guarantee that email providers will accept incoming email from any given user. Users will have to work directly with email providers to fix any message delivery or SPAM filtering issues that involve specific providers.

For pay-as-you-go subscriptions that were created after November 15, 2017, there will be technical restrictions that block email that's sent directly from VMs within the subscriptions. If you want to be able to send email from Azure VMs directly to external email providers (without using an authenticated SMTP relay) and you have an account in good standing with a payment history, you can request to have the restriction removed. You can do so in the **Connectivity** section of the **Diagnose and Solve** blade for an Azure Virtual Network resource in the Azure portal. If your request is accepted, your subscription will be enabled or you'll receive instructions for next steps.

After a pay-as-you-go subscription is exempted and the VMs are stopped and restarted in the Azure portal, all VMs in that subscription are exempted going forward. The exemption applies only to the subscription requested

and only to VM traffic that's routed directly to the internet.

**NOTE**

Microsoft reserves the right to revoke these exemptions if it's determined that a violation of terms of service has occurred.

## MSDN, Azure Pass, Azure in Open, Education, Azure for Students, Visual Studio, and Free Trial

If you created one of the following subscription types after November 15, 2017, you'll have technical restrictions that block email that's sent from VMs within the subscription directly to email providers:

- MSDN
- Azure Pass
- Azure in Open
- Education
- Azure for Students
- Free Trial
- Any Visual Studio subscription

The restrictions are in place to prevent abuse. Requests to remove these restrictions won't be granted.

If you're using these subscription types, we encourage you to use SMTP relay services, as outlined earlier in this article, or to change your subscription type.

## Cloud Solution Provider

If you're using Azure resources through a Cloud Solution Provider, you can make a request to remove the restriction in the **Connectivity** section of the **Diagnose and Solve** pane for a virtual network resource in the Azure portal. If your request is accepted, your subscription will be enabled or you'll receive instructions for next steps.

## Microsoft Partner Network, BizSpark Plus, or Azure Sponsorship

For subscriptions of the following types that were created after November 15, 2017, there will be technical restrictions that block email that's sent directly from VMs within the subscriptions:

- Microsoft Partner Network (MPN)
- BizSpark Plus
- Azure Sponsorship

If you want to be able to send email from Azure VMs directly to external email providers (without using an authenticated SMTP relay), you can make a request by opening a support case by using the following issue type: **Technical > Virtual Network > Connectivity > Cannot send email (SMTP/Port 25)**. Be sure to add details about why your deployment has to send mail directly to mail providers instead of using an authenticated relay. Requests will be reviewed and approved at the discretion of Microsoft. Requests will be granted only after additional antifraud checks are completed.

After a subscription is exempted and the VMs have been stopped and restarted in the Azure portal, all VMs in that subscription are exempted going forward. The exemption applies only to the subscription requested and only to VM traffic that's routed directly to the internet.

## Need help? Contact support

If you still need help, [contact support](#) to get your problem resolved quickly. Use this issue type: Technical > Virtual Network > Connectivity > Cannot send email (SMTP/Port 25).

# What is IP address 168.63.129.16?

2/18/2021 • 2 minutes to read • [Edit Online](#)

IP address 168.63.129.16 is a virtual public IP address that is used to facilitate a communication channel to Azure platform resources. Customers can define any address space for their private virtual network in Azure. Therefore, the Azure platform resources must be presented as a unique public IP address. This virtual public IP address facilitates the following things:

- Enables the VM Agent to communicate with the Azure platform to signal that it is in a "Ready" state.
- Enables communication with the DNS virtual server to provide filtered name resolution to the resources (such as VM) that do not have a custom DNS server. This filtering makes sure that customers can resolve only the hostnames of their resources.
- Enables [health probes from Azure load balancer](#) to determine the health state of VMs.
- Enables the VM to obtain a dynamic IP address from the DHCP service in Azure.
- Enables Guest Agent heartbeat messages for the PaaS role.

## NOTE

In a non-virtual network scenario (Classic), a private IP address is used instead of 168.63.129.16. This private IP address is dynamically discovered through DHCP. Firewall rules specific to 168.63.129.16 need to be adjusted as appropriate.

## Scope of IP address 168.63.129.16

The public IP address 168.63.129.16 is used in all regions and all national clouds. This special public IP address is owned by Microsoft and will not change. We recommend that you allow this IP address in any local (in the VM) firewall policies (outbound direction). The communication between this special IP address and the resources is safe because only the internal Azure platform can source a message from this IP address. If this address is blocked, unexpected behavior can occur in a variety of scenarios. 168.63.129.16 is a [virtual IP of the host node](#) and as such it is not subject to user defined routes.

- The VM Agent requires outbound communication over ports 80/tcp and 32526/tcp with WireServer (168.63.129.16). These should be open in the local firewall on the VM. The communication on these ports with 168.63.129.16 is not subject to the configured network security groups.
- 168.63.129.16 can provide DNS services to the VM. If this is not desired, outbound traffic to 168.63.129.16 ports 53/udp and 53/tcp can be blocked in the local firewall on the VM.

By default DNS communication is not subject to the configured network security groups unless specifically targeted leveraging the [AzurePlatformDNS](#) service tag. To block DNS traffic to Azure DNS through NSG, create an outbound rule to deny traffic to [AzurePlatformDNS](#), and specify "\*" as "Destination port ranges" and "Any" as protocol.

- When the VM is part of a load balancer backend pool, [health probe](#) communication should be allowed to originate from 168.63.129.16. The default network security group configuration has a rule that allows this communication. This rule leverages the [AzureLoadBalancer](#) service tag. If desired this traffic can be blocked by configuring the network security group however this will result in probes that fail.

## Next steps

- [Security groups](#)

- Create, change, or delete a network security group

# Troubleshoot Azure VM connectivity problems

2/18/2021 • 6 minutes to read • [Edit Online](#)

This article helps administrators diagnose and resolve connectivity problems that affect Azure virtual machines (VMs).

## Problems

- An Azure VM that's deployed by using Resource Manager can't connect to another Azure VM in same virtual network.
- An Azure VM can't connect to the second network adapter of an Azure VM in same virtual network.
- An Azure VM can't connect to the internet.

To resolve these problems, follow the steps in the following section.

## Resolution

### Azure VM cannot connect to another Azure VM in same virtual network

#### Step 1: Verify that VMs can communicate with each other.

1. Download TCping to your source VM.
2. Open a Command Prompt window.
3. Navigate to the folder in which you downloaded TCping.
4. Ping the destination from the source VM by using the following command:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\>tcping64.exe -t 10.0.0.9 3389
** Pinging continuously. Press control-c to stop **
```

```
tcping64.exe -t <destination VM address> 3389
```

#### TIP

If the ping test is successful, go to Step 3. Otherwise, go to the next step.

#### Step 2: Check the Network security group settings.

For each VM, check for default Inbound port rules ("Allow VNet Inbound" and "Allow Load Balancer Inbound"). Make sure to also check that there are no matching blocking rules that are listed below a lower-priority rule.

#### NOTE

Rules that have a lower number will be matched first. For example, if you have a rule that has priority 1000 and 6500, the rule that has priority 1000 will be matched first.

After that, try to ping the destination from the source VM again:

```
tcping64.exe -t <destination VM address> 3389
```

### Step 3: Check whether you can connect to the destination VM by using Remote Desktop or SSH.

To connect by using Remote Desktop, follow these steps.

#### Windows:

1. Sign in to the Azure portal.
2. In the left menu, select **Virtual Machines**.
3. Select the virtual machine in the list.
4. On the page for the virtual machine, select **Connect**.

For more information, see [How to connect and sign on to an Azure virtual machine running Windows](#).

#### Linux:

For more information, see [Connect to a Linux VM in Azure](#).

If the Remote Desktop or SSH connection is successful, go to next step.

### Step 4: Perform a connectivity check.

Run a connectivity check on the source VM, and check the response.

**Windows:** [Check connectivity with Azure Network Watcher using PowerShell](#)

**Linux:** [Check connectivity with Azure Network Watcher using Azure CLI 2.0](#)

The following is an example response:

```
ConnectionStatus : Unreachable
AvgLatencyInMs :
MinLatencyInMs :
MaxLatencyInMs :
ProbesSent : 100
ProbesFailed : 100
Hops : [
 {
 "Type": "Source",
 "Id": "c5222ea0-3213-4f85-a642-cee63217c2f3",
 "Address": "10.1.1.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/appNic0/ipConfigurations/ipconfig1",
 "NextHopIds": [
 "9283a9f0-cc5e-4239-8f5e-ae0f3c19fbaa"
],
 "Issues": []
 },
 {
 "Type": "VirtualAppliance",
 "Id": "9283a9f0-cc5e-4239-8f5e-ae0f3c19fbaa",
 "Address": "10.1.2.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/fwNic/ipConfigurations/ipconfig1",
 "NextHopIds": [
 "0f1500cd-c512-4d43-b431-7267e4e67017"
],
 "Issues": []
 }
],
```

### Step 5: Fix the issue in the connectivity check result.

- In the **Hops** section of the connectivity check response that you received, check the listed **issues**.

```

 },
 {
 "Type": "VirtualAppliance",
 "Id": "0f1500cd-c512-4d43-b431-7267e4e67017",
 "Address": "10.1.3.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/auNetworking-ipconfig1",
 "NextHopIds": [
 "a88940f8-5fbe-40da-8d99-1dee89240f64"
],
 "Issues": [
 {
 "Origin": "Outbound",
 "Severity": "Error",
 "Type": "NetworkSecurityRule",
 "Context": [
 {
 "key": "RuleName",
 "value": "UserRule_Port80"
 }
]
 }
],
 {
 "Type": "VnetLocal",
 "Id": "a88940f8-5fbe-40da-8d99-1dee89240f64",
 "Address": "10.1.4.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/dbNetworking-ipconfig1"
 }
]
 }
}

```

- Find the corresponding resolution in the following table, and follow the indicated steps to resolve the issues.

ISSUE TYPE	VALUE	RESOLUTION ACTION
NetworkSecurityRule	Name of the blocking NSG	You can <a href="#">delete the NSG rule</a> or modify the rule as described <a href="#">here</a> .
UserDefinedRoute	Name of the blocking UDR	If you do not require this route, delete the UDR. If you can't delete the route, update the route by using the appropriate address prefix and next hop. You can also adjust the Network Virtual Appliance to forward traffic appropriately. For more information, see: <a href="#">Virtual network traffic routing</a> and <a href="#">Route network traffic with a route table using PowerShell</a> .
CPU	Usage	Follow these recommendations that describe in <a href="#">Generic performance troubleshooting for Azure Virtual Machine running Linux or Windows</a> .

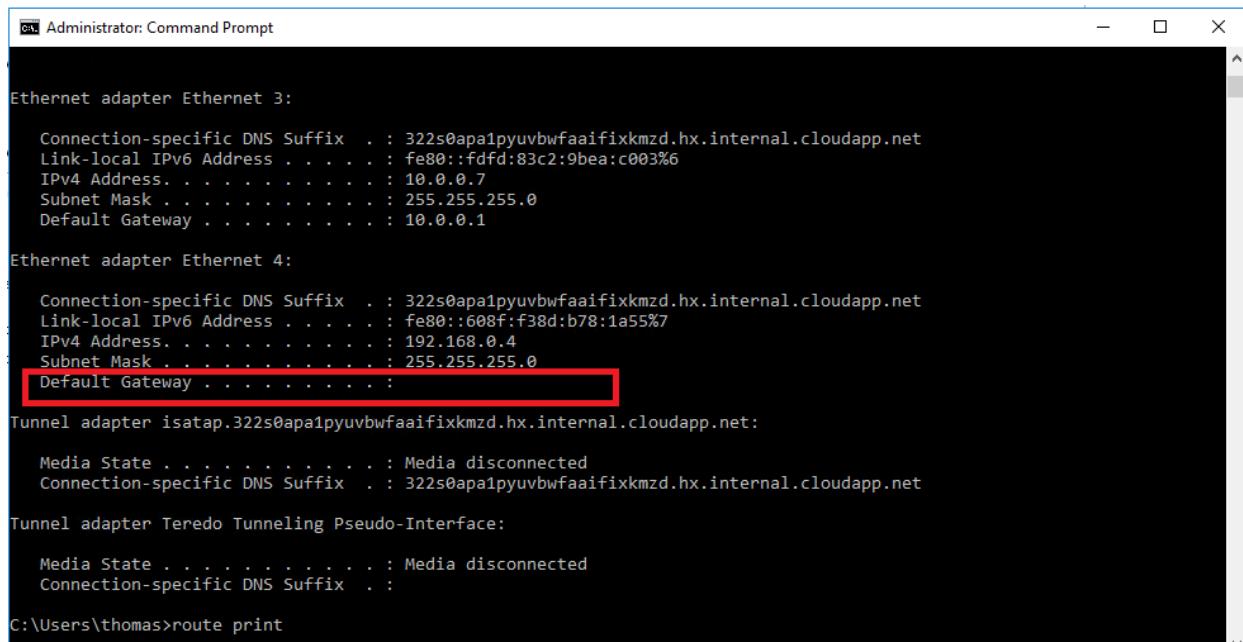
Issue Type	Value	Resolution Action
Memory	Usage	Follow the recommendations that are described in <a href="#">Generic performance troubleshooting for Azure Virtual Machine running Linux or Windows</a> .
Guest Firewall	Name of the firewall blocking	Follow these steps: <a href="#">Turn Windows Defender Firewall on or off</a> .
DNS Resolution	Name of the DNS	Follow these steps: <a href="#">Azure DNS troubleshooting guide</a> and <a href="#">Name resolution for resources in Azure virtual networks</a> .
Socket Error	Not applicable	The specified port is already in use by another application. Try to use a different port.

3. Run the connectivity check again to determine whether the problem is resolved.

### Azure VM cannot connect to the second network adapter of an Azure VM in same virtual network

#### Step 1: Make sure that the second network adapter is enabled to talk outside the subnet.

By default, secondary network adapters (also known as network interface cards, or network adapters) are not configured to have a default gateway. Therefore, the traffic flow on the secondary adapter will be limited to the same subnet.



```

Administrator: Command Prompt
Ethernet adapter Ethernet 3:

Connection-specific DNS Suffix . : 322s0apa1pyuvbwfaaifixkmzd.hx.internal.cloudapp.net
Link-local IPv6 Address : fe80::fdff:83c2:9bea:c003%6
IPv4 Address : 10.0.0.7
Subnet Mask : 255.255.255.0
Default Gateway : 10.0.0.1

Ethernet adapter Ethernet 4:

Connection-specific DNS Suffix . : 322s0apa1pyuvbwfaaifixkmzd.hx.internal.cloudapp.net
Link-local IPv6 Address : fe80::608f:f38d:b78:1a55%7
IPv4 Address : 192.168.0.4
Subnet Mask : 255.255.255.0
Default Gateway : [REDACTED]

Tunnel adapter isatap.322s0apa1pyuvbwfaaifixkmzd.hx.internal.cloudapp.net:

Media State : Media disconnected
Connection-specific DNS Suffix . : 322s0apa1pyuvbwfaaifixkmzd.hx.internal.cloudapp.net

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Media State : Media disconnected
Connection-specific DNS Suffix . :

C:\Users\thomas>route print

```

If users want to enable secondary network adapters to talk outside their own subnet, they must add an entry to the routing table to configure the gateway. To do this, follow these steps:

1. On the VM that has the second network adapter configured, open a Command Prompt window as an administrator.
2. Run the following command to add the entry in routing table:

```
Route add 0.0.0.0 mask 0.0.0.0 -p <Gateway IP>
```

For example, if the second IP address is 192.168.0.4, the gateway IP should be 192.168.0.1. You have to run the following command:

```
Route add 0.0.0.0 mask 0.0.0.0 -p 192.168.0.1
```

3. Run route print. If the entry is added successfully, you will see an entry that resembles the following:

```
Administrator: Command Prompt
IPv4 Route Table
=====
Active Routes:
Network Destination Netmask Gateway Interface Metric
 0.0.0.0 0.0.0.0 10.0.0.1 10.0.0.7 10
 0.0.0.0 0.0.0.0 192.168.0.1 192.168.0.4 266
 10.0.0.0 255.255.255.0 On-link 10.0.0.7 266
 10.0.0.7 255.255.255.255 On-link 10.0.0.7 266
 10.0.0.255 255.255.255.255 On-link 10.0.0.7 266
 127.0.0.0 255.0.0.0 On-link 127.0.0.1 331
 127.0.0.1 255.255.255.255 On-link 127.0.0.1 331
127.255.255.255 255.255.255.255 On-link 127.0.0.1 331
 168.63.129.16 255.255.255.255 On-link 10.0.0.1 11
 169.254.169.254 255.255.255.255 On-link 10.0.0.1 11
 192.168.0.0 255.255.255.0 On-link 192.168.0.4 266
 192.168.0.4 255.255.255.255 On-link 192.168.0.4 266
192.168.0.255 255.255.255.255 On-link 192.168.0.4 266
 224.0.0.0 240.0.0.0 On-link 127.0.0.1 331
 224.0.0.0 240.0.0.0 On-link 10.0.0.7 266
 224.0.0.0 240.0.0.0 On-link 192.168.0.4 266
255.255.255.255 255.255.255.255 On-link 127.0.0.1 331
255.255.255.255 255.255.255.255 On-link 10.0.0.7 266
 255.255.255.255 255.255.255.255 On-link 192.168.0.4 266
=====
Persistent Routes:
Network Address Netmask Gateway Address Metric
 0.0.0.0 0.0.0.0 192.168.0.1 Default
```

Now, try to connect to secondary network adapter. If the connection is still unsuccessful, go to next step.

#### **Step 2: Check NSG settings for the network adapters.**

For both the primary and secondary network adapters, check the default Inbound port rules (Allow VNet Inbound, Allow Load Balancer) to inbound on both network adapters. You should also make sure that there are no matching blocking rules that have a lower-priority rule above them.

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	
1000	default-allow-rdp	3389	TCP	Any	Any	<input checked="" type="checkbox"/> Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Allow	...
65001	AllowAzureLoadBalancerIn...	Any	Any	AzureLoadBal...	Any	<input checked="" type="checkbox"/> Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	<input checked="" type="checkbox"/> Deny	...

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Allow	...
65001	AllowInternetOutBound	Any	Any	Any	Internet	<input checked="" type="checkbox"/> Allow	...

#### **Step 3: Run a connectivity check to the secondary network adapter.**

1. Run a connectivity check to the secondary network adapter.
2. Run a connectivity check across the environment to make sure that the process works end to end.

For more information about how to run the connectivity check, see the following articles:

**Windows:** [Check connectivity with Azure Network Watcher using PowerShell](#)

**Linux:** [Check connectivity with Azure Network Watcher using Azure CLI 2.0.](#)

The following is an example response:

```
ConnectionStatus : Unreachable
AvgLatencyInMs :
MinLatencyInMs :
MaxLatencyInMs :
ProbesSent : 100
ProbesFailed : 100
Hops : [
 {
 "Type": "Source",
 "Id": "c5222ea0-3213-4f85-a642-cee63217c2f3",
 "Address": "10.1.1.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/appNic0/ipConfigurations/ipconfig1",
 "NextHopIds": [
 "9283a9f0-cc5e-4239-8f5e-ae0f3c19fbaa"
],
 "Issues": []
 },
 {
 "Type": "VirtualAppliance",
 "Id": "9283a9f0-cc5e-4239-8f5e-ae0f3c19fbaa",
 "Address": "10.1.2.4",
 "ResourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ContosoRG/providers/Microsoft.Network/networkInterfaces/fwNic/ipConfigurations/ipconfig1",
 "NextHopIds": [
 "0f1500cd-c512-4d43-b431-7267e4e67017"
],
 "Issues": []
 }
],
```

**Step 4:** Refer the table under [Step 5](#), and follow these steps to resolve the issues.

#### Azure VM cannot connect to the internet

**Step 1: Check whether the network adapter is in a failed state.**

Follow these steps to check the state of the NIC:

1. Log in to the Resource Explorer portal.
2. In the left pane, select **Subscriptions**.
3. In the left pane, select the resource group that the affected network adapter or VM belongs to.
4. Go to the **Microsoft Network**.
5. Select the **Network Interfaces** option.
6. Select the affected network interface.
7. Select the **Read/Write** option at the top of the portal.
8. Select the **Edit** option.

The screenshot shows the Azure Resource Explorer interface. On the left, a tree view lists resource groups and their components. Under 'testRG', 'Microsoft.Network' is selected, and 'testvm429' is selected under it. On the right, a detailed view of 'testvm429' is shown with tabs for 'Data (GET, PUT)', 'Actions (POST, DELETE)', 'Create', 'Documentation', 'PowerShell', and 'Azure CLI 2.0'. The 'Edit' button is highlighted with a red box. Below the tabs, a code editor displays the JSON representation of the resource, specifically focusing on the 'ipConfigurations' section.

#### NOTE

After you select the **Edit** option, the "Get" option changes to a **Put** option.

This screenshot shows the 'PUT' option highlighted with a red box in the top navigation bar of the Azure Resource Explorer. The JSON code editor below shows the properties of a network interface named 'nic2'. The 'ipConfigurations' section is expanded, and the 'provisioningState' field is highlighted with a red box, indicating it is the target for modification.

9. Select the affected network interface, and then select the **Put** option.

#### NOTE

After you make this selection, the **ProvisioningState** is displayed as **Updating**. The same result is shown on the regular Azure Resource Manager portal. If the operation is completed successfully, the **ProvisioningState** value changes to **Succeeded**, as shown.

10. Refresh your portal. The network adapter should be in a success state.

**Step 2:** Follow [Step 4](#) to run a connectivity check.

**Step 3:** Refer the table under [Step 5](#), and follow the steps to resolve the issues.

## Next steps

[Troubleshooting connectivity problems between Azure VMs](#)

# Azure Policy built-in definitions for Azure Virtual Network

2/18/2021 • 10 minutes to read • [Edit Online](#)

This page is an index of [Azure Policy](#) built-in policy definitions for Azure Virtual Network. For additional Azure Policy built-ins for other services, see [Azure Policy built-in definitions](#).

The name of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Version** column to view the source on the [Azure Policy GitHub repo](#).

## Azure Virtual Network

Name (Azure Portal)	Description	Effect(s)	Version (GitHub)
<a href="#">A custom IPsec/IKE policy must be applied to all Azure virtual network gateway connections</a>	This policy ensures that all Azure virtual network gateway connections use a custom Internet Protocol Security(IPsec)/Internet Key Exchange(IKE) policy. Supported algorithms and key strengths - <a href="https://aka.ms/AA62kb0">https://aka.ms/AA62kb0</a>	Audit, Disabled	1.0.0
<a href="#">All Internet traffic should be routed via your deployed Azure Firewall</a>	Azure Security Center has identified that some of your subnets aren't protected with a next generation firewall. Protect your subnets from potential threats by restricting access to them with Azure Firewall or a supported next generation firewall	AuditIfNotExists, Disabled	3.0.0-preview
<a href="#">App Service should use a virtual network service endpoint</a>	This policy audits any App Service not configured to use a virtual network service endpoint.	AuditIfNotExists, Disabled	1.0.0
<a href="#">Azure VPN gateways should not use 'basic' SKU</a>	This policy ensures that VPN gateways do not use 'basic' SKU.	Audit, Disabled	1.0.0
<a href="#">Container Registry should use a virtual network service endpoint</a>	This policy audits any Container Registry not configured to use a virtual network service endpoint.	Audit, Disabled	1.0.0-preview
<a href="#">Cosmos DB should use a virtual network service endpoint</a>	This policy audits any Cosmos DB not configured to use a virtual network service endpoint.	Audit, Disabled	1.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
Deploy a flow log resource with target network security group	Configures flow log for specific network security group. It will allow to log information about IP traffic flowing through an network security group. Flow log helps to identify unknown or undesired traffic, verify network isolation and compliance with enterprise access rules, analyze network flows from compromised IPs and network interfaces.	deployIfNotExists	1.0.0
Deploy network watcher when virtual networks are created	This policy creates a network watcher resource in regions with virtual networks. You need to ensure existence of a resource group named networkWatcherRG, which will be used to deploy network watcher instances.	DeployIfNotExists	1.0.0
Event Hub should use a virtual network service endpoint	This policy audits any Event Hub not configured to use a virtual network service endpoint.	AuditIfExists, Disabled	1.0.0
Flow log should be configured for every network security group	Audit for network security groups to verify if flow log resource is configured. Flow log allows to log information about IP traffic flowing through network security group. It can be used for optimizing network flows, monitoring throughput, verifying compliance, detecting intrusions and more.	audit	1.0.0
Gateway subnets should not be configured with a network security group	This policy denies if a gateway subnet is configured with a network security group. Assigning a network security group to a gateway subnet will cause the gateway to stop functioning.	deny	1.0.0
Key Vault should use a virtual network service endpoint	This policy audits any Key Vault not configured to use a virtual network service endpoint.	Audit, Disabled	1.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
Network interfaces should disable IP forwarding	This policy denies the network interfaces which enabled IP forwarding. The setting of IP forwarding disables Azure's check of the source and destination for a network interface. This should be reviewed by the network security team.	deny	1.0.0
Network interfaces should not have public IPs	This policy denies the network interfaces which are configured with any public IP. Public IP addresses allow internet resources to communicate inbound to Azure resources, and Azure resources to communicate outbound to the internet. This should be reviewed by the network security team.	deny	1.0.0
Network Watcher should be enabled	Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level in, to, and from Azure. Scenario level monitoring enables you to diagnose problems at an end to end network level view. Network diagnostic and visualization tools available with Network Watcher help you understand, diagnose, and gain insights to your network in Azure.	auditIfNotExists	1.1.0
RDP access from the Internet should be blocked	This policy audits any network security rule that allows RDP access from Internet	Audit, Disabled	2.0.0
Service Bus should use a virtual network service endpoint	This policy audits any Service Bus not configured to use a virtual network service endpoint.	AuditIfNotExists, Disabled	1.0.0
SQL Server should use a virtual network service endpoint	This policy audits any SQL Server not configured to use a virtual network service endpoint.	AuditIfNotExists, Disabled	1.0.0
SSH access from the Internet should be blocked	This policy audits any network security rule that allows SSH access from Internet	Audit, Disabled	2.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
<a href="#">Storage Accounts should use a virtual network service endpoint</a>	This policy audits any Storage Account not configured to use a virtual network service endpoint.	Audit, Disabled	1.0.0
<a href="#">Virtual machines should be connected to an approved virtual network</a>	This policy audits any virtual machine connected to a virtual network that is not approved.	Audit, Deny, Disabled	1.0.0
<a href="#">Virtual networks should use specified virtual network gateway</a>	This policy audits any virtual network if the default route does not point to the specified virtual network gateway.	AuditIfNotExists, Disabled	1.0.0
<a href="#">Web Application Firewall (WAF) should be enabled for Application Gateway</a>	Deploy Azure Web Application Firewall (WAF) in front of public facing web applications for additional inspection of incoming traffic. Web Application Firewall (WAF) provides centralized protection of your web applications from common exploits and vulnerabilities such as SQL injections, Cross-Site Scripting, local and remote file executions. You can also restrict access to your web applications by countries, IP address ranges, and other http(s) parameters via custom rules.	Audit, Deny, Disabled	1.0.1
<a href="#">Web Application Firewall (WAF) should be enabled for Azure Front Door Service service</a>	Deploy Azure Web Application Firewall (WAF) in front of public facing web applications for additional inspection of incoming traffic. Web Application Firewall (WAF) provides centralized protection of your web applications from common exploits and vulnerabilities such as SQL injections, Cross-Site Scripting, local and remote file executions. You can also restrict access to your web applications by countries, IP address ranges, and other http(s) parameters via custom rules.	Audit, Deny, Disabled	1.0.1

NAME	DESCRIPTION	EFFECT(S)	VERSION
<a href="#">Web Application Firewall (WAF) should use the specified mode for Application Gateway</a>	Mandates the use of 'Detection' or 'Prevention' mode to be active on all Web Application Firewall policies for Application Gateway.	Audit, Deny, Disabled	<a href="#">1.0.0</a>
<a href="#">Web Application Firewall (WAF) should use the specified mode for Azure Front Door Service</a>	Mandates the use of 'Detection' or 'Prevention' mode to be active on all Web Application Firewall policies for Azure Front Door Service.	Audit, Deny, Disabled	<a href="#">1.0.0</a>

## Tags

NAME (AZURE PORTAL)	DESCRIPTION	EFFECT(S)	VERSION (GITHUB)
<a href="#">Add a tag to resource groups</a>	Adds the specified tag and value when any resource group missing this tag is created or updated. Existing resource groups can be remediated by triggering a remediation task. If the tag exists with a different value it will not be changed.	modify	<a href="#">1.0.0</a>
<a href="#">Add a tag to resources</a>	Adds the specified tag and value when any resource missing this tag is created or updated. Existing resources can be remediated by triggering a remediation task. If the tag exists with a different value it will not be changed. Does not modify tags on resource groups.	modify	<a href="#">1.0.0</a>
<a href="#">Add a tag to subscriptions</a>	Adds the specified tag and value to subscriptions via a remediation task. If the tag exists with a different value it will not be changed. See <a href="https://aka.ms/azurepolicyremediation">https://aka.ms/azurepolicyremediation</a> for more information on policy remediation.	modify	<a href="#">1.0.0</a>

NAME	DESCRIPTION	EFFECT(S)	VERSION
Add or replace a tag on resource groups	Adds or replaces the specified tag and value when any resource group is created or updated. Existing resource groups can be remediated by triggering a remediation task.	modify	1.0.0
Add or replace a tag on resources	Adds or replaces the specified tag and value when any resource is created or updated. Existing resources can be remediated by triggering a remediation task. Does not modify tags on resource groups.	modify	1.0.0
Add or replace a tag on subscriptions	Adds or replaces the specified tag and value on subscriptions via a remediation task. Existing resource groups can be remediated by triggering a remediation task. See <a href="https://aka.ms/azurepolicyremediation">https://aka.ms/azurepolicyremediation</a> for more information on policy remediation.	modify	1.0.0
Append a tag and its value from the resource group	Appends the specified tag with its value from the resource group when any resource which is missing this tag is created or updated. Does not modify the tags of resources created before this policy was applied until those resources are changed. New 'modify' effect policies are available that support remediation of tags on existing resources (see <a href="https://aka.ms/modifydoc">https://aka.ms/modifydoc</a> ).	append	1.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
Append a tag and its value to resource groups	Appends the specified tag and value when any resource group which is missing this tag is created or updated. Does not modify the tags of resource groups created before this policy was applied until those resource groups are changed. New 'modify' effect policies are available that support remediation of tags on existing resources (see <a href="https://aka.ms/modifydoc">https://aka.ms/modifydoc</a> ).	append	1.0.0
Append a tag and its value to resources	Appends the specified tag and value when any resource which is missing this tag is created or updated. Does not modify the tags of resources created before this policy was applied until those resources are changed. Does not apply to resource groups. New 'modify' effect policies are available that support remediation of tags on existing resources (see <a href="https://aka.ms/modifydoc">https://aka.ms/modifydoc</a> ).	append	1.0.1
Inherit a tag from the resource group	Adds or replaces the specified tag and value from the parent resource group when any resource is created or updated. Existing resources can be remediated by triggering a remediation task.	modify	1.0.0
Inherit a tag from the resource group if missing	Adds the specified tag with its value from the parent resource group when any resource missing this tag is created or updated. Existing resources can be remediated by triggering a remediation task. If the tag exists with a different value it will not be changed.	modify	1.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
Inherit a tag from the subscription	Adds or replaces the specified tag and value from the containing subscription when any resource is created or updated. Existing resources can be remediated by triggering a remediation task.	modify	1.0.0
Inherit a tag from the subscription if missing	Adds the specified tag with its value from the containing subscription when any resource missing this tag is created or updated. Existing resources can be remediated by triggering a remediation task. If the tag exists with a different value it will not be changed.	modify	1.0.0
Require a tag and its value on resource groups	Enforces a required tag and its value on resource groups.	deny	1.0.0
Require a tag and its value on resources	Enforces a required tag and its value. Does not apply to resource groups.	deny	1.0.1
Require a tag on resource groups	Enforces existence of a tag on resource groups.	deny	1.0.0
Require a tag on resources	Enforces existence of a tag. Does not apply to resource groups.	deny	1.0.1

## General

NAME (AZURE PORTAL)	DESCRIPTION	EFFECT(S)	VERSION (GITHUB)
Allowed locations	This policy enables you to restrict the locations your organization can specify when deploying resources. Use to enforce your geo-compliance requirements. Excludes resource groups, Microsoft.AzureActiveDirectory/b2cDirectories, and resources that use the 'global' region.	deny	1.0.0

NAME	DESCRIPTION	EFFECT(S)	VERSION
Allowed locations for resource groups	This policy enables you to restrict the locations your organization can create resource groups in. Use to enforce your geo-compliance requirements.	deny	1.0.0
Allowed resource types	This policy enables you to specify the resource types that your organization can deploy. Only resource types that support 'tags' and 'location' will be affected by this policy. To restrict all resources please duplicate this policy and change the 'mode' to 'All'.	deny	1.0.0
Audit resource location matches resource group location	Audit that the resource location matches its resource group location	audit	2.0.0
Audit usage of custom RBAC rules	Audit built-in roles such as 'Owner, Contributor, Reader' instead of custom RBAC roles, which are error prone. Using custom roles is treated as an exception and requires a rigorous review and threat modeling	Audit, Disabled	1.0.0
Custom subscription owner roles should not exist	This policy ensures that no custom subscription owner roles exist.	Audit, Disabled	2.0.0
Not allowed resource types	This policy enables you to specify the resource types that your organization cannot deploy.	Deny	1.0.0

## Next steps

- See the built-ins on the [Azure Policy GitHub repo](#).
- Review the [Azure Policy definition structure](#).
- Review [Understanding policy effects](#).

# Azure Virtual Network frequently asked questions (FAQ)

2/18/2021 • 29 minutes to read • [Edit Online](#)

## Virtual Network basics

### What is an Azure Virtual Network (VNet)?

An Azure Virtual Network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can use VNets to provision and manage virtual private networks (VPNs) in Azure and, optionally, link the VNets with other VNets in Azure, or with your on-premises IT infrastructure to create hybrid or cross-premises solutions. Each VNet you create has its own CIDR block and can be linked to other VNets and on-premises networks as long as the CIDR blocks do not overlap. You also have control of DNS server settings for VNets, and segmentation of the VNet into subnets.

Use VNets to:

- Create a dedicated private cloud-only VNet. Sometimes you don't require a cross-premises configuration for your solution. When you create a VNet, your services and VMs within your VNet can communicate directly and securely with each other in the cloud. You can still configure endpoint connections for the VMs and services that require Internet communication, as part of your solution.
- Securely extend your data center. With VNets, you can build traditional site-to-site (S2S) VPNs to securely scale your datacenter capacity. S2S VPNs use IPSEC to provide a secure connection between your corporate VPN gateway and Azure.
- Enable hybrid cloud scenarios. VNets give you the flexibility to support a range of hybrid cloud scenarios. You can securely connect cloud-based applications to any type of on-premises system such as mainframes and Unix systems.

### How do I get started?

Visit the [Virtual network documentation](#) to get started. This content provides overview and deployment information for all of the VNet features.

### Can I use VNets without cross-premises connectivity?

Yes. You can use a VNet without connecting it to your premises. For example, you could run Microsoft Windows Server Active Directory domain controllers and SharePoint farms solely in an Azure VNet.

### Can I perform WAN optimization between VNets or a VNet and my on-premises data center?

Yes. You can deploy a [WAN optimization network virtual appliance](#) from several vendors through the Azure Marketplace.

## Configuration

### What tools do I use to create a VNet?

You can use the following tools to create or configure a VNet:

- Azure portal
- PowerShell
- Azure CLI
- A network configuration file (netcfg - for classic VNets only). See the [Configure a VNet using a network](#)

[configuration file](#) article.

## What address ranges can I use in my VNets?

We recommend that you use the address ranges enumerated in [RFC 1918](#), which have been set aside by the IETF for private, non-routable address spaces:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Other address spaces may work but may have undesirable side effects.

In addition, you cannot add the following address ranges:

- 224.0.0.0/4 (Multicast)
- 255.255.255.255/32 (Broadcast)
- 127.0.0.0/8 (Loopback)
- 169.254.0.0/16 (Link-local)
- 168.63.129.16/32 (Internal DNS)

## Can I have public IP addresses in my VNets?

Yes. For more information about public IP address ranges, see [Create a virtual network](#). Public IP addresses are not directly accessible from the internet.

## Is there a limit to the number of subnets in my VNet?

Yes. See [Azure limits](#) for details. Subnet address spaces cannot overlap one another.

## Are there any restrictions on using IP addresses within these subnets?

Yes. Azure reserves 5 IP addresses within each subnet. These are x.x.x.0-x.x.x.3 and the last address of the subnet. x.x.x.1-x.x.x.3 is reserved in each subnet for Azure services.

- x.x.x.0: Network address
- x.x.x.1: Reserved by Azure for the default gateway
- x.x.x.2, x.x.x.3: Reserved by Azure to map the Azure DNS IPs to the VNet space
- x.x.x.255: Network broadcast address

## How small and how large can VNets and subnets be?

The smallest supported IPv4 subnet is /29, and the largest is /8 (using CIDR subnet definitions). IPv6 subnets must be exactly /64 in size.

## Can I bring my VLANs to Azure using VNets?

No. VNets are Layer-3 overlays. Azure does not support any Layer-2 semantics.

## Can I specify custom routing policies on my VNets and subnets?

Yes. You can create a route table and associate it to a subnet. For more information about routing in Azure, see [Routing overview](#).

## Do VNets support multicast or broadcast?

No. Multicast and broadcast are not supported.

## What protocols can I use within VNets?

You can use TCP, UDP, and ICMP TCP/IP protocols within VNets. Unicast is supported within VNets, with the exception of Dynamic Host Configuration Protocol (DHCP) via Unicast (source port UDP/68 / destination port UDP/67) and UDP source port 65330 which is reserved for the host. Multicast, broadcast, IP-in-IP encapsulated packets, and Generic Routing Encapsulation (GRE) packets are blocked within VNets.

**Can I ping my default routers within a VNet?**

No.

**Can I use tracert to diagnose connectivity?**

No.

**Can I add subnets after the VNet is created?**

Yes. Subnets can be added to VNets at any time as long as the subnet address range is not part of another subnet and there is available space left in the virtual network's address range.

**Can I modify the size of my subnet after I create it?**

Yes. You can add, remove, expand, or shrink a subnet if there are no VMs or services deployed within it.

**Can I modify Vnet after I created them?**

Yes. You can add, remove, and modify the CIDR blocks used by a VNet.

**If I am running my services in a VNet, can I connect to the internet?**

Yes. All services deployed within a VNet can connect outbound to the internet. To learn more about outbound internet connections in Azure, see [Outbound connections](#). If you want to connect inbound to a resource deployed through Resource Manager, the resource must have a public IP address assigned to it. To learn more about public IP addresses, see [Public IP addresses](#). Every Azure Cloud Service deployed in Azure has a publicly addressable VIP assigned to it. You define input endpoints for PaaS roles and endpoints for virtual machines to enable these services to accept connections from the internet.

**Do VNets support IPv6?**

Yes, VNets can be IPv4-only or dual stack (IPv4+IPv6). For details, see [Overview of IPv6 for Azure Virtual Networks](#).

**Can a VNet span regions?**

No. A VNet is limited to a single region. A virtual network does, however, span availability zones. To learn more about availability zones, see [Availability zones overview](#). You can connect virtual networks in different regions with virtual network peering. For details, see [Virtual network peering overview](#)

**Can I connect a VNet to another VNet in Azure?**

Yes. You can connect one VNet to another VNet using either:

- **Virtual network peering**: For details, see [VNet peering overview](#)
- **An Azure VPN Gateway**: For details, see [Configure a VNet-to-VNet connection](#).

## Name Resolution (DNS)

**What are my DNS options for VNets?**

Use the decision table on the [Name Resolution for VMs and Role Instances](#) page to guide you through all the DNS options available.

**Can I specify DNS servers for a VNet?**

Yes. You can specify DNS server IP addresses in the VNet settings. The setting is applied as the default DNS server(s) for all VMs in the VNet.

**How many DNS servers can I specify?**

Reference [Azure limits](#).

**Can I modify my DNS servers after I have created the network?**

Yes. You can change the DNS server list for your VNet at any time. If you change your DNS server list, you need to perform a DHCP lease renewal on all affected VMs in the VNet, for the new DNS settings to take effect. For

VMs running Windows OS you can do this by typing `ipconfig /renew` directly on the VM. For other OS types, refer to the DHCP lease renewal documentation for the specific OS type.

### What is Azure-provided DNS and does it work with VNets?

Azure-provided DNS is a multi-tenant DNS service offered by Microsoft. Azure registers all of your VMs and cloud service role instances in this service. This service provides name resolution by hostname for VMs and role instances contained within the same cloud service, and by FQDN for VMs and role instances in the same VNet. To learn more about DNS, see [Name Resolution for VMs and Cloud Services role instances](#).

There is a limitation to the first 100 cloud services in a VNet for cross-tenant name resolution using Azure-provided DNS. If you are using your own DNS server, this limitation does not apply.

### Can I override my DNS settings on a per-VM or cloud service basis?

Yes. You can set DNS servers per VM or cloud service to override the default network settings. However, it's recommended that you use network-wide DNS as much as possible.

### Can I bring my own DNS suffix?

No. You cannot specify a custom DNS suffix for your VNets.

## Connecting virtual machines

### Can I deploy VMs to a VNet?

Yes. All network interfaces (NIC) attached to a VM deployed through the Resource Manager deployment model must be connected to a VNet. VMs deployed through the classic deployment model can optionally be connected to a VNet.

### What are the different types of IP addresses I can assign to VMs?

- **Private:** Assigned to each NIC within each VM. The address is assigned using either the static or dynamic method. Private IP addresses are assigned from the range that you specified in the subnet settings of your VNet. Resources deployed through the classic deployment model are assigned private IP addresses, even if they're not connected to a VNet. The behavior of the allocation method is different depending on whether a resource was deployed with the Resource Manager or classic deployment model:
  - **Resource Manager:** A private IP address assigned with the dynamic or static method remains assigned to a virtual machine (Resource Manager) until the resource is deleted. The difference is that you select the address to assign when using static, and Azure chooses when using dynamic.
  - **Classic:** A private IP address assigned with the dynamic method may change when a virtual machine (classic) VM is restarted after having been in the stopped (deallocated) state. If you need to ensure that the private IP address for a resource deployed through the classic deployment model never changes, assign a private IP address with the static method.
- **Public:** Optionally assigned to NICs attached to VMs deployed through the Azure Resource Manager deployment model. The address can be assigned with the static or dynamic allocation method. All VMs and Cloud Services role instances deployed through the classic deployment model exist within a cloud service, which is assigned a *dynamic*, public virtual IP (VIP) address. A public *static* IP address, called a [Reserved IP address](#), can optionally be assigned as a VIP. You can assign public IP addresses to individual VMs or Cloud Services role instances deployed through the classic deployment model. These addresses are called [Instance level public IP \(ILPIP\)](#) addresses and can be assigned dynamically.

### Can I reserve a private IP address for a VM that I will create at a later time?

No. You cannot reserve a private IP address. If a private IP address is available, it is assigned to a VM or role instance by the DHCP server. The VM may or may not be the one that you want the private IP address assigned to. You can, however, change the private IP address of an already created VM, to any available private IP address.

### Do private IP addresses change for VMs in a VNet?

It depends. If the VM was deployed through Resource Manager, no, regardless of whether the IP address was assigned with the static or dynamic allocation method. If the VM was deployed through the classic deployment model, dynamic IP addresses can change when a VM is started after having been in the stopped (deallocated) state. The address is released from a VM deployed through either deployment model when the VM is deleted.

### **Can I manually assign IP addresses to NICs within the VM operating system?**

Yes, but it's not recommended unless necessary, such as when assigning multiple IP addresses to a virtual machine. For details, see [Adding multiple IP addresses to a virtual machine](#). If the IP address assigned to an Azure NIC attached to a VM changes, and the IP address within the VM operating system is different, you lose connectivity to the VM.

### **If I stop a Cloud Service deployment slot or shutdown a VM from within the operating system, what happens to my IP addresses?**

Nothing. The IP addresses (public VIP, public, and private) remain assigned to the cloud service deployment slot or VM.

### **Can I move VMs from one subnet to another subnet in a VNet without redeploying?**

Yes. You can find more information in the [How to move a VM or role instance to a different subnet](#) article.

### **Can I configure a static MAC address for my VM?**

No. A MAC address cannot be statically configured.

### **Will the MAC address remain the same for my VM once it's created?**

Yes, the MAC address remains the same for a VM deployed through both the Resource Manager and classic deployment models until it's deleted. Previously, the MAC address was released if the VM was stopped (deallocated), but now the MAC address is retained even when the VM is in the deallocated state. The MAC address remains assigned to the network interface until the network interface is deleted or the private IP address assigned to the primary IP configuration of the primary network interface is changed.

### **Can I connect to the internet from a VM in a VNet?**

Yes. All VMs and Cloud Services role instances deployed within a VNet can connect to the Internet.

## Azure services that connect to VNets

### **Can I use Azure App Service Web Apps with a VNet?**

Yes. You can deploy Web Apps inside a VNet using an ASE (App Service Environment), connect the backend of your apps to your VNets with VNet Integration, and lock down inbound traffic to your app with service endpoints. For more information, see the following articles:

- [App Service networking features](#)
- [Creating Web Apps in an App Service Environment](#)
- [Integrate your app with an Azure Virtual Network](#)
- [App Service access restrictions](#)

### **Can I deploy Cloud Services with web and worker roles (PaaS) in a VNet?**

Yes. You can (optionally) deploy Cloud Services role instances within VNets. To do so, you specify the VNet name and the role/subnet mappings in the network configuration section of your service configuration. You do not need to update any of your binaries.

### **Can I connect a virtual machine scale set to a VNet?**

Yes. You must connect a virtual machine scale set to a VNet.

### **Is there a complete list of Azure services that can I deploy resources from into a VNet?**

Yes, For details, see [Virtual network integration for Azure services](#).

## **How can I restrict access to Azure PaaS resources from a VNet?**

Resources deployed through some Azure PaaS services (such as Azure Storage and Azure SQL Database), can restrict network access to VNet through the use of virtual network service endpoints or Azure Private Link. For details, see [Virtual network service endpoints overview](#), [Azure Private Link overview](#)

## **Can I move my services in and out of VNets?**

No. You cannot move services in and out of VNets. To move a resource to another VNet, you have to delete and redeploy the resource.

# Security

## **What is the security model for VNets?**

VNets are isolated from one another, and other services hosted in the Azure infrastructure. A VNet is a trust boundary.

## **Can I restrict inbound or outbound traffic flow to VNet-connected resources?**

Yes. You can apply [Network Security Groups](#) to individual subnets within a VNet, NICs attached to a VNet, or both.

## **Can I implement a firewall between VNet-connected resources?**

Yes. You can deploy a [firewall network virtual appliance](#) from several vendors through the Azure Marketplace.

## **Is there information available about securing VNets?**

Yes. For details, see [Azure Network Security Overview](#).

## **Do Virtual Networks store customer data?**

No. Virtual Networks doesn't store any customer data.

# APIs, schemas, and tools

## **Can I manage VNets from code?**

Yes. You can use REST APIs for VNets in the [Azure Resource Manager](#) and [classic](#) deployment models.

## **Is there tooling support for VNets?**

Yes. Learn more about using:

- The Azure portal to deploy VNets through the [Azure Resource Manager](#) and [classic](#) deployment models.
- PowerShell to manage VNets deployed through the [Resource Manager](#) and [classic](#) deployment models.
- The Azure command-line interface (CLI) to deploy and manage VNets deployed through the [Resource Manager](#) and [classic](#) deployment models.

# VNet peering

## **What is VNet peering?**

VNet peering (or virtual network peering) enables you to connect virtual networks. A VNet peering connection between virtual networks enables you to route traffic between them privately through IPv4 addresses. Virtual machines in the peered VNets can communicate with each other as if they are within the same network. These virtual networks can be in the same region or in different regions (also known as Global VNet Peering). VNet peering connections can also be created across Azure subscriptions.

## **Can I create a peering connection to a VNet in a different region?**

Yes. Global VNet peering enables you to peer VNets in different regions. Global VNet peering is available in all Azure public regions, China cloud regions, and Government cloud regions. You cannot globally peer from Azure

public regions to national cloud regions.

### **What are the constraints related to Global VNet Peering and Load Balancers?**

If the two virtual networks in two different regions are peered over Global VNet Peering, you cannot connect to resources that are behind a Basic Load Balancer through the Front End IP of the Load Balancer. This restriction does not exist for a Standard Load Balancer. The following resources can use Basic Load Balancers which means you cannot reach them through the Load Balancer's Front End IP over Global VNet Peering. You can however use Global VNet peering to reach the resources directly through their private VNet IPs, if permitted.

- VMs behind Basic Load Balancers
- Virtual machine scale sets with Basic Load Balancers
- Redis Cache
- Application Gateway (v1) SKU
- Service Fabric
- API Management
- Active Directory Domain Service (ADDS)
- Logic Apps
- HDInsight
- Azure Batch
- App Service Environment

You can connect to these resources via ExpressRoute or VNet-to-VNet through VNet Gateways.

### **Can I enable VNet Peering if my virtual networks belong to subscriptions within different Azure Active Directory tenants?**

Yes. It is possible to establish VNet Peering (whether local or global) if your subscriptions belong to different Azure Active Directory tenants. You can do this via Portal, PowerShell or CLI.

### **My VNet peering connection is in *Initiated* state, why can't I connect?**

If your peering connection is in an *Initiated* state, this means you have created only one link. A bidirectional link must be created in order to establish a successful connection. For example, to peer VNet A to VNet B, a link must be created from VNetA to VNetB and from VNetB to VNetA. Creating both links will change the state to *Connected*.

### **My VNet peering connection is in *Disconnected* state, why can't I create a peering connection?**

If your VNet peering connection is in a *Disconnected* state, it means one of the links created was deleted. In order to re-establish a peering connection, you will need to delete the link and recreate it.

### **Can I peer my VNet with a VNet in a different subscription?**

Yes. You can peer VNets across subscriptions and across regions.

### **Can I peer two VNets with matching or overlapping address ranges?**

No. Address spaces must not overlap to enable VNet Peering.

### **Can I peer a VNet to two different VNets with the the 'Use Remote Gateway' option enabled on both the peerings?**

No. You can only enable the 'Use Remote Gateway' option on one peering to one of the VNets.

### **How much do VNet peering links cost?**

There is no charge for creating a VNet peering connection. Data transfer across peering connections is charged. [See here](#).

### **Is VNet peering traffic encrypted?**

When Azure traffic moves between datacenters (outside physical boundaries not controlled by Microsoft or on

behalf of Microsoft), [MACsec data-link layer encryption](#) is utilized on the underlying network hardware. This is applicable to VNet peering traffic.

#### **Why is my peering connection in a *Disconnected* state?**

VNet peering connections go into *Disconnected* state when one VNet peering link is deleted. You must delete both links in order to reestablish a successful peering connection.

#### **If I peer VNetA to VNetB and I peer VNetB to VNetC, does that mean VNetA and VNetC are peered?**

No. Transitive peering is not supported. You must peer VNetA and VNetC for this to take place.

#### **Are there any bandwidth limitations for peering connections?**

No. VNet peering, whether local or global, does not impose any bandwidth restrictions. Bandwidth is only limited by the VM or the compute resource.

#### **How can I troubleshoot VNet Peering issues?**

Here is a [troubleshooter guide](#) you can try.

## **Virtual network TAP**

#### **Which Azure regions are available for virtual network TAP?**

Virtual network TAP preview is available in all Azure regions. The monitored network interfaces, the virtual network TAP resource, and the collector or analytics solution must be deployed in the same region.

#### **Does Virtual Network TAP support any filtering capabilities on the mirrored packets?**

Filtering capabilities are not supported with the virtual network TAP preview. When a TAP configuration is added to a network interface a deep copy of all the ingress and egress traffic on the network interface is streamed to the TAP destination.

#### **Can multiple TAP configurations be added to a monitored network interface?**

A monitored network interface can have only one TAP configuration. Check with the individual [partner solution](#) for the capability to stream multiple copies of the TAP traffic to the analytics tools of your choice.

#### **Can the same virtual network TAP resource aggregate traffic from monitored network interfaces in more than one virtual network?**

Yes. The same virtual network TAP resource can be used to aggregate mirrored traffic from monitored network interfaces in peered virtual networks in the same subscription or a different subscription. The virtual network TAP resource and the destination load balancer or destination network interface must be in the same subscription. All subscriptions must be under the same Azure Active Directory tenant.

#### **Are there any performance considerations on production traffic if I enable a virtual network TAP configuration on a network interface?**

Virtual network TAP is in preview. During preview, there is no service level agreement. The capability should not be used for production workloads. When a virtual machine network interface is enabled with a TAP configuration, the same resources on the Azure host allocated to the virtual machine to send the production traffic is used to perform the mirroring function and send the mirrored packets. Select the correct [Linux](#) or [Windows](#) virtual machine size to ensure that sufficient resources are available for the virtual machine to send the production traffic and the mirrored traffic.

#### **Is accelerated networking for [Linux](#) or [Windows](#) supported with virtual network TAP?**

You will be able to add a TAP configuration on a network interface attached to a virtual machine that is enabled with accelerated networking. But the performance and latency on the virtual machine will be affected by adding TAP configuration since the offload for mirroring traffic is currently not supported by Azure accelerated networking.

# Virtual network service endpoints

## What is the right sequence of operations to set up service endpoints to an Azure service?

There are two steps to secure an Azure service resource through service endpoints:

1. Turn on service endpoints for the Azure service.
2. Set up VNet ACLs on the Azure service.

The first step is a network side operation and the second step is a service resource side operation. Both steps can be performed either by the same administrator or different administrators based on the Azure RBAC permissions granted to the administrator role. We recommend that you first turn on service endpoints for your virtual network prior to setting up VNet ACLs on Azure service side. Hence, the steps must be performed in the sequence listed above to set up VNet service endpoints.

### NOTE

Both the operations described above must be completed before you can limit the Azure service access to the allowed VNet and subnet. Only turning on service endpoints for the Azure service on the network side does not provide you the limited access. In addition, you must also set up VNet ACLs on the Azure service side.

Certain services (such as SQL and CosmosDB) allow exceptions to the above sequence through the `IgnoreMissingVnetServiceEndpoint` flag. Once the flag is set to `True`, VNet ACLs can be set on the Azure service side prior to setting up the service endpoints on the network side. Azure services provide this flag to help customers in cases where the specific IP firewalls are configured on Azure services and turning on the service endpoints on the network side can lead to a connectivity drop since the source IP changes from a public IPv4 address to a private address. Setting up VNet ACLs on the Azure service side before setting service endpoints on the network side can help avoid a connectivity drop.

## Do all Azure services reside in the Azure virtual network provided by the customer? How does VNet service endpoint work with Azure services?

No, not all Azure services reside in the customer's virtual network. The majority of Azure data services such as Azure Storage, Azure SQL, and Azure Cosmos DB, are multi-tenant services that can be accessed over public IP addresses. You can learn more about virtual network integration for Azure services [here](#).

When you use the VNet service endpoints feature (turning on VNet service endpoint on the network side and setting up appropriate VNet ACLs on the Azure service side), access to an Azure service is restricted from an allowed VNet and subnet.

## How does VNet service endpoint provide security?

The VNet service endpoint feature (turning on VNet service endpoint on the network side and setting up appropriate VNet ACLs on the Azure service side) limits the Azure service access to the allowed VNet and subnet, thus providing a network level security and isolation of the Azure service traffic. All traffic using VNet service endpoints flows over Microsoft backbone, thus providing another layer of isolation from the public internet. Moreover, customers can choose to fully remove public Internet access to the Azure service resources and allow traffic only from their virtual network through a combination of IP firewall and VNet ACLs, thus protecting the Azure service resources from unauthorized access.

## What does the VNet service endpoint protect - VNet resources or Azure service?

VNet service endpoints help protect Azure service resources. VNet resources are protected through Network Security Groups (NSGs).

## Is there any cost for using VNet service endpoints?

No, there is no additional cost for using VNet service endpoints.

## Can I turn on VNet service endpoints and set up VNet ACLs if the virtual network and the Azure service

## **resources belong to different subscriptions?**

Yes, it is possible. Virtual networks and Azure service resources can be either in the same or different subscriptions. The only requirement is that both the virtual network and Azure service resources must be under the same Active Directory (AD) tenant.

## **Can I turn on VNet service endpoints and set up VNet ACLs if the virtual network and the Azure service resources belong to different AD tenants?**

Yes, it is possible when using service endpoints for Azure Storage and Azure Key Vault. For rest of services, VNet service endpoints and VNet ACLs are not supported across AD tenants.

## **Can an on-premises device's IP address that is connected through Azure Virtual Network gateway (VPN) or ExpressRoute gateway access Azure PaaS Service over VNet service endpoints?**

By default, Azure service resources secured to virtual networks are not reachable from on-premises networks. If you want to allow traffic from on-premises, you must also allow public (typically, NAT) IP addresses from your on-premises or ExpressRoute. These IP addresses can be added through the IP firewall configuration for the Azure service resources.

## **Can I use VNet Service Endpoint feature to secure Azure service to multiple subnets within a virtual network or across multiple virtual networks?**

To secure Azure services to multiple subnets within a virtual network or across multiple virtual networks, enable service endpoints on the network side on each of the subnets independently and then secure Azure service resources to all of the subnets by setting up appropriate VNet ACLs on the Azure service side.

## **How can I filter outbound traffic from a virtual network to Azure services and still use service endpoints?**

If you want to inspect or filter the traffic destined to an Azure service from a virtual network, you can deploy a network virtual appliance within the virtual network. You can then apply service endpoints to the subnet where the network virtual appliance is deployed and secure Azure service resources only to this subnet through VNet ACLs. This scenario might also be helpful if you wish to restrict Azure service access from your virtual network only to specific Azure resources using network virtual appliance filtering. For more information, see [egress with network virtual appliances](#).

## **What happens when you access an Azure service account that has a virtual network access control list (ACL) enabled from outside the VNet?**

The HTTP 403 or HTTP 404 error is returned.

## **Are subnets of a virtual network created in different regions allowed to access an Azure service account in another region?**

Yes, for most of the Azure services, virtual networks created in different regions can access Azure services in another region through the VNet service endpoints. For example, if an Azure Cosmos DB account is in West US or East US and virtual networks are in multiple regions, the virtual network can access Azure Cosmos DB. Storage and SQL are exceptions and are regional in nature and both the virtual network and the Azure service need to be in the same region.

## **Can an Azure service have both a VNet ACL and an IP firewall?**

Yes, a VNet ACL and an IP firewall can co-exist. Both features complement each other to ensure isolation and security.

## **What happens if you delete a virtual network or subnet that has service endpoint turned on for Azure service?**

Deletion of VNets and subnets are independent operations and are supported even when service endpoints are turned on for Azure services. In cases where the Azure services have VNet ACLs set up, for those VNets and subnets, the VNet ACL information associated with that Azure service is disabled when a VNet or subnet that has VNet service endpoint turned on is deleted.

## **What happens if an Azure service account that has a VNet Service endpoint enabled is deleted?**

The deletion of an Azure service account is an independent operation and is supported even when the service

endpoint is enabled on the network side and VNet ACLs are set up on Azure service side.

### **What happens to the source IP address of a resource (like a VM in a subnet) that has VNet service endpoint enabled?**

When virtual network service endpoints are enabled, the source IP addresses of the resources in your virtual network's subnet switches from using public IPV4 addresses to the Azure virtual network's private IP addresses for traffic to Azure service. Note that this can cause specific IP firewalls that are set to public IPV4 address earlier on the Azure services to fail.

### **Does the service endpoint route always take precedence?**

Service endpoints add a system route which takes precedence over BGP routes and provides optimum routing for the service endpoint traffic. Service endpoints always take service traffic directly from your virtual network to the service on the Microsoft Azure backbone network. For more information about how Azure selects a route, see [Azure Virtual network traffic routing](#).

### **Do service endpoints work with ICMP?**

No, ICMP traffic that is sourced from a subnet with service endpoints enabled will not take the service tunnel path to the desired endpoint. Service endpoints will only handle TCP traffic. This means that if you want to test latency or connectivity to an endpoint via service endpoints, tools like ping and tracert will not show the true path that the resources within the subnet will take.

### **How does NSG on a subnet work with service endpoints?**

To reach the Azure service, NSGs need to allow outbound connectivity. If your NSGs are opened to all Internet outbound traffic, then the service endpoint traffic should work. You can also limit the outbound traffic to service IPs only using the Service tags.

### **What permissions do I need to set up service endpoints?**

Service endpoints can be configured on a virtual network independently by a user with write access to the virtual network. To secure Azure service resources to a VNet, the user must have permission

**Microsoft.Network/virtualNetworks/subnets/joinViaServiceEndpoint/action** for the subnets being added. This permission is included in the built-in service administrator role by default and can be modified by creating custom roles. Learn more about built-in roles and assigning specific permissions to [custom roles](#).

### **Can I filter virtual network traffic to Azure services, allowing only specific azure service resources, over VNet service endpoints?**

Virtual network (VNet) service endpoint policies allow you to filter virtual network traffic to Azure services, allowing only specific Azure service resources over the service endpoints. Endpoint policies provide granular access control from the virtual network traffic to the Azure services. You can learn more about the service endpoint policies [here](#).

### **Does Azure Active Directory (Azure AD) support VNet service endpoints?**

Azure Active Directory (Azure AD) doesn't support service endpoints natively. Complete list of Azure Services supporting VNet service endpoints can be seen [here](#). Note that the "Microsoft.AzureActiveDirectory" tag listed under services supporting service endpoints is used for supporting service endpoints to ADLS Gen 1. For ADLS Gen 1, virtual network integration for Azure Data Lake Storage Gen1 makes use of the virtual network service endpoint security between your virtual network and Azure Active Directory (Azure AD) to generate additional security claims in the access token. These claims are then used to authenticate your virtual network to your Data Lake Storage Gen1 account and allow access. Learn more about [Azure Data Lake Store Gen 1 VNet Integration](#)

### **Are there any limits on how many VNet service endpoints I can set up from my VNet?**

There is no limit on the total number of VNet service endpoints in a virtual network. For an Azure service resource (such as an Azure Storage account), services may enforce limits on the number of subnets used for securing the resource. The following table shows some example limits:

AZURE SERVICE	LIMITS ON VNET RULES
Azure Storage	100
Azure SQL	128
Azure Synapse Analytics	128
Azure KeyVault	127
Azure Cosmos DB	64
Azure Event Hub	128
Azure Service Bus	128
Azure Data Lake Store V1	100

**NOTE**

The limits are subjected to changes at the discretion of the Azure service. Refer to the respective service documentation for services details.