

Contents

VPN Gateway Documentation

Overview

About VPN Gateway

Tutorials

Create and manage a VPN gateway

Create and manage S2S VPN connections

Samples

Azure PowerShell

Concepts

Planning and design for VPN Gateway

About VPN Gateway settings

About VPN devices

About cryptographic requirements

About BGP and VPN Gateway

About highly available connections

About Point-to-Site connections

About Point-to-Site VPN routing

How-to guides

Configure Site-to-Site connections

Azure portal

Azure PowerShell

Azure CLI

Download VPN device configuration scripts

Configure Point-to-Site connections - native Azure certificate authentication

Configure a P2S VPN

Azure portal

Azure PowerShell

P2S certificates and clients

Generate self-signed certificates

- Azure PowerShell
- Makecert
- Linux
 - Install client certificates
 - Create and install VPN client configuration files
- Configure Point-to-Site connections - RADIUS authentication
 - Configure a P2S VPN
 - Azure PowerShell
 - Create and install VPN client configuration files
 - Integrate P2S VPN RADIUS authentication with NPS server
 - Configure VNet-to-VNet connections
 - Azure portal
 - Azure PowerShell
 - Azure CLI
 - Configure a VNet-to-VNet connection between deployment models
 - Azure portal
 - Azure PowerShell
 - Configure Site-to-Site and ExpressRoute coexisting connections
 - Azure PowerShell
 - Configure multiple Site-to-Site connections
 - Azure portal
 - Connect multiple policy-based VPN devices
 - Azure PowerShell
 - Configure IPsec/IKE policies on connections
 - Azure PowerShell
 - Configure highly available active-active connections
 - Azure PowerShell
 - Create a zone-redundant VNet gateway in Azure Availability Zones
 - Routing, BGP, and VNet Peering
 - Configure BGP for a VPN gateway
 - Azure PowerShell
 - Azure CLI

- [Configure forced tunneling](#)
 - [Azure PowerShell](#)
 - [Azure PowerShell \(classic\)](#)
- [Configure gateway transit for VNet peering](#)
- [Modify local network gateway settings](#)
 - [Azure portal](#)
 - [Azure PowerShell](#)
 - [Azure CLI](#)
- [Create a route-based VPN gateway](#)
 - [Azure portal](#)
 - [Azure PowerShell](#)
 - [Azure CLI](#)
- [Verify a VPN gateway connection](#)
- [Reset a VPN gateway](#)
- [Delete a VPN gateway](#)
 - [Azure portal](#)
 - [Azure PowerShell](#)
- [Gateway SKUs \(legacy\)](#)
- [Configure third-party VPN devices](#)
 - [Overview & Azure configuration](#)
 - [Sample: Cisco ASA device \(IKEv2/no BGP\)](#)
- [Troubleshoot](#)
 - [Community-suggested VPN or firewall device settings](#)
 - [Configure and validate VNet or VPN connections](#)
 - [Validate VPN throughput to a VNet](#)
 - [Point-to-Site connections](#)
 - [Point-to-Site connection problems](#)
 - [Point-to-Site connection problems - Mac OS X VPN client](#)
 - [Site-to-Site connection issues](#)
 - [Site-to-Site connections](#)
 - [Site-to-Site connection disconnects intermittently](#)
- [Classic deployment model articles](#)
- [Configure a Site-to-Site connection](#)

- - [Configure a Point-to-Site connection](#)
 - [Configure a VNet-to-VNet connection](#)
 - [Configure forced tunneling](#)
 - [Delete a VPN gateway](#)
 - [Configure multiple S2S connections](#)
 - [Configure a VPN gateway](#)
 - [Classic to Resource Manager migration](#)

Reference

- [Azure PowerShell](#)

- [Azure PowerShell \(classic\)](#)

- [REST](#)

- [REST \(classic\)](#)

- [Azure CLI](#)

Resources

- [VPN Gateway FAQ](#)

- [Azure Roadmap](#)

- [Blog](#)

- [Forum](#)

- [Subscription and service limits](#)

- [Pricing](#)

- [Pricing calculator](#)

- [SLA](#)

- [Videos](#)

What is VPN Gateway?

4/25/2018 • 11 minutes to read • [Edit Online](#)

A VPN gateway is a specific type of virtual network gateway that is used to send encrypted traffic between an Azure virtual network and an on-premises location over the public Internet. You can also use a VPN gateway to send encrypted traffic between Azure virtual networks over the Microsoft network. Each virtual network can have only one VPN gateway. However, you can create multiple connections to the same VPN gateway. When you create multiple connections to the same VPN gateway, all VPN tunnels share the available gateway bandwidth.

What is a virtual network gateway?

A virtual network gateway is composed of two or more virtual machines that are deployed to a specific subnet you create, which is called the *gateway subnet*. The VMs that are located in the gateway subnet are created when you create the virtual network gateway. Virtual network gateway VMs are configured to contain routing tables and gateway services specific to the gateway. You can't directly configure the VMs that are part of the virtual network gateway and you should never deploy additional resources to the gateway subnet.

Creating a virtual network gateway can take up to 45 minutes to complete. When you create a virtual network gateway, gateway VMs are deployed to the gateway subnet and configured with the settings that you specify. One of the settings you configure is the gateway type. The gateway type 'vpn' specifies that the type of virtual network gateway created is a VPN gateway. After you create a VPN gateway, you can create an IPsec/IKE VPN tunnel connection between that VPN gateway and another VPN gateway (VNet-to-VNet), or create a cross-premises IPsec/IKE VPN tunnel connection between the VPN gateway and an on-premises VPN device (Site-to-Site). You can also create a Point-to-Site VPN connection (VPN over IKEv2 or SSTP), which lets you connect to your virtual network from a remote location, such as from a conference or from home.

Configuring a VPN Gateway

A VPN gateway connection relies on multiple resources that are configured with specific settings. Most of the resources can be configured separately, although some resources must be configured in a certain order.

Settings

The settings that you chose for each resource are critical to creating a successful connection. For information about individual resources and settings for VPN Gateway, see [About VPN Gateway settings](#). The article contains information to help you understand gateway types, gateway SKUs, VPN types, connection types, gateway subnets, local network gateways, and various other resource settings that you may want to consider.

Deployment tools

You can start out creating and configuring resources using one configuration tool, such as the Azure portal. You can later decide to switch to another tool, such as PowerShell, to configure additional resources, or modify existing resources when applicable. Currently, you can't configure every resource and resource setting in the Azure portal. The instructions in the articles for each connection topology specify when a specific configuration tool is needed.

Deployment model

There are currently two deployment models for Azure. When you configure a VPN gateway, the steps you take depend on the deployment model that you used to create your virtual network. For example, if you created your VNet using the classic deployment model, you use the guidelines and instructions for the classic deployment model to create and configure your VPN gateway settings. For more information about deployment models, see [Understanding Resource Manager and classic deployment models](#).

Planning table

The following table can help you decide the best connectivity option for your solution.

	POINT-TO-SITE	SITE-TO-SITE	EXPRESSROUTE
Azure Supported Services	Cloud Services and Virtual Machines	Cloud Services and Virtual Machines	Services list
Typical Bandwidths	Based on the gateway SKU	Typically < 1 Gbps aggregate	50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps
Protocols Supported	Secure Sockets Tunneling Protocol (SSTP) and IPsec	IPsec	Direct connection over VLANs, NSP's VPN technologies (MPLS, VPLS,...)
Routing	RouteBased (dynamic)	We support PolicyBased (static routing) and RouteBased (dynamic routing VPN)	BGP
Connection resiliency	active-passive	active-passive or active-active	active-active
Typical use case	Prototyping, dev / test / lab scenarios for cloud services and virtual machines	Dev / test / lab scenarios and small scale production workloads for cloud services and virtual machines	Access to all Azure services (validated list), Enterprise-class and mission critical workloads, Backup, Big Data, Azure as a DR site
SLA	SLA	SLA	SLA
Pricing	Pricing	Pricing	Pricing
Technical Documentation	VPN Gateway Documentation	VPN Gateway Documentation	ExpressRoute Documentation
FAQ	VPN Gateway FAQ	VPN Gateway FAQ	ExpressRoute FAQ

Gateway SKUs

When you create a virtual network gateway, you specify the gateway SKU that you want to use. Select the SKU that satisfies your requirements based on the types of workloads, throughputs, features, and SLAs. For more information about gateway SKUs, including supported features, production and dev-test, and configuration steps, see [Gateway SKUs](#).

Gateway SKUs by tunnel, connection, and throughput

SKU	S2S/VNET-TO-VNET TUNNELS	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK
VpnGw1	Max. 30*	Max. 128**	650 Mbps
VpnGw2	Max. 30*	Max. 128**	1 Gbps
VpnGw3	Max. 30*	Max. 128**	1.25 Gbps

SKU	S2S/VNET-TO-VNET TUNNELS	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK
Basic	Max. 10	Max. 128	100 Mbps

- (*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.
- (**) Contact support if additional connections are needed. This applies to IKEv2 only, number of connections for SSTP cannot be increased.
- Aggregate Throughput Benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- VpnGw1, VpnGw2, and VpnGw3 are supported for VPN gateways using the Resource Manager deployment model only.

Connection topology diagrams

It's important to know that there are different configurations available for VPN gateway connections. You need to determine which configuration best fits your needs. In the sections below, you can view information and topology diagrams about the following VPN gateway connections: The following sections contain tables which list:

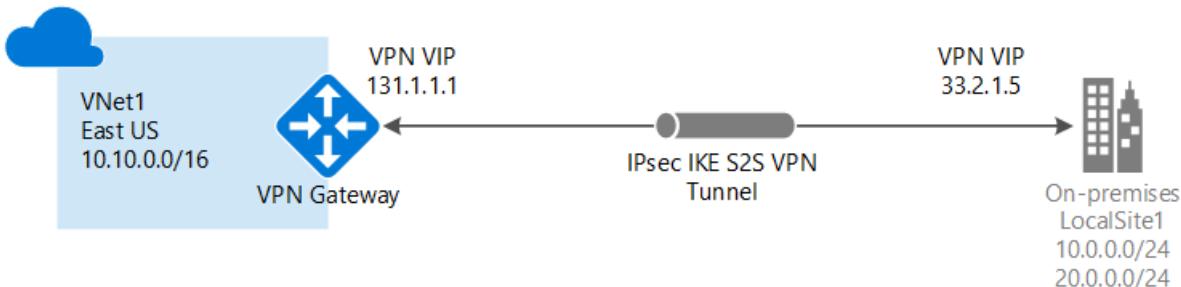
- Available deployment model
- Available configuration tools
- Links that take you directly to an article, if available

Use the diagrams and descriptions to help select the connection topology to match your requirements. The diagrams show the main baseline topologies, but it's possible to build more complex configurations using the diagrams as a guideline.

Site-to-Site and Multi-Site (IPsec/IKE VPN tunnel)

Site-to-Site

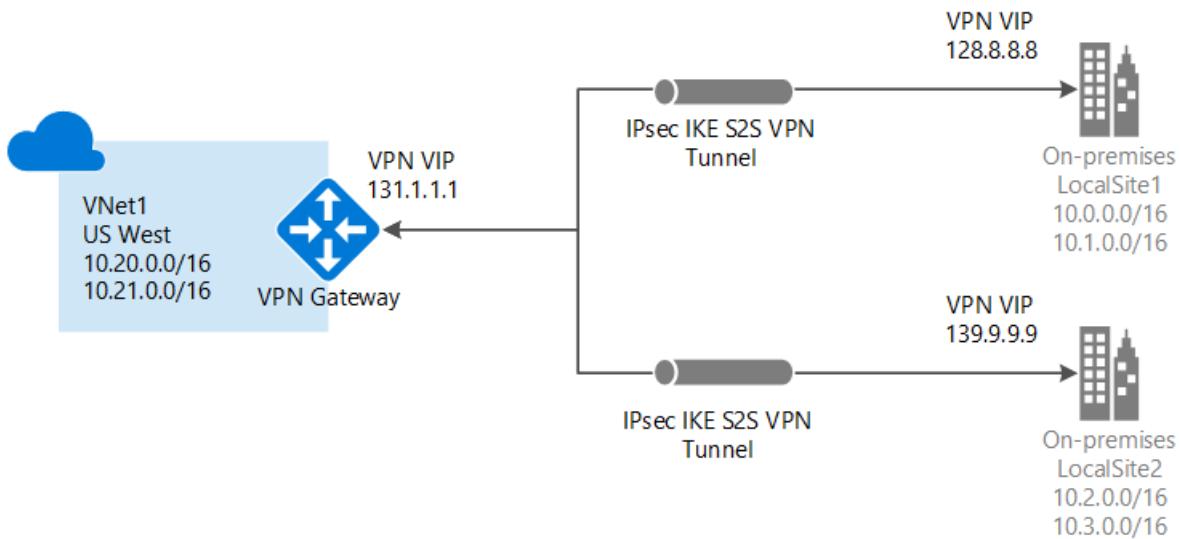
A Site-to-Site (S2S) VPN gateway connection is a connection over IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. S2S connections can be used for cross-premises and hybrid configurations. A S2S connection requires a VPN device located on-premises that has a public IP address assigned to it and is not located behind a NAT. For information about selecting a VPN device, see the [VPN Gateway FAQ - VPN devices](#).



Multi-Site

This type of connection is a variation of the Site-to-Site connection. You create more than one VPN connection from your virtual network gateway, typically connecting to multiple on-premises sites. When working with multiple connections, you must use a RouteBased VPN type (known as a dynamic gateway when working with

classic VNets). Because each virtual network can only have one VPN gateway, all connections through the gateway share the available bandwidth. This type of connection is often called a "multi-site" connection.



Deployment models and methods for Site-to-Site and Multi-Site

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL	AZURE CLI
Resource Manager	Article Article+	Article	Article
Classic	Article**	Article+	Not Supported

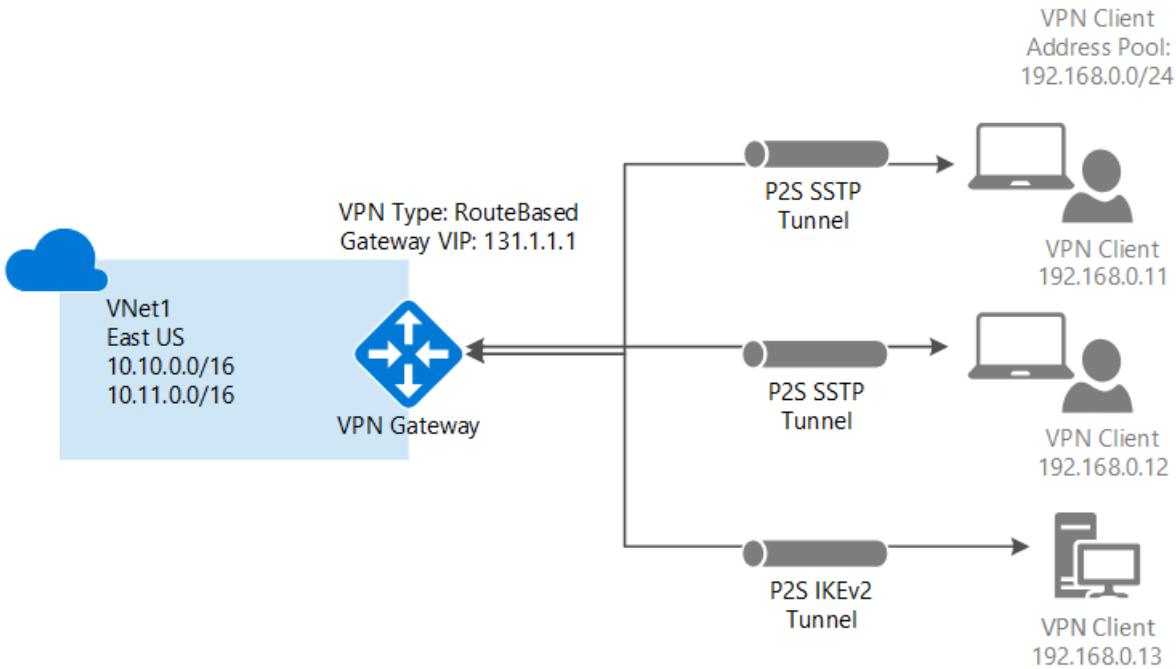
(**) denotes that this method contains steps that require PowerShell.

(+) denotes that this article is written for multi-site connections.

Point-to-Site (VPN over IKEv2 or SSTP)

A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer. A P2S connection is established by starting it from the client computer. This solution is useful for telecommuters who want to connect to Azure VNets from a remote location, such as from home or a conference. P2S VPN is also a useful solution to use instead of S2S VPN when you have only a few clients that need to connect to a VNet.

Unlike S2S connections, P2S connections do not require an on-premises public-facing IP address or a VPN device. P2S connections can be used with S2S connections through the same VPN gateway, as long as all the configuration requirements for both connections are compatible. For more information about Point-to-Site connections, see [About Point-to-Site VPN](#).



Deployment models and methods for P2S

Azure native certificate authentication

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Article	Article
Classic	Article	Supported

RADIUS authentication

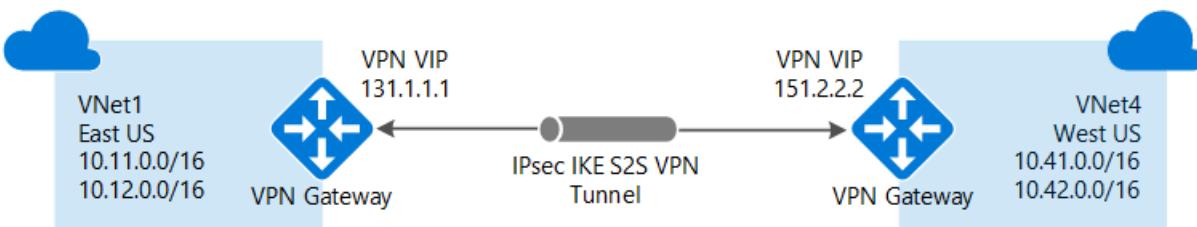
DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Supported	Article
Classic	Not Supported	Not Supported

VNet-to-VNet connections (IPsec/IKE VPN tunnel)

Connecting a virtual network to another virtual network (VNet-to-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 even combine VNet-to-VNet communication with multi-site connection configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

The VNets you connect can be:

- in the same or different regions
- in the same or different subscriptions
- in the same or different deployment models



Connections between deployment models

Azure currently has two deployment models: classic and Resource Manager. If you have been using Azure for some time, you probably have Azure VMs and instance roles running in a classic VNet. Your newer VMs and role instances may be running in a VNet created in Resource Manager. You can create a connection between the VNets to allow the resources in one VNet to communicate directly with resources in another.

VNet peering

You may be able to use VNet peering to create your connection, as long as your virtual network meets certain requirements. VNet peering does not use a virtual network gateway. For more information, see [VNet peering](#).

Deployment models and methods for VNet-to-VNet

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL	AZURE CLI
Classic	Article*	Supported	Not Supported
Resource Manager	Article+	Article	Article
Connections between different deployment models	Article*	Article	Not Supported

(+) denotes this deployment method is available only for VNets in the same subscription.

(*) denotes that this deployment method also requires PowerShell.

ExpressRoute (private connection)

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Office 365, and CRM Online. Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility.

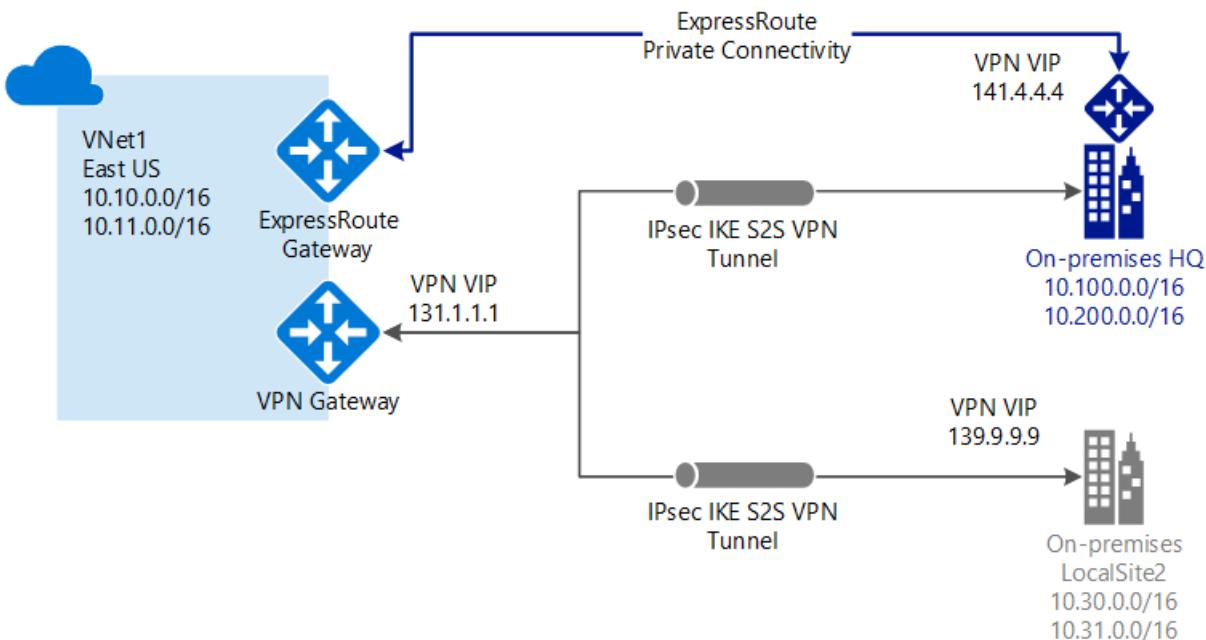
ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

An ExpressRoute connection uses a virtual network gateway as part of its required configuration. In an ExpressRoute connection, the virtual network gateway is configured with the gateway type 'ExpressRoute', rather than 'Vpn'. While traffic that travels over an ExpressRoute circuit is not encrypted by default, it is possible to create a solution that allows you to send encrypted traffic over an ExpressRoute circuit. For more information about ExpressRoute, see the [ExpressRoute technical overview](#).

Site-to-Site and ExpressRoute coexisting connections

ExpressRoute is a direct, private connection from your WAN (not over the public Internet) to Microsoft Services, including Azure. Site-to-Site VPN traffic travels encrypted over the public Internet. Being able to configure Site-to-Site VPN and ExpressRoute connections for the same virtual network has several advantages.

You can configure a Site-to-Site VPN as a secure failover path for ExpressRoute, or use Site-to-Site VPNs to connect to sites that are not part of your network, but that are connected through ExpressRoute. Notice that this configuration requires two virtual network gateways for the same virtual network, one using the gateway type 'Vpn', and the other using the gateway type 'ExpressRoute'.



Deployment models and methods for S2S and ExpressRoute coexist

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Not Supported	Article
Classic	Not Supported	Article

Pricing

You pay for two things: the hourly compute costs for the virtual network gateway, and the egress data transfer from the virtual network gateway. Pricing information can be found on the [Pricing](#) page.

Virtual network gateway compute costs

Each virtual network gateway has an hourly compute cost. The price is based on the gateway SKU that you specify when you create a virtual network gateway. The cost is for the gateway itself and is in addition to the data transfer that flows through the gateway.

Data transfer costs

Data transfer costs are calculated based on egress traffic from the source virtual network gateway.

- If you are sending traffic to your on-premises VPN device, it will be charged with the Internet egress data transfer rate.
- If you are sending traffic between virtual networks in different regions, the pricing is based on the region.
- If you are sending traffic only between virtual networks that are in the same region, there are no data costs. Traffic between VNets in the same region is free.

For more information about gateway SKUs for VPN Gateway, see [Gateway SKUs](#).

FAQ

For frequently asked questions about VPN gateway, see the [VPN Gateway FAQ](#).

Next steps

- Plan your VPN gateway configuration. See [VPN Gateway Planning and Design](#).
- View the [VPN Gateway FAQ](#) for additional information.
- View the [Subscription and service limits](#).
- Learn about some of the other key [networking capabilities](#) of Azure.

Create and Manage VPN gateway with the Azure PowerShell module

8/31/2018 • 5 minutes to read • [Edit Online](#)

Azure VPN gateways provide cross-premises connectivity between customer premises and Azure. This tutorial covers basic Azure VPN gateway deployment items such as creating and managing a VPN gateway. You learn how to:

- Create a VPN gateway
- Resize a VPN gateway
- Reset a VPN gateway

The following diagram shows the virtual network and the VPN gateway created as part of this tutorial.



Azure Cloud Shell and Azure PowerShell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

Click Try It in the upper right corner of a code block.	 A screenshot of a code block from a Microsoft documentation page. The 'Try It' button is highlighted with a red box.
Open Cloud Shell in your browser.	 A screenshot of the Azure portal's top navigation bar. The 'Launch Cloud Shell' button is highlighted with a red box.
Click the Cloud Shell button on the menu in the upper right of the Azure portal.	 A screenshot of the Azure portal's top navigation bar. The 'Cloud Shell' button is highlighted with a red box.

If you choose to install and use the PowerShell locally, this tutorial requires the Azure PowerShell module version 5.3 or later. Run `Get-Module -ListAvailable AzureRM` to find the version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Login-AzureRmAccount` to create a connection with Azure.

Common network parameter values

Change the values below based on your environment and network setup.

```

$RG1      = "TestRG1"
$VNet1    = "VNet1"
$Location1 = "East US"
$FESubnet1 = "FrontEnd"
$BESubnet1 = "Backend"
$GwSubnet1 = "GatewaySubnet"
$VNet1Prefix = "10.1.0.0/16"
$FEPrefix1  = "10.1.0.0/24"
$BEPrefix1  = "10.1.1.0/24"
$GwPrefix1  = "10.1.255.0/27"
$VNet1ASN   = 65010
$DNS1       = "8.8.8.8"
$Gw1        = "VNet1GW"
$GwIP1      = "VNet1GWIP"
$GwIPConf1  = "gwipconf1"

```

Create resource group

Create a resource group with the [New-AzureRmResourceGroup](#) command. An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created first. In the following example, a resource group named *TestRG1* is created in the *East US* region:

```
New-AzureRmResourceGroup -ResourceGroupName $RG1 -Location $Location1
```

Create a virtual network

Azure VPN gateway provides cross-premises connectivity and P2S VPN server functionality for your virtual network. Add the VPN gateway to an existing virtual network or create a new virtual network and the gateway. This example creates a new virtual network with three subnets: Frontend, Backend, and GatewaySubnet using [New-AzureRmVirtualNetworkSubnetConfig](#) and [New-AzureRmVirtualNetwork](#):

```

$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubnet1 -AddressPrefix $FEPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubnet1 -AddressPrefix $BEPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GwSubnet1 -AddressPrefix $GwPrefix1
$vnet   = New-AzureRmVirtualNetwork ` 
          -Name $VNet1 ` 
          -ResourceGroupName $RG1 ` 
          -Location $Location1 ` 
          -AddressPrefix $VNet1Prefix ` 
          -Subnet $fesub1,$besub1,$gwsb1

```

Request a public IP address for the VPN gateway

Azure VPN gateways communicate with your on-premises VPN devices over the Internet to performs IKE (Internet Key Exchange) negotiation and establish IPsec tunnels. Create and assign a public IP address to your VPN gateway as shown in the example below with [New-AzureRmPublicIpAddress](#) and [New-AzureRmVirtualNetworkGatewayIpConfig](#):

IMPORTANT

Currently, you can only use a Dynamic public IP address for the gateway. Static IP address is not supported on Azure VPN gateways.

```
$gwpip = New-AzureRmPublicIpAddress -Name $GwIP1 -ResourceGroupName $RG1 `  
    -Location $Location1 -AllocationMethod Dynamic  
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' `  
    -VirtualNetwork $vnet  
$gwipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GwIPConf1 `  
    -Subnet $subnet -PublicIpAddress $gwpip
```

Create VPN gateway

A VPN gateway can take 45 minutes or more to create. Once the gateway creation has completed, you can create a connection between your virtual network and another VNet. Or create a connection between your virtual network and an on-premises location. Create a VPN gateway using the [New-AzureRmVirtualNetworkGateway](#) cmdlet.

```
New-AzureRmVirtualNetworkGateway -Name $Gw1 -ResourceGroupName $RG1 `  
    -Location $Location1 -IpConfigurations $gwipconf -GatewayType Vpn `  
    -VpnType RouteBased -GatewaySku VpnGw1
```

Key parameter values:

- **GatewayType:** Use **Vpn** for site-to-site and VNet-to-VNet connections
- **VpnType:** Use **RouteBased** to interact with wider range of VPN devices and more routing features
- **GatewaySku:** **VpnGw1** is the default; change it to VpnGw2 or VpnGw3 if you need higher throughputs or more connections. For more information, see [Gateway SKUs](#).

Once the gateway creation has completed, you can create a connection between your virtual network and another VNet, or create a connection between your virtual network and an on-premises location. You can also configure a P2S connection to your VNet from a client computer.

Resize VPN gateway

You can change the VPN gateway SKU after the gateway is created. Different gateway SKUs support different specifications such as throughputs, number of connections, etc. The following example uses [Resize-AzureRmVirtualNetworkGateway](#) to resize your gateway from VpnGw1 to VpnGw2. For more information, see [Gateway SKUs](#).

```
$gw = Get-AzureRmVirtualNetworkGateway -Name $Gw1 -ResourceGroup $RG1  
Resize-AzureRmVirtualNetworkGateway -GatewaySku VpnGw2 -VirtualNetworkGateway $gateway
```

Resizing a VPN gateway also takes about 30 to 45 minutes, although this operation will **not** interrupt or remove existing connections and configurations.

Reset VPN gateway

As part of the troubleshooting steps, you can reset your Azure VPN gateway to force the VPN gateway to restart the IPsec/IKE tunnel configurations. Use [Reset-AzureRmVirtualNetworkGateway](#) to reset your gateway.

```
$gw = Get-AzureRmVirtualNetworkGateway -Name $Gw1 -ResourceGroup $RG1  
Reset-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gateway
```

For more information, see [Reset a VPN gateway](#).

Get the gateway public IP address

If you know the name of the public IP address, use [Get-AzureRmPublicIpAddress](#) to show the public IP address assigned to the gateway.

```
$myGwIp = Get-AzureRmPublicIpAddress -Name $GwIP1 -ResourceGroup $RG1  
$myGwIp.IpAddress
```

Delete VPN gateway

A complete configuration of cross-premises and VNet-to-VNet connectivity requires multipel resource types in addition to VPN gateway. Delete the connections associated with the VPN gateway before deleting the gateway itself. Once the gateway is deleted, you can then delete the public IP address(es) for the gateway. See [Delete a VPN gateway](#) for the detailed steps.

If the gateway is part of a prototype or proof-of-concept deployment, you can use [Remove-AzureRmResourceGroup](#) command to remove the resource group, the VPN gateway, and all related resources.

```
Remove-AzureRmResourceGroup -Name $RG1
```

Next steps

In this tutorial, you learned about basic VPN gateway creation and management such as how to:

- Create a VPN gateway
- Resize a VPN gateway
- Reset a VPN gateway

Advance to the following tutorials to learn about S2S, VNet-to-VNet, and P2S connections.

- [Create S2S connections](#)
- [Create VNet-to-VNet connections](#)
- [Create P2S connections](#)

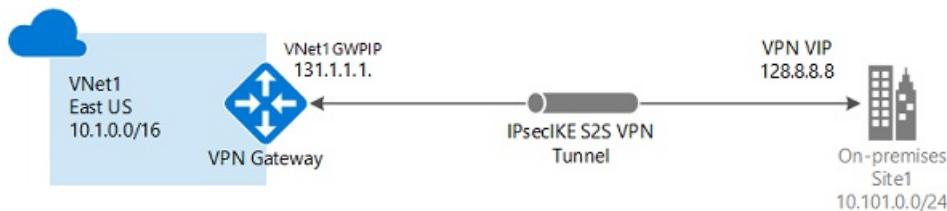
Create and Manage S2S VPN connections with the Azure PowerShell module

8/31/2018 • 5 minutes to read • [Edit Online](#)

Azure S2S VPN connections provide secure, cross-premises connectivity between customer premises and Azure. This tutorial walks through IPsec S2S VPN connection life cycles such as creating and managing a S2S VPN connection. You learn how to:

- Create an S2S VPN connection
- Update the connection property: pre-shared key, BGP, IPsec/IKE policy
- Add more VPN connections
- Delete a VPN connection

The following diagram shows the topology for this tutorial:



Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

Click Try It in the upper right corner of a code block.	
Open Cloud Shell in your browser.	
Click the Cloud Shell button on the menu in the upper right of the Azure portal.	

If you choose to install and use the PowerShell locally, this tutorial requires the Azure PowerShell module version 5.3 or later. Run `Get-Module -ListAvailable AzureRM` to find the version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Login-AzureRmAccount` to create a connection with Azure.

Requirements

Complete the first tutorial: "[Create VPN gateway with Azure PowerShell](#)" to create the following resources:

1. Resource group (TestRG1), virtual network (VNet1), and GatewaySubnet
2. VPN gateway (VNet1GW)

The virtual network parameter values are listed below. Note the additional values for the local network gateway to represent your on-premises network. Change the values based on your environment and network setup.

```
# Virtual network
$RG1      = "TestRG1"
$VNet1    = "VNet1"
$Location1 = "East US"
$VNet1Prefix = "10.1.0.0/16"
$VNet1ASN   = 65010
$Gw1       = "VNet1GW"

# On-premises network
$LNG1      = "VPNSite1"
$LNGprefix1 = "10.101.0.0/24"
$LNGprefix2 = "10.101.1.0/24"
$LNGIP1    = "YourDevicePublicIP"

# Optional - on-premises BGP properties
$LNGASN1   = 65011
$BGPPeerIP1 = "10.101.1.254"

# Connection
$Connection1 = "VNet1ToSite1"
```

The workflow to create an S2S VPN connection is straightforward:

1. Create a local network gateway to represent your on-premises network
2. Create a connection between your Azure VPN gateway and the local network gateway

Create a local network gateway

A local network gateway represents your on-premises network. You can specify the properties of your on-premises network in the local network gateway, including:

- Public IP address of your VPN device
- On-premises address space
- (Optional) BGP attributes (BGP peer IP address and AS number)

Create a local network gateway with the [New-AzureRmLocalNetworkGateway](#) command.

```
New-AzureRmLocalNetworkGateway -Name $LNG1 -ResourceGroupName $RG1 ` 
-Location 'East US' -GatewayIpAddress $LNGIP1 -AddressPrefix $LNGprefix1,$LNGprefix2
```

Create a S2S VPN connection

Next, create a Site-to-Site VPN connection between your virtual network gateway and your VPN device with the [New-AzureRmVirtualNetworkGatewayConnection](#). Notice that the '-ConnectionType' for Site-to-Site VPN is *IPsec*.

```
$vng1 = Get-AzureRmVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1
$lng1 = Get-AzureRmLocalNetworkGateway -Name $LNG1 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection1 -ResourceGroupName $RG1 ` 
-Location $Location1 -VirtualNetworkGateway1 $vng1 -LocalNetworkGateway2 $lng1 ` 
-ConnectionType IPsec -SharedKey "Azure@!b2C3"
```

Add the optional "**-EnableBGP \$True**" property to enable BGP for the connection if you are using BGP. It is disabled by default.

Update the VPN connection pre-shared key, BGP, and IPsec/IKE policy

View and update your pre-shared key

Azure S2S VPN connection uses a pre-shared key (secret) to authenticate between your on-premises VPN device and the Azure VPN gateway. You can view and update the pre-shared key for a connection with [Get-AzureRmVirtualNetworkGatewayConnectionSharedKey](#) and [Set-AzureRmVirtualNetworkGatewayConnectionSharedKey](#).

IMPORTANT

The pre-shared key is a string of **printable ASCII characters** no longer than 128 in length.

This command shows the pre-shared key for the connection:

```
Get-AzureRmVirtualNetworkGatewayConnectionSharedKey `  
-Name $Connection1 -ResourceGroupName $RG1
```

The output will be "**Azure@!b2C3**" following the example above. Use the command below to change the pre-shared key value to "**Azure@!_b2=C3**":

```
Set-AzureRmVirtualNetworkGatewayConnectionSharedKey `  
-Name $Connection1 -ResourceGroupName $RG1 `  
-Value "Azure@!_b2=C3"
```

Enable BGP on VPN connection

Azure VPN gateway supports BGP dynamic routing protocol. You can enable BGP on each individual connection, depending on whether you are using BGP in your on-premises networks and devices. Specify the following BGP properties before enabling BGP on the connection:

- Azure VPN ASN (Autonomous System Number)
- On-premises local network gateway ASN
- On-premises local network gateway BGP peer IP address

If you have not configured the BGP properties, use the following commands to add these properties to your VPN gateway and local network gateway: [Set-AzureRmVirtualNetworkGateway](#) and [Set-AzureRmLocalNetworkGateway](#).

```
$vng1 = Get-AzureRmVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1  
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $vng1 -Asn $VNet1ASN  
  
$lng1 = Get-AzureRmLocalNetworkGateway -Name $LNG1 -ResourceGroupName $RG1  
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $lng1 `  
-Asn $LNGASN1 -BgpPeeringAddress $BGPPeerIP1
```

Enable BGP with [Set-AzureRmVirtualNetworkGatewayConnection](#).

```
$connection = Get-AzureRmVirtualNetworkGatewayConnection `  
-Name $Connection1 -ResourceGroupName $RG1  
  
Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection `  
-EnableBGP $True
```

You can disable BGP by changing the "-EnableBGP" property value to **\$False**. Refer to [BGP on Azure VPN gateways](#) for more detailed explanations of BGP on Azure VPN gateways.

Apply a custom IPsec/IKE policy on the connection

You can apply an optional IPsec/IKE policy to specify the exact combination of IPsec/IKE cryptographic algorithms and key strengths on the connection, instead of using the [default proposals](#). The following sample script creates a different IPsec/IKE policy with the following algorithms and parameters:

- IKEv2: AES256, SHA256, DHGroup14
- IPsec: AES128, SHA1, PFS14, SA Lifetime 14,400 seconds & 102,400,000 KB

```
$connection = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection1 ` 
    -ResourceGroupName $RG1
$newpolicy = New-AzureRmIpsecPolicy ` 
    -IkeEncryption AES256 -IkeIntegrity SHA256 -DhGroup DHGroup14 ` 
    -IpsecEncryption AES128 -IpsecIntegrity SHA1 -PfsGroup PFS2048 ` 
    -SALifeTimeSeconds 14400 -SADataSizeKilobytes 102400000

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection ` 
    -IpsecPolicies $newpolicy
```

Refer to [IPsec/IKE policy for S2S or VNet-to-VNet connections](#) for a complete list of algorithms and instructions.

Add another S2S VPN connection

To add an additional S2S VPN connection to the same VPN gateway, create another local network gateway, and create a new connection between the new local network gateway and the VPN gateway. Following the example in this article.

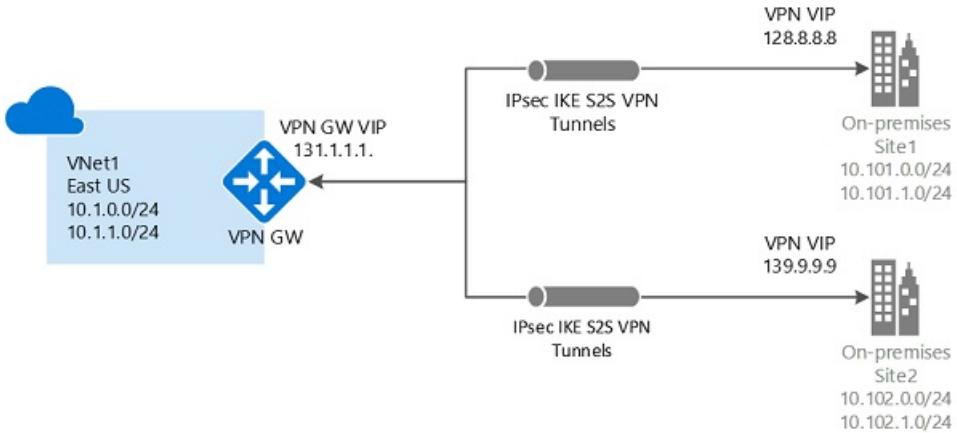
```
# On-premises network
$LNG2      = "VPNsite2"
$Location2  = "West US"
$LNGprefix21 = "10.102.0.0/24"
$LNGprefix22 = "10.102.1.0/24"
$LNGIP2     = "YourDevicePublicIP"
$Connection2 = "VNet1ToSite2"

New-AzureRmLocalNetworkGateway -Name $LNG2 -ResourceGroupName $RG1 ` 
    -Location $Location2 -GatewayIpAddress $LNGIP2 -AddressPrefix $LNGprefix21,$LNGprefix22

$vng1 = Get-AzureRmVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1
$lng2 = Get-AzureRmLocalNetworkGateway -Name $LNG2 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection2 -ResourceGroupName $RG1 ` 
    -Location $Location1 -VirtualNetworkGateway1 $vng1 -LocalNetworkGateway2 $lng2 ` 
    -ConnectionType IPsec -SharedKey "AzureA1%b2_C3+"
```

There are now two S2S VPN connections to your Azure VPN gateway.



Delete a S2S VPN connection

Delete a S2S VPN connection with [Remove-AzureRmVirtualNetworkGatewayConnection](#).

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name $Connection2 -ResourceGroupName $RG1
```

Delete the local network gateway if you no longer need it. You cannot delete a local network gateway if there are other connections associated with it.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name $LNG2 -ResourceGroupName $RG1
```

Next steps

In this tutorial, you learned about creating and managing S2S VPN connections such as how to:

- Create an S2S VPN connection
- Update the connection property: pre-shared key, BGP, IPsec/IKE policy
- Add more VPN connections
- Delete a VPN connection

Advance to the following tutorials to learn about S2S, VNet-to-VNet, and P2S connections.

- [Create VNet-to-VNet connections](#)
- [Create P2S connections](#)

Azure PowerShell samples for VPN Gateway

5/21/2018 • 2 minutes to read • [Edit Online](#)

The following table includes links to Azure Powershell scripts:

Create a VPN gateway	Creates a route-based VPN gateway.
Create a VPN gateway and P2S configuration - RADIUS	Creates a route-based VPN gateway and a P2S configuration that uses RADIUS username/password authentication.
Create a VPN gateway and P2S configuration - certificate authentication	Creates a route-based VPN gateway and a P2S configuration that uses native Azure certificate authentication.
Create a VPN gateway and Site-to-Site connection	Creates a route-based VPN gateway and a S2S connection.
Create vnet-to-vnet connections	Create vnet-to-vnet connections.
Download VPN device template	Download VPN device template.

Planning and design for VPN Gateway

8/15/2017 • 9 minutes to read • [Edit Online](#)

Planning and designing your cross-premises and VNet-to-VNet configurations can be either simple, or complicated, depending on your networking needs. This article walks you through basic planning and design considerations.

Planning

Cross-premises connectivity options

If you want to connect your on-premises sites securely to a virtual network, you have three different ways to do so: Site-to-Site, Point-to-Site, and ExpressRoute. Compare the different cross-premises connections that are available. The option you choose can depend on various considerations, such as:

- What kind of throughput does your solution require?
- Do you want to communicate over the public Internet via secure VPN, or over a private connection?
- Do you have a public IP address available to use?
- Are you planning to use a VPN device? If so, is it compatible?
- Are you connecting just a few computers, or do you want a persistent connection for your site?
- What type of VPN gateway is required for the solution you want to create?
- Which gateway SKU should you use?

Planning table

The following table can help you decide the best connectivity option for your solution.

	POINT-TO-SITE	SITE-TO-SITE	EXPRESSROUTE
Azure Supported Services	Cloud Services and Virtual Machines	Cloud Services and Virtual Machines	Services list
Typical Bandwidths	Based on the gateway SKU	Typically < 1 Gbps aggregate	50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps
Protocols Supported	Secure Sockets Tunneling Protocol (SSTP) and IPsec	IPsec	Direct connection over VLANs, NSP's VPN technologies (MPLS, VPLS,...)
Routing	RouteBased (dynamic)	We support PolicyBased (static routing) and RouteBased (dynamic routing VPN)	BGP
Connection resiliency	active-passive	active-passive or active-active	active-active
Typical use case	Prototyping, dev / test / lab scenarios for cloud services and virtual machines	Dev / test / lab scenarios and small scale production workloads for cloud services and virtual machines	Access to all Azure services (validated list), Enterprise-class and mission critical workloads, Backup, Big Data, Azure as a DR site

	POINT-TO-SITE	SITE-TO-SITE	EXPRESSROUTE
SLA	SLA	SLA	SLA
Pricing	Pricing	Pricing	Pricing
Technical Documentation	VPN Gateway Documentation	VPN Gateway Documentation	ExpressRoute Documentation
FAQ	VPN Gateway FAQ	VPN Gateway FAQ	ExpressRoute FAQ

Gateway SKUs

SKU	S2S/VNET-TO-VNET TUNNELS	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK
VpnGw1	Max. 30*	Max. 128**	650 Mbps
VpnGw2	Max. 30*	Max. 128**	1 Gbps
VpnGw3	Max. 30*	Max. 128**	1.25 Gbps
Basic	Max. 10	Max. 128	100 Mbps

- (*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.
- (**) Contact support if additional connections are needed. This applies to IKEv2 only, number of connections for SSTP cannot be increased.
- Aggregate Throughput Benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- VpnGw1, VpnGw2, and VpnGw3 are supported for VPN gateways using the Resource Manager deployment model only.

Workflow

The following list outlines the common workflow for cloud connectivity:

1. Design and plan your connectivity topology and list the address spaces for all networks you want to connect.
2. Create an Azure virtual network.
3. Create a VPN gateway for the virtual network.
4. Create and configure connections to on-premises networks or other virtual networks (as needed).
5. Create and configure a Point-to-Site connection for your Azure VPN gateway (as needed).

Design

Connection topologies

Start by looking at the diagrams in the [About VPN Gateway](#) article. The article contains basic diagrams, the deployment models for each topology, and the available deployment tools you can use to deploy your configuration.

Design basics

The following sections discuss the VPN gateway basics.

Networking services limits

Scroll through the tables to view [networking services limits](#). The limits listed may impact your design.

About subnets

When you are creating connections, you must consider your subnet ranges. You cannot have overlapping subnet address ranges. An overlapping subnet is when one virtual network or on-premises location contains the same address space that the other location contains. This means that you need your network engineers for your local on-premises networks to carve out a range for you to use for your Azure IP addressing space/subnets. You need address space that is not being used on the local on-premises network.

Avoiding overlapping subnets is also important when you are working with VNet-to-VNet connections. If your subnets overlap and an IP address exists in both the sending and destination VNets, VNet-to-VNet connections fail. Azure can't route the data to the other VNet because the destination address is part of the sending VNet.

VPN Gateways require a specific subnet called a gateway subnet. All gateway subnets must be named GatewaySubnet to work properly. Be sure not to name your gateway subnet a different name, and don't deploy VMs or anything else to the gateway subnet. See [Gateway Subnets](#).

About local network gateways

The local network gateway typically refers to your on-premises location. In the classic deployment model, the local network gateway is referred to as a Local Network Site. When you configure a local network gateway, you give it a name, specify the public IP address of the on-premises VPN device, and specify the address prefixes that are in the on-premises location. Azure looks at the destination address prefixes for network traffic, consults the configuration that you have specified for the local network gateway, and routes packets accordingly. You can modify the address prefixes as needed. For more information, see [Local network gateways](#).

About gateway types

Selecting the correct gateway type for your topology is critical. If you select the wrong type, your gateway won't work properly. The gateway type specifies how the gateway itself connects and is a required configuration setting for the Resource Manager deployment model.

The gateway types are:

- Vpn
- ExpressRoute

About connection types

Each configuration requires a specific connection type. The connection types are:

- IPsec
- Vnet2Vnet
- ExpressRoute
- VPNClient

About VPN types

Each configuration requires a specific VPN type. If you are combining two configurations, such as creating a Site-to-Site connection and a Point-to-Site connection to the same VNet, you must use a VPN type that satisfies both connection requirements.

- **PolicyBased:** PolicyBased VPNs were previously called static routing gateways in the classic deployment model. Policy-based VPNs encrypt and direct packets through IPsec tunnels based on the IPsec policies configured with the combinations of address prefixes between your on-premises network and the Azure VNet. The policy (or traffic selector) is usually defined as an access list in the VPN device configuration. The value for a PolicyBased VPN type is *PolicyBased*. When using a PolicyBased VPN, keep in mind the

following limitations:

- PolicyBased VPNs can **only** be used on the Basic gateway SKU. This VPN type is not compatible with other gateway SKUs.
- You can have only 1 tunnel when using a PolicyBased VPN.
- You can only use PolicyBased VPNs for S2S connections, and only for certain configurations. Most VPN Gateway configurations require a RouteBased VPN.
- **RouteBased:** RouteBased VPNs were previously called dynamic routing gateways in the classic deployment model. RouteBased VPNs use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels. The policy (or traffic selector) for RouteBased VPNs are configured as any-to-any (or wild cards). The value for a RouteBased VPN type is *RouteBased*.

The following tables show the VPN type as it maps to each connection configuration. Make sure the VPN type for your gateway matches the configuration that you want to create.

VPN type - Resource Manager deployment model

	ROUTEBASED	POLICYBASED
Site-to-Site	Supported	Supported
VNet-to-VNet	Supported	Not Supported
Multi-Site	Supported	Not Supported
S2S and ExpressRoute coexist	Supported	Not Supported
Point-to-Site	Supported	Not Supported
Classic to Resource Manager	Supported	Not Supported

VPN type - classic deployment model

	DYNAMIC	STATIC
Site-to-Site	Supported	Supported
VNet-to-VNet	Supported	Not Supported
Multi-Site	Supported	Not Supported
S2S and ExpressRoute coexist	Supported	Not Supported
Point-to-Site	Supported	Not Supported
Classic to Resource Manager	Supported	Not Supported

VPN devices for Site-to-Site connections

To configure a Site-to-Site connection, regardless of deployment model, you need the following items:

- A VPN device that is compatible with Azure VPN gateways
- A public-facing IPv4 IP address that is not behind a NAT

You need to have experience configuring your VPN device, or have someone that can configure the device for you.

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](#).

Consider forced tunnel routing

For most configurations, you can configure forced tunneling. Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies.

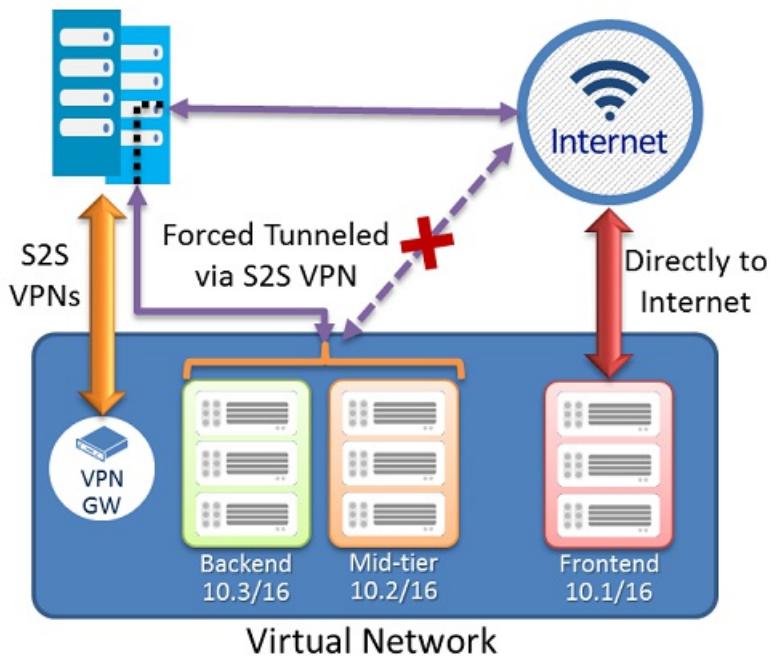
Without forced tunneling, Internet-bound traffic from your VMs in Azure will always traverse from Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic.

Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches.

A forced tunneling connection can be configured in both deployment models and by using different tools. For more information, see [Configure forced tunneling](#).

Forced tunneling diagram

On Premises



Next steps

See the [VPN Gateway FAQ](#) and [About VPN Gateway](#) articles for more information to help you with your design.

For more information about specific gateway settings, see [About VPN Gateway Settings](#).

About VPN Gateway configuration settings

6/6/2018 • 12 minutes to read • [Edit Online](#)

A VPN gateway is a type of virtual network gateway that sends encrypted traffic between your virtual network and your on-premises location across a public connection. You can also use a VPN gateway to send traffic between virtual networks across the Azure backbone.

A VPN gateway connection relies on the configuration of multiple resources, each of which contains configurable settings. The sections in this article discuss the resources and settings that relate to a VPN gateway for a virtual network created in Resource Manager deployment model. You can find descriptions and topology diagrams for each connection solution in the [About VPN Gateway](#) article.

NOTE

The values in this article apply to virtual network gateways that use the `-GatewayType 'Vpn'`. This is why these particular virtual network gateways are referred to as VPN gateways. The values for ExpressRoute gateways are not the same values that you use for VPN gateways.

For values that apply to `-GatewayType 'ExpressRoute'`, see [Virtual Network Gateways for ExpressRoute](#).

Gateway types

Each virtual network can only have one virtual network gateway of each type. When you are creating a virtual network gateway, you must make sure that the gateway type is correct for your configuration.

The available values for `-GatewayType` are:

- Vpn
- ExpressRoute

A VPN gateway requires the `-GatewayType Vpn`.

Example:

```
New-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg `  
-Location 'West US' -IpConfigurations $gwipconfig -GatewayType Vpn `  
-VpnType RouteBased
```

Gateway SKUs

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. Select the SKU that satisfies your requirements based on the types of workloads, throughputs, features, and SLAs.

Gateway SKUs by tunnel, connection, and throughput

SKU	S2S/VNET-TO-VNET TUNNELS	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK
VpnGw1	Max. 30*	Max. 128**	650 Mbps
VpnGw2	Max. 30*	Max. 128**	1 Gbps

SKU	S2S/VNET-TO-VNET TUNNELS	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK
VpnGw3	Max. 30*	Max. 128**	1.25 Gbps
Basic	Max. 10	Max. 128	100 Mbps

- (*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.
- (**) Contact support if additional connections are needed. This applies to IKEv2 only, number of connections for SSTP cannot be increased.
- Aggregate Throughput Benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- VpnGw1, VpnGw2, and VpnGw3 are supported for VPN gateways using the Resource Manager deployment model only.

NOTE

The new VPN gateway SKUs (VpnGw1, VpnGw2, and VpnGw3) are supported for the Resource Manager deployment model only. Classic virtual networks should continue to use the old (legacy) SKUs.

- For information about working with the legacy gateway SKUs (Basic, Standard, and HighPerformance), see [Working with VPN gateway SKUs \(legacy SKUs\)](#).
- For ExpressRoute gateway SKUs, see [Virtual Network Gateways for ExpressRoute](#).

Gateway SKUs by feature set

The new VPN gateway SKUs streamline the feature sets offered on the gateways:

SKU	FEATURES
Basic (**)	Route-based VPN: 10 tunnels with P2S; no RADIUS authentication for P2S; no IKEv2 for P2S Policy-based VPN: (IKEv1): 1 tunnel; no P2S
VpnGw1, VpnGw2, and VpnGw3	Route-based VPN: up to 30 tunnels (*), P2S, BGP, active-active, custom IPsec/IKE policy, ExpressRoute/VPN coexistence

(*) You can configure "PolicyBasedTrafficSelectors" to connect a route-based VPN gateway (VpnGw1, VpnGw2, VpnGw3) to multiple on-premises policy-based firewall devices. Refer to [Connect VPN gateways to multiple on-premises policy-based VPN devices using PowerShell](#) for details.

(**) The Basic SKU is considered a legacy SKU. The Basic SKU has certain feature limitations. You can't resize a gateway that uses a Basic SKU to one of the new gateway SKUs, you must instead change to a new SKU, which involves deleting and recreating your VPN gateway.

Gateway SKUs - Production vs. Dev-Test Workloads

Due to the differences in SLAs and feature sets, we recommend the following SKUs for production vs. dev-test:

WORKLOAD	SKUS
Production, critical workloads	VpnGw1, VpnGw2, VpnGw3
Dev-test or proof of concept	Basic (**)

(**) The Basic SKU is considered a legacy SKU and has feature limitations. Verify that the feature that you need is supported before you use the Basic SKU.

If you are using the old SKUs (legacy), the production SKU recommendations are Standard and HighPerformance. For information and instructions for old SKUs, see [Gateway SKUs \(legacy\)](#).

Configure a gateway SKU

Azure portal

If you use the Azure portal to create a Resource Manager virtual network gateway, you can select the gateway SKU by using the dropdown. The options you are presented with correspond to the Gateway type and VPN type that you select.

PowerShell

The following PowerShell example specifies the `-GatewaySku` as VpnGw1. When using PowerShell to create a gateway, you have to first create the IP configuration, then use a variable to refer to it. In this example, the configuration variable is \$gwipconfig.

```
New-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1 ` 
-Location 'US East' -IpConfigurations $gwipconfig -GatewaySku VpnGw1 ` 
-GatewayType Vpn -VpnType RouteBased
```

Azure CLI

```
az network vnet-gateway create --name VNet1GW --public-ip-address VNet1GWPIP --resource-group TestRG1 --vnet 
VNet1 --gateway-type Vpn --vpn-type RouteBased --sku VpnGw1 --no-wait
```

Resizing or changing a SKU

If you have a VPN gateway and you want to use a different gateway SKU, your options are to either resize your gateway SKU, or to change to another SKU. When you change to another gateway SKU, you delete the existing gateway entirely and build a new one. This could take up to 45 minutes to build. In comparison, when you resize a gateway SKU, you will have very little downtime because you do not have to delete and rebuild the gateway. If you have the option to resize your gateway SKU, rather than change it, you will want to do that. However, there are rules regarding resizing:

1. You can resize between VpnGw1, VpnGw2, and VpnGw3 SKUs.
2. When working with the old gateway SKUs, you can resize between Basic, Standard, and HighPerformance SKUs.
3. You **cannot** resize from Basic/Standard/HighPerformance SKUs to the new VpnGw1/VpnGw2/VpnGw3 SKUs. You must instead, [change](#) to the new SKUs.

To resize a gateway

For the current SKUs (VpnGw1, VpnGw2, and VPNGW3) you want to resize your gateway SKU to upgrade to a more powerful one, you can use the `Resize-AzureRmVirtualNetworkGateway` PowerShell cmdlet. You can also downgrade the gateway SKU size using this cmdlet. If you are using the Basic gateway SKU, [use these instructions instead](#) to resize your gateway.

The following PowerShell example shows a gateway SKU being resized to VpnGw2.

```
$gw = Get-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg  
Resize-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -GatewaySku VpnGw2
```

You can also resize a gateway in the Azure portal by going to the **Configuration** page for your virtual network gateway and selecting a different SKU from the dropdown.

To change from an old (legacy) SKU to a new SKU

If you are working with the Resource Manager deployment model, you can change to the new gateway SKUs. When you change from a legacy gateway SKU to a new SKU, you delete the existing VPN gateway and create a new VPN gateway.

Workflow:

1. Remove any connections to the virtual network gateway.
2. Delete the old VPN gateway.
3. Create the new VPN gateway.
4. Update your on-premises VPN devices with the new VPN gateway IP address (for Site-to-Site connections).
5. Update the gateway IP address value for any VNet-to-VNet local network gateways that will connect to this gateway.
6. Download new client VPN configuration packages for P2S clients connecting to the virtual network through this VPN gateway.
7. Recreate the connections to the virtual network gateway.

Considerations:

- To move to the new SKUs, your VPN gateway must be in the Resource Manager deployment model.
- If you have a classic VPN gateway, you must continue using the older legacy SKUs for that gateway, however, you can resize between the legacy SKUs. You cannot change to the new SKUs.
- You will have connectivity downtime when you change from a legacy SKU to a new SKU.
- When changing to a new gateway SKU, the public IP address for your VPN gateway will change. This happens even if you specify the same public IP address object that you used previously.

Connection types

In the Resource Manager deployment model, each configuration requires a specific virtual network gateway connection type. The available Resource Manager PowerShell values for `-ConnectionType` are:

- IPsec
- Vnet2Vnet
- ExpressRoute
- VPNCClient

In the following PowerShell example, we create a S2S connection that requires the connection type *IPsec*.

```
New-AzureRmVirtualNetworkGatewayConnection -Name localtovon -ResourceGroupName testrg `  
-Location 'West US' -VirtualNetworkGateway1 $gateway1 -LocalNetworkGateway2 $local `  
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

VPN types

When you create the virtual network gateway for a VPN gateway configuration, you must specify a VPN type. The VPN type that you choose depends on the connection topology that you want to create. For example, a P2S connection requires a RouteBased VPN type. A VPN type can also depend on the hardware that you are using.

S2S configurations require a VPN device. Some VPN devices only support a certain VPN type.

The VPN type you select must satisfy all the connection requirements for the solution you want to create. For example, if you want to create a S2S VPN gateway connection and a P2S VPN gateway connection for the same virtual network, you would use VPN type *RouteBased* because P2S requires a RouteBased VPN type. You would also need to verify that your VPN device supported a RouteBased VPN connection.

Once a virtual network gateway has been created, you can't change the VPN type. You have to delete the virtual network gateway and create a new one. There are two VPN types:

- **PolicyBased:** PolicyBased VPNs were previously called static routing gateways in the classic deployment model. Policy-based VPNs encrypt and direct packets through IPsec tunnels based on the IPsec policies configured with the combinations of address prefixes between your on-premises network and the Azure VNet. The policy (or traffic selector) is usually defined as an access list in the VPN device configuration. The value for a PolicyBased VPN type is *PolicyBased*. When using a PolicyBased VPN, keep in mind the following limitations:
 - PolicyBased VPNs can **only** be used on the Basic gateway SKU. This VPN type is not compatible with other gateway SKUs.
 - You can have only 1 tunnel when using a PolicyBased VPN.
 - You can only use PolicyBased VPNs for S2S connections, and only for certain configurations. Most VPN Gateway configurations require a RouteBased VPN.
- **RouteBased:** RouteBased VPNs were previously called dynamic routing gateways in the classic deployment model. RouteBased VPNs use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels. The policy (or traffic selector) for RouteBased VPNs are configured as any-to-any (or wild cards). The value for a RouteBased VPN type is *RouteBased*.

The following PowerShell example specifies the `-VpnType` as *RouteBased*. When you are creating a gateway, you must make sure that the `-VpnType` is correct for your configuration.

```
New-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg `  
-Location 'West US' -IpConfigurations $gwpipconfig `  
-GatewayType Vpn -VpnType RouteBased
```

Gateway requirements

The following table lists the requirements for PolicyBased and RouteBased VPN gateways. This table applies to both the Resource Manager and classic deployment models. For the classic model, PolicyBased VPN gateways are the same as Static gateways, and Route-based gateways are the same as Dynamic gateways.

	POLICYBASED BASIC VPN GATEWAY	ROUTEBASED BASIC VPN GATEWAY	ROUTEBASED STANDARD VPN GATEWAY	ROUTEBASED HIGH PERFORMANCE VPN GATEWAY
Site-to-Site connectivity (S2S)	PolicyBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration
Point-to-Site connectivity (P2S)	Not supported	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)
Authentication method	Pre-shared key	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity

	POLICYBASED BASIC VPN GATEWAY	ROUTEBASED BASIC VPN GATEWAY	ROUTEBASED STANDARD VPN GATEWAY	ROUTEBASED HIGH PERFORMANCE VPN GATEWAY
Maximum number of S2S connections	1	10	10	30
Maximum number of P2S connections	Not supported	128	128	128
Active routing support (BGP) (*)	Not supported	Not supported	Supported	Supported

(*) BGP is not supported for the classic deployment model.

Gateway subnet

Before you create a VPN gateway, you must create a gateway subnet. The gateway subnet contains the IP addresses that the virtual network gateway VMs and services use. When you create your virtual network gateway, gateway VMs are deployed to the gateway subnet and configured with the required VPN gateway settings. You must never deploy anything else (for example, additional VMs) to the gateway subnet. The gateway subnet must be named 'GatewaySubnet' to work properly. Naming the gateway subnet 'GatewaySubnet' lets Azure know that this is the subnet to deploy the virtual network gateway VMs and services to.

NOTE

Do not associate a route table that includes a route with a destination of 0.0.0.0/0 to the gateway subnet. Doing so prevents the gateway from functioning properly.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The IP addresses in the gateway subnet are allocated to the gateway VMs and gateway services. Some configurations require more IP addresses than others. Look at the instructions for the configuration that you want to create and verify that the gateway subnet you want to create meets those requirements. Additionally, you may want to make sure your gateway subnet contains enough IP addresses to accommodate possible future additional configurations. While you can create a gateway subnet as small as /29, we recommend that you create a gateway subnet of /28 or larger (/28, /27, /26 etc.). That way, if you add functionality in the future, you won't have to tear your gateway, then delete and recreate the gateway subnet to allow for more IP addresses.

The following Resource Manager PowerShell example shows a gateway subnet named GatewaySubnet. You can see the CIDR notation specifies a /27, which allows for enough IP addresses for most configurations that currently exist.

```
Add-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.0.3.0/27
```

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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

Local network gateways

When creating a VPN gateway configuration, the local network gateway often represents your on-premises

location. In the classic deployment model, the local network gateway was referred to as a Local Site.

You give the local network gateway a name, the public IP address of the on-premises VPN device, and specify the address prefixes that are located on the on-premises location. Azure looks at the destination address prefixes for network traffic, consults the configuration that you have specified for your local network gateway, and routes packets accordingly. You also specify local network gateways for VNet-to-VNet configurations that use a VPN gateway connection.

The following PowerShell example creates a new local network gateway:

```
New-AzureRmLocalNetworkGateway -Name LocalSite -ResourceGroupName testrg `  
-Location 'West US' -GatewayIpAddress '23.99.221.164' -AddressPrefix '10.5.51.0/24'
```

Sometimes you need to modify the local network gateway settings. For example, when you add or modify the address range, or if the IP address of the VPN device changes. See [Modify local network gateway settings using PowerShell](#).

REST APIs, PowerShell cmdlets, and CLI

For additional technical resources and specific syntax requirements when using REST APIs, PowerShell cmdlets, or Azure CLI for VPN Gateway configurations, see the following pages:

CLASSIC	RESOURCE MANAGER
PowerShell	PowerShell
REST API	REST API
Not supported	Azure CLI

Next steps

For more information about available connection configurations, see [About VPN Gateway](#).

About VPN devices and IPsec/IKE parameters for Site-to-Site VPN Gateway connections

8/27/2018 • 7 minutes to read • [Edit Online](#)

A VPN device is required to configure a Site-to-Site (S2S) cross-premises VPN connection using a VPN gateway. Site-to-Site connections can be used to create a hybrid solution, or whenever you want secure connections between your on-premises networks and your virtual networks. This article provides a list of validated VPN devices and a list of IPsec/IKE parameters for VPN gateways.

IMPORTANT

If you are experiencing connectivity issues between your on-premises VPN devices and VPN gateways, refer to [Known device compatibility issues](#).

Items to note when viewing the tables:

- There has been a terminology change for Azure VPN gateways. Only the names have changed. There is no functionality change.
 - Static Routing = PolicyBased
 - Dynamic Routing = RouteBased
- Specifications for HighPerformance VPN gateway and RouteBased VPN gateway are the same, unless otherwise noted. For example, the validated VPN devices that are compatible with RouteBased VPN gateways are also compatible with the HighPerformance VPN gateway.

Validated VPN devices and device configuration guides

NOTE

When configuring a Site-to-Site connection, a public-facing IPv4 IP address is required for your VPN device.

In partnership with device vendors, we have validated a set of standard VPN devices. All of the devices in the device families in the following list should work with VPN gateways. See [About VPN Gateway Settings](#) to understand the VPN type use (PolicyBased or RouteBased) for the VPN Gateway solution you want to configure.

To help configure your VPN device, refer to the links that correspond to appropriate device family. The links to configuration instructions are provided on a best-effort basis. For VPN device support, contact your device manufacturer.

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED CONFIGURATION INSTRUCTIONS	ROUTEBASED CONFIGURATION INSTRUCTIONS
A10 Networks, Inc.	Thunder CFW	ACOS 4.1.1	Not compatible	Configuration guide
Allied Telesis	AR Series VPN Routers	2.9.2	Coming soon	Not compatible
Barracuda Networks, Inc.	Barracuda NextGen Firewall F-series	PolicyBased: 5.4.3 RouteBased: 6.2.0	Configuration guide	Configuration guide

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED CONFIGURATION INSTRUCTIONS	ROUTEBASED CONFIGURATION INSTRUCTIONS
Barracuda Networks, Inc.	Barracuda NextGen Firewall X-series	Barracuda Firewall 6.5	Configuration guide	Not compatible
Brocade	Vyatta 5400 vRouter	Virtual Router 6.6R3 GA	Configuration guide	Not compatible
Check Point	Security Gateway	R77.30	Configuration guide	Configuration guide
Cisco	ASA	8.3 8.4+ (IKEv2*)	Configuration samples	Configuration guide*
Cisco	ASR	PolicyBased: IOS 15.1 RouteBased: IOS 15.2	Configuration samples	Configuration samples
Cisco	ISR	PolicyBased: IOS 15.0 RouteBased*: IOS 15.1	Configuration samples	Configuration samples**
Cisco	Meraki	N/A	Not compatible	Not compatible
Citrix	NetScaler MPX, SDX, VPX	10.1 and above	Configuration guide	Not compatible
F5	BIG-IP series	12.0	Configuration guide	Configuration guide
Fortinet	FortiGate	FortiOS 5.6		Configuration guide
Internet Initiative Japan (IIJ)	SEIL Series	SEIL/X 4.60 SEIL/B1 4.60 SEIL/x86 3.20	Configuration guide	Not compatible
Juniper	SRX	PolicyBased: JunOS 10.2 Routebased: JunOS 11.4	Configuration samples	Configuration samples
Juniper	J-Series	PolicyBased: JunOS 10.4r9 RouteBased: JunOS 11.4	Configuration samples	Configuration samples
Juniper	ISG	ScreenOS 6.3	Configuration samples	Configuration samples
Juniper	SSG	ScreenOS 6.2	Configuration samples	Configuration samples
Microsoft	Routing and Remote Access Service	Windows Server 2012	Not compatible	Configuration samples
Open Systems AG	Mission Control Security Gateway	N/A	Configuration guide	Not compatible

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED CONFIGURATION INSTRUCTIONS	ROUTEBASED CONFIGURATION INSTRUCTIONS
Palo Alto Networks	All devices running PAN-OS	PAN-OS PolicyBased: 6.1.5 or later RouteBased: 7.1.4	Configuration guide	Configuration guide
ShareTech	Next Generation UTM (NU series)	9.0.1.3	Not compatible	Configuration guide
SonicWall	TZ Series, NSA Series SuperMassive Series E-Class NSA Series	SonicOS 5.8.x SonicOS 5.9.x SonicOS 6.x	Not compatible	Configuration guide
Sophos	XG Next Gen Firewall	XG v17		Configuration guide
Ubiquiti	EdgeRouter	EdgeOS v1.10		BGP over IKEv2/IPsec VTI over IKEv2/IPsec
WatchGuard	All	Fireware XTM PolicyBased: v11.11.x RouteBased: v11.12.x	Configuration guide	Configuration guide

NOTE

(*) Cisco ASA versions 8.4+ add IKEv2 support, can connect to Azure VPN gateway using custom IPsec/IKE policy with "UsePolicyBasedTrafficSelectors" option. Refer to this [how-to article](#).

(**) ISR 7200 Series routers only support PolicyBased VPNs.

Download VPN device configuration scripts from Azure

For certain devices, you can download configuration scripts directly from Azure. For more information and download instructions, see [Download VPN device configuration scripts](#).

Devices with available configuration scripts

VENDOR	DEVICE FAMILY	FIRMWARE VERSION
Cisco	ISR	IOS 15.1 (Preview)
Cisco	ASA	ASA (*) RouteBased (IKEv2- No BGP) for ASA below 9.8
Cisco	ASA	ASA RouteBased (IKEv2 - No BGP) for ASA 9.8+
Juniper	SRX_GA	12.x
Juniper	SSG_GA	ScreenOS 6.2.x
Juniper	JSeries_GA	JunOS 12.x

VENDOR	DEVICE FAMILY	FIRMWARE VERSION
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased VTI
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased BGP

NOTE

(*) Required: NarrowAzureTrafficSelectors and CustomAzurePolicies (IKE/IPsec)

Non-validated VPN devices

If you don't see your device listed in the Validated VPN devices table, your device still may work with a Site-to-Site connection. Contact your device manufacturer for additional support and configuration instructions.

Editing device configuration samples

After you download the provided VPN device configuration sample, you'll need to replace some of the values to reflect the settings for your environment.

To edit a sample:

1. Open the sample using Notepad.
2. Search and replace all <text> strings with the values that pertain to your environment. Be sure to include < and >. When a name is specified, the name you select should be unique. If a command does not work, consult your device manufacturer documentation.

SAMPLE TEXT	CHANGE TO
<RP_OnPremisesNetwork>	Your chosen name for this object. Example: myOnPremisesNetwork
<RP_AzureNetwork>	Your chosen name for this object. Example: myAzureNetwork
<RP_AccessList>	Your chosen name for this object. Example: myAzureAccessList
<RP_IPSecTransformSet>	Your chosen name for this object. Example: myIPSecTransformSet
<RP_IPSecCryptoMap>	Your chosen name for this object. Example: myIPSecCryptoMap
<SP_AzureNetworkIpRange>	Specify range. Example: 192.168.0.0
<SP_AzureNetworkSubnetMask>	Specify subnet mask. Example: 255.255.0.0
<SP_OnPremisesNetworkIpRange>	Specify on-premises range. Example: 10.2.1.0
<SP_OnPremisesNetworkSubnetMask>	Specify on-premises subnet mask. Example: 255.255.255.0
<SP_AzureGatewayIpAddress>	This information specific to your virtual network and is located in the Management Portal as Gateway IP address .

SAMPLE TEXT	CHANGE TO
<SP_PresharedKey>	This information is specific to your virtual network and is located in the Management Portal as Manage Key.

IPsec/IKE parameters

IMPORTANT

1. The tables below contain the combinations of algorithms and parameters Azure VPN gateways use in default configuration. For route-based VPN gateways created using the Azure Resource Management deployment model, you can specify a custom policy on each individual connection. Please refer to [Configure IPsec/IKE policy](#) for detailed instructions.
2. In addition, you must clamp TCP **MSS** at **1350**. Or if your VPN devices do not support MSS clamping, you can alternatively set the **MTU** on the tunnel interface to **1400** bytes instead.

In the following tables:

- SA = Security Association
- IKE Phase 1 is also called "Main Mode"
- IKE Phase 2 is also called "Quick Mode"

IKE Phase 1 (Main Mode) parameters

PROPERTY	POLICYBASED	ROUTEBASED
IKE Version	IKEv1	IKEv2
Diffie-Hellman Group	Group 2 (1024 bit)	Group 2 (1024 bit)
Authentication Method	Pre-Shared Key	Pre-Shared Key
Encryption & Hashing Algorithms	1. AES256, SHA256 2. AES256, SHA1 3. AES128, SHA1 4. 3DES, SHA1	1. AES256, SHA1 2. AES256, SHA256 3. AES128, SHA1 4. AES128, SHA256 5. 3DES, SHA1 6. 3DES, SHA256
SA Lifetime	28,800 seconds	28,800 seconds

IKE Phase 2 (Quick Mode) parameters

PROPERTY	POLICYBASED	ROUTEBASED
IKE Version	IKEv1	IKEv2
Encryption & Hashing Algorithms	1. AES256, SHA256 2. AES256, SHA1 3. AES128, SHA1 4. 3DES, SHA1	RouteBased QM SA Offers
SA Lifetime (Time)	3,600 seconds	27,000 seconds

PROPERTY	POLICYBASED	ROUTEBASED
SA Lifetime (Bytes)	102,400,000 KB	-
Perfect Forward Secrecy (PFS)	No	RouteBased QM SA Offers
Dead Peer Detection (DPD)	Not supported	Supported

RouteBased VPN IPsec Security Association (IKE Quick Mode SA) Offers

The following table lists IPsec SA (IKE Quick Mode) Offers. Offers are listed the order of preference that the offer is presented or accepted.

Azure Gateway as initiator

-	ENCRYPTION	AUTHENTICATION	PFS GROUP
1	GCM AES256	GCM (AES256)	None
2	AES256	SHA1	None
3	3DES	SHA1	None
4	AES256	SHA256	None
5	AES128	SHA1	None
6	3DES	SHA256	None

Azure Gateway as responder

-	ENCRYPTION	AUTHENTICATION	PFS GROUP
1	GCM AES256	GCM (AES256)	None
2	AES256	SHA1	None
3	3DES	SHA1	None
4	AES256	SHA256	None
5	AES128	SHA1	None
6	3DES	SHA256	None
7	DES	SHA1	None
8	AES256	SHA1	1
9	AES256	SHA1	2
10	AES256	SHA1	14
11	AES128	SHA1	1

-	ENCRYPTION	AUTHENTICATION	PFS GROUP
12	AES128	SHA1	2
13	AES128	SHA1	14
14	3DES	SHA1	1
15	3DES	SHA1	2
16	3DES	SHA256	2
17	AES256	SHA256	1
18	AES256	SHA256	2
19	AES256	SHA256	14
20	AES256	SHA1	24
21	AES256	SHA256	24
22	AES128	SHA256	None
23	AES128	SHA256	1
24	AES128	SHA256	2
25	AES128	SHA256	14
26	3DES	SHA1	14

- You can specify IPsec ESP NULL encryption with RouteBased and HighPerformance VPN gateways. Null based encryption does not provide protection to data in transit, and should only be used when maximum throughput and minimum latency is required. Clients may choose to use this in VNet-to-VNet communication scenarios, or when encryption is being applied elsewhere in the solution.
- For cross-premises connectivity through the Internet, use the default Azure VPN gateway settings with encryption and hashing algorithms listed in the tables above to ensure security of your critical communication.

Known device compatibility issues

IMPORTANT

These are the known compatibility issues between third-party VPN devices and Azure VPN gateways. The Azure team is actively working with the vendors to address the issues listed here. Once the issues are resolved, this page will be updated with the most up-to-date information. Please check back periodically.

Feb. 16, 2017

Palo Alto Networks devices with version prior to 7.1.4 for Azure route-based VPN: If you are using VPN devices from Palo Alto Networks with PAN-OS version prior to 7.1.4 and are experiencing connectivity issues to Azure route-based VPN gateways, perform the following steps:

1. Check the firmware version of your Palo Alto Networks device. If your PAN-OS version is older than 7.1.4, upgrade to 7.1.4.
2. On the Palo Alto Networks device, change the Phase 2 SA (or Quick Mode SA) lifetime to 28,800 seconds (8 hours) when connecting to the Azure VPN gateway.
3. If you are still experiencing connectivity issues, open a support request from the Azure portal.

About cryptographic requirements and Azure VPN gateways

8/31/2017 • 6 minutes to read • [Edit Online](#)

This article discusses how you can configure Azure VPN gateways to satisfy your cryptographic requirements for both cross-premises S2S VPN tunnels and VNet-to-VNet connections within Azure.

About IPsec and IKE policy parameters for Azure VPN gateways

IPsec and IKE protocol standard supports a wide range of cryptographic algorithms in various combinations. If customers do not request a specific combination of cryptographic algorithms and parameters, Azure VPN gateways use a set of default proposals. The default policy sets were chosen to maximize interoperability with a wide range of third-party VPN devices in default configurations. As a result, the policies and the number of proposals cannot cover all possible combinations of available cryptographic algorithms and key strengths.

The default policy set for Azure VPN gateway is listed in the document: [About VPN devices and IPsec/IKE parameters for Site-to-Site VPN Gateway connections](#).

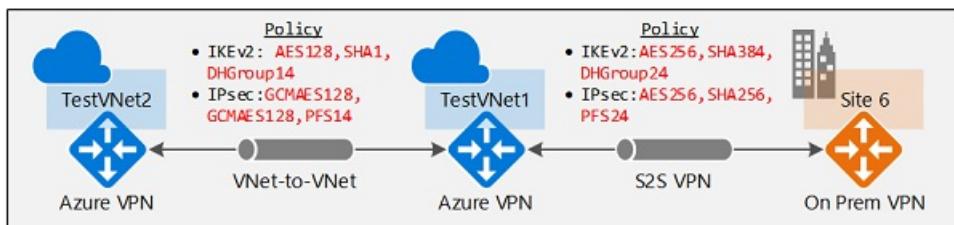
Cryptographic requirements

For communications that require specific cryptographic algorithms or parameters, typically due to compliance or security requirements, customers can now configure their Azure VPN gateways to use a custom IPsec/IKE policy with specific cryptographic algorithms and key strengths, rather than the Azure default policy sets.

For example, the IKEv2 main mode policies for Azure VPN gateways utilize only Diffie-Hellman Group 2 (1024 bits), whereas customers may need to specify stronger groups to be used in IKE, such as Group 14 (2048-bit), Group 24 (2048-bit MODP Group), or ECP (elliptic curve groups) 256 or 384 bit (Group 19 and Group 20, respectively). Similar requirements apply to IPsec quick mode policies as well.

Custom IPsec/IKE policy with Azure VPN gateways

Azure VPN gateways now support per-connection, custom IPsec/IKE policy. For a Site-to-Site or VNet-to-VNet connection, you can choose a specific combination of cryptographic algorithms for IPsec and IKE with the desired key strength, as shown in the following example:



You can create an IPsec/IKE policy and apply to a new or existing connection.

Workflow

1. Create the virtual networks, VPN gateways, or local network gateways for your connectivity topology as described in other how-to documents
2. Create an IPsec/IKE policy
3. You can apply the policy when you create a S2S or VNet-to-VNet connection
4. If the connection is already created, you can apply or update the policy to an existing connection

IPsec/IKE policy FAQ

Is Custom IPsec/IKE policy supported on all Azure VPN Gateway SKUs?

Custom IPsec/IKE policy is supported on Azure **VpnGw1**, **VpnGw2**, **VpnGw3**, **Standard**, and **HighPerformance** VPN gateways. The **Basic** SKU is **not** supported.

How many policies can I specify on a connection?

You can only specify **one** policy combination for a given connection.

Can I specify a partial policy on a connection? (for example, only IKE algorithms, but not IPsec)

No, you must specify all algorithms and parameters for both IKE (Main Mode) and IPsec (Quick Mode). Partial policy specification is not allowed.

What are the algorithms and key strengths supported in the custom policy?

The following table lists the supported cryptographic algorithms and key strengths configurable by the customers. You must select one option for every field.

IPSEC/IKEV2	OPTIONS
IKEv2 Encryption	AES256, AES192, AES128, DES3, DES
IKEv2 Integrity	SHA384, SHA256, SHA1, MD5
DH Group	DHGroup24, ECP384, ECP256, DHGroup14 (DHGroup2048), DHGroup2, DHGroup1, None
IPsec Encryption	GCMAES256, GCMAES192, GCMAES128, AES256, AES192, AES128, DES3, DES, None
IPsec Integrity	GCMAES256, GCMAES192, GCMAES128, SHA256, SHA1, MD5
PFS Group	PFS24, ECP384, ECP256, PFS2048, PFS2, PFS1, None
QM SA Lifetime	Seconds (integer; min. 300 /default 27000 seconds) KBytes (integer; min. 1024 /default 102400000 KBytes)
Traffic Selector	UsePolicyBasedTrafficSelectors (\$True/\$False; default \$False)

IMPORTANT

1. DHGroup2048 & PFS2048 are the same as Diffie-Hellman Group **14** in IKE and IPsec PFS. See [Diffie-Hellman Groups](#) for the complete mappings.
2. For GCMAES algorithms, you must specify the same GCMAES algorithm and key length for both IPsec Encryption and Integrity.
3. IKEv2 Main Mode SA lifetime is fixed at 28,800 seconds on the Azure VPN gateways
4. QM SA Lifetimes are optional parameters. If none was specified, default values of 27,000 seconds (7.5 hrs) and 102400000 KBytes (102GB) are used.
5. UsePolicyBasedTrafficSelector is an option parameter on the connection. See the next FAQ item for "UsePolicyBasedTrafficSelectors"

Does everything need to match between the Azure VPN gateway policy and my on-premises VPN device configurations?

Your on-premises VPN device configuration must match or contain the following algorithms and parameters that you specify on the Azure IPsec/IKE policy:

- IKE encryption algorithm
- IKE integrity algorithm
- DH Group
- IPsec encryption algorithm
- IPsec integrity algorithm
- PFS Group
- Traffic Selector (*)

The SA lifetimes are local specifications only, do not need to match.

If you enable **UsePolicyBasedTrafficSelectors**, you need to ensure your VPN device has the matching traffic selectors defined with all combinations of your on-premises network (local network gateway) prefixes to/from the Azure virtual network prefixes, instead of any-to-any. For example, if your on-premises network prefixes are 10.1.0.0/16 and 10.2.0.0/16, and your virtual network prefixes are 192.168.0.0/16 and 172.16.0.0/16, you need to specify the following traffic selectors:

- 10.1.0.0/16 <=====> 192.168.0.0/16
- 10.1.0.0/16 <=====> 172.16.0.0/16
- 10.2.0.0/16 <=====> 192.168.0.0/16
- 10.2.0.0/16 <=====> 172.16.0.0/16

For more information, see [Connect multiple on-premises policy-based VPN devices](#).

Which Diffie-Hellman Groups are supported?

The table below lists the supported Diffie-Hellman Groups for IKE (DHGroup) and IPsec (PFSGroup):

DIFFIE-HELLMAN GROUP	DHGROUP	PFSGROUP	KEY LENGTH
1	DHGroup1	PFS1	768-bit MODP
2	DHGroup2	PFS2	1024-bit MODP
14	DHGroup14 DHGroup2048	PFS2048	2048-bit MODP
19	ECP256	ECP256	256-bit ECP
20	ECP384	ECP284	384-bit ECP
24	DHGroup24	PFS24	2048-bit MODP

For more information, see [RFC3526](#) and [RFC5114](#).

Does the custom policy replace the default IPsec/IKE policy sets for Azure VPN gateways?

Yes, once a custom policy is specified on a connection, Azure VPN gateway will only use the policy on the connection, both as IKE initiator and IKE responder.

If I remove a custom IPsec/IKE policy, does the connection become unprotected?

No, the connection will still be protected by IPsec/IKE. Once you remove the custom policy from a connection, the Azure VPN gateway reverts back to the [default list of IPsec/IKE proposals](#) and restart the IKE handshake again.

with your on-premises VPN device.

Would adding or updating an IPsec/IKE policy disrupt my VPN connection?

Yes, it could cause a small disruption (a few seconds) as the Azure VPN gateway tears down the existing connection and restarts the IKE handshake to re-establish the IPsec tunnel with the new cryptographic algorithms and parameters. Ensure your on-premises VPN device is also configured with the matching algorithms and key strengths to minimize the disruption.

Can I use different policies on different connections?

Yes. Custom policy is applied on a per-connection basis. You can create and apply different IPsec/IKE policies on different connections. You can also choose to apply custom policies on a subset of connections. The remaining ones use the Azure default IPsec/IKE policy sets.

Can I use the custom policy on VNet-to-VNet connection as well?

Yes, you can apply custom policy on both IPsec cross-premises connections or VNet-to-VNet connections.

Do I need to specify the same policy on both VNet-to-VNet connection resources?

Yes. A VNet-to-VNet tunnel consists of two connection resources in Azure, one for each direction. Make sure both connection resources have the same policy, otherwise the VNet-to-VNet connection won't establish.

Does custom IPsec/IKE policy work on ExpressRoute connection?

No. IPsec/IKE policy only works on S2S VPN and VNet-to-VNet connections via the Azure VPN gateways.

Next steps

See [Configure IPsec/IKE policy](#) for step-by-step instructions on configuring custom IPsec/IKE policy on a connection.

See also [Connect multiple policy-based VPN devices](#) to learn more about the UsePolicyBasedTrafficSelectors option.

Overview of BGP with Azure VPN Gateways

8/31/2017 • 7 minutes to read • [Edit Online](#)

This article provides an overview of BGP (Border Gateway Protocol) support in Azure VPN Gateways.

BGP is the standard routing protocol commonly used in the Internet to exchange routing and reachability information between two or more networks. When used in the context of Azure Virtual Networks, BGP enables the Azure VPN Gateways and your on-premises VPN devices, called BGP peers or neighbors, to 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 a BGP gateway learns from one BGP peer to all other BGP peers.

Why use BGP?

BGP is an optional feature you can use with Azure Route-Based VPN gateways. You should also make sure your on-premises VPN devices support BGP before you enable the feature. You can continue to use Azure VPN gateways and your on-premises VPN devices without BGP. It is the equivalent of using static routes (without BGP) vs. using dynamic routing with BGP between your networks and Azure.

There are several advantages and new capabilities with BGP:

Support automatic and flexible prefix updates

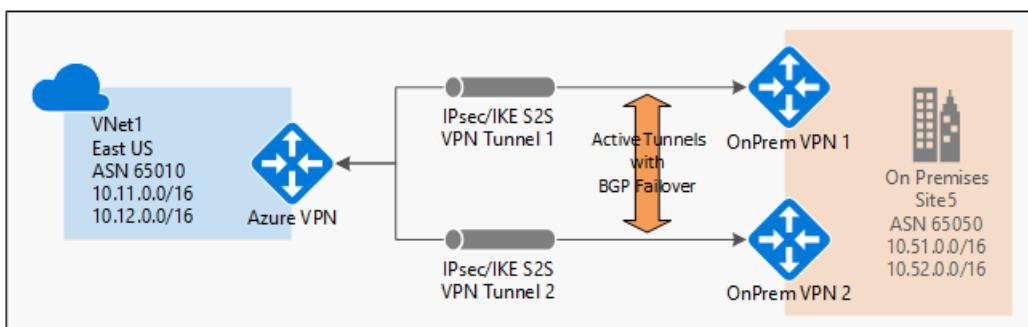
With BGP, you only need to declare a minimum prefix to a specific BGP peer over the IPsec S2S VPN tunnel. It can be as small as a host prefix (/32) of the BGP peer IP address of your on-premises VPN device. You can control which on-premises network prefixes you want to advertise to Azure to allow your Azure Virtual Network to access.

You can also advertise larger prefixes that may include some of your VNet address prefixes, such as a large private IP address space (for example, 10.0.0.0/8). Note though the prefixes cannot be identical with any one of your VNet prefixes. Those routes identical to your VNet prefixes will be rejected.

Support multiple tunnels between a VNet and an on-premises site with automatic failover based on BGP

You can establish multiple connections between your Azure VNet and your on-premises VPN devices in the same location. This capability provides multiple tunnels (paths) between the two networks in an active-active configuration. If one of the tunnels is disconnected, the corresponding routes will be withdrawn via BGP and the traffic automatically shifts to the remaining tunnels.

The following diagram shows a simple example of this highly available setup:

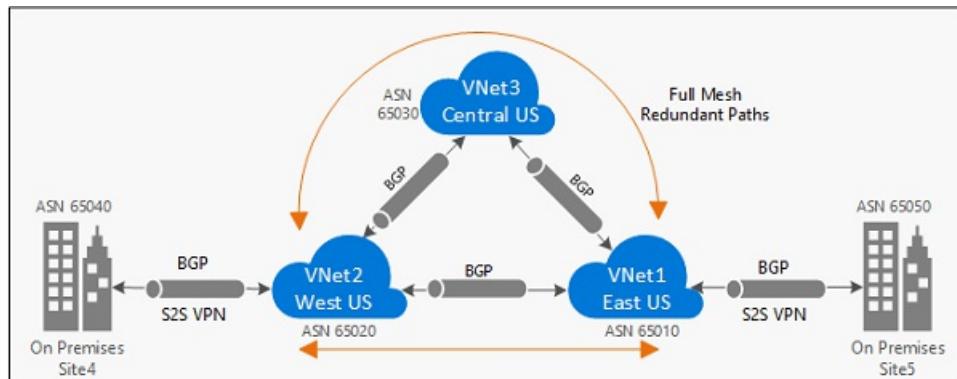


Support transit routing between your on-premises networks and multiple Azure VNets

BGP enables multiple gateways to learn and propagate prefixes from different networks, whether they are directly or indirectly connected. This can enable transit routing with Azure VPN gateways between your on-premises sites

or across multiple Azure Virtual Networks.

The following diagram shows an example of a multi-hop topology with multiple paths that can transit traffic between the two on-premises networks through Azure VPN gateways within the Microsoft Networks:



BGP FAQ

Is BGP supported on all Azure VPN Gateway SKUs?

No, BGP is supported on Azure **VpnGw1**, **VpnGw2**, **VpnGw3**, **Standard** and **HighPerformance** VPN gateways. **Basic** SKU is NOT supported.

Can I use BGP with Azure Policy-Based VPN gateways?

No, BGP is supported on Route-Based VPN gateways only.

Can I use private ASNs (Autonomous System Numbers)?

Yes, you can use your own public ASNs or private ASNs for both your on-premises networks and Azure virtual networks.

Can I use 32-bit ASNs (Autonomous System Numbers)?

No, the Azure VPN Gateways support 16-Bit ASNs today.

Are there ASNs reserved by Azure?

Yes, the following ASNs are reserved by Azure for both internal and external peerings:

- Public ASNs: 8074, 8075, 12076
- Private ASNs: 65515, 65517, 65518, 65519, 65520

You cannot specify these ASNs for your on-premises VPN devices when connecting to Azure VPN gateways.

Are there any other ASNs that I can't use?

Yes, the following ASNs are [reserved by IANA](#) and can't be configured on your Azure VPN Gateway:

23456, 64496-64511, 65535-65551 and 429496729

Can I use the same ASN for both on-premises VPN networks and Azure VNets?

No, you must assign different ASNs between your on-premises networks and your Azure VNets if you are connecting them together with BGP. Azure VPN Gateways have a default ASN of 65515 assigned, whether BGP is enabled or not for your cross-premises connectivity. You can override this default by assigning a different ASN when creating the VPN gateway, or change the ASN after the gateway is created. You will need to assign your on-premises ASNs to the corresponding Azure Local Network Gateways.

What address prefixes will Azure VPN gateways advertise to me?

Azure VPN gateway will advertise the following routes to your on-premises BGP devices:

- Your VNet address prefixes

- Address prefixes for each Local Network Gateways connected to the Azure VPN gateway
- Routes learned from other BGP peering sessions connected to the Azure VPN gateway, **except default route or routes overlapped with any VNet prefix.**

Can I advertise default route (0.0.0.0/0) to Azure VPN gateways?

Yes.

Please note this will force all VNet egress traffic towards your on-premises site, and will prevent the VNet VMs from accepting public communication from the Internet directly, such RDP or SSH from the Internet to the VMs.

Can I advertise the exact prefixes as my Virtual Network prefixes?

No, advertising the same prefixes as any one of your Virtual Network address prefixes will be blocked or filtered by the Azure platform. However you can advertise a prefix that is a superset of what you have inside your Virtual Network.

For example, if your virtual network used the address space 10.0.0.0/16, you could advertise 10.0.0.0/8. But you cannot advertise 10.0.0.0/16 or 10.0.0.0/24.

Can I use BGP with my VNet-to-VNet connections?

Yes, you can use BGP for both cross-premises connections and VNet-to-VNet connections.

Can I mix BGP with non-BGP connections for my Azure VPN gateways?

Yes, you can mix both BGP and non-BGP connections for the same Azure VPN gateway.

Does Azure VPN gateway support BGP transit routing?

Yes, BGP transit routing is supported, with the exception that Azure VPN gateways will **NOT** advertise default routes to other BGP peers. To enable transit routing across multiple Azure VPN gateways, you must enable BGP on all intermediate VNet-to-VNet connections.

Can I have more than one tunnel between Azure VPN gateway and my on-premises network?

Yes, you can establish more than one S2S VPN tunnel between an Azure VPN gateway and your on-premises network. Please note that all these tunnels will be counted against the total number of tunnels for your Azure VPN gateways and you must enable BGP on both tunnels.

For example, if you have two redundant tunnels between your Azure VPN gateway and one of your on-premises networks, they will consume 2 tunnels out of the total quota for your Azure VPN gateway (10 for Standard and 30 for HighPerformance).

Can I have multiple tunnels between two Azure VNets with BGP?

Yes, but at least one of the virtual network gateways must be in active-active configuration.

Can I use BGP for S2S VPN in an ExpressRoute/S2S VPN co-existence configuration?

Yes.

What address does Azure VPN gateway use for BGP Peer IP?

The Azure VPN gateway will allocate a single IP address from the GatewaySubnet range defined for the virtual network. By default, it is the second last address of the range. For example, if your GatewaySubnet is 10.12.255.0/27, ranging from 10.12.255.0 to 10.12.255.31, the BGP Peer IP address on the Azure VPN gateway will be 10.12.255.30. You can find this information when you list the Azure VPN gateway information.

What are the requirements for the BGP Peer IP addresses on my VPN device?

Your on-premises BGP peer address **MUST NOT** be the same as the public IP address of your VPN device. Use a different IP address on the VPN device for your BGP Peer IP. It can be an address assigned to the loopback interface on the device, but please note that it cannot be an APIPA (169.254.x.x) address. Specify this address in the corresponding Local Network Gateway representing the location.

What should I specify as my address prefixes for the Local Network Gateway when I use BGP?

Azure Local Network Gateway specifies the initial address prefixes for the on-premises network. With BGP, you must allocate the host prefix (/32 prefix) of your BGP Peer IP address as the address space for that on-premises network. If your BGP Peer IP is 10.52.255.254, you should specify "10.52.255.254/32" as the localNetworkAddressSpace of the Local Network Gateway representing this on-premises network. This is to ensure that the Azure VPN gateway establishes the BGP session through the S2S VPN tunnel.

What should I add to my on-premises VPN device for the BGP peering session?

You should add a host route of the Azure BGP Peer IP address on your VPN device pointing to the IPsec S2S VPN tunnel. For example, if the Azure VPN Peer IP is "10.12.255.30", you should add a host route for "10.12.255.30" with a nexthop interface of the matching IPsec tunnel interface on your VPN device.

Next steps

See [Getting started with BGP on Azure VPN gateways](#) for steps to configure BGP for your cross-premises and VNet-to-VNet connections.

Highly Available Cross-Premises and VNet-to-VNet Connectivity

8/9/2018 • 5 minutes to read • [Edit Online](#)

This article provides an overview of Highly Available configuration options for your cross-premises and VNet-to-VNet connectivity using Azure VPN gateways.

About Azure VPN gateway redundancy

Every Azure VPN gateway consists of two instances in an active-standby configuration. For any planned maintenance or unplanned disruption that happens to the active instance, the standby instance would take over (failover) automatically, and resume the S2S VPN or VNet-to-VNet connections. The switch over will cause a brief interruption. For planned maintenance, the connectivity should be restored within 10 to 15 seconds. For unplanned issues, the connection recovery will be longer, about 1 minute to 1 and a half minutes in the worst case. For P2S VPN client connections to the gateway, the P2S connections will be disconnected and the users will need to reconnect from the client machines.



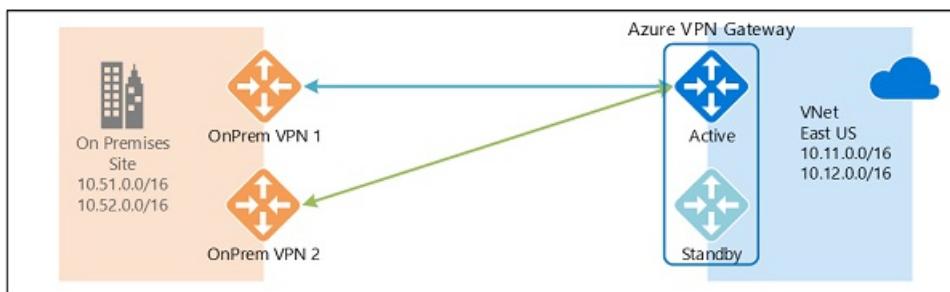
Highly Available Cross-Premises Connectivity

To provide better availability for your cross premises connections, there are a couple of options available:

- Multiple on-premises VPN devices
- Active-active Azure VPN gateway
- Combination of both

Multiple on-premises VPN devices

You can use multiple VPN devices from your on-premises network to connect to your Azure VPN gateway, as shown in the following diagram:



This configuration provides multiple active tunnels from the same Azure VPN gateway to your on-premises devices in the same location. There are some requirements and constraints:

1. You need to create multiple S2S VPN connections from your VPN devices to Azure. When you connect

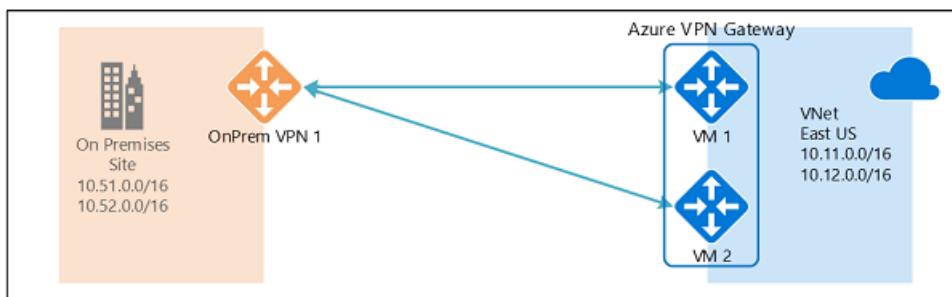
multiple VPN devices from the same on-premises network to Azure, you need to create one local network gateway for each VPN device, and one connection from your Azure VPN gateway to each local network gateway.

2. The local network gateways corresponding to your VPN devices must have unique public IP addresses in the "GatewayIpAddress" property.
3. BGP is required for this configuration. Each local network gateway representing a VPN device must have a unique BGP peer IP address specified in the "BgpPeerIpAddress" property.
4. The AddressPrefix property field in each local network gateway must not overlap. You should specify the "BgpPeerIpAddress" in /32 CIDR format in the AddressPrefix field, for example, 10.200.200.254/32.
5. You should use BGP to advertise the same prefixes of the same on-premises network prefixes to your Azure VPN gateway, and the traffic will be forwarded through these tunnels simultaneously.
6. Each connection is counted against the maximum number of tunnels for your Azure VPN gateway, 10 for Basic and Standard SKUs, and 30 for HighPerformance SKU.

In this configuration, the Azure VPN gateway is still in active-standby mode, so the same failover behavior and brief interruption will still happen as described [above](#). But this setup guards against failures or interruptions on your on-premises network and VPN devices.

Active-active Azure VPN gateway

You can now create an Azure VPN gateway in an active-active configuration, where both instances of the gateway VMs will establish S2S VPN tunnels to your on-premises VPN device, as shown the following diagram:



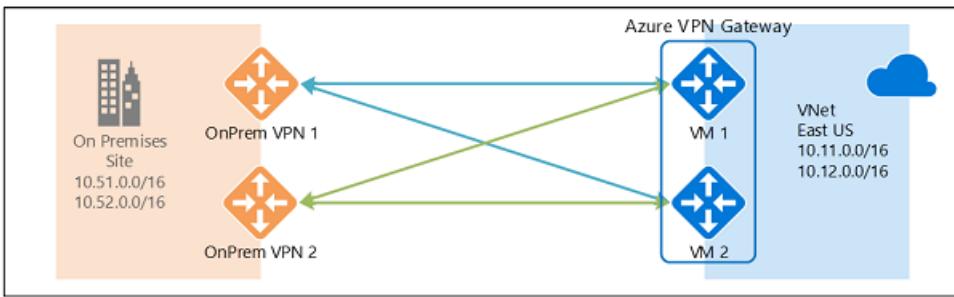
In this configuration, each Azure gateway instance will have a unique public IP address, and each will establish an IPsec/IKE S2S VPN tunnel to your on-premises VPN device specified in your local network gateway and connection. Note that both VPN tunnels are actually part of the same connection. You will still need to configure your on-premises VPN device to accept or establish two S2S VPN tunnels to those two Azure VPN gateway public IP addresses.

Because the Azure gateway instances are in active-active configuration, the traffic from your Azure virtual network to your on-premises network will be routed through both tunnels simultaneously, even if your on-premises VPN device may favor one tunnel over the other. Note though the same TCP or UDP flow will always traverse the same tunnel or path, unless a maintenance event happens on one of the instances.

When a planned maintenance or unplanned event happens to one gateway instance, the IPsec tunnel from that instance to your on-premises VPN device will be disconnected. The corresponding routes on your VPN devices should be removed or withdrawn automatically so that the traffic will be switched over to the other active IPsec tunnel. On the Azure side, the switch over will happen automatically from the affected instance to the active instance.

Dual-redundancy: active-active VPN gateways for both Azure and on-premises networks

The most reliable option is to combine the active-active gateways on both your network and Azure, as shown in the diagram below.



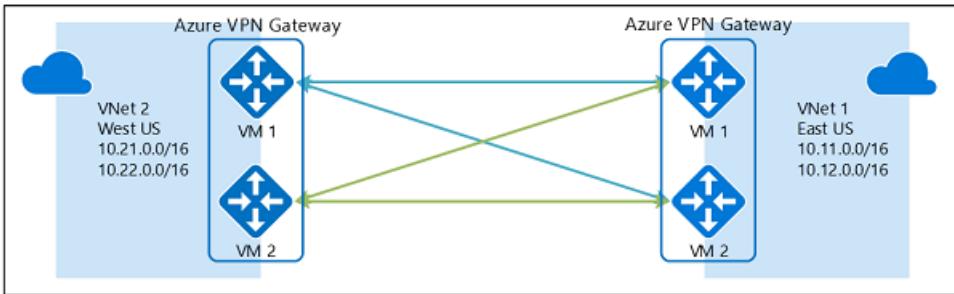
Here you create and setup the Azure VPN gateway in an active-active configuration, and create two local network gateways and two connections for your two on-premises VPN devices as described above. The result is a full mesh connectivity of 4 IPsec tunnels between your Azure virtual network and your on-premises network.

All gateways and tunnels are active from the Azure side, so the traffic will be spread among all 4 tunnels simultaneously, although each TCP or UDP flow will again follow the same tunnel or path from the Azure side. Even though by spreading the traffic, you may see slightly better throughput over the IPsec tunnels, the primary goal of this configuration is for high availability. And due to the statistical nature of the spreading, it is difficult to provide the measurement on how different application traffic conditions will affect the aggregate throughput.

This topology will require two local network gateways and two connections to support the pair of on-premises VPN devices, and BGP is required to allow the two connections to the same on-premises network. These requirements are the same as the [above](#).

Highly Available VNet-to-VNet Connectivity through Azure VPN Gateways

The same active-active configuration can also apply to Azure VNet-to-VNet connections. You can create active-active VPN gateways for both virtual networks, and connect them together to form the same full mesh connectivity of 4 tunnels between the two VNets, as shown in the diagram below:



This ensures there are always a pair of tunnels between the two virtual networks for any planned maintenance events, providing even better availability. Even though the same topology for cross-premises connectivity requires two connections, the VNet-to-VNet topology shown above will need only one connection for each gateway. Additionally, BGP is optional unless transit routing over the VNet-to-VNet connection is required.

Next steps

See [Configuring Active-Active VPN Gateways for Cross-Premises and VNet-to-VNet Connections](#) for steps to configure active-active cross-premises and VNet-to-VNet connections.

About Point-to-Site VPN

7/12/2018 • 15 minutes to read • [Edit Online](#)

A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer. A P2S connection is established by starting it from the client computer. This solution is useful for telecommuters who want to connect to Azure VNets from a remote location, such as from home or a conference. P2S VPN is also a useful solution to use instead of S2S VPN when you have only a few clients that need to connect to a VNet. This article applies to the Resource Manager deployment model.

What protocol does P2S use?

Point-to-site VPN can use one of the following protocols:

- Secure Socket Tunneling Protocol (SSTP), a proprietary SSL-based VPN protocol. An SSL VPN solution can penetrate firewalls, since most firewalls open TCP port 443, which SSL uses. SSTP is only supported on Windows devices. Azure supports all versions of Windows that have SSTP (Windows 7 and later).
- IKEv2 VPN, a standards-based IPsec VPN solution. IKEv2 VPN can be used to connect from Mac devices (OSX versions 10.11 and above).

If you have a mixed client environment consisting of Windows and Mac devices, configure both SSTP and IKEv2.

NOTE

IKEv2 for P2S is available for the Resource Manager deployment model only. It is not available for the classic deployment model.

How are P2S VPN clients authenticated?

Before Azure accepts a P2S VPN connection, the user has to be authenticated first. There are two mechanisms that Azure offers to authenticate a connecting user.

Authenticate using native Azure certificate authentication

When using the native Azure certificate authentication, a client certificate that is present on the device is used to authenticate the connecting user. Client certificates are generated from a trusted root certificate and then installed on each client computer. You can use a root certificate that was generated using an Enterprise solution, or you can generate a self-signed certificate.

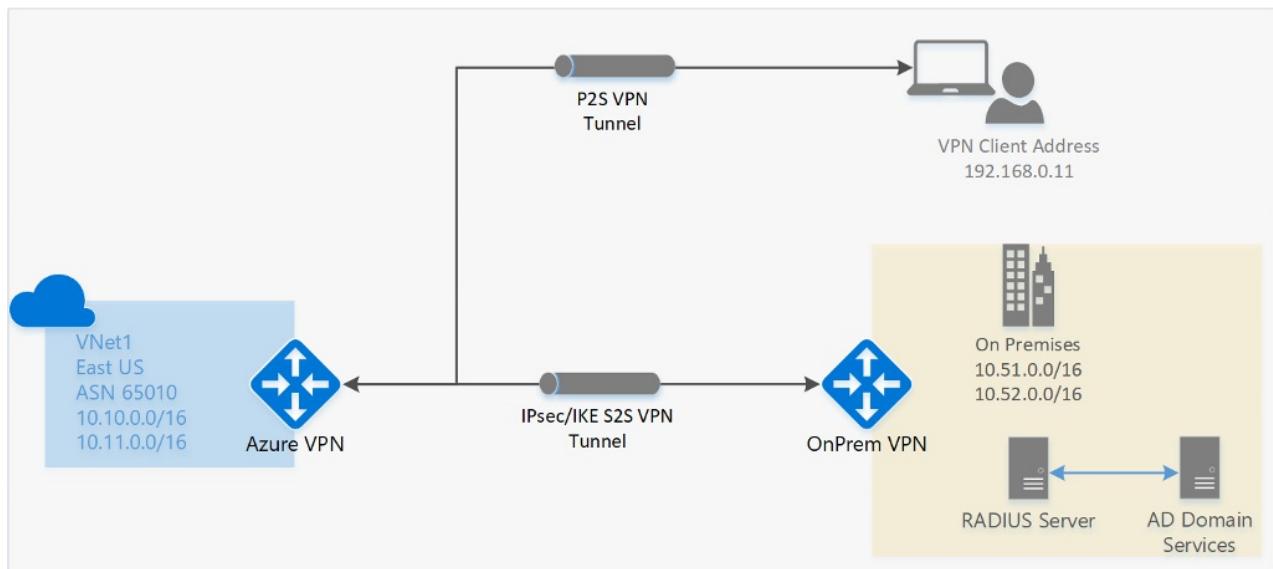
The validation of the client certificate is performed by the VPN gateway and happens during establishment of the P2S VPN connection. The root certificate is required for the validation and must be uploaded to Azure.

Authenticate using Active Directory (AD) Domain Server

AD Domain authentication allows users to connect to Azure using their organization domain credentials. It requires a RADIUS server that integrates with the AD server. Organizations can also leverage their existing RADIUS deployment. The RADIUS server could be deployed on-premises or in your Azure VNET. During authentication, the Azure VPN Gateway acts as a pass through and forwards authentication messages back and forth between the RADIUS server and the connecting device. So Gateway reachability to the RADIUS server is important. If the RADIUS server is present on-premises, then a VPN S2S connection from Azure to the on-premises site is required for reachability. The RADIUS server can also integrate with AD certificate services. This lets you use the RADIUS server and your enterprise certificate deployment for P2S certificate authentication as an alternative to the Azure certificate authentication. The advantage is that you don't need to upload root certificates

and revoked certificates to Azure.

A RADIUS server can also integrate with other external identity systems. This opens up plenty of authentication options for P2S VPN, including multi-factor options.



What are the client configuration requirements?

NOTE

For Windows clients, you must have administrator rights on the client device in order to initiate the VPN connection from the client device to Azure.

Users use the native VPN clients on Windows and Mac devices for P2S. Azure provides a VPN client configuration zip file that contains settings required by these native clients to connect to Azure.

- For Windows devices, the VPN client configuration consists of an installer package that users install on their devices.
- For Mac devices, it consists of the mobileconfig file that users install on their devices.

The zip file also provides the values of some of the important settings on the Azure side that you can use to create your own profile for these devices. Some of the values include the VPN gateway address, configured tunnel types, routes, and the root certificate for gateway validation.

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. Only point-to-site connections are impacted; site-to-site connections will not be affected. If you're using TLS for point-to-site VPNs on Windows 10 clients, you don't need to take any action. If you are using TLS for point-to-site connections on Windows 7 and Windows 8 clients, see the [VPN Gateway FAQ](#) for update instructions.

Which Gateway SKUs Support P2S VPN?

SKU	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	RADIUS AUTHENTICATION	IKEV2 P2S VPN
VpnGw1	128	650 Mbps	Supported	Supported

SKU	P2S CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	RADIUS AUTHENTICATION	IKEV2 P2S VPN
VpnGw2	128	1Gbps	Supported	Supported
VpnGw3	128	1.25 Gbps	Supported	Supported
Basic	128	100 Mbps	Not Supported	Not Supported

- Aggregate Throughput Benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to internet traffic conditions and your application behaviors.
- Pricing information can be found on the [Pricing page](#)
- SLA (Service Level Agreement) information can be found on the [SLA page](#).

NOTE

The Basic SKU does not support IKEv2 or RADIUS authentication.

How do I configure a P2S connection?

A P2S configuration requires quite a few specific steps. The following articles contain the steps to walk you through P2S configuration, and links to configure the VPN client devices:

- [Configure a P2S connection - RADIUS authentication](#)
- [Configure a P2S connection - Azure native certificate authentication](#)

FAQ for native Azure certificate authentication

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:
 - [KB3140245](#)
 - [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2. Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\
IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

What tools can I use to create certificates?

You can use your Enterprise PKI solution (your internal PKI), Azure PowerShell, MakeCert, and OpenSSL.

Are there instructions for certificate settings and parameters?

- **Internal PKI/Enterprise PKI solution:** See the steps to [Generate certificates](#).
- **Azure PowerShell:** See the [Azure PowerShell](#) article for steps.
- **MakeCert:** See the [MakeCert](#) article for steps.
- **OpenSSL:**
 - When exporting certificates, be sure to convert the root certificate to Base64.
 - For the client certificate:
 - When creating the private key, specify the length as 4096.

- When creating the certificate, for the `-extensions` parameter, specify `usr_cert`.

FAQ for RADIUS authentication

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2. Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\
IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the

Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Is RADIUS authentication supported on all Azure VPN Gateway SKUs?

RADIUS authentication is supported for VpnGw1, VpnGw2, and VpnGw3 SKUs. If you are using legacy SKUs, RADIUS authentication is supported on Standard and High Performance SKUs. It is not supported on the Basic Gateway SKU.

Is RADIUS authentication supported for the classic deployment model?

No. RADIUS authentication is not supported for the classic deployment model.

Are 3rd-party RADIUS servers supported?

Yes, 3rd-party RADIUS servers are supported.

What are the connectivity requirements to ensure that the Azure gateway is able to reach an on-premises RADIUS server?

A VPN Site-to-Site connection to the on-premises site, with the proper routes configured, is required.

Can traffic to an on-premises RADIUS server (from the Azure VPN gateway) be routed over an ExpressRoute connection?

No. It can only be routed over a Site-to-Site connection.

Is there a change in the number of SSTP connections supported with RADIUS authentication? What is the maximum number of SSTP and IKEv2 connections supported?

There is no change in the maximum number of SSTP connections supported on a gateway with RADIUS authentication. It remains 128. The maximum number of connections supported is 128, irrespective of whether the gateway is configured for SSTP, IKEv2, or both.

What is the difference between doing certificate authentication using a RADIUS server vs. using Azure native certificate authentication (by uploading a trusted certificate to Azure)?

In RADIUS certificate authentication, the authentication request is forwarded to a RADIUS server that handles the actual certificate validation. This option is useful if you want to integrate with a certificate authentication infrastructure that you already have through RADIUS.

When using Azure for certificate authentication, the Azure VPN gateway performs the validation of the certificate. You need to upload your certificate public key to the gateway. You can also specify list of revoked certificates that shouldn't be allowed to connect.

Does RADIUS authentication work with both IKEv2, and SSTP VPN?

Yes, RADIUS authentication is supported for both IKEv2, and SSTP VPN.

Next Steps

- [Configure a P2S connection - RADIUS authentication](#)
- [Configure a P2S connection - Azure native certificate authentication](#)

About Point-to-Site VPN routing

5/31/2018 • 6 minutes to read • [Edit Online](#)

This article helps you understand how Azure Point-to-Site VPN routing behaves. P2S VPN routing behavior is dependent on the client OS, the protocol used for the VPN connection, and how the virtual networks (VNets) are connected to each other.

Azure currently supports two protocols for remote access, IKEv2 and SSTP. IKEv2 is supported on many client operating systems including Windows, Linux, MacOS, Android, and iOS. SSTP is only supported on Windows. If you make a change to the topology of your network and have Windows VPN clients, the VPN client package for Windows clients must be downloaded and installed again in order for the changes to be applied to the client.

NOTE

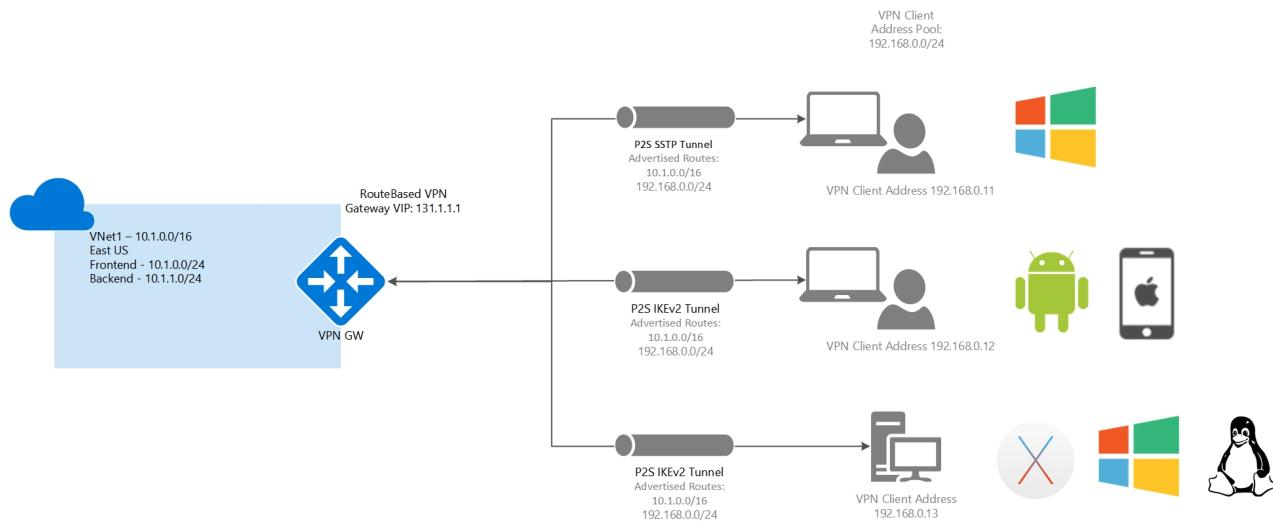
This article applies to IKEv2 only.

About the diagrams

There are a number of different diagrams in this article. Each section shows a different topology or configuration. For the purposes of this article, Site-to-Site (S2S) and VNet-to-VNet connections function the same way, as both are IPsec tunnels. All VPN gateways in this article are route-based.

One isolated VNet

The Point-to-Site VPN gateway connection in this example is for a VNet that is not connected or peered with any other virtual network (VNet1). In this example, clients using SSTP or IKEv2 can access VNet1.



Address space

- VNet1: 10.1.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to non-Windows clients: 10.1.0.0/16, 192.168.0.0/24

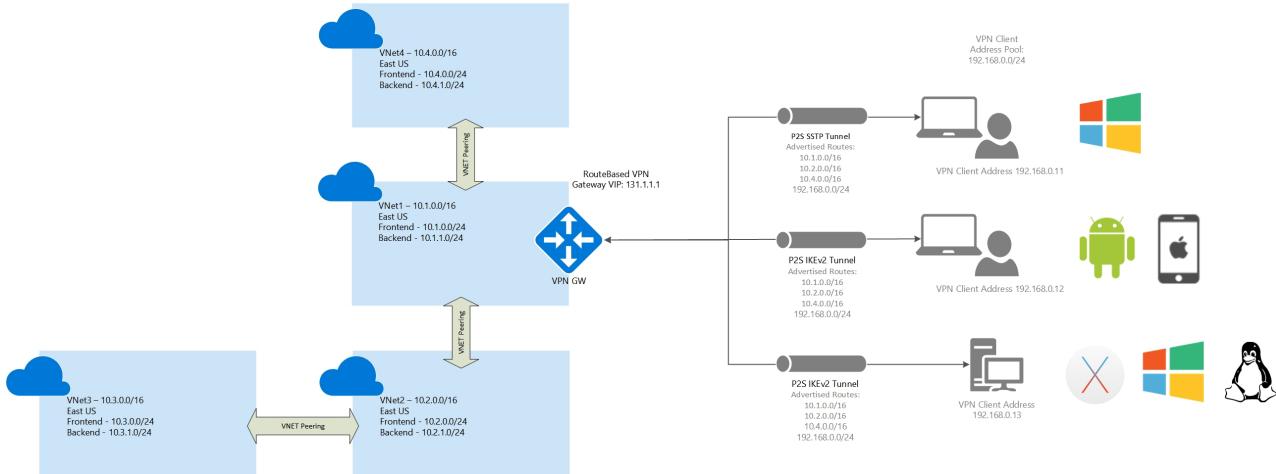
Access

- Windows clients can access VNet1
- Non-Windows clients can access VNet1

Multiple peered VNets

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is peered with VNet2. VNet 2 is peered with VNet3. VNet1 is peered with VNet4. There is no direct peering between VNet1 and VNet3. VNet1 has "Allow gateway transit" and VNet2 has "Use remote gateways" enabled.

Clients using Windows can access directly peered VNets, but the VPN client must be downloaded again if any changes are made to VNet peering or the network topology. Non-Windows clients can access directly peered VNets. Access is not transitive and is limited to only directly peered VNets.



Address space:

- VNet1: 10.1.0.0/16
- VNet2: 10.2.0.0/16
- VNet3: 10.3.0.0/16
- VNet4: 10.4.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16, 10.2.0.0/16, 10.4.0.0/16, 192.168.0.0/24
- Routes added to non-Windows clients: 10.1.0.0/16, 10.2.0.0/16, 10.4.0.0/16, 192.168.0.0/24

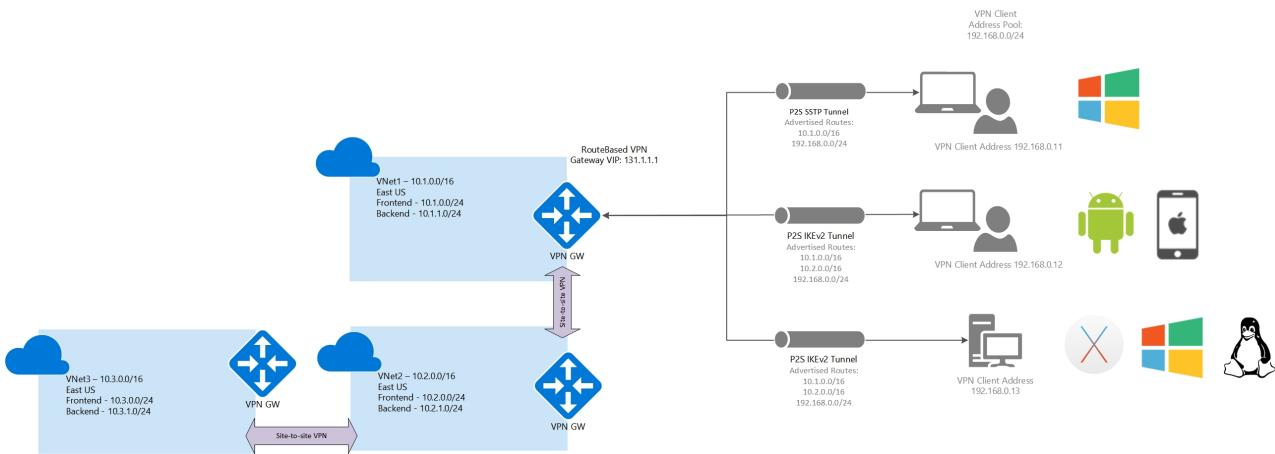
Access

- Windows clients can access VNet1, VNet2, and VNet4, but the VPN client must be downloaded again for any topology changes to take effect.
- Non-Windows clients can access VNet1, VNet2, and VNet4

Multiple VNets connected using an S2S VPN

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is connected to VNet2 using a Site-to-Site VPN connection. VNet2 is connected to VNet3 using a Site-to-Site VPN connection. There is no direct peering or Site-to-Site VPN connection between VNet1 and VNet3. All Site-to-Site connections are not running BGP for routing.

Clients using Windows, or another supported OS, can only access VNet1. To access additional VNets, BGP must be used.



Address space

- VNet1: 10.1.0.0/16
- VNet2: 10.2.0.0/16
- VNet3: 10.3.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.2.0.0/16, 192.168.0.0/24

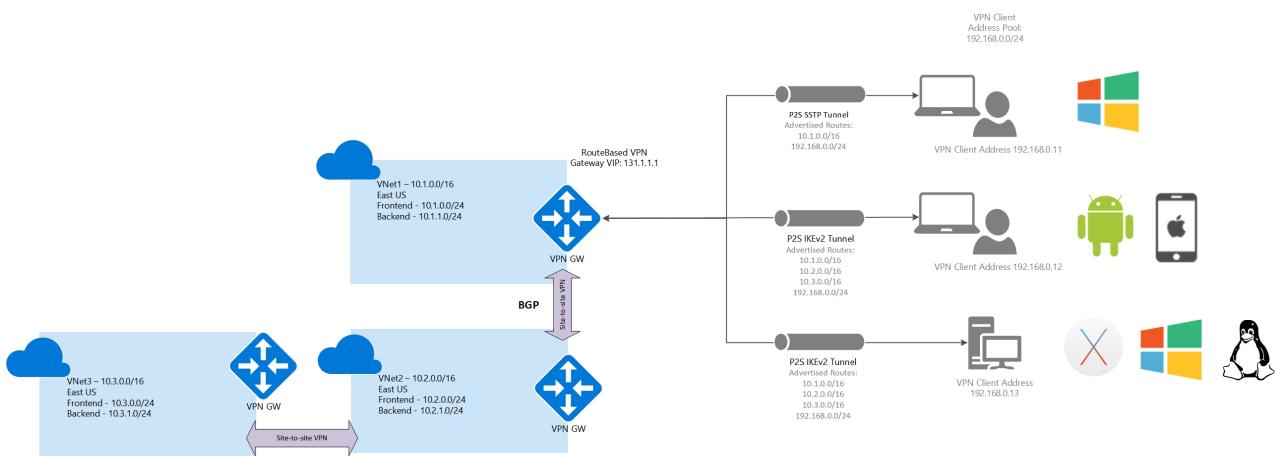
Access

- Windows clients can only access VNet1
- Non-Windows clients can access VNet1 only

Multiple VNets connected using an S2S VPN (BGP)

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is connected to VNet2 using a Site-to-Site VPN connection. VNet2 is connected to VNet3 using a Site-to-Site VPN connection. There is no direct peering or Site-to-Site VPN connection between VNet1 and VNet3. All Site-to-Site connections are running BGP for routing.

Clients using Windows, or another supported OS, can access all VNets that are connected using a Site-to-Site VPN connection, but routes to connected VNets have to be manually added to the Windows clients.



Address space

- VNet1: 10.1.0.0/16
- VNet2: 10.2.0.0/16

- VNet3: 10.3.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16, 192.168.0.0/24

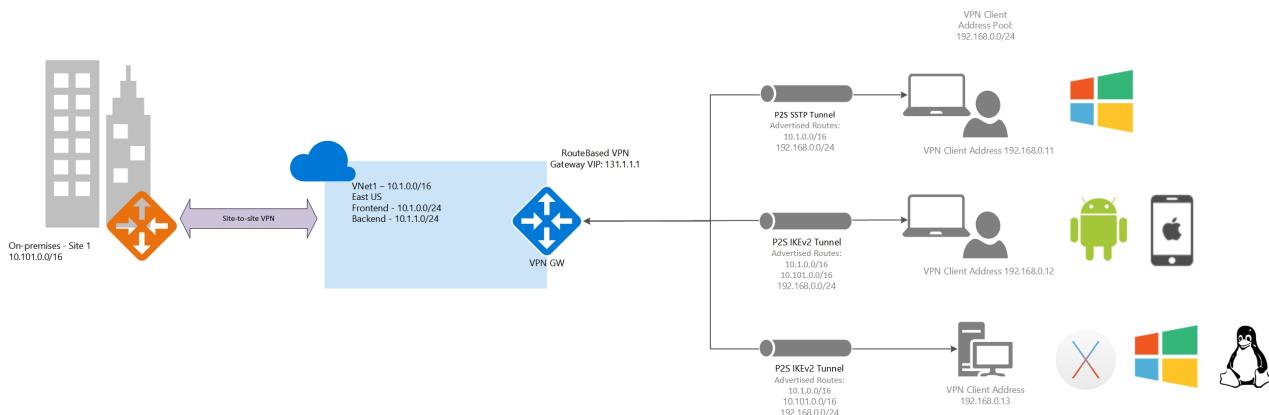
Access

- Windows clients can access VNet1, VNet2, and VNet3, but routes to VNet2 and VNet3 will have to be manually added.
- Non-Windows clients can access VNet1, VNet2, and VNet3

One VNet and a branch office

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is not connected/ peered with any other virtual network, but is connected to an on-premises site through a Site-to-Site VPN connection that is not running BGP.

Windows clients can access VNet1 and the branch office (Site1), but the routes to Site1 must be manually added to the client. Non-Windows clients can access VNet1, as well as the on-premises Site1.



Address space

- VNet1: 10.1.0.0/16
- Site1: 10.101.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.101.0.0/16, 192.168.0.0/24

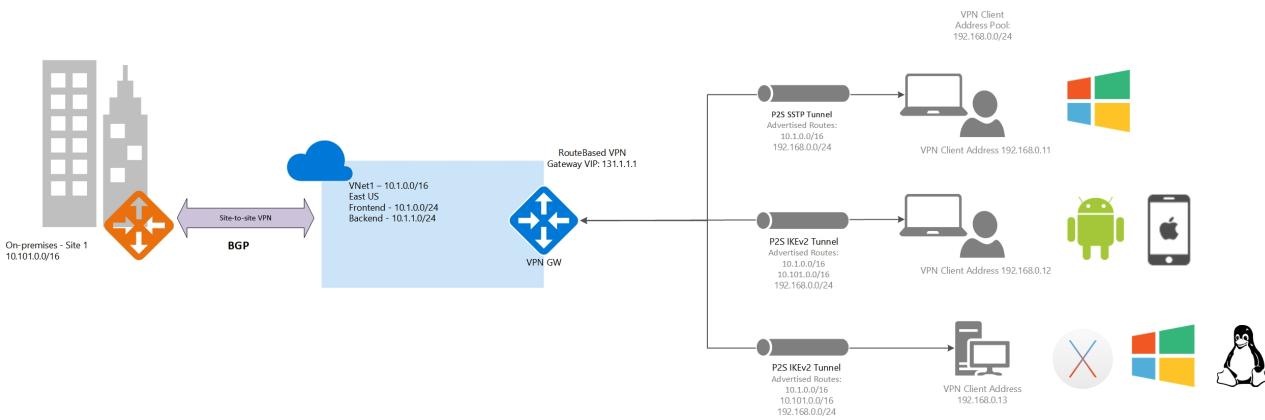
Access

- Windows clients can access only VNet1
- Non-Windows clients can access VNet1 only

One VNet and a branch office (BGP)

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is not connected or peered with any other virtual network, but is connected to an on-premises site (Site1) through a Site-to-Site VPN connection running BGP.

Windows clients can access the VNet and the branch office (Site1), but the routes to Site1 must be manually added to the client. Non-Windows clients can access the VNet as well as the on-premises branch office.



Address space

- VNet1: 10.1.0.0/16
- Site1: 10.101.0.0/16

Routes added

- Routes added to Windows clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.101.0.0/16, 192.168.0.0/24

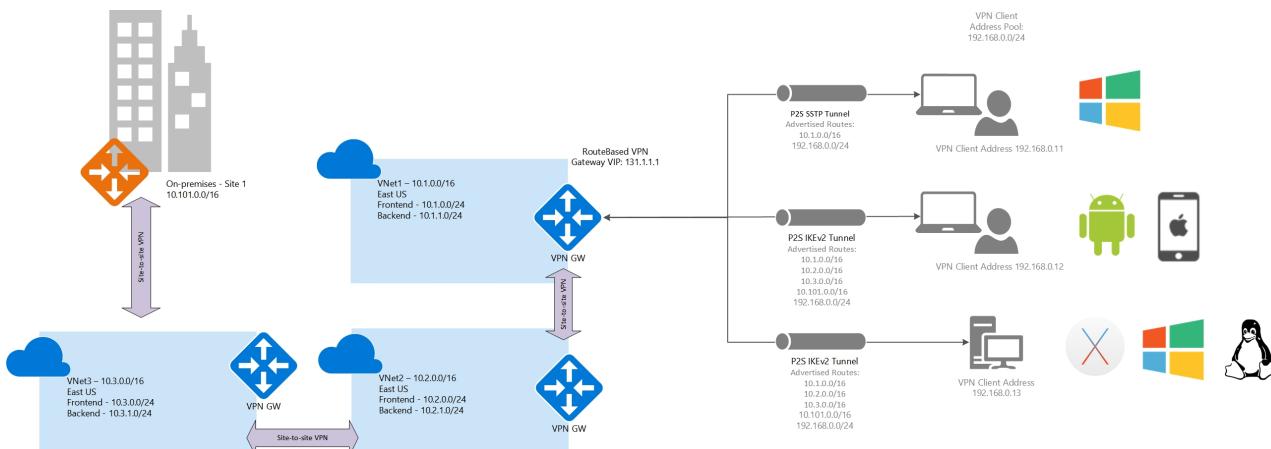
Access

- Windows clients can access VNet1 and Site1, but routes to Site1 will have to be manually added.
- Non-Windows clients can access VNet1 and Site1.

Multiple VNets connected using S2S and a branch office

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is connected to VNet2 using a Site-to-Site VPN connection. VNet2 is connected to VNet3 using a Site-to-Site VPN connection. There is no direct peering or Site-to-Site VPN tunnel between the VNet1 and VNet3 networks. VNet3 is connected to a branch office (Site1) using a Site-to-Site VPN connection. All VPN connections are not running BGP.

All clients can access VNet1 only.



Address space

- VNet1: 10.1.0.0/16
- VNet2: 10.2.0.0/16
- VNet3: 10.3.0.0/16
- Site1: 10.101.0.0/16

Routes added

- Routes added clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16, 10.101.0.0/16, 192.168.0.0/24

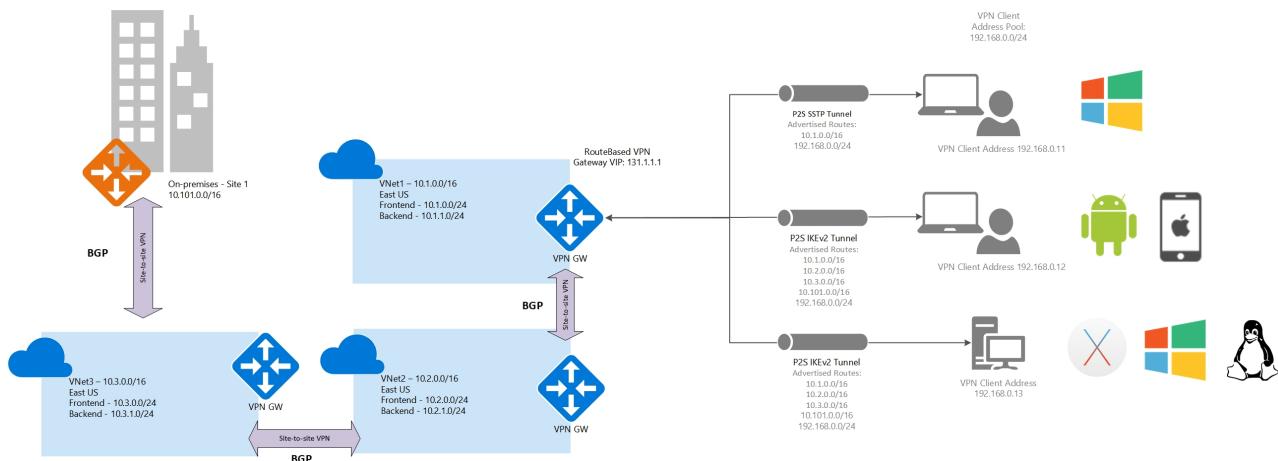
Access

- The Windows clients can access VNet1 only
- Non-Windows clients can access VNet1 only

Multiple VNets connected using S2S and a branch office (BGP)

In this example, the Point-to-Site VPN gateway connection is for VNet1. VNet1 is connected to VNet2 using a Site-to-Site VPN connection. VNet2 is connected to VNet3 using a Site-to-Site VPN connection. There is no direct peering or Site-to-Site VPN tunnel between the VNet1 and VNet3 networks. VNet3 is connected to a branch office (Site1) using a Site-to-Site VPN connection. All VPN connections are running BGP.

Clients using Windows can access VNets and sites that are connected using a Site-to-Site VPN connection, but the routes to VNet2, VNet3 and Site1 must be manually added to the client. Non-Windows clients can access VNets and sites that are connected using a Site-to-Site VPN connection without any manual intervention. The access is transitive, and clients can access resources in all connected VNets and sites (on-premises).



Address space

- VNet1: 10.1.0.0/16
- VNet2: 10.2.0.0/16
- VNet3: 10.3.0.0/16
- Site1: 10.101.0.0/16

Routes added

- Routes added clients: 10.1.0.0/16, 192.168.0.0/24
- Routes added to Non-Windows clients: 10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16, 10.101.0.0/16, 192.168.0.0/24

Access

- The Windows clients can access VNet1, VNet2, VNet3, and Site1, but routes to VNet2, VNet3 and Site1 must be manually added to the client.
- Non-Windows clients can access VNet1, VNet2, VNet3, and Site1.

Next steps

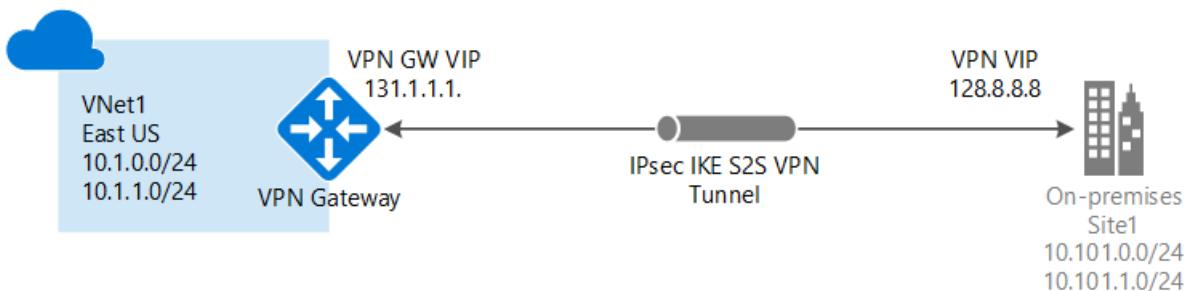
See [Create a P2S VPN using the Azure portal](#) to begin creating your P2S VPN.

Create a Site-to-Site connection in the Azure portal

4/9/2018 • 19 minutes to read • [Edit Online](#)

This article 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. The steps in this article apply to the Resource Manager deployment model. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it. For more information about VPN gateways, see [About VPN gateway](#).



Before you begin

Verify that you have met the following criteria before beginning your configuration:

- 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. This IP address cannot be located behind a NAT.
- 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.

Example values

The examples in this article use the following values. You can use these values to create a test environment, or refer to them to better understand the examples in this article. For more information about VPN Gateway settings in general, see [About VPN Gateway Settings](#).

- **VNet Name:** TestVNet1
- **Address Space:** 10.1.0.0/16
- **Subscription:** The subscription you want to use
- **Resource Group:** TestRG1
- **Location:** East US
- **Subnet:** FrontEnd: 10.1.0.0/24, BackEnd: 10.1.1.0/24 (optional for this exercise)
- **Gateway Subnet name:** GatewaySubnet (this will auto-fill in the portal)
- **Gateway Subnet address range:** 10.1.255.0/27
- **DNS Server:** 8.8.8.8 - Optional. The IP address of your DNS server.

- **Virtual Network Gateway Name:** VNet1GW
- **Public IP:** VNet1GWIP
- **VPN Type:** Route-based
- **Connection Type:** Site-to-site (IPsec)
- **Gateway Type:** VPN
- **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. Create a virtual network

To create a VNet in the Resource Manager deployment model by using the Azure portal, follow the steps below. Use the [example values](#) if you are using these steps as a tutorial. If you are not doing these steps as a tutorial, be sure to replace the values with your own. For more information about working with virtual networks, see the [Virtual Network Overview](#).

NOTE

In order for this VNet to connect to an on-premises location you need 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 does not route the way you may expect it to. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Take care to plan your network configuration accordingly.

1. From a browser, navigate to the [Azure portal](#) and sign in with your Azure account.
2. Click **Create a resource**. In the **Search the marketplace** field, type 'virtual network'. Locate **Virtual network** from the returned list and click to open the **Virtual Network** page.
3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, select **Resource Manager**, and then click **Create**. This opens the 'Create virtual network' page.

Create virtual network X

* Name
VNet1 ✓

* Address space !
10.1.0.0/16 ✓
10.1.0.0 - 10.1.255.255 (65536 addresses)

* Subscription
Windows Azure Internal Consumption ▼

* Resource group
 Create new Use existing
TestRG1 ✓

* Location
East US ▼

Subnet

* Name
Frontend ✓

* Address range !
10.1.0.0/24 ✓
10.1.0.0 - 10.1.0.255 (256 addresses)

Service endpoints !
 Disabled Enabled

Pin to dashboard

Create Automation options

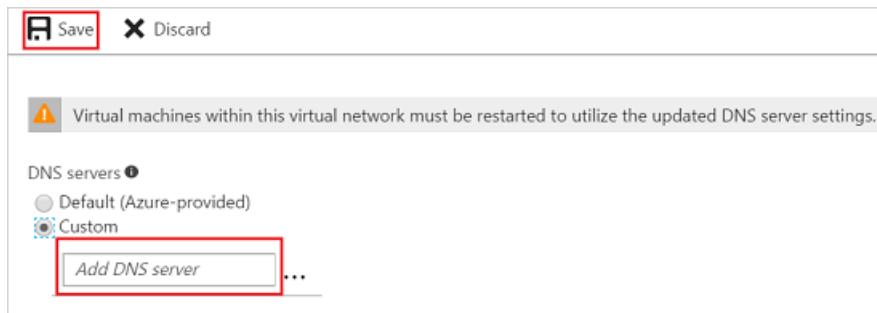
- On the **Create virtual network** page, configure the VNet settings. When you fill in the fields, the red exclamation mark becomes a green check mark when the characters entered in the field are valid.
 - Name:** Enter the name for your virtual network. In this example, we use VNet1.
 - Address space:** Enter the address space. If you have multiple address spaces to add, add your first address space. You can add additional address spaces later, after creating the VNet. Make sure that the address space that you specify does not overlap with the address space for your on-premises location.

- **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 create a new one by typing a name for your new resource group. If you are creating a new group, name the resource group according to your planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](#).
 - **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
 - **Subnet:** Add the first subnet name and subnet address range. You can add additional subnets and the gateway subnet later, after creating this VNet.
5. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**. After clicking **Create**, you will see a tile on your dashboard that will reflect the progress of your VNet. The tile changes as the VNet is being created.

2. Specify a DNS server

DNS is not required to create a Site-to-Site connection. However, if you want to have name resolution for resources that are deployed to your virtual network, you should specify a DNS server. This setting lets you specify the DNS server that you want to use for name resolution for this virtual network. It does not create a DNS server. For more information about name resolution, see [Name Resolution for VMs and role instances](#).

1. On the **Settings** page for your virtual network, navigate to **DNS Servers** and click to open the **DNS servers** page.



- **DNS Servers:** Select **Custom**.
 - **Add DNS server:** Enter the IP address of the DNS server that you want to use for name resolution.
2. When you are done adding DNS servers, click **Save** at the top of the page.

3. Create 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. The subnet must be named 'GatewaySubnet' in order for Azure to deploy the gateway resources. You can't specify a different subnet to deploy the gateway resources to. If you don't have a subnet named 'GatewaySubnet', when you create your VPN gateway, it will fail.

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.

1. In the portal, navigate to the virtual network for which you want to create a virtual network gateway.
2. In the **Settings** section of your VNet page, click **Subnets** to expand the Subnets page.
3. On the **Subnets** page, click **+Gateway subnet** at the top to open the **Add subnet** page.

The screenshot shows the 'Add subnet' interface. At the top, there are two buttons: '+ Subnet' and '+ Gateway subnet'. The '+ Gateway subnet' button is highlighted with a red border. Below these buttons is a search bar labeled 'Search subnets'. The main area has three columns: 'NAME', 'ADDRESS RANGE', and 'AVAILABLE ADDRESSES'. The 'NAME' column contains the placeholder text 'GatewaySubnet'.

4. The **Name** for your subnet is automatically filled in with the value 'GatewaySubnet'. The **GatewaySubnet** value is required in order for Azure to recognize the subnet as the gateway subnet. Adjust the auto-filled **Address range** values to match your configuration requirements.

The screenshot shows the 'Add subnet' interface for 'VNet1'. The 'Name' field is set to 'GatewaySubnet'. The 'Address range (CIDR block)' field is set to '10.1.255.0/27', which is described as '10.1.1.0 - 10.1.1.255 (251 + 5 Azure reserved addresses)'. Below these fields, there are sections for 'Route table' (set to 'None') and 'Service endpoints' (with a dropdown menu showing '0 selected').

5. To create the subnet, click **OK** at the bottom of the page.

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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

4. Create the VPN gateway

1. On the left side of the portal page, click **+** and type 'Virtual Network Gateway' in search. In **Results**, locate and click **Virtual network gateway**.
2. At the bottom of the 'Virtual network gateway' page, click **Create**. This opens the **Create virtual network gateway** page.
3. On the **Create virtual network gateway** page, specify the values for your virtual network gateway.

Create virtual network gateway

* Name

✓

Gateway type ⓘ

 VPN ExpressRoute

VPN type ⓘ

 Route-based Policy-based

* SKU ⓘ

▼

Enable active-active mode ⓘ

* Virtual network ⓘ

 >

* First IP configuration

 >

Configure BGP ASN

* Subscription

▼

Resource group ⓘ

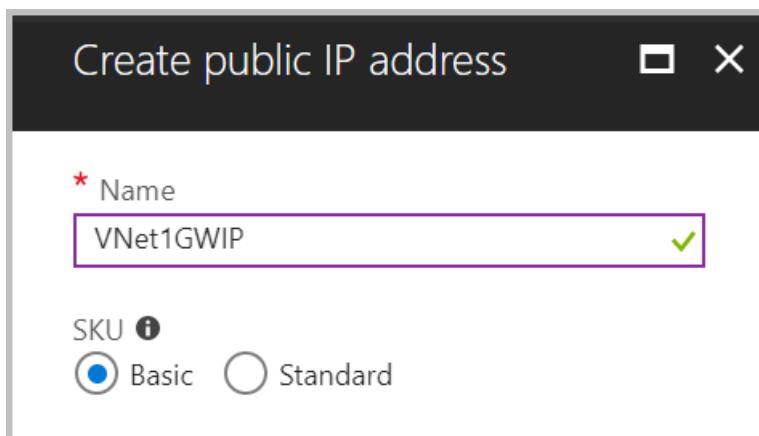
Pin to dashboard

Provisioning a virtual network gateway may take up to 45 minutes.

- **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **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](#).

- **Location:** You may need to scroll to see Location. Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, when you select a virtual network in the next step, it will not appear in the drop-down list.
- **Virtual network:** Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the 'Choose a virtual network' page. Select the VNet. If you don't see your VNet, make sure the Location field is pointing to the region in which your virtual network is located.
- **Gateway subnet address range:** You will only see this setting if you did not previously create a gateway subnet for your virtual network. If you previously created a valid gateway subnet, this setting will not appear.
- **First IP configuration:** The 'Choose public IP address' page creates a 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. VPN Gateway currently only supports *Dynamic* Public IP address allocation. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.
 - First, click **Create gateway IP configuration** to open the 'Choose public IP address' page, then click **+Create new** to open the 'Create public IP address' page.
 - Next, input a **Name** for your public IP address. Leave the SKU as **Basic** unless there is a specific reason to change it to something else, then click **OK** at the bottom of this page to save your changes.



4. Verify the settings. You can select **Pin to dashboard** at the bottom of the page if you want your gateway to appear on the dashboard.
5. Click **Create** to begin creating the VPN gateway. The settings are validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.

After the gateway is created, 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. You can click the connected device (your virtual network gateway) to view more information.

5. Create the local network gateway

The local network gateway typically refers to your on-premises location. 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.

1. In the portal, click **+Create a resource**.
2. In the search box, type **Local network gateway**, then press **Enter** to search. This will return a list of results. Click **Local network gateway**, then click the **Create** button to open the **Create local network gateway** page.

Create local network gateway

* Name
Site1 ✓

* IP address ⓘ
128.8.8.8 ✓

Address space ⓘ
10.101.1.0/24 ...
10.101.0.0/24 ...
Add additional address range ...

Configure BGP settings

* Subscription
Windows Azure Internal Consumption ▾

* Resource group ⓘ
 Create new Use existing
TestRG1 ▾

* Location
East US ▾

Pin to dashboard

Create Automation options

3. On the **Create local network gateway page**, specify the values for your local network gateway.

- **Name:** Specify a name for your local network gateway object.

- **IP address:** This is the public IP address of the VPN device that you want Azure to connect to. Specify a valid public IP address. The IP address cannot be behind NAT and has to be reachable by Azure. 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:** 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.

4. When you have finished specifying the values, click the **Create** button at the bottom of the page to create the local network gateway.

6. 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:

- 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 click 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,

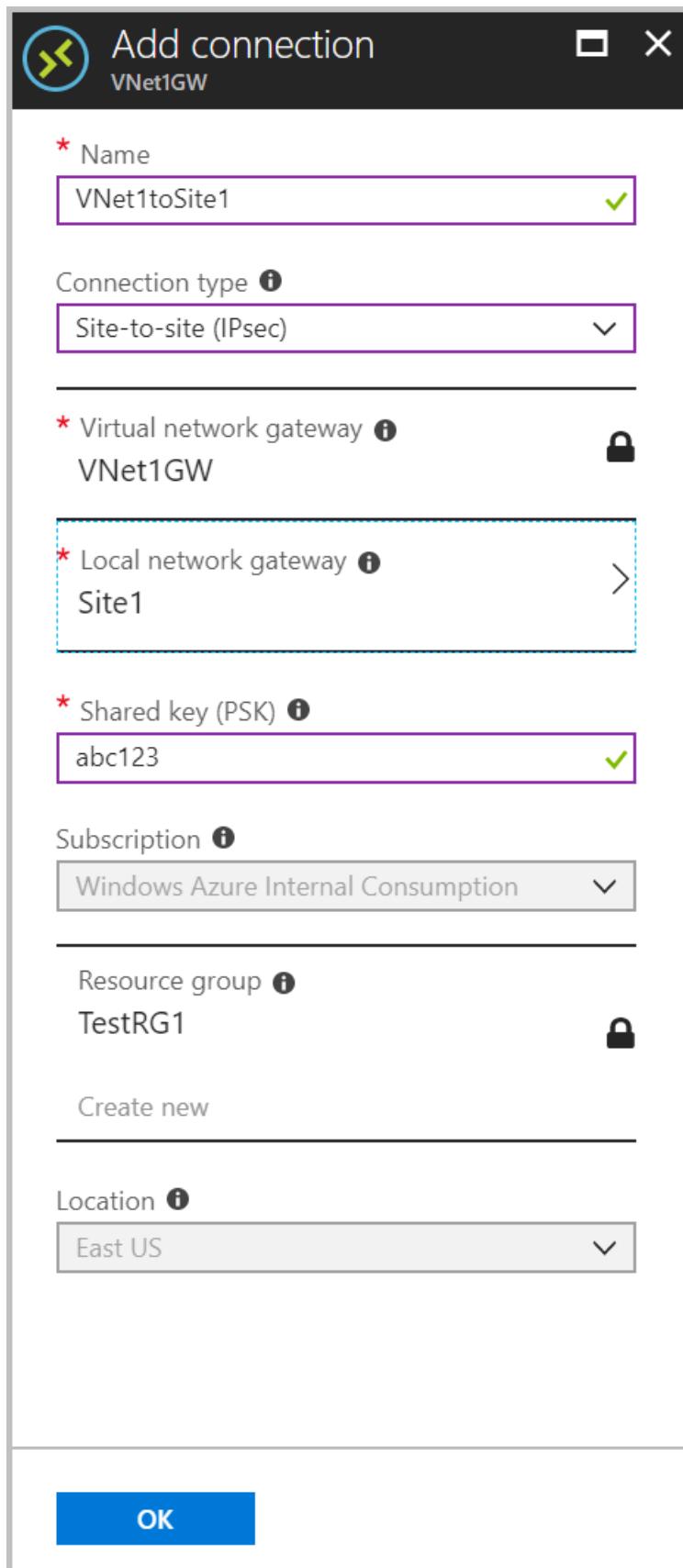
Authentication method, encryption and hashing algorithms, SA lifetime, PFS, and DPD, in addition to other parameter information that you need to complete your configuration.

- 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](#).

7. Create the VPN connection

Create the Site-to-Site VPN connection between your virtual network gateway and your on-premises VPN device.

1. Navigate to and open the page for your virtual network gateway. There are multiple ways to navigate. You can navigate to the gateway 'VNet1GW' by going to **TestVNet1 -> Overview -> Connected devices -> VNet1GW**.
2. On the page for VNet1GW, click **Connections**. At the top of the Connections page, click **+Add** to open the **Add connection** page.



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:** Click **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.

- The remaining values for **Subscription**, **Resource Group**, and **Location** are fixed.
4. Click **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*.

8. 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](#), click **All resources** and navigate to your virtual network gateway.
2. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
3. 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.

Essentials ^	
Resource group	Data in 2.35 KB
Status Connected	Data out 3.14 KB
Location East US	Virtual network
Subscription name	Virtual network gateway
Subscription ID	Local network gateway

To 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 in multiple ways. Below, we show the steps for the Azure portal and for 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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | 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 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.

To troubleshoot an RDP connection to a VM

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](#).

How to reset a VPN 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. For steps, see [Reset a VPN gateway](#).

How to change a gateway SKU (resize a gateway)

For the steps to change a gateway SKU, see [Gateway SKUs](#).

How to add an additional connection to a VPN 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 click **Connections** to open the Connections page.
2. Click **+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 click **OK** to create the connection.

Next steps

- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).
- For information about forced tunneling, see [About forced tunneling](#).

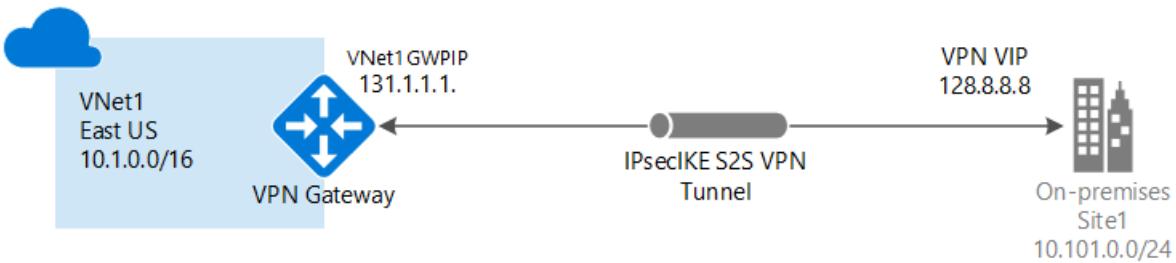
- For information about Highly Available Active-Active connections, see [Highly Available cross-premises and VNet-to-VNet connectivity](#).
- For information about how to limit network traffic to resources in a virtual network, see [Network Security](#).
- For information about how Azure routes traffic between Azure, on-premises, and Internet resources, see [Virtual network traffic routing](#).
- For information about creating a Site-to-Site VPN connection using Azure Resource Manager template, see [Create a Site-to-Site VPN Connection](#).
- For information about creating a Vnet-to-Vnet VPN connection using Azure Resource Manager template, see [Deploy HBase geo replication](#).

Create a VNet with a Site-to-Site VPN connection using PowerShell

4/18/2018 • 17 minutes to read • [Edit Online](#)

This article shows you how to use PowerShell to create a Site-to-Site VPN gateway connection from your on-premises network to the VNet. The steps in this article apply to the Resource Manager deployment model. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it. For more information about VPN gateways, see [About VPN gateway](#).



Before you begin

Verify that you have met the following criteria before beginning your configuration:

- 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. This IP address cannot be located behind a NAT.
- 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.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

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

Azure PowerShell

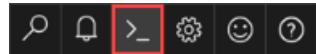
Copy

Try It

Open Cloud Shell in your browser.

Launch Cloud Shell

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



Running PowerShell locally

If you choose to install and use the PowerShell locally, install the latest version of the Azure Resource Manager PowerShell cmdlets. PowerShell cmdlets are updated frequently and you will typically need to update your PowerShell cmdlets to get the latest feature functionality. If you don't update your PowerShell cmdlets, the values specify may fail.

To find the version you are using, run 'Get-Module -ListAvailable AzureRM'. If you need to upgrade, see [Install the Azure PowerShell module](#). For more information, see [How to install and configure Azure PowerShell](#). If you are running PowerShell locally, you also need to run 'Connect-AzureRmAccount' to create a connection with Azure.

Example values

The examples in this article use the following values. You can use these values to create a test environment, or refer to them to better understand the examples in this article.

```
#Example values

$VnetName          = "VNet1"
$ResourceGroup      = "TestRG1"
$Location           = "East US"
$AddressSpace        = "10.1.0.0/16"
$SubnetName          = "Frontend"
$Subnet              = "10.1.0.0/24"
$GatewaySubnet       = "10.1.255.0/27"
$LocalNetworkGatewayName = "Site1"
$LNG Public IP       = <On-premises VPN device IP address>
$Local Address Prefixes = "10.101.0.0/24, 10.101.1.0/24"
$Gateway Name         = "VNet1GW"
$PublicIP             = "VNet1GWPPIP"
$Gateway IP Config    = "gwipconfig1"
$VPNType              = "RouteBased"
$GatewayType           = "Vpn"
$ConnectionName        = "VNet1toSite1"
```

1. Create a virtual network and a gateway subnet

If you don't already have a virtual network, create one. When creating a virtual network, make sure that the address spaces you specify don't overlap any of the address spaces that you have on your on-premises network.

NOTE

In order for this VNet to connect to an on-premises location, you need 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 does not route the way you may expect it to. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Take care to plan your network configuration accordingly.

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. The subnet must be named

'GatewaySubnet' in order for Azure to deploy the gateway resources. You can't specify a different subnet to deploy the gateway resources to. If you don't have a subnet named 'GatewaySubnet', when you create your VPN gateway, it will fail.

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.

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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

To create a virtual network and a gateway subnet

This example creates a virtual network and a gateway subnet. If you already have a virtual network that you need to add a gateway subnet to, see [To add a gateway subnet to a virtual network you have already created](#).

Create a resource group:

```
New-AzureRmResourceGroup -Name TestRG1 -Location 'East US'
```

Create your virtual network.

1. Set the variables.

```
$subnet1 = New-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27  
$subnet2 = New-AzureRmVirtualNetworkSubnetConfig -Name 'Frontend' -AddressPrefix 10.1.0.0/24
```

2. Create the VNet.

```
New-AzureRmVirtualNetwork -Name VNet1 -ResourceGroupName TestRG1 `  
-Location 'East US' -AddressPrefix 10.1.0.0/16 -Subnet $subnet1, $subnet2
```

To add a gateway subnet to a virtual network you have already created

Use the steps in this section if you already have a virtual network, but need to add a gateway subnet.

1. Set the variables.

```
$vnet = Get-AzureRmVirtualNetwork -ResourceGroupName TestRG1 -Name TestVet1
```

2. Create the gateway subnet.

```
Add-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27 -  
VirtualNetwork $vnet
```

3. Set the configuration.

```
Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

2. Create the local network gateway

The local network gateway typically refers to your on-premises location. 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, you can easily update the prefixes.

Use the following values:

- The *GatewayIPAddress* is the IP address of your on-premises VPN device. Your VPN device cannot be located behind a NAT.
- The *AddressPrefix* is your on-premises address space.

To add a local network gateway with a single address prefix:

```
New-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
-Location 'East US' -GatewayIpAddress '23.99.221.164' -AddressPrefix '10.101.0.0/24'
```

To add a local network gateway with multiple address prefixes:

```
New-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
-Location 'East US' -GatewayIpAddress '23.99.221.164' -AddressPrefix @('10.101.0.0/24', '10.101.1.0/24')
```

To modify IP address prefixes for your local network gateway:

Sometimes your local network gateway prefixes change. The steps you take to modify your IP address prefixes depend on whether you have created a VPN gateway connection. See the [Modify IP address prefixes for a local network gateway](#) section of this article.

3. Request a Public IP address

A VPN gateway must have a Public IP address. You first request the IP address resource, and then refer to it when creating your virtual network gateway. The IP address is dynamically assigned to the resource when the VPN gateway is created. VPN Gateway currently only supports *Dynamic* Public IP address allocation. You cannot request a Static Public IP address assignment. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.

Request a Public IP address that will be assigned to your virtual network VPN gateway.

```
$gwpip= New-AzureRmPublicIpAddress -Name VNet1GWPIP -ResourceGroupName TestRG1 -Location 'East US' -  
AllocationMethod Dynamic
```

4. Create the gateway IP addressing configuration

The gateway configuration defines the subnet and the public IP address to use. Use the following example to create your gateway configuration:

```
$vnet = Get-AzureRmVirtualNetwork -Name VNet1 -ResourceGroupName TestRG1
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gwipconfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name gwipconfig1 -SubnetId $subnet.Id -
PublicIpAddressId $gwip.Id
```

5. Create the VPN gateway

Create the virtual network VPN gateway.

Use the following values:

- The `-GatewayType` for a Site-to-Site configuration is *Vpn*. The gateway type is always specific to the configuration that you are implementing. For example, other gateway configurations may require `-GatewayType ExpressRoute`.
- The `-VpnType` can be *RouteBased* (referred to as a Dynamic Gateway in some documentation), or *PolicyBased* (referred to as a Static Gateway in some documentation). For more information about VPN gateway types, see [About VPN Gateway](#).
- Select the Gateway SKU that you want to use. There are configuration limitations for certain SKUs. For more information, see [Gateway SKUs](#). If you get an error when creating the VPN gateway regarding the `-GatewaySku`, verify that you have installed the latest version of the PowerShell cmdlets.

```
New-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1 ` 
-Location 'East US' -IpConfigurations $gwipconfig -GatewayType Vpn ` 
-VpnType RouteBased -GatewaySku VpnGw1
```

After running this command, it can take up to 45 minutes for the gateway configuration to complete.

6. 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:

- 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 virtual network gateway using PowerShell, use the following example:

```
Get-AzureRmPublicIpAddress -Name GW1PublicIP -ResourceGroupName TestRG1
```

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](#).

7. Create the VPN connection

Next, create the Site-to-Site VPN connection between your virtual network gateway and your VPN device. Be sure to replace the values with your own. The shared key must match the value you used for your VPN device configuration. Notice that the '-ConnectionType' for Site-to-Site is */Psec*.

1. Set the variables.

```
$gateway1 = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1
```

2. Create the connection.

```
New-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1 ` 
-Location 'East US' -VirtualNetworkGateway1 $gateway1 -LocalNetworkGateway2 $local ` 
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

After a short while, the connection will be established.

8. Verify the VPN connection

There are a few different ways to verify your VPN connection.

You can verify that your connection succeeded by using the 'Get-AzureRmVirtualNetworkGatewayConnection' 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-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

- 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
```

To 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.

- Locate the private IP address. You can find the private IP address of a VM in multiple ways. Below, we show the steps for the Azure portal and for 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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | 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"
}
```

- Verify that you are connected to your VNet using the VPN connection.
- 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.
- In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

To troubleshoot an RDP connection to a VM

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](#).

To modify IP address prefixes for a local network gateway

If the IP address prefixes that you want routed to your on-premises location change, you can modify the local network gateway. Two sets of instructions are provided. The instructions you choose depend on whether you have already created your gateway connection.

To modify local network gateway IP address prefixes - no gateway connection

To add additional address prefixes:

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24','10.101.1.0/24','10.101.2.0/24')
```

To remove address prefixes:

Leave out the prefixes that you no longer need. In this example, we no longer need prefix 10.101.2.0/24 (from the previous example), so we update the local network gateway, excluding that prefix.

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24','10.101.1.0/24')
```

To modify local network gateway IP address prefixes - existing gateway connection

If you have a gateway connection and want to add or remove the IP address prefixes contained in your local network gateway, you need to do the following steps, in order. This results in some downtime for your VPN connection. When modifying IP address prefixes, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. Modify the address prefixes for your local network gateway.

Set the variable for the LocalNetworkGateway.

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1
```

Modify the prefixes.

```
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24','10.101.1.0/24')
```

3. Create the connection. In this example, we configure an IPsec connection type. When you recreate your connection, use the connection type that is specified for your configuration. For additional connection types, see the [PowerShell cmdlet](#) page.

Set the variable for the VirtualNetworkGateway.

```
$gateway1 = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection. This example uses the variable \$local that you set in step 2.

```
New-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 `  
-ResourceGroupName TestRG1 -Location 'East US' `  
-VirtualNetworkGateway1 $gateway1 -LocalNetworkGateway2 $local `  
-ConnectionType IPsec `  
-RoutingWeight 10 -SharedKey 'abc123'
```

To modify the gateway IP address for a local network gateway

To modify the local network gateway 'GatewayIpAddress' - no gateway connection

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. Use the example to modify a local network gateway that does not have a gateway connection.

When modifying this value, you can also modify the address prefixes at the same time. Be sure to use the existing name of your local network gateway in order to overwrite the current settings. If you use a different name, you create a new local network gateway, instead of overwriting the existing one.

```
New-AzureRmLocalNetworkGateway -Name Site1 `  
-Location "East US" -AddressPrefix @('10.101.0.0/24','10.101.1.0/24') `  
-GatewayIpAddress "5.4.3.2" -ResourceGroupName TestRG1
```

To modify the local network gateway 'GatewayIpAddress' - existing gateway connection

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. If a gateway connection already exists, you first need to remove the connection. After the connection is removed, you can modify the gateway IP address and recreate a new connection. You can also modify the address prefixes at the same time. This results in some downtime for your VPN connection. When modifying the gateway IP address, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection. You can find the name of your connection by using the 'Get-AzureRmVirtualNetworkGatewayConnection' cmdlet.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 `  
-ResourceGroupName TestRG1
```

2. Modify the 'GatewayIpAddress' value. You can also modify the address prefixes at the same time. Be sure to use the existing name of your local network gateway to overwrite the current settings. If you don't, you create a new local network gateway, instead of overwriting the existing one.

```
New-AzureRmLocalNetworkGateway -Name Site1 `  
-Location "East US" -AddressPrefix @('10.101.0.0/24','10.101.1.0/24') `  
-GatewayIpAddress "104.40.81.124" -ResourceGroupName TestRG1
```

3. Create the connection. In this example, we configure an IPsec connection type. When you recreate your connection, use the connection type that is specified for your configuration. For additional connection types, see the [PowerShell cmdlet](#) page. To obtain the VirtualNetworkGateway name, you can run the 'Get-AzureRmVirtualNetworkGateway' cmdlet.

Set the variables.

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
$vnetgw = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection.

```
New-AzureRmVirtualNetworkGatewayConnection -Name VNet1Site1 -ResourceGroupName TestRG1 `  
-Location "East US" `  
-VirtualNetworkGateway1 $vnetgw `  
-LocalNetworkGateway2 $local `  
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

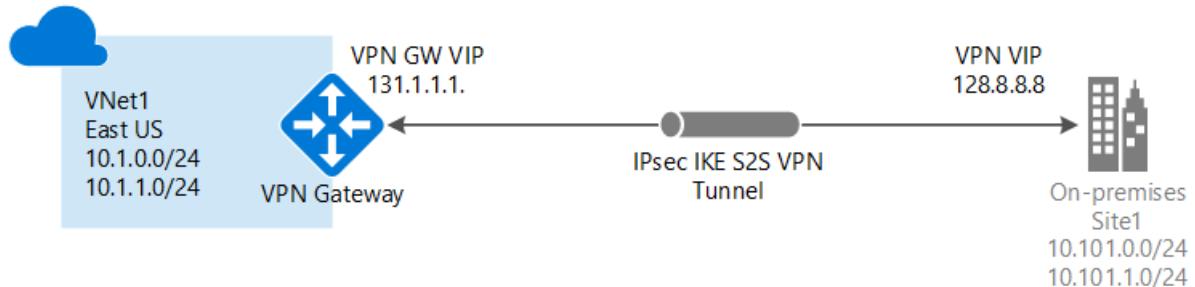
Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).
- For information about creating a site-to-site VPN connection using Azure Resource Manager template, see [Create a Site-to-Site VPN Connection](#).
- For information about creating a vnet-to-vnet VPN connection using Azure Resource Manager template, see [Deploy HBase geo replication](#).

Create a virtual network with a Site-to-Site VPN connection using CLI

3/14/2018 • 15 minutes to read • [Edit Online](#)

This article shows you how to use the Azure CLI to create a Site-to-Site VPN gateway connection from your on-premises network to the VNet. The steps in this article apply to the Resource Manager deployment model. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:



A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it. For more information about VPN gateways, see [About VPN gateway](#).

Before you begin

Verify that you have met the following criteria before beginning configuration:

- 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. This IP address cannot be located behind a NAT.
- 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.
- Verify that you have installed latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install Azure CLI 2.0](#) and [Get Started with Azure CLI 2.0](#).

Example values

You can use the following values to create a test environment, or refer to these values to better understand the examples in this article:

```
#Example values

VnetName          = TestVNet1
ResourceGroup     = TestRG1
Location          = eastus
AddressSpace      = 10.11.0.0/16
SubnetName        = Subnet1
Subnet            = 10.11.0.0/24
GatewaySubnet    = 10.11.255.0/27
LocalNetworkGatewayName = Site2
LNG Public IP     = <VPN device IP address>
LocalAddrPrefix1  = 10.0.0.0/24
LocalAddrPrefix2  = 20.0.0.0/24
GatewayName       = VNet1GW
PublicIP          = VNet1GWIP
VPNTYPE           = RouteBased
GatewayType       = Vpn
ConnectionName    = VNet1toSite2
```

1. Connect to your subscription

Sign in to your Azure subscription with the `az login` command and follow the on-screen directions. For more information about signing in, see [Get Started with Azure CLI 2.0](#).

```
az login
```

If you have more than one Azure subscription, list the subscriptions for the account.

```
az account list --all
```

Specify the subscription that you want to use.

```
az account set --subscription <replace_with_your_subscription_id>
```

2. Create a resource group

The following example creates a resource group named 'TestRG1' in the 'eastus' location. If you already have a resource group in the region that you want to create your VNet, you can use that one instead.

```
az group create --name TestRG1 --location eastus
```

3. Create a virtual network

If you don't already have a virtual network, create one using the `az network vnet create` command. When creating a virtual network, make sure that the address spaces you specify don't overlap any of the address spaces that you have on your on-premises network.

NOTE

In order for this VNet to connect to an on-premises location, you need 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 does not route the way you may expect it to. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Take care to plan your network configuration accordingly.

The following example creates a virtual network named 'TestVNet1' and a subnet, 'Subnet1'.

```
az network vnet create --name TestVNet1 --resource-group TestRG1 --address-prefix 10.11.0.0/16 --location eastus --subnet-name Subnet1 --subnet-prefix 10.11.0.0/24
```

4. Create 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. The subnet must be named 'GatewaySubnet' in order for Azure to deploy the gateway resources. You can't specify a different subnet to deploy the gateway resources to. If you don't have a subnet named 'GatewaySubnet', when you create your VPN gateway, it will fail.

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.

Use the `az network vnet subnet create` command to create the gateway subnet.

```
az network vnet subnet create --address-prefix 10.11.255.0/27 --name GatewaySubnet --resource-group TestRG1 --vnet-name TestVNet1
```

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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

5. Create the local network gateway

The local network gateway typically refers to your on-premises location. 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, you can easily update the prefixes.

Use the following values:

- The `--gateway-ip-address` is the IP address of your on-premises VPN device. Your VPN device cannot be located behind a NAT.
- The `--local-address-prefixes` are your on-premises address spaces.

Use the [az network local-gateway create](#) command to add a local network gateway with multiple address prefixes:

```
az network local-gateway create --gateway-ip-address 23.99.221.164 --name Site2 --resource-group TestRG1 --local-address-prefixes 10.0.0.0/24 20.0.0.0/24
```

6. Request a Public IP address

A VPN gateway must have a Public IP address. You first request the IP address resource, and then refer to it when creating your virtual network gateway. The IP address is dynamically assigned to the resource when the VPN gateway is created. VPN Gateway currently only supports *Dynamic* Public IP address allocation. You cannot request a Static Public IP address assignment. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.

Use the [az network public-ip create](#) command to request a Dynamic Public IP address.

```
az network public-ip create --name VNet1GWIP --resource-group TestRG1 --allocation-method Dynamic
```

7. Create the VPN gateway

Create the virtual network VPN gateway. Creating a VPN gateway can take up to 45 minutes or more to complete.

Use the following values:

- The `--gateway-type` for a Site-to-Site configuration is *Vpn*. The gateway type is always specific to the configuration that you are implementing. For more information, see [Gateway types](#).
- The `--vpn-type` can be *RouteBased* (referred to as a Dynamic Gateway in some documentation), or *PolicyBased* (referred to as a Static Gateway in some documentation). The setting is specific to requirements of the device that you are connecting to. For more information about VPN gateway types, see [About VPN Gateway configuration settings](#).
- Select the Gateway SKU that you want to use. There are configuration limitations for certain SKUs. For more information, see [Gateway SKUs](#).

Create the VPN gateway using the [az network vnet-gateway create](#) command. If you run this command using the `--no-wait` parameter, you don't see any feedback or output. This parameter allows the gateway to create in the background. It takes around 45 minutes to create a gateway.

```
az network vnet-gateway create --name VNet1GW --public-ip-address VNet1GWIP --resource-group TestRG1 --vnet TestVNet1 --gateway-type Vpn --vpn-type RouteBased --sku VpnGw1 --no-wait
```

8. 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:

- 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 virtual network gateway, use the [az network public-ip list](#) command. For easy reading, the output is formatted to display the list of public IPs in table format.

```
az network public-ip list --resource-group TestRG1 --output table
```

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](#).

9. Create the VPN connection

Create the Site-to-Site VPN connection between your virtual network gateway and your on-premises VPN device. Pay particular attention to the shared key value, which must match the configured shared key value for your VPN device.

Create the connection using the [az network vpn-connection create](#) command.

```
az network vpn-connection create --name VNet1toSite2 -resource-group TestRG1 --vnet-gateway1 VNet1GW -l eastus --shared-key abc123 --local-gateway2 Site2
```

After a short while, the connection will be established.

10. Verify the VPN connection

You can verify that your connection succeeded by using the `az network vpn-connection show` command. In the example,'--name'refers to the name of the connection that you want to test. When the connection is in the process of being established, its connection status shows 'Connecting'. Once the connection is established, the status changes to 'Connected'.

```
az network vpn-connection show --name VNet1toSite2 --resource-group TestRG1
```

If you want to use another method to verify your connection, see [Verify a VPN Gateway connection](#).

To 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 in multiple ways. Below, we show the steps for the Azure portal and for 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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | 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 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.

To troubleshoot an RDP connection to a VM

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](#).

Common tasks

This section contains common commands that are helpful when working with site-to-site configurations. For the full list of CLI networking commands, see [Azure CLI - Networking](#).

To view local network gateways

To view a list of the local network gateways, use the [az network local-gateway list](#) command.

```
az network local-gateway list --resource-group TestRG1
```

To modify local network gateway IP address prefixes - no gateway connection

If you don't have a gateway connection and you want to add or remove IP address prefixes, you use the same command that you use to create the local network gateway, [az network local-gateway create](#). You can also use this command to update the gateway IP address for the VPN device. To overwrite the current settings, use the existing name of your local network gateway. If you use a different name, you create a new local network gateway, instead of overwriting the existing one.

Each time you make a change, the entire list of prefixes must be specified, not just the prefixes that you want to change. Specify only the prefixes that you want to keep. In this case, 10.0.0.0/24 and 20.0.0.0/24

```
az network local-gateway create --gateway-ip-address 23.99.221.164 --name Site2 -g TestRG1 --local-address-prefixes 10.0.0.0/24 20.0.0.0/24
```

To modify local network gateway IP address prefixes - existing gateway connection

If you have a gateway connection and want to add or remove IP address prefixes, you can update the prefixes using [az network local-gateway update](#). This results in some downtime for your VPN connection. When modifying the IP address prefixes, you don't need to delete the VPN gateway.

Each time you make a change, the entire list of prefixes must be specified, not just the prefixes that you want to change. In this example, 10.0.0.0/24 and 20.0.0.0/24 are already present. We add the prefixes 30.0.0.0/24 and 40.0.0.0/24 and specify all 4 of the prefixes when updating.

```
az network local-gateway update --local-address-prefixes 10.0.0.0/24 20.0.0.0/24 30.0.0.0/24 40.0.0.0/24 --name VNet1toSite2 -g TestRG1
```

To modify the local network gateway 'gatewayIpAddress'

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. The gateway IP address can be changed without removing an existing VPN gateway connection (if you have one). To modify the gateway IP address, replace the values 'Site2' and 'TestRG1' with your own using the [az network local-gateway update](#) command.

```
az network local-gateway update --gateway-ip-address 23.99.222.170 --name Site2 --resource-group TestRG1
```

Verify that the IP address is correct in the output:

```
"gatewayIpAddress": "23.99.222.170",
```

To verify the shared key values

Verify that the shared key value is the same value that you used for your VPN device configuration. If it is not, either run the connection again using the value from the device, or update the device with the value from the return. The values must match. To view the shared key, use the [az network vpn-connection-list](#).

```
az network vpn-connection shared-key show --connection-name VNet1toSite2 --resource-group TestRG1
```

To view the VPN gateway Public IP address

To find the public IP address of your virtual network gateway, use the [az network public-ip list](#) command. For easy reading, the output for this example is formatted to display the list of public IPs in table format.

```
az network public-ip list --resource-group TestRG1 --output table
```

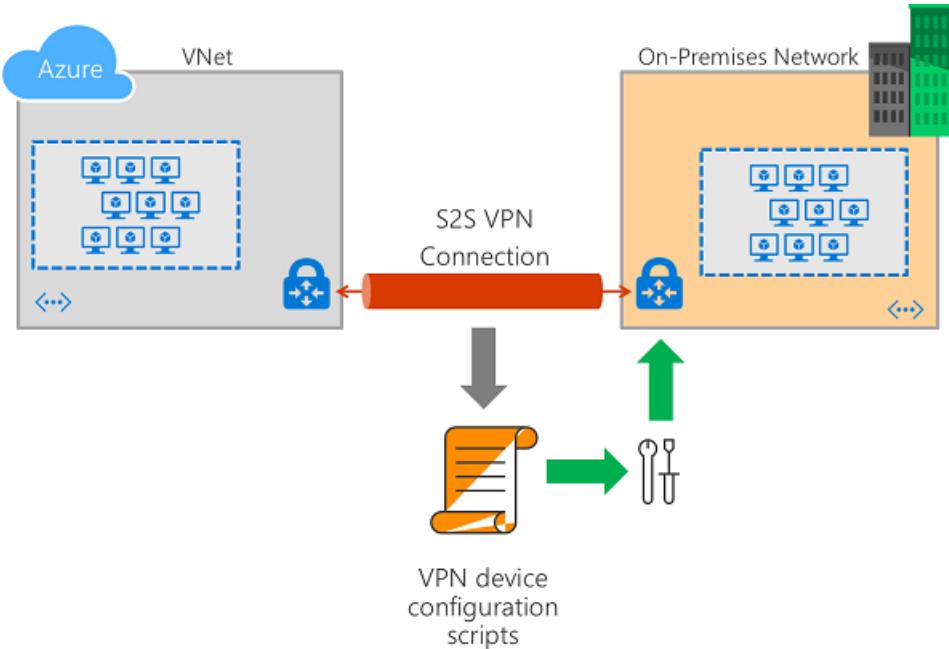
Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).
- 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 a list of networking Azure CLI commands, see [Azure CLI](#).
- For information about creating a site-to-site VPN connection using Azure Resource Manager template, see [Create a Site-to-Site VPN Connection](#).
- For information about creating a vnet-to-vnet VPN connection using Azure Resource Manager template, see [Deploy HBase geo replication](#).

Download VPN device configuration scripts for S2S VPN connections

4/9/2018 • 3 minutes to read • [Edit Online](#)

This article walks you through downloading VPN device configuration scripts for S2S VPN connections with Azure VPN Gateways using Azure Resource Manager. The following diagram shows the high-level workflow.



The following devices have available scripts:

VENDOR	DEVICE FAMILY	FIRMWARE VERSION
Cisco	ISR	IOS 15.1 (Preview)
Cisco	ASA	ASA (*) RouteBased (IKEv2- No BGP) for ASA below 9.8
Cisco	ASA	ASA RouteBased (IKEv2 - No BGP) for ASA 9.8+
Juniper	SRX_GA	12.x
Juniper	SSG_GA	ScreenOS 6.2.x
Juniper	JSeries_GA	JunOS 12.x
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased VTI
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased BGP

NOTE

(*) Required: NarrowAzureTrafficSelectors and CustomAzurePolicies (IKE/IPsec)

About VPN device configuration scripts

A cross-premises VPN connection consists of an Azure VPN gateway, an on-premises VPN device, and an IPsec S2S VPN tunnel connecting the two. The typical work flow includes the following steps:

1. Create and configure an Azure VPN gateway (virtual network gateway)
2. Create and configure an Azure local network gateway that represents your on-premises network and VPN device
3. Create and configure an Azure VPN connection between the Azure VPN gateway and the local network gateway
4. Configure the on-premises VPN device represented by the local network gateway to establish the actual S2S VPN tunnel with the Azure VPN gateway

You can complete steps 1 through 3 using the Azure [portal](#), [PowerShell](#), or [CLI](#). The last step involves configuring the on-premises VPN devices outside of Azure. This feature allows you to download a configuration script for your VPN device with the corresponding values of your Azure VPN gateway, virtual network, and on-premises network address prefixes, and VPN connection properties, etc. already filled in. You can use the script as a starting point, or apply the script directly to your on-premises VPN devices via the configuration console.

IMPORTANT

- The syntax for each VPN device configuration script is different, and heavily dependent on the models and firmware versions. Pay special attention to your device model and version information against the available templates.
- Some parameter values must be unique on the device, and cannot be determined without accessing the device. The Azure-generated configuration scripts pre-fill these values, but you need to ensure the provided values are valid on your device. For examples:
 - Interface numbers
 - Access control list numbers
 - Policy names or numbers, etc.
- Look for the keyword, "**REPLACE**", embedded in the script to find the parameters you need to verify before applying the script.
- Some templates include a "**CLEANUP**" section you can apply to remove the configurations. The cleanup sections are commented out by default.

Download the configuration script from Azure portal

Create an Azure VPN gateway, local network gateway, and a connection resource connecting the two. The following page guides you through the steps:

- [Create a Site-to-Site connection in the Azure portal](#)

Once the connection resource is created, follow the instructions below to download the VPN device configuration scripts:

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account
2. Go to the connection resource you created. You can find the list of all connection resources by clicking "All services", then "NETWORKING", and "Connections."

The screenshot shows the Microsoft Azure Connections page. The left sidebar includes links for 'Create a resource', 'All services', 'FAVORITES' (Dashboard, Resource groups, All resources, Virtual machines, Virtual network gateways, Virtual networks), and 'Connections'. The main area displays a table titled 'Subscriptions: 2 of 44 selected' with columns: NAME, STATUS, PEER 1, PEER 2, RESOURCE, LOCATION, and SUBSCRIPTION. One item is listed: 'VNet1toSite5' with STATUS 'Connecting', PEER 1 'VNet1GW', PEER 2 'Site5', RESOURCE 'TestRG1', LOCATION 'East US 2', and SUBSCRIPTION '...'. Filter options at the top include 'Filter by name...', '2 subscriptions', 'All resource groups', 'All locations', and 'No grouping'.

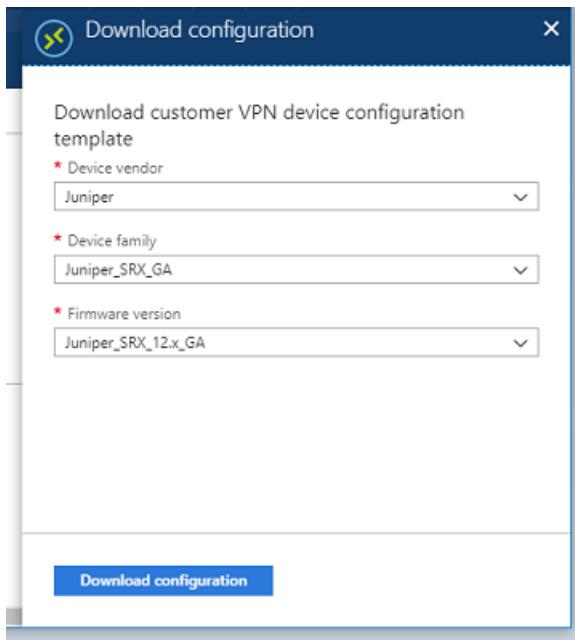
3. Click on the connection you want to configure.

The screenshot shows the 'VNet1toSite5 Connection' overview page. The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, SETTINGS (Shared key, Configuration, Properties), and a search bar. The main area shows connection details: Resource group (TestRG1), Location (East US 2), Subscription (TestVNet1, TestVNet1), Subscription ID, Data in (0 B), Data out (0 B), Virtual network (VNet1GW (40.70.3.155)), and Local network gateway (Site5 (131.107.5.8)). A red box highlights the 'Download configuration' link in the top right.

4. Click on the "Download configuration" link as highlighted in red in the Connection overview page; this opens the "Download configuration" page.

The screenshot shows the 'Download configuration' page. The left sidebar has 'Move', 'Download configuration' (which is highlighted with a red box), and 'Delete' buttons. The main area is titled 'Download customer VPN device configuration template' and contains three dropdown fields: 'Device vendor' (selected), 'Device family' (empty), and 'Firmware version' (empty). At the bottom is a 'Download configuration' button.

5. Select the model family and firmware version for your VPN device, then click on the "Download configuration" button.



6. You are prompted to save the downloaded script (a text file) from your browser.
7. Once you downloaded the configuration script, open it with a text editor and search for the keyword "REPLACE" to identify and examine the parameters that may need to be replaced.

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
File Edit View Insert Tools Plugins Window Help
VNettoSite5.txt
238 set security ipsec proposal azure-ipsec2-proposal-VNettoSite5-40.70.3.155 lifetime-seconds 3600
239 ! DEFINING THE IPSEC (PHASE 2) POLICY FOR AZURE
240 set security ipsec policy azure-YOUR-policy-VNettoSite5-40.70.3.155 proposals
241
242 ! DEFINING THE IPSEC (PHASE 2) VPN FOR AZURE: Binding to the secure tunnel is
243 set security ipsec YOUR azure-ipsec2-YOUR-VNettoSite5-40.70.3.155 bind-interface
244
245 ! Note: Please REPLACE the destination-ip below by the IP address of an active
246 ! (e) for the VPM Monitor functionality (Phase 2 liveliness) of your SRX
247
248 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 VPN-monitors :
249 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 ike gateway
250 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 ike proxy-ids
251 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 ike proxy-ids
252 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 ike proxy-ids
253 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 ike ipsec-poi
254 set security ipsec YEG azure-ipsec2-YEG-VNettoSite5-40.70.3.155 establish-tun
255
256 ! SETTING THE SECURITY ZONES FOR AZURE
257 ! =====
258 ! NOTE: The zones are defined as:
259 ! 1) "Internal" from the ingress (internal facing) on-premises network behavior shown above.
260 ! "External" from the egress (external facing) Azure network, which the external interface is connected to.
261 ! REPLACE these as needed.
262
263 ! 2) The on-premises network is tied to a virtual interface on the SRX labeled "vlan.1".
264 ! The external (public facing) interface and port on the SRX for Azure is labeled "fx-0/0/0.0".
265 ! REPLACE these as needed.
266
267
268
269
270
271
272
273
274
275
276
277 ! The physical LAN interface (fx-0/0/1.0) for the on-premises network. REPLACE the INT as needed.
278 set security zones security-zone Internal address-book address onprex-networks-VNettoSite5-40.70.3.155 10.51.0.0/16
279 set security zones security-zone Internal address-book address onprex-networks-VNettoSite5-40.70.3.155 10.52.0.0/16
280
281 ! The VLAN (vlan.1) representing the on-premises network. REPLACE the INT as needed.
282 set security zones security-zone Internal host-inbound-traffic system-services all
283 set security zones security-zone Internal interfaces vlan.1
284 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services ping
285 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services DHCP
286 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services HTTP
287 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services HTTPS
288 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services ssh
289 set security zones security-zone Internal interfaces vlan.1 host-inbound-traffic system-services telnet
290
291 ! INTERNET ZONE DEFINITION
292
293
294
295
296
297
298
299
300
301
302
303

```

Download the configuration script using Azure PowerShell

You can also download the configuration script using Azure PowerShell, as shown in the following example:

```
$RG          = "TestRG1"
$GWName     = "VNet1GW"
$Connection  = "VNet1toSite1"

# List the available VPN device models and versions
Get-AzureRmVirtualNetworkGatewaySupportedVpnDevice -Name $GWName -ResourceGroupName $RG

# Download the configuration script for the connection
Get-AzureRmVirtualNetworkGatewayConnectionVpnDeviceConfigScript -Name $Connection -ResourceGroupName $RG -
DeviceVendor Juniper -DeviceFamily Juniper_SRX_GA -FirmwareVersion Juniper_SRX_12.x_GA
```

Apply the configuration script to your VPN device

After you have downloaded and validated the configuration script, the next step is to apply the script to your VPN device. The actual procedure varies based on your VPN device makes and models. Consult the operation manuals or the instruction pages for your VPN devices.

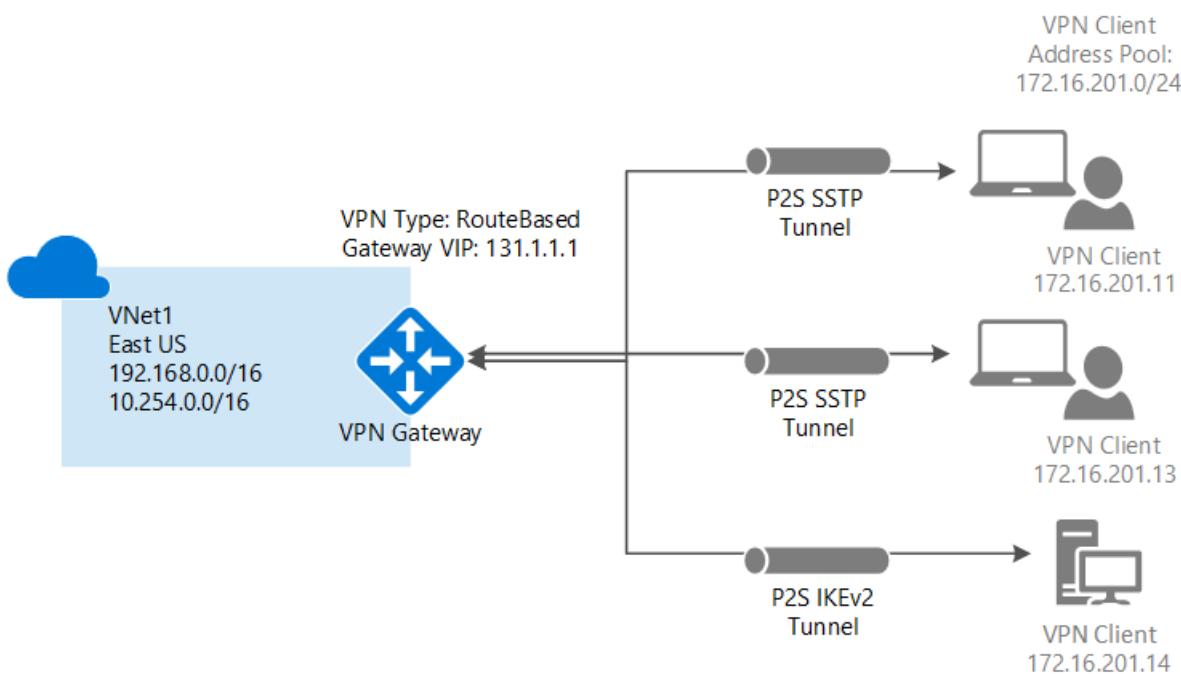
Next steps

Continue configuring your [Site-to-Site connection](#).

Configure a Point-to-Site connection to a VNet using native Azure certificate authentication: Azure portal

9/10/2018 • 28 minutes to read • [Edit Online](#)

This article helps you securely connect individual clients running Windows, Linux, or Mac OS X to an Azure VNet. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location, such as when you are telecommuting from home or a conference. You can also use P2S instead of a Site-to-Site VPN when you have only a few clients that need to connect to a VNet. Point-to-Site connections do not require a VPN device or a public-facing IP address. P2S creates the VPN connection over either SSTP (Secure Socket Tunneling Protocol), or IKEv2. For more information about Point-to-Site VPN, see [About Point-to-Site VPN](#).



Architecture

Point-to-Site native Azure certificate authentication connections use the following items, which you configure in this exercise:

- A RouteBased VPN gateway.
- The public key (.cer file) for a root certificate, which is uploaded to Azure. Once the certificate is uploaded, it is considered a trusted certificate and is used for authentication.
- A client certificate that is generated from the root certificate. The client certificate installed on each client computer that will connect to the VNet. This certificate is used for client authentication.
- A VPN client configuration. The VPN client configuration files contain the necessary information for the client to connect to the VNet. The files configure the existing VPN client that is native to the operating system. Each client that connects must be configured using the settings in the configuration files.

Example values

You can use the following values to create a test environment, or refer to these values to better understand the examples in this article:

- **VNet Name:** VNet1
- **Address space:** 192.168.0.0/16

For this example, we use only one address space. You can have more than one address space for your VNet.

- **Subnet name:** FrontEnd
- **Subnet address range:** 192.168.1.0/24
- **Subscription:** If you have more than one subscription, verify that you are using the correct one.
- **Resource Group:** TestRG
- **Location:** East US
- **GatewaySubnet:** 192.168.200.0/24
- **DNS Server:** (optional) IP address of the DNS server that you want to use for name resolution.
- **Virtual network gateway name:** VNet1GW
- **Gateway type:** VPN
- **VPN type:** Route-based
- **Public IP address name:** VNet1GWpip
- **Connection type:** Point-to-site
- **Client address pool:** 172.16.201.0/24

VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the client address pool.

1. Create a virtual network

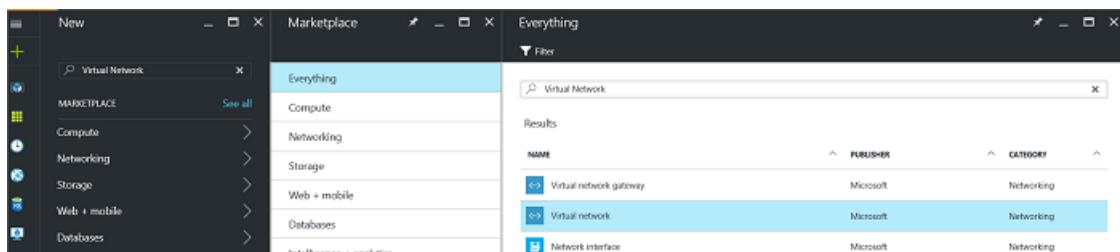
Before beginning, verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).

To create a VNet in the Resource Manager deployment model by using the Azure portal, follow the steps below. The screenshots are provided as examples. Be sure to replace the values with your own. For more information about working with virtual networks, see the [Virtual Network Overview](#).

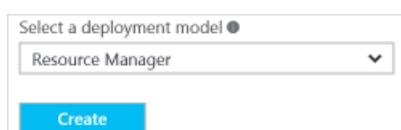
NOTE

If you want this VNet to connect to an on-premises location (in addition to creating a P2S configuration), you need 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 does not route the way you may expect it to. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Take care to plan your network configuration accordingly.

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **+**. In the **Search the marketplace** field, type "Virtual Network". Locate **Virtual Network** from the returned list and click to open the **Virtual Network** page.



3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, select **Resource Manager**, and then click **Create**.



- On the **Create virtual network** page, configure the VNet settings. When you fill in the fields, the red exclamation mark becomes a green check mark when the characters entered in the field are valid. There may be values that are auto-filled. If so, replace the values with your own. The **Create virtual network** page looks similar to the following example:

The screenshot shows the 'Create virtual network' dialog box. It includes the following fields:

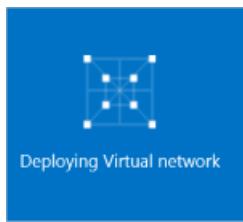
- Name:** VNet1 (marked with a green checkmark)
- Address space:** 192.168.0.0/16 (marked with a green checkmark)
192.168.0.0 - 192.168.255.255 (65536 addresses)
- Subscription:** Windows Azure Internal Consumption
- Resource group:** TestRG (radio button selected for 'Create new')
- Location:** East US
- Subnet:**
 - Name:** FrontEnd (marked with a green checkmark)
 - Address range:** 192.168.1.0/24 (marked with a green checkmark)
192.168.1.0 - 192.168.1.255 (256 addresses)

- Name:** Enter the name for your Virtual Network.
- Address space:** Enter the address space. If you have multiple address spaces to add, add your first address space. You can add additional address spaces later, after creating the VNet.
- 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 create a new one by typing a name for your new resource group. If you are creating a new group, name the resource group according to your planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](#).
- Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
- Subnet:** Add the subnet name and subnet address range. You can add additional subnets later, after creating the VNet.
- Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.

The screenshot shows the 'Create' button section, which includes:

- A checkbox labeled 'Pin to dashboard' with a checked checkedmark icon.
- A large blue rectangular button labeled 'Create'.

- After clicking **Create**, you will see a tile on your dashboard that will reflect the progress of your VNet. The tile changes as the VNet is being created.



2. Add a gateway subnet

Before connecting your virtual network to a gateway, you first need to create the gateway subnet for the virtual network to which you want to connect. The gateway services use the IP addresses specified in the gateway subnet. If possible, create a gateway subnet using a CIDR block of /28 or /27 to provide enough IP addresses to accommodate additional future configuration requirements.

1. In the [portal](#), navigate to the Resource Manager virtual network for which you want to create a virtual network gateway.
2. In the **Settings** section of your VNet page, click **Subnets** to expand the **Subnets** page.
3. On the **Subnets** page, click **+Gateway subnet** to open the **Add subnet** page.

A screenshot of the Azure portal's Subnets page. At the top, there are two buttons: "+ Subnet" and "+ Gateway subnet", with "+ Gateway subnet" highlighted by a red box. Below the buttons is a search bar labeled "Search subnets". Underneath is a table with columns: NAME, ADDRESS RANGE, and AVAILABLE ADDRESSES. The table has a header row and no data rows.

4. The **Name** for your subnet is automatically filled in with the value 'GatewaySubnet'. This value is required in order for Azure to recognize the subnet as the gateway subnet. Adjust the auto-filled **Address range** values to match your configuration requirements, then click **OK** at the bottom of the page to create the subnet.

A screenshot of the "Add subnet" dialog box. It shows a "VNet1" header. The "Name" field is set to "GatewaySubnet". The "Address range (CIDR block)" field is set to "192.168.200.0/24", with a note below it stating "192.168.200.0 - 192.168.200.255 (256 addresses)". The "Route table" field is set to "None". At the bottom right is a "OK" button.

3. Specify a DNS server (optional)

After you create your virtual network, you can add the IP address of a DNS server to handle name resolution. The DNS server is optional for this configuration, but required if you want name resolution. Specifying a value does not create a new DNS server. The DNS server IP address that you specify should be a DNS server that can resolve the names for the resources you are connecting to. For this example, we used a private IP address, but it is likely that this is not the IP address of your DNS server. Be sure to use your own values. The value you specify is used by the resources that you deploy to the VNet, not by the P2S connection or the VPN client.

1. On the **Settings** page for your virtual network, navigate to **DNS Servers** and click to open the **DNS servers** page.

Save **X Discard**

Virtual machines within this virtual network must be restarted to utilize the updated DNS server settings.

DNS servers i

Default (Azure-provided)

Custom

Add DNS server ...

- **DNS Servers:** Select **Custom**.
 - **Add DNS server:** Enter the IP address of the DNS server that you want to use for name resolution.
2. When you are done adding DNS servers, click **Save** at the top of the page.

4. Create a virtual network gateway

1. In the portal, on the left side, click **+ Create a resource** and type 'Virtual Network Gateway' in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create** at the bottom of the page to open the **Create virtual network gateway** page.
2. On the **Create virtual network gateway** page, fill in the values for your virtual network gateway.

Create virtual network gateway...

* Name
VNet1GW

Gateway type i
 VPN ExpressRoute

VPN type i
 Route-based Policy-based

* SKU i
VpnGw1

Enable active-active mode i

* Virtual network i
VNet1

* First IP configuration i
(new) VNet1GWpip

Configure BGP ASN

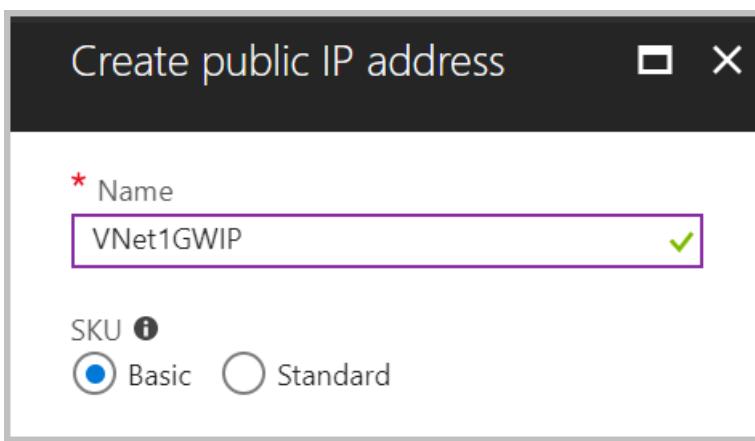
* Subscription
Windows Azure Internal Consumption

Resource group i
TestRG

* Location i
East US

3. On the **Create virtual network gateway** page, specify the values for your virtual network gateway.

- **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **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](#).
- **Location:** You may need to scroll to see Location. Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, when you select a virtual network in the next step, it will not appear in the drop-down list.
- **Virtual network:** Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the 'Choose a virtual network' page. Select the VNet. If you don't see your VNet, make sure the Location field is pointing to the region in which your virtual network is located.
- **Gateway subnet address range:** You will only see this setting if you did not previously create a gateway subnet for your virtual network. If you previously created a valid gateway subnet, this setting will not appear.
- **First IP configuration:** The 'Choose public IP address' page creates a 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. VPN Gateway currently only supports *Dynamic* Public IP address allocation. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.
 - First, click **Create gateway IP configuration** to open the 'Choose public IP address' page, then click **+Create new** to open the 'Create public IP address' page.
 - Next, input a **Name** for your public IP address. Leave the SKU as **Basic** unless there is a specific reason to change it to something else, then click **OK** at the bottom of this page to save your changes.



4. Verify the settings. You can select **Pin to dashboard** at the bottom of the page if you want your gateway to appear on the dashboard.
5. Click **Create** to begin creating the VPN gateway. The settings are validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.

After the gateway is created, 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. You can click the connected device (your virtual network gateway) to view more information.

NOTE

The Basic SKU does not support IKEv2 or RADIUS authentication.

5. Generate certificates

Certificates are used by Azure to authenticate clients connecting to a VNet over a Point-to-Site VPN connection. Once you obtain a root certificate, you [upload](#) the public key information to Azure. The root certificate is then considered 'trusted' by Azure for connection over P2S to the virtual network. You also generate client certificates from the trusted root certificate, and then install them on each client computer. The client certificate is used to authenticate the client when it initiates a connection to the VNet.

1. Obtain the .cer file for the root certificate

You can use either a root certificate that was generated using an enterprise solution (recommended), or you can generate a self-signed certificate. After creating the root certificate, export the public certificate data (not the private key) as a Base-64 encoded X.509 .cer file and upload the public certificate data to Azure.

- **Enterprise certificate:** If you are using an enterprise solution, you can use your existing certificate chain. Obtain the .cer file for the root certificate that you want to use.
- **Self-signed root certificate:** If you aren't using an enterprise certificate solution, you need to create a self-signed root certificate. It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the certificates you create won't be compatible with P2S connections and clients receive a connection error when trying to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the provided articles generate a compatible certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [Linux instructions](#)

2. Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. The client certificate is generated from the root certificate and installed on each client computer. If a valid client certificate is not installed and the client tries to connect to the VNet, authentication fails.

You can either generate a unique certificate for each client, or you can use the same certificate for multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate. Otherwise, if multiple clients are using the same client certificate and you need to revoke it, you have to generate and install new certificates for all the clients that use that certificate to authenticate.

You can generate client certificates using the following methods:

- **Enterprise certificate:**
 - If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
 - Make sure the client certificate is based on the 'User' certificate template that has 'Client Authentication' as the first item in the use list, rather than Smart Card Logon, etc. You can check the certificate by double-clicking the client certificate and viewing **Details > Enhanced Key Usage**.
- **Self-signed root certificate:** It's important that you follow the steps in one of the P2S certificate articles

below. Otherwise, the client certificates you create won't be compatible with P2S connections and clients receive an error when trying to connect. The steps in either of the following articles generate a compatible client certificate:

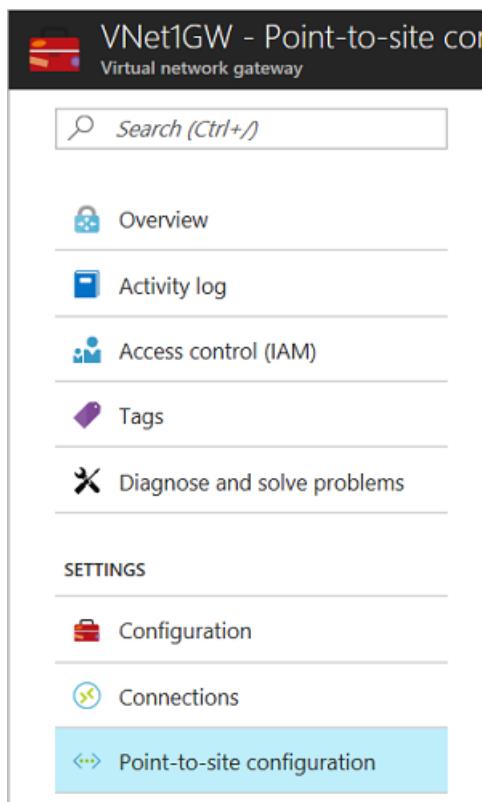
- [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. The certificates that are generated can be installed on any supported P2S client.
- [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. The certificates that are generated can be installed on any supported P2S client.
- [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate using the preceding instructions, it's automatically installed on the computer that you used to generate it. If you want to install a client certificate on another client computer, you need to export it as a .pfx, along with the entire certificate chain. This creates a .pfx file that contains the root certificate information that is required for the client to successfully authenticate. For steps to export a certificate, see [Certificates - export a client certificate](#).

6. Add the client address pool

The client address pool is a range of private IP addresses that you specify. The clients that connect over a Point-to-Site VPN dynamically receive an IP address from this range. Use a private IP address range that does not overlap with the on-premises location that you connect from, or the VNet that you want to connect to.

1. Once the virtual network gateway has been created, navigate to the **Settings** section of the virtual network gateway page. In the **Settings** section, click **Point-to-site configuration**.



2. Click **Configure now** to open the configuration page.



3. On the **Point-to-site** configuration page, in the **Address pool** box, add the private IP address range that you want to use. VPN clients dynamically receive an IP address from the range that you specify. Click

Save to validate and save the setting.

Address pool
172.16.201.0/24 ✓

Tunnel type
SSL VPN (SSTP)
IKEv2 VPN

Authentication type
 Azure certificate RADIUS authentication

NOTE

If you don't see Tunnel type or Authentication type in the portal on this page, your gateway is using the Basic SKU. The Basic SKU does not support IKEv2 or RADIUS authentication.

7. Configure tunnel type

You can select the tunnel type. The two tunnel options are SSTP and IKEv2. The strongSwan client on Android and Linux and the native IKEv2 VPN client on iOS and OSX will use only IKEv2 tunnel to connect. Windows clients try IKEv2 first and if that doesn't connect, they fall back to SSTP. You can choose to enable one of them or both. Select the checkboxes that your solution requires.

Address pool
172.16.201.0/24 ✓

Tunnel type
SSL VPN (SSTP)
IKEv2 VPN

Authentication type
 Azure certificate RADIUS authentication

8. Configure authentication type

Select **Azure certificate**.

Address pool
172.16.201.0/24

Tunnel type
SSL VPN (SSTP)
IKEv2 VPN

Authentication type
 Azure certificate RADIUS authentication

9. Upload the root certificate public certificate data

You can upload additional trusted root certificates up to a total of 20. Once the public certificate data is uploaded, Azure can use it to authenticate clients that have installed a client certificate generated from the trusted root certificate. Upload the public key information for the root certificate to Azure.

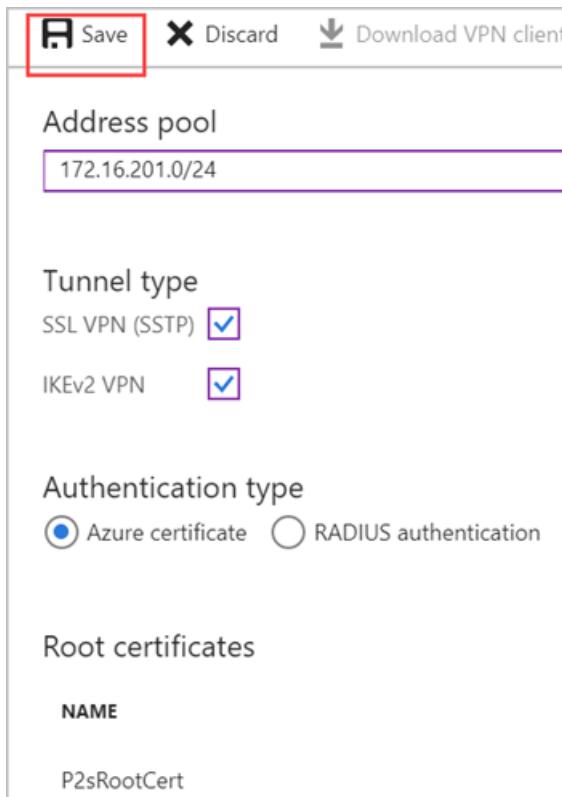
1. Certificates are added on the **Point-to-site configuration** page in the **Root certificate** section.
2. Make sure that you exported the root certificate as a Base-64 encoded X.509 (.cer) file. You need to export the certificate in this format so you can open the certificate with text editor.
3. Open the certificate with a text editor, such as Notepad. When copying the certificate data, make sure that you copy the text as one continuous line without carriage returns or line feeds. You may need to modify your view in the text editor to 'Show Symbol/Show all characters' to see the carriage returns and line feeds. Copy only the following section as one continuous line:

```
P2SRootCert.cer - Notepad
File Edit Format View Help
-----BEGIN CERTIFICATE-----
MIIC6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY
MRYwFAYDVQQDDA1QMINSb290Q2VydDEwM84XDTE3MDgwNzIxNTg0N1oXDTE4MDgw
NzIxMTg0N1owGDEWMBQGA1UEAwNUJDUMvdeENlcnQxMDCCAS1wDQYJKoZIhvcN
AQEBBQAQDggEPADCCAQcggEB8ANw4PjxpJKPnYhbToxn4+YE178CP8HzIsZqvzqwv
Uvgov0hQ2lQnxweUI27arHaZf9fjaJ9ACOUgT/XKC2gnq3mDej42CdDpZG7Hgpfe
wZZzAUdAeUh1D9ngnxpsVCuCrRiuHYoT9kyh9zwRYDHQa1z/tatJb3fP7cxPJ1
KSpvdvm5esZpwypPphVN83KAHuGK4eV рука
File Edit Format View Help
-----BEGIN CERTIFICATE-----
MIIC6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY
MRYwFAYDVQQDDA1QMINSb290Q2VydDEwM84XDTE3MDgwNzIxNTg0N1oXDTE4MDgw
NzIxMTg0N1owGDEWMBQGA1UEAwNUJDUMvdeENlcnQxMDCCAS1wDQYJKoZIhvcN
AQEBBQAQDggEPADCCAQcggEB8ANw4PjxpJKPnYhbToxn4+YE178CP8HzIsZqvzqwv
Uvgov0hQ2lQnxweUI27arHaZf9fjaJ9ACOUgT/XKC2gnq3mDej42CdDpZG7Hgpfe
wZZzAUdAeUh1D9ngnxpsVCuCrRiuHYoT9kyh9zwRYDHQa1z/tatJb3fP7cxPJ1
KSpvdvm5esZpwypPphVN83KAHuGK4eV рука
-----END CERTIFICATE-----
```

4. Paste the certificate data into the **Public Certificate Data** field. **Name** the certificate, and then click **Save**. You can add up to 20 trusted root certificates.

Root certificates				
<table border="1"> <thead> <tr> <th>NAME</th> <th>PUBLIC CERTIFICATE DATA</th> </tr> </thead> <tbody> <tr> <td>P2SRootCert</td> <td>MIIIC6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY MRYwFAYDVQQDDA1QMINSb290Q2V...</td> </tr> </tbody> </table>	NAME	PUBLIC CERTIFICATE DATA	P2SRootCert	MIIIC6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY MRYwFAYDVQQDDA1QMINSb290Q2V...
NAME	PUBLIC CERTIFICATE DATA			
P2SRootCert	MIIIC6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY MRYwFAYDVQQDDA1QMINSb290Q2V...			

5. Click **Save** at the top of the page to save all of the configuration settings.



10. Install an exported client certificate

If you want to create a P2S connection from a client computer other than the one you used to generate the client certificates, you need to install a client certificate. When installing a client certificate, you need the password that was created when the client certificate was exported.

Make sure the client certificate was exported as a .pfx along with the entire certificate chain (which is the default). Otherwise, the root certificate information isn't present on the client computer and the client won't be able to authenticate properly.

For install steps, see [Install a client certificate](#).

11. Generate and install the VPN client configuration package

The VPN client configuration files contain settings to configure devices to connect to a VNet over a P2S connection. For instructions to generate and install VPN client configuration files, see [Create and install VPN client configuration files for native Azure certificate authentication P2S configurations](#).

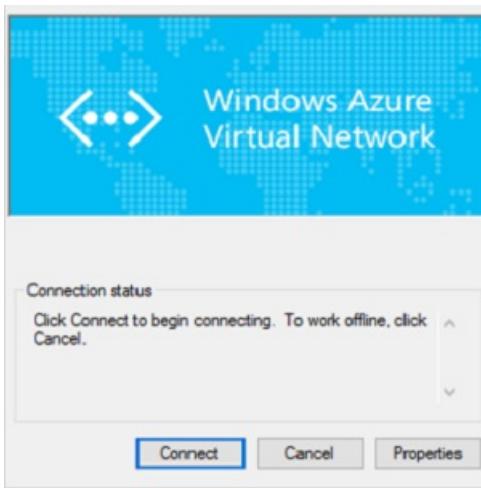
12. Connect to Azure

To connect from a Windows VPN client

NOTE

You must have Administrator rights on the Windows client computer from which you are connecting.

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. Click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection is established.



Troubleshoot Windows P2S connections

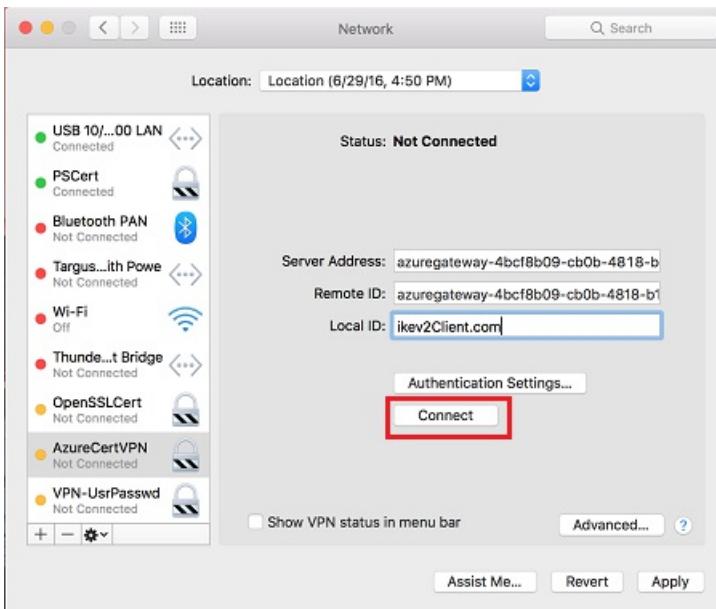
If you are having trouble connecting, check the following items:

- If you exported a client certificate, make sure that you exported it as a .pfx file using the default value 'Include all certificates in the certification path if possible'. When you export it using this value, the root certificate information is also exported. When the certificate is installed on the client computer, the root certificate which is contained in the .pfx file is then also installed on the client computer. The client computer must have the root certificate information installed. To check, go to **Manage user certificates** and navigate to **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed. The root certificate must be present in order for authentication to work.
- If you are using a certificate that was issued using an Enterprise CA solution and are having trouble authenticating, check the authentication order on the client certificate. You can check the authentication list order by double-clicking the client certificate, and going to **Details > Enhanced Key Usage**. Make sure the list shows 'Client Authentication' as the first item. If not, you need to issue a client certificate based on the User template that has Client Authentication as the first item in the list.
- For additional P2S troubleshooting information, see [Troubleshoot P2S connections](#).

To connect from a Mac VPN client

From the Network dialog box, locate the client profile that you want to use, specify the settings from the `VpnSettings.xml`, and then click **Connect**.

Please check [Install - Mac \(OS X\)](#) for detailed instructions.



To verify your connection

These instructions apply to Windows clients.

1. To verify that your VPN connection is active, open an elevated command prompt, and run `ipconfig/all`.
2. View the results. Notice that the IP address you received is one of the addresses within the Point-to-Site VPN Client Address Pool that you specified in your configuration. The results are similar to this example:

```
PPP adapter VNet1:  
  Connection-specific DNS Suffix .:  
  Description.....: VNet1  
  Physical Address.....:  
  DHCP Enabled.....: No  
  Autoconfiguration Enabled....: Yes  
  IPv4 Address.....: 172.16.201.3(Preferred)  
  Subnet Mask.....: 255.255.255.255  
  Default Gateway.....:  
  NetBIOS over Tcpip.....: Enabled
```

To connect to a virtual machine

These instructions apply to Windows clients.

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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | 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.

To troubleshoot an RDP connection to a VM

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.
- Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VpnClientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
- 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](#).
- Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.
- For more information about RDP connections, see [Troubleshoot Remote Desktop connections to a VM](#).

To add or remove trusted root certificates

You can add and remove trusted root certificates from Azure. When you remove a root certificate, clients that have a certificate generated from that root won't be able to authenticate, and thus will not be able to connect. If you want a client to authenticate and connect, you need to install a new client certificate generated from a root certificate that is trusted (uploaded) to Azure.

To add a trusted root certificate

You can add up to 20 trusted root certificate .cer files to Azure. For instructions, see the section [Upload a trusted root certificate](#) in this article.

To remove a trusted root certificate

1. To remove a trusted root certificate, navigate to the **Point-to-site configuration** page for your virtual network gateway.
2. In the **Root certificate** section of the page, locate the certificate that you want to remove.
3. Click the ellipsis next to the certificate, and then click 'Remove'.

To revoke a client certificate

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. This is different than removing a trusted root certificate. If you remove a trusted root certificate .cer from Azure, it revokes the access for all client certificates generated/signed by the revoked root certificate. Revoking a client certificate, rather than the root certificate, allows the other certificates that were generated from the root certificate to continue to be used for authentication.

The common practice is to use the root certificate to manage access at team or organization levels, while using revoked client certificates for fine-grained access control on individual users.

Revoke a client certificate

You can revoke a client certificate by adding the thumbprint to the revocation list.

1. Retrieve the client certificate thumbprint. For more information, see [How to retrieve the Thumbprint of a Certificate](#).
2. Copy the information to a text editor and remove all spaces so that it is a continuous string.
3. Navigate to the virtual network gateway **Point-to-site-configuration** page. This is the same page that you used to [upload a trusted root certificate](#).
4. In the **Revoked certificates** section, input a friendly name for the certificate (it doesn't have to be the certificate CN).
5. Copy and paste the thumbprint string to the **Thumbprint** field.
6. The thumbprint validates and is automatically added to the revocation list. A message appears on the screen that the list is updating.
7. After updating has completed, the certificate can no longer be used to connect. Clients that try to connect using this certificate receive a message saying that the certificate is no longer valid.

Point-to-Site FAQ

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2. Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\
IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

What tools can I use to create certificates?

You can use your Enterprise PKI solution (your internal PKI), Azure PowerShell, MakeCert, and OpenSSL.

Are there instructions for certificate settings and parameters?

- **Internal PKI/Enterprise PKI solution:** See the steps to [Generate certificates](#).
- **Azure PowerShell:** See the [Azure PowerShell](#) article for steps.
- **MakeCert:** See the [MakeCert](#) article for steps.
- **OpenSSL:**
 - When exporting certificates, be sure to convert the root certificate to Base64.
 - For the client certificate:
 - When creating the private key, specify the length as 4096.
 - When creating the certificate, for the `-extensions` parameter, specify `usr_cert`.

Next steps

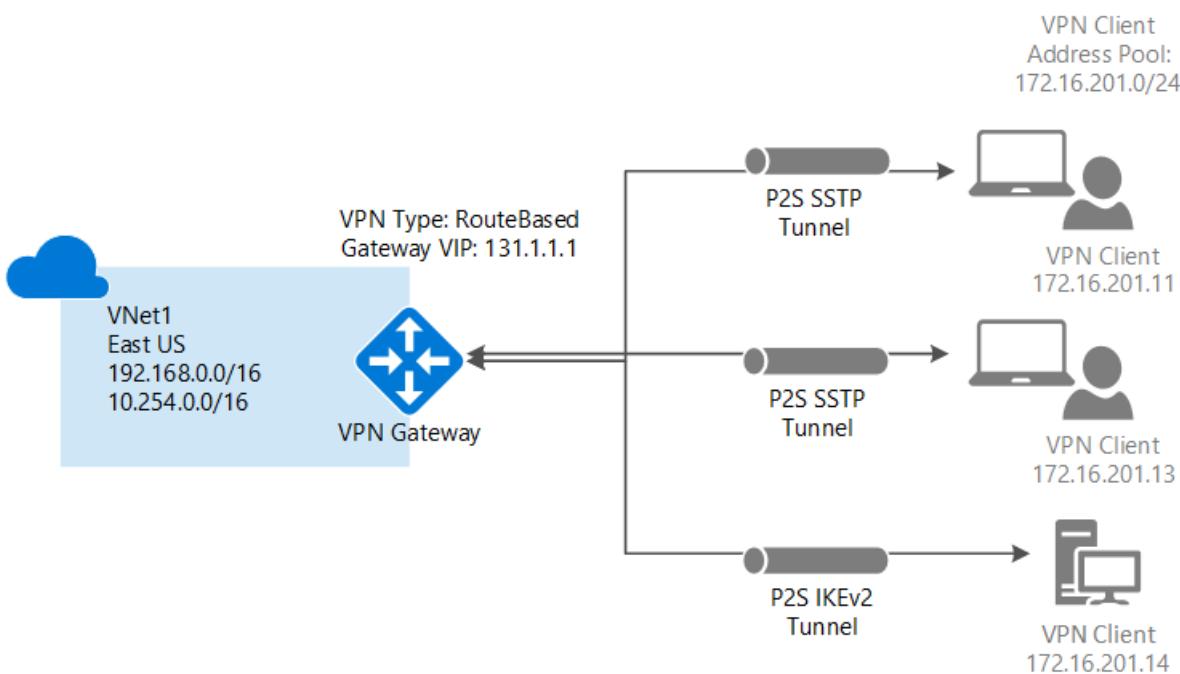
Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#). To understand more about networking and virtual machines, see [Azure and Linux VM network overview](#).

For P2S troubleshooting information, [Troubleshooting Azure point-to-site connections](#).

Configure a Point-to-Site connection to a VNet using native Azure certificate authentication: PowerShell

4/18/2018 • 25 minutes to read • [Edit Online](#)

This article helps you securely connect individual clients running Windows or Mac OS X to an Azure VNet. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location, such when you are telecommuting from home or a conference. You can also use P2S instead of a Site-to-Site VPN when you have only a few clients that need to connect to a VNet. Point-to-Site connections do not require a VPN device or a public-facing IP address. P2S creates the VPN connection over either SSTP (Secure Socket Tunneling Protocol), or IKEv2. For more information about Point-to-Site VPN, see [About Point-to-Site VPN](#).



Architecture

Point-to-Site native Azure certificate authentication connections use the following items, which you configure in this exercise:

- A RouteBased VPN gateway.
- The public key (.cer file) for a root certificate, which is uploaded to Azure. Once the certificate is uploaded, it is considered a trusted certificate and is used for authentication.
- A client certificate that is generated from the root certificate. The client certificate installed on each client computer that will connect to the VNet. This certificate is used for client authentication.
- A VPN client configuration. The VPN client configuration files contain the necessary information for the client to connect to the VNet. The files configure the existing VPN client that is native to the operating system. Each client that connects must be configured using the settings in the configuration files.

Before you begin

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- Install the latest version of the Resource Manager PowerShell cmdlets. For more information about installing

PowerShell cmdlets, see [How to install and configure Azure PowerShell](#). This is important because earlier versions of the cmdlets do not contain the current values that you need for this exercise.

Example values

You can use the example values to create a test environment, or refer to these values to better understand the examples in this article. The variables are set in section 1 of the article. You can either use the steps as a walk-through and use the values without changing them, or change them to reflect your environment.

- **Name: VNet1**

- **Address space: 192.168.0.0/16 and 10.254.0.0/16**

This example uses more than one address space to illustrate that this configuration works with multiple address spaces. However, multiple address spaces are not required for this configuration.

- **Subnet name: FrontEnd**

- **Subnet address range: 192.168.1.0/24**

- **Subnet name: BackEnd**

- **Subnet address range: 10.254.1.0/24**

- **Subnet name: GatewaySubnet**

The Subnet name *GatewaySubnet* is mandatory for the VPN gateway to work.

- **GatewaySubnet address range: 192.168.200.0/24**

- **VPN client address pool: 172.16.201.0/24**

VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the VPN client address pool.

- **Subscription:** If you have more than one subscription, verify that you are using the correct one.

- **Resource Group: TestRG**

- **Location: East US**

- **DNS Server: IP address** of the DNS server that you want to use for name resolution. (optional)

- **GW Name: Vnet1GW**

- **Public IP name: VNet1GWPIP**

- **VpnType: RouteBased**

1. Log in and set variables

In this section, you log in and declare the values used for this configuration. The declared values are used in the sample scripts. Change the values to reflect your own environment. Or, you can use the declared values and go through the steps as an exercise.

1. Open your PowerShell console with elevated privileges, and log in to your Azure account. This cmdlet prompts you for the login credentials. After logging in, it downloads your account settings so that they are available to Azure PowerShell.

```
Connect-AzureRmAccount
```

2. Get a list of your Azure subscriptions.

```
Get-AzureRmSubscription
```

3. Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Name of subscription"
```

4. Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary.

```
$VNetName = "VNet1"
$FESubName = "FrontEnd"
$BESubName = "Backend"
$GWSubName = "GatewaySubnet"
$VNetPrefix1 = "192.168.0.0/16"
$VNetPrefix2 = "10.254.0.0/16"
$FESubPrefix = "192.168.1.0/24"
$BESubPrefix = "10.254.1.0/24"
$GWSubPrefix = "192.168.200.0/26"
$VPNClientAddressPool = "172.16.201.0/24"
$RG = "TestRG"
$Location = "East US"
$GWName = "VNet1GW"
$GWIPName = "VNet1GWPiP"
$GWIPconfName = "gwipconf"
```

2. Configure a VNet

1. Create a resource group.

```
New-AzureRmResourceGroup -Name $RG -Location $Location
```

2. Create the subnet configurations for the virtual network, naming them *FrontEnd*, *BackEnd*, and *GatewaySubnet*. These prefixes must be part of the VNet address space that you declared.

```
$fesub = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName -AddressPrefix $FESubPrefix
$besub = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName -AddressPrefix $BESubPrefix
$gbsub = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName -AddressPrefix $GWSubPrefix
```

3. Create the virtual network.

In this example, the `-DnsServer` server parameter is optional. Specifying a value does not create a new DNS server. The DNS server IP address that you specify should be a DNS server that can resolve the names for the resources you are connecting to from your VNet. This example uses a private IP address, but it is likely that this is not the IP address of your DNS server. Be sure to use your own values. The value you specify is used by the resources that you deploy to the VNet, not by the P2S connection or the VPN client.

```
New-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG -Location $Location -AddressPrefix
$VNetPrefix1,$VNetPrefix2 -Subnet $fesub, $besub, $gbsub -DnsServer 10.2.1.3
```

4. Specify the variables for the virtual network you created.

```
$vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
```

5. A VPN gateway must have a Public IP address. You first request the IP address resource, and then refer to it when creating your virtual network gateway. The IP address is dynamically assigned to the resource when the VPN gateway is created. VPN Gateway currently only supports *Dynamic* Public IP address allocation. You cannot request a Static Public IP address assignment. However, it doesn't mean that the IP address changes after it has been assigned to your VPN gateway. 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.

Request a dynamically assigned public IP address.

```
$pip = New-AzureRmPublicIpAddress -Name $GWIPName -ResourceGroupName $RG -Location $Location -  
AllocationMethod Dynamic  
$ipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress  
$pip
```

3. Create the VPN gateway

Configure and create the virtual network gateway for your VNet.

- The `-GatewayType` must be **Vpn** and the `-VpnType` must be **RouteBased**.
- The `-VpnClientProtocol` is used to specify the types of tunnels that you would like to enable. The two tunnel options are **SSTP** and **IKEv2**. You can choose to enable one of them or both. If you want to enable both, then specify both the names separated by a comma. The strongSwan client on Android and Linux and the native IKEv2 VPN client on iOS and OSX will use only the IKEv2 tunnel to connect. Windows clients try IKEv2 first and if that doesn't connect, they fall back to SSTP.
- A VPN gateway can take up to 45 minutes to complete, depending on the [gateway sku](#) you select. This example uses IKEv2.

```
New-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG `  
-Location $Location -IpConfigurations $ipconf -GatewayType Vpn `  
-VpnType RouteBased -EnableBgp $false -GatewaySku VpnGw1 -VpnClientProtocol "IKEv2"
```

4. Add the VPN client address pool

After the VPN gateway finishes creating, you can add the VPN client address pool. The VPN client address pool is the range from which the VPN clients receive an IP address when connecting. Use a private IP address range that does not overlap with the on-premises location that you connect from, or with the VNet that you want to connect to. In this example, the VPN client address pool is declared as a [variable](#) in Step 1.

```
$Gateway = Get-AzureRmVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientAddressPool $VPNclientAddressPool
```

5. Generate certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You upload the public key information of the root certificate to Azure. The public key is then considered 'trusted'. Client certificates must be generated from the trusted root certificate, and then installed on each client computer in the Certificates-Current User/Personal certificate store. The certificate is used to authenticate the client when it initiates a connection to the VNet.

If you use self-signed certificates, they must be created using specific parameters. You can create a self-signed certificate using the instructions for [PowerShell and Windows 10](#), or, if you don't have Windows 10, you can use [MakeCert](#). It's important that you follow the steps in the instructions when generating self-signed root certificates and client certificates. Otherwise, the certificates you generate will not be compatible with P2S connections and you receive a connection error.

1. Obtain the .cer file for the root certificate

You can use either a root certificate that was generated using an enterprise solution (recommended), or you can generate a self-signed certificate. After creating the root certificate, export the public certificate data (not the

private key) as a Base-64 encoded X.509 .cer file and upload the public certificate data to Azure.

- **Enterprise certificate:** If you are using an enterprise solution, you can use your existing certificate chain. Obtain the .cer file for the root certificate that you want to use.
- **Self-signed root certificate:** If you aren't using an enterprise certificate solution, you need to create a self-signed root certificate. It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the certificates you create won't be compatible with P2S connections and clients receive a connection error when trying to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the provided articles generate a compatible certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [Linux instructions](#)

2. Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. The client certificate is generated from the root certificate and installed on each client computer. If a valid client certificate is not installed and the client tries to connect to the VNet, authentication fails.

You can either generate a unique certificate for each client, or you can use the same certificate for multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate. Otherwise, if multiple clients are using the same client certificate and you need to revoke it, you have to generate and install new certificates for all the clients that use that certificate to authenticate.

You can generate client certificates using the following methods:

- **Enterprise certificate:**
 - If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
 - Make sure the client certificate is based on the 'User' certificate template that has 'Client Authentication' as the first item in the use list, rather than Smart Card Logon, etc. You can check the certificate by double-clicking the client certificate and viewing **Details > Enhanced Key Usage**.
- **Self-signed root certificate:** It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the client certificates you create won't be compatible with P2S connections and clients receive an error when trying to connect. The steps in either of the following articles generate a compatible client certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. The certificates that are generated can be installed on any supported P2S client.
 - [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. The certificates that are generated can be installed on any supported P2S client.
 - [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate using the preceding instructions, it's automatically installed on the computer that you used to generate it. If you want to install a client certificate on another client computer, you need to export it as a .pfx, along with the entire certificate chain. This creates a .pfx file that contains the root certificate information that is required for the client to successfully authenticate. For steps to export a certificate, see [Certificates - export a client certificate](#).

6. Upload the root certificate public key information

Verify that your VPN gateway has finished creating. Once it has completed, you can upload the .cer file (which contains the public key information) for a trusted root certificate to Azure. Once a.cer file is uploaded, Azure can use it to authenticate clients that have installed a client certificate generated from the trusted root certificate. You can upload additional trusted root certificate files - up to a total of 20 - later, if needed.

1. Declare the variable for your certificate name, replacing the value with your own.

```
$P2SRootCertName = "P2SRootCert.cer"
```

2. Replace the file path with your own, and then run the cmdlets.

```
$filePathForCert = "C:\cert\P2SRootCert.cer"
$cert = new-object System.Security.Cryptography.X509Certificates.X509Certificate2($filePathForCert)
$CertBase64 = [system.convert]::ToBase64String($cert.RawData)
$p2srootcert = New-AzureRmVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $CertBase64
```

3. Upload the public key information to Azure. Once the certificate information is uploaded, Azure considers this to be a trusted root certificate.

```
Add-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName -
VirtualNetworkGatewayname "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData $CertBase64
```

7. Install an exported client certificate

If you want to create a P2S connection from a client computer other than the one you used to generate the client certificates, you need to install a client certificate. When installing a client certificate, you need the password that was created when the client certificate was exported.

Make sure the client certificate was exported as a .pfx along with the entire certificate chain (which is the default). Otherwise, the root certificate information isn't present on the client computer and the client won't be able to authenticate properly.

For install steps, see [Install a client certificate](#).

8. Configure the native VPN client

The VPN client configuration files contain settings to configure devices to connect to a VNet over a P2S connection. For instructions to generate and install VPN client configuration files, see [Create and install VPN client configuration files for native Azure certificate authentication P2S configurations](#).

9. Connect to Azure

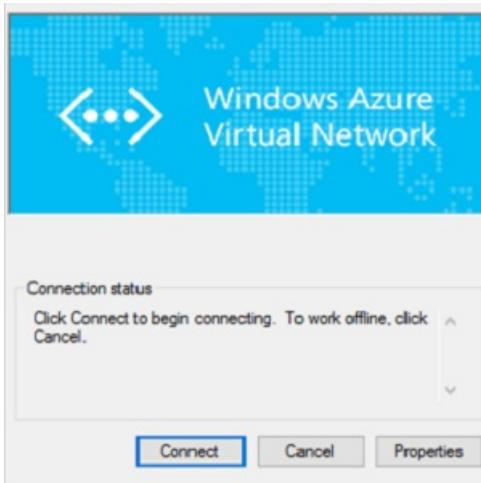
To connect from a Windows VPN client

NOTE

You must have Administrator rights on the Windows client computer from which you are connecting.

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. Click **Continue** to use elevated privileges.

2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection is established.



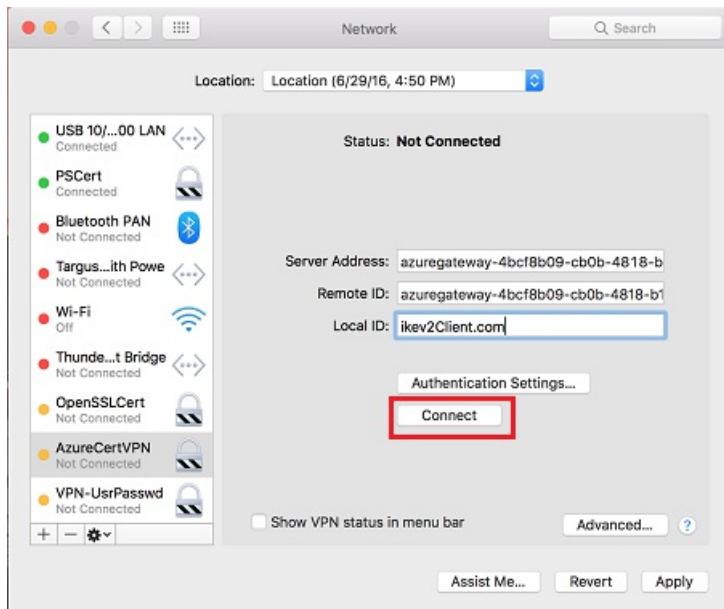
Troubleshooting Windows client P2S connections

If you are having trouble connecting, check the following items:

- If you exported a client certificate, make sure that you exported it as a .pfx file using the default value 'Include all certificates in the certification path if possible'. When you export it using this value, the root certificate information is also exported. When the certificate is installed on the client computer, the root certificate which is contained in the .pfx file is then also installed on the client computer. The client computer must have the root certificate information installed. To check, go to **Manage user certificates** and navigate to **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed. The root certificate must be present in order for authentication to work.
- If you are using a certificate that was issued using an Enterprise CA solution and are having trouble authenticating, check the authentication order on the client certificate. You can check the authentication list order by double-clicking the client certificate, and going to **Details > Enhanced Key Usage**. Make sure the list shows 'Client Authentication' as the first item. If not, you need to issue a client certificate based on the User template that has Client Authentication as the first item in the list.
- For additional P2S troubleshooting information, see [Troubleshoot P2S connections](#).

To connect from a Mac VPN client

From the Network dialog box, locate the client profile that you want to use, then click **Connect**.



To verify your connection

These instructions apply to Windows clients.

1. To verify that your VPN connection is active, open an elevated command prompt, and run `ipconfig/all`.
2. View the results. Notice that the IP address you received is one of the addresses within the Point-to-Site VPN Client Address Pool that you specified in your configuration. The results are similar to this example:

```
PPP adapter VNet1:  
  Connection-specific DNS Suffix .:  
  Description.....: VNet1  
  Physical Address.....:  
  DHCP Enabled.....: No  
  Autoconfiguration Enabled.....: Yes  
  IPv4 Address.....: 172.16.201.3(Preferred)  
  Subnet Mask.....: 255.255.255.255  
  Default Gateway.....:  
  NetBIOS over Tcpip.....: Enabled
```

To connect to a virtual machine

These instructions apply to Windows clients.

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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object 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.

To troubleshoot an RDP connection to a VM

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.
- Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VPNClientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
- 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](#).
- Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.
- For more information about RDP connections, see [Troubleshoot Remote Desktop connections to a VM](#).

To add or remove a root certificate

You can add and remove trusted root certificates from Azure. When you remove a root certificate, clients that have a certificate generated from the root certificate can't authenticate and won't be able to connect. If you want a client to authenticate and connect, you need to install a new client certificate generated from a root certificate that is trusted (uploaded) to Azure.

To add a trusted root certificate

You can add up to 20 root certificate .cer files to Azure. The following steps help you add a root certificate:

Method 1

This is the most efficient method to upload a root certificate.

1. Prepare the .cer file to upload:

```

$filePathForCert = "C:\cert\P2SRootCert3.cer"
$cert = new-object System.Security.Cryptography.X509Certificates.X509Certificate2($filePathForCert)
$CertBase64_3 = [system.convert]::ToBase64String($cert.RawData)
$p2srootcert = New-AzureRmVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $CertBase64_3

```

- Upload the file. You can only upload one file at a time.

```
Add-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName -VirtualNetworkGatewayname "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData $CertBase64_3
```

- To verify that the certificate file uploaded:

```
Get-AzureRmVpnClientRootCertificate -ResourceGroupName "TestRG" -VirtualNetworkGatewayName "VNet1GW"
```

Method 2

This method has more steps than Method 1, but has the same result. It is included in case you need to view the certificate data.

- Create and prepare the new root certificate to add to Azure. Export the public key as a Base-64 encoded X.509 (.CER) and open it with a text editor. Copy the values, as shown in the following example:

```
-----BEGIN CERTIFICATE-----
MIIDBTCAfgwIBAgIQvH9orou8J1MSox6ZBK83jAJBgUrDgMCHQUAMBx0GDAw
BgNVBAMTD0FSTVayU1jv3RDZXJ0jAeFw0xNjAAjUyMAxN0BwFw0z0TEyh1zEy
MzU5MTAhBoxDWBgNVBAHTD0FSTVayU1jv3RDZXJ0MjCCAS10QYJKozIhvCN
AQEBBQAQDgEPADCCAQggEAL5vQ1gPR2r0b17xBGxGHeesQtuo1HFrkL1Cxv/
7c1q5ExuNtMFwQjWmt8ejhMwERpnQvr5kPVNlb5hdLB7qz1Klm1VKEcVx/
K9LHfcMiuHohh2bjq0idcpq8ZD02XuDF45SDWAmhCgycFSS8XP/BHkpzmbAR5
Ymnkx3XfPadkHv1qV7rR8QaKKs1sy3C2fQov92gpmI0Mj7hxLLikPbkjgv1+Un
EJF/rCgXXR1VPgobLSqDCA1/skhQ2abzCy3actLtc3jPhbxwulfcdT+njtgt1MdI
XGfMSMDPVXzdodR8JU1WY9nLH8aoKHP47xv12gSLMk0CAwEAAlPMEBwsWvD
VR8BQEoWqoQArkg3CNURnw5ZxsCByKEcHBoxGDwIBgNvBAMTD0FSTVayU1jv
-----END CERTIFICATE-----
```

NOTE

When copying the certificate data, make sure that you copy the text as one continuous line without carriage returns or line feeds. You may need to modify your view in the text editor to 'Show Symbol/Show all characters' to see the carriage returns and line feeds.

- Specify the certificate name and key information as a variable. Replace the information with your own, as shown in the following example:

```
$P2SRootCertName2 = "ARMP2SRootCert2.cer"
$MyP2SCertPubKeyBase64_2 =
"MIIC/zCCAeugAwIBAgIQKazxxFjMkp9JRIx+tkTfSzAJBgUrDgMCHQUAMBgxFjAUBgNVBAMTDU15UDJTUm9vdEN1cnQwHhcNMTUxMj
E5MDI1MTIxWhcNMzkxMjMxMjM1OTU5WjAYMRWwFAYDVQQDEw1NeVayU1jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIIBC
gKCAQEAYjIXoWy8xE/GF10SiVua0bxbjZ1PJfcXkMwsHPzvhWc2esOKrVQtfgDz4ggAnOUFEkFaszjihDnxV3mjzE2SpmAVIZPf2/
yPWqkoHwmrp6Bp0vNV0pKxaGPOuK8+dql1xcL0eCkt69g41xy0FGRFkBcS1gVTViS9wjuuS7LPo5+0XgyFkAY3pSDiMzQCKrgNfgw5
WGMHRDAiruDQF1ciLNojAQCsDdLnI3pDYsvRW73HZEmh0qRRnJQe6VekvBYKLvnKaxUTKhFIYwuymHBB96nMFdRUKCIIwRIy8Hc8+s
QEaML2EItAjQv4+fqgYiFdSwqnQCPf/7IZbotgQIDAQABo00wSzBJBgnVHQEEQjBAgBAkuVrwFsCJAk5pb/eoCnRowGDEWMQBGA
1UEAxMNTx1QM1NSb290Q2VydIIQKazxxFjMkp9JRIx+tkTfSzAJBgUrDgMCHQUAA4IBAQa223veAZEiar9N12ubNH2+HwZASNzDVNqs
pkPKD97TXfkH1P1icS43Ta9kTz38eVrwI6E0yDk4jAuPaKnpuPYFRj9w540SvY6Pd0UwDoEqpIcAVp+b4VYwxPL6oyEQ8wnOYuoAK1h
hh201Cbo8h9mMy9ofU+RP6HJ71TquplFxID/XevI8tW6Dm+C/wCeV3EmI109KUob1D/e24z1o3Yz0tbyXwTIh34T0f0/zQvuubqZMc
IPFM1cDvcqiEFLWvWKOAnxbzckye2uk1gH052d8AVL3mGiX8wBjkjc/pMdxeVvCzJklBmqxtTM6XjdJALuVh16qFlqgTWCICb7ju"
```

- Add the new root certificate. You can only add one certificate at a time.

```
Add-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -VirtualNetworkGatewayname "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData
$MyP2SCertPubKeyBase64_2
```

- You can verify that the new certificate was added correctly by using the following example:

```
Get-AzureRmVpnClientRootCertificate -ResourceGroupName "TestRG" `  
-VirtualNetworkGatewayName "VNet1GW"
```

To remove a root certificate

1. Declare the variables.

```
$GWName = "Name_of_virtual_network_gateway"  
$RG = "Name_of_resource_group"  
$P2SRootCertName2 = "ARMP2SRootCert2.cer"  
$MyP2SCertPubKeyBase64_2 =  
"MIIC/zCCAeugAwIBAgIQKazxzFjMkp9JRIx+tkTfSzAJBgUrDgMCHQUAMBgxFjAUBgNVBAMTDU15UDJTUm9vdENlcnQwHcNMTUxMj  
E5MDI1MTIxWhcNMzKxMjMxMjM1OTU5WjAYMRYwFAYDVQDEw1NeVAyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIIBC  
gKCAQEAyjIXoIy8xE/GF10SiVuaA0bxBjZ1PJfcXkMwsHPzvhWc2esOKrVQtgFgdz4ggAnOUFEkFaszjiHdnXv3mjzE2SpmAVIZPF2/  
yPwqkoHwkmrp6Bp0NVNOpKxaGPOuK8+dql1xcL0eCkt69g41xy0FGRFkBcSIgVTViS9wjuuS7LPo5+OXgyFkAY3pSDiMzQCKRGNFgw5  
WGMHRDAiruDQF1ciLN0jaAQCsDdLnI3pDYsvRW73HZehmOqRRnJQe6VekvBYKLvnKaxUTKhFIYwuymHBB96nMFdRUKCZIiWRIy8Hc8+s  
QEaML2EItAjQv4+fqgYiFdSwqnQCPf/7IZbotgQIDAQABo0wSzBJBgvNHQEEQjBAGBAkuVrWvFsCJAdK5pb/eoCnRowGDEWMBQGA  
1UEAxMNTXlQM1NSb290Q2VydIIQKazxzFjMkp9JRIx+tkTfSzAJBgUrDgMCHQUAA4IBAQAA223veAZEiar9N12ubNH2+HwZASNzDVNqs  
pkPKD97TXfkH1PlIcs43TaYkTz38eVrwI6E0yDk4jAuPaKnPuPYFRj9w540SvY6Pd0UwDoEqpIcAVp+b4VYwxPL6oyEQ8wnOYuoAK1h  
hh201Cbo8h9mMy9ofu+RP6HJ71TquplfXdID/Xevi8tW6Dm+C/wCeV3EmI109KUob1D/e24zlo3Yz0tbyXwTIh34T0f0/zQvuBqZMc  
IPfM1cDvqcqiEFLWvWKOAnxbzckye2uk1gH052d8AVL3mGiX8wBJkjc/pMdxrEvvCzJk1tBmqxTM6XjDJALuVh16qFlqgTWCICb7ju"
```

2. Remove the certificate.

```
Remove-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -  
VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -PublicCertData $MyP2SCertPubKeyBase64_2
```

3. Use the following example to verify that the certificate was removed successfully.

```
Get-AzureRmVpnClientRootCertificate -ResourceGroupName "TestRG" `  
-VirtualNetworkGatewayName "VNet1GW"
```

To revoke a client certificate

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. This is different than removing a trusted root certificate. If you remove a trusted root certificate .cer from Azure, it revokes the access for all client certificates generated/signed by the revoked root certificate. Revoking a client certificate, rather than the root certificate, allows the other certificates that were generated from the root certificate to continue to be used for authentication.

The common practice is to use the root certificate to manage access at team or organization levels, while using revoked client certificates for fine-grained access control on individual users.

Revoke a client certificate

1. Retrieve the client certificate thumbprint. For more information, see [How to retrieve the Thumbprint of a Certificate](#).
2. Copy the information to a text editor and remove all spaces so that it is a continuous string. This string is declared as a variable in the next step.
3. Declare the variables. Make sure to declare the thumbprint you retrieved in the previous step.

```
$RevokedClientCert1 = "NameofCertificate"  
$RevokedThumbprint1 = "51ab1edd8da4cfed77e20061c5eb6d2ef2f778c7"  
$GWName = "Name_of_virtual_network_gateway"  
$RG = "Name_of_resource_group"
```

4. Add the thumbprint to the list of revoked certificates. You see "Succeeded" when the thumbprint has been added.

```
Add-AzureRmVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 `  
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG `  
-Thumbprint $RevokedThumbprint1
```

5. Verify that the thumbprint was added to the certificate revocation list.

```
Get-AzureRmVpnClientRevokedCertificate -VirtualNetworkGatewayName $GWName -ResourceGroupName $RG
```

6. After the thumbprint has been added, the certificate can no longer be used to connect. Clients that try to connect using this certificate receive a message saying that the certificate is no longer valid.

To reinstate a client certificate

You can reinstate a client certificate by removing the thumbprint from the list of revoked client certificates.

1. Declare the variables. Make sure you declare the correct thumbprint for the certificate that you want to reinstate.

```
$RevokedClientCert1 = "NameofCertificate"  
$RevokedThumbprint1 = "51ab1edd8da4cfed77e20061c5eb6d2ef2f778c7"  
$GWName = "Name_of_virtual_network_gateway"  
$RG = "Name_of_resource_group"
```

2. Remove the certificate thumbprint from the certificate revocation list.

```
Remove-AzureRmVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 `  
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -Thumbprint $RevokedThumbprint1
```

3. Check if the thumbprint is removed from the revoked list.

```
Get-AzureRmVpnClientRevokedCertificate -VirtualNetworkGatewayName $GWName -ResourceGroupName $RG
```

Point-to-Site FAQ

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)

- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2.

Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\
IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

What tools can I use to create certificates?

You can use your Enterprise PKI solution (your internal PKI), Azure PowerShell, MakeCert, and OpenSSL.

Are there instructions for certificate settings and parameters?

- **Internal PKI/Enterprise PKI solution:** See the steps to [Generate certificates](#).
- **Azure PowerShell:** See the [Azure PowerShell](#) article for steps.

- **MakeCert:** See the [MakeCert](#) article for steps.
- **OpenSSL:**
 - When exporting certificates, be sure to convert the root certificate to Base64.
 - For the client certificate:
 - When creating the private key, specify the length as 4096.
 - When creating the certificate, for the `-extensions` parameter, specify `usr_cert`.

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#). To understand more about networking and virtual machines, see [Azure and Linux VM network overview](#).

For P2S troubleshooting information, [Troubleshooting: Azure point-to-site connection problems](#).

Generate and export certificates for Point-to-Site using PowerShell

9/10/2018 • 7 minutes to read • [Edit Online](#)

Point-to-Site connections use certificates to authenticate. This article shows you how to create a self-signed root certificate and generate client certificates using PowerShell on Windows 10 or Windows Server 2016. If you are looking for different certificate instructions, see [Certificates - Linux](#) or [Certificates - MakeCert](#).

You must perform the steps in this article on a computer running Windows 10 or Windows Server 2016. The PowerShell cmdlets that you use to generate certificates are part of the operating system and do not work on other versions of Windows. The Windows 10 or Windows Server 2016 computer is only needed to generate the certificates. Once the certificates are generated, you can upload them, or install them on any supported client operating system.

If you do not have access to a Windows 10 or Windows Server 2016 computer, you can use [MakeCert](#) to generate certificates. The certificates that you generate using either method can be installed on any [supported](#) client operating system.

1. Create a self-signed root certificate

Use the `New-SelfSignedCertificate` cmdlet to create a self-signed root certificate. For additional parameter information, see [New-SelfSignedCertificate](#).

1. From a computer running Windows 10 or Windows Server 2016, open a Windows PowerShell console with elevated privileges.
2. Use the following example to create the self-signed root certificate. The following example creates a self-signed root certificate named 'P2SRootCert' that is automatically installed in 'Certificates-Current User\Personal\Certificates'. You can view the certificate by opening `certmgr.msc`, or *Manage User Certificates*.

```
$cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature `  
-Subject "CN=P2SRootCert" -KeyExportPolicy Exportable `  
-HashAlgorithm sha256 -KeyLength 2048 `  
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -KeyUsage CertSign
```

2. Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. You generate a client certificate from the self-signed root certificate, and then export and install the client certificate. If the client certificate is not installed, authentication fails.

The following steps walk you through generating a client certificate from a self-signed root certificate. You may generate multiple client certificates from the same root certificate. When you generate client certificates using the steps below, the client certificate is automatically installed on the computer that you used to generate the certificate. If you want to install a client certificate on another client computer, you can export the certificate.

The examples use the `New-SelfSignedCertificate` cmdlet to generate a client certificate that expires in one year. For additional parameter information, such as setting a different expiration value for the client certificate, see [New-SelfSignedCertificate](#).

Example 1

This example uses the declared '\$cert' variable from the previous section. If you closed the PowerShell console after creating the self-signed root certificate, or are creating additional client certificates in a new PowerShell console session, use the steps in [Example 2](#).

Modify and run the example to generate a client certificate. If you run the following example without modifying it, the result is a client certificate named 'P2SChildCert'. If you want to name the child certificate something else, modify the CN value. Do not change the TextExtension when running this example. The client certificate that you generate is automatically installed in 'Certificates - Current User\Personal\Certificates' on your computer.

```
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec Signature `  
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable `  
-HashAlgorithm sha256 -KeyLength 2048 `  
-CertStoreLocation "Cert:\CurrentUser\My" `  
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

Example 2

If you are creating additional client certificates, or are not using the same PowerShell session that you used to create your self-signed root certificate, use the following steps:

1. Identify the self-signed root certificate that is installed on the computer. This cmdlet returns a list of certificates that are installed on your computer.

```
Get-ChildItem -Path "Cert:\CurrentUser\My"
```

2. Locate the subject name from the returned list, then copy the thumbprint that is located next to it to a text file. In the following example, there are two certificates. The CN name is the name of the self-signed root certificate from which you want to generate a child certificate. In this case, 'P2SRootCert'.

Thumbprint	Subject
AED812AD883826FF76B4D1D5A77B3C08EFA79F3F	CN=P2SChildCert4
7181AA8C1B4D34EEDB2F3D3BEC5839F3FE52D655	CN=P2SRootCert

3. Declare a variable for the root certificate using the thumbprint from the previous step. Replace THUMBPRINT with the thumbprint of the root certificate from which you want to generate a child certificate.

```
$cert = Get-ChildItem -Path "Cert:\CurrentUser\My\THUMBPRINT"
```

For example, using the thumbprint for P2SRootCert in the previous step, the variable looks like this:

```
$cert = Get-ChildItem -Path "Cert:\CurrentUser\My\7181AA8C1B4D34EEDB2F3D3BEC5839F3FE52D655"
```

4. Modify and run the example to generate a client certificate. If you run the following example without modifying it, the result is a client certificate named 'P2SChildCert'. If you want to name the child certificate something else, modify the CN value. Do not change the TextExtension when running this example. The client certificate that you generate is automatically installed in 'Certificates - Current User\Personal\Certificates' on your computer.

```

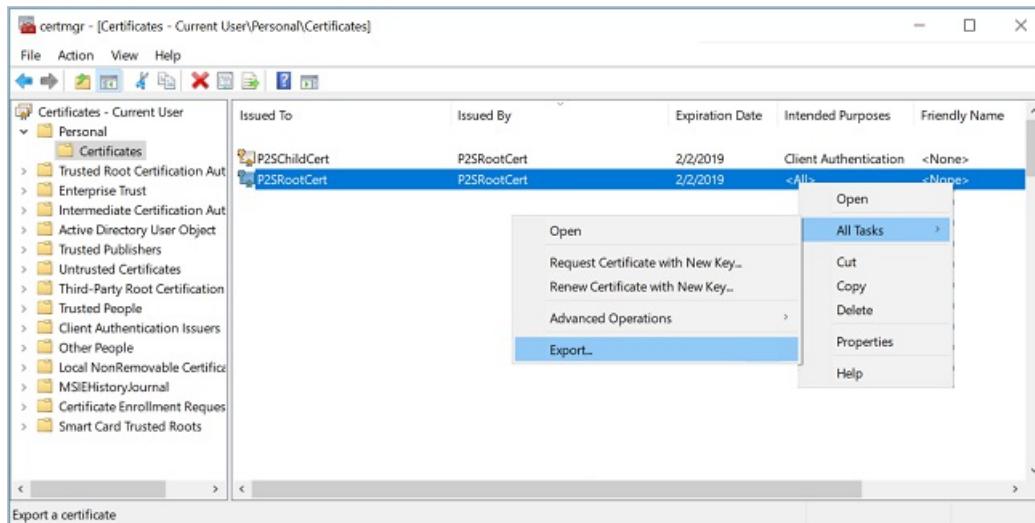
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec Signature ` 
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable ` 
-HashAlgorithm sha256 -KeyLength 2048 ` 
-CertStoreLocation "Cert:\CurrentUser\My" ` 
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")

```

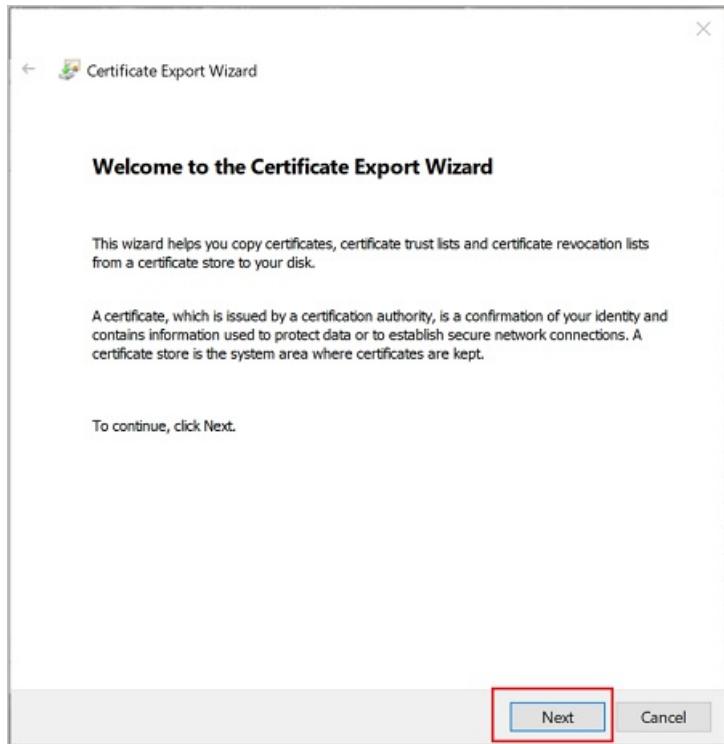
3. Export the root certificate public key (.cer)

After creating a self-signed root certificate, export the root certificate public key .cer file (not the private key). You will later upload this file to Azure. The following steps help you export the .cer file for your self-signed root certificate:

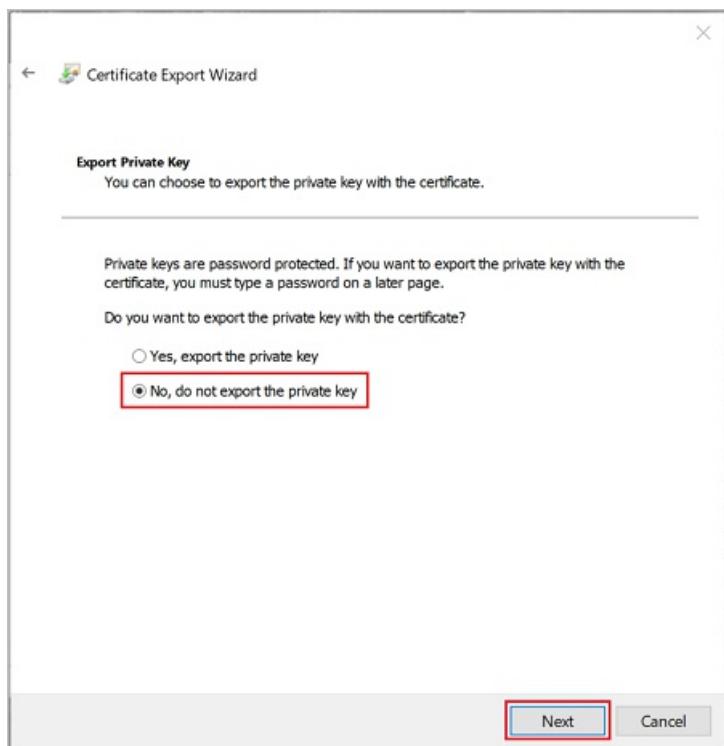
1. To obtain a .cer file from the certificate, open **Manage user certificates**. Locate the self-signed root certificate, typically in 'Certificates - Current User\Personal\Certificates', and right-click. Click **All Tasks**, and then click **Export**. This opens the **Certificate Export Wizard**. If you can't find the certificate under Current User\Personal\Certificates it could be that you opened Certificates Manager for the local computer certificates (title will be "Certificates - Local Computer" as opposed to "Certificates - Current User"). To open Certificates Manager in current user scope start it from the same PowerShell where the certificates were created by typing `certmgr`.



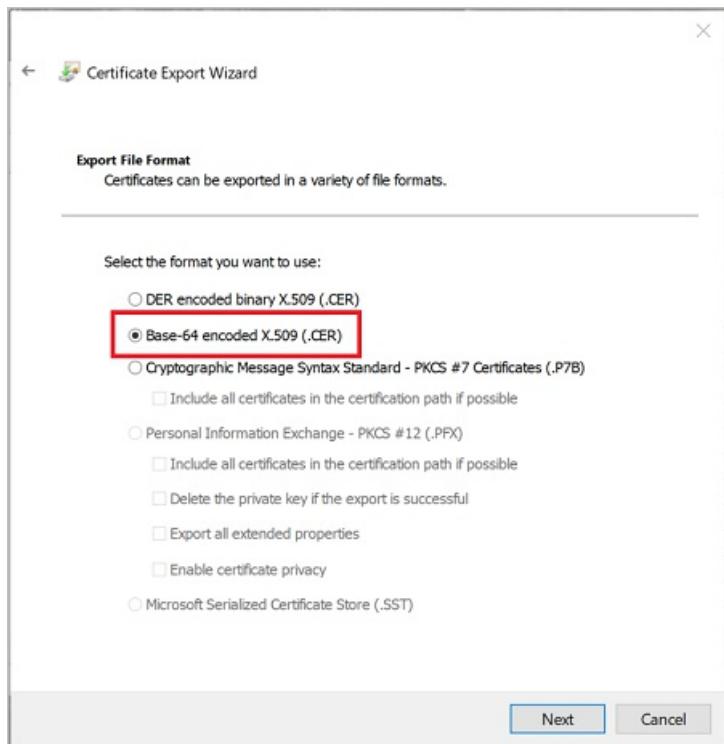
2. In the Wizard, click **Next**.



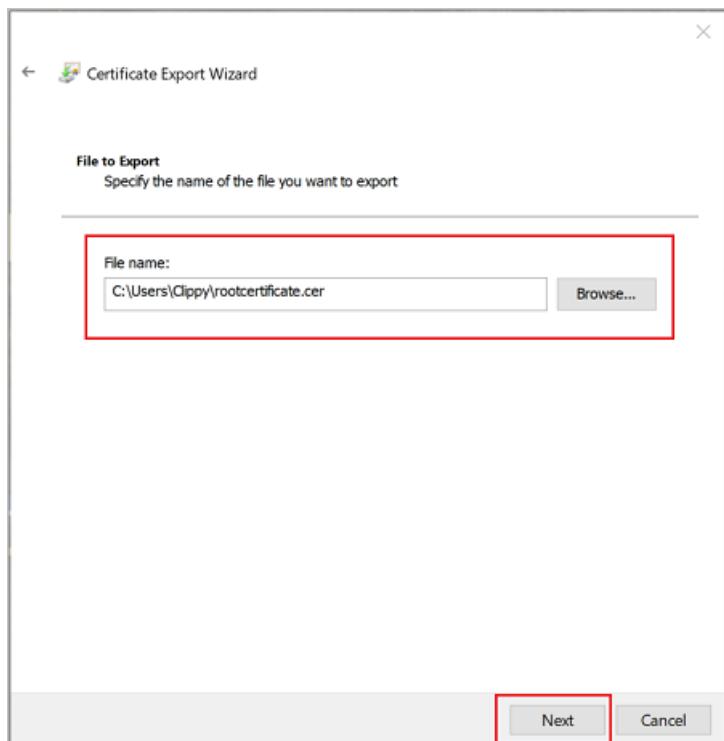
3. Select **No, do not export the private key**, and then click **Next**.



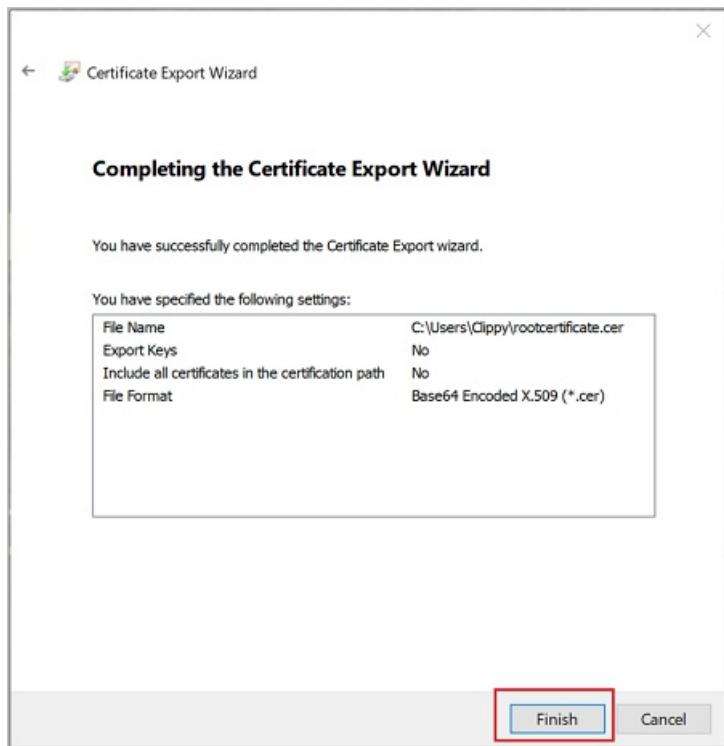
4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**, and then click **Next**.



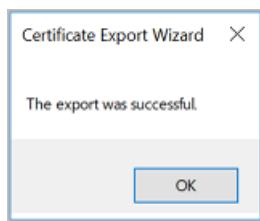
5. For **File to Export**, **Browse** to the location to which you want to export the certificate. For **File name**, name the certificate file. Then, click **Next**.



6. Click **Finish** to export the certificate.



7. Your certificate is successfully exported.



8. The exported certificate looks similar to this:



9. If you open the exported certificate using Notepad, you see something similar to this example. The section in blue contains the information that is uploaded to Azure. If you open your certificate with Notepad and it does not look similar to this, typically this means you did not export it using the Base-64 encoded X.509(CER) format. Additionally, if you want to use a different text editor, understand that some editors can introduce unintended formatting in the background. This can create problems when uploaded the text from this certificate to Azure.

```

rootcertificate.cer - Notepad
File Edit Format View Help
-----BEGIN CERTIFICATE-----
MIIC5zCCAc+gAwIBAgIQRtNVXFGwtIhK4R7RaOak0jANBgkqhkiG9w0BAQsFADAW
MRQwEgYDVQQDDAtQM1NSb290Q2VydDAeFw0xODAyMDIxOTM5MThaFw0xOTAyMDIx
OTU5MThaMBYxFDASBgNVBAMMC1AyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEAvrvt56dNhYGXF01/NUS2GcCCQ9mSKQeMCsoMZgyEC8nP
1ZioGeFwd23Thb9+k00k08sPbM4Jm42riyNsmtVNyCviP5pC/V2NyQr/F61+K5X
0GurFxSm/mv6wf0xf/FHvu5PojX7Z5/oEbeYB11GVVPgq6QWSrx331W1zmD2FeuA
QE46dMPSPHFwnc6P2hfthzs3+tv1R4dg02Wr5drVNvr1cVOHqoSbf/dqjV2thzz
ZSSN5kew2G/3H7Mc2ScZD+AYXWRzuiIrKrJSZiuIfJxLpQJaLQTByEU+wT8R8Rnq
GKmyKuUCPoXGYY3TRPmBXgA0800RsKFrXPWp5SiuuQIDAQABozEwLzAOBgNVHQ8B
Af8EBAMCAgQwHQYDVR0OBByEFCk6GjsuuTSfxMNz4ATy77DEbyCwMA0GCSqGSIb3
DQEBCwUA4IBAQBzZQCC4SEXqDgR2BL3uj17XDoscR/52U9rVxLorW1Z4Wu4kRA0
EA4IppBNmQep9eaCCqNfc6sbXf4QWjkBv1TBja20rFzt5cTAkwYG6Y0aWT1L//fw
u9goi2RihBs6IeBwc621u1Lo0Lw5htCOv0XPSS0lmAm5R6//IqyHZRcAK/TffitC
EIYTfcKdavxe9CgY/TtzEMCS7gLARDpHh/nDrxtIeKx1vvUfnOoeXeaSsQwHtumq
GFH3+BgzxEGB8v4oLlQzzbvj+xAb1WKpYqXFbp/ulhd6Ao2qn9sIVuKRkJjBj78
o45M2omAFZZaAVQrUa/fprKr3es/6IYrPT8J
-----END CERTIFICATE-----

```

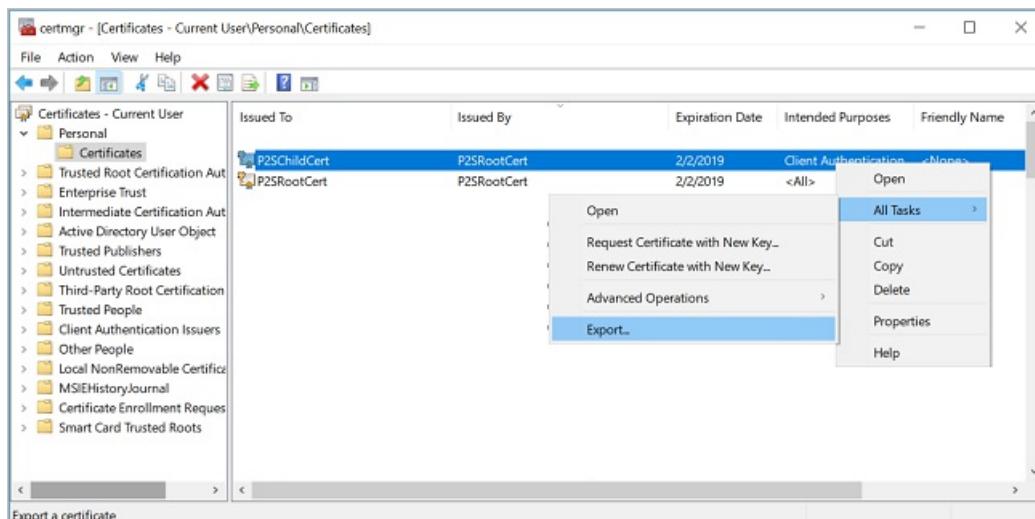
Export the self-signed root certificate and private key to store it (optional)

You may want to export the self-signed root certificate and store it safely as backup. If need be, you can later install it on another computer and generate more client certificates. To export the self-signed root certificate as a .pfx, select the root certificate and use the same steps as described in [Export a client certificate](#).

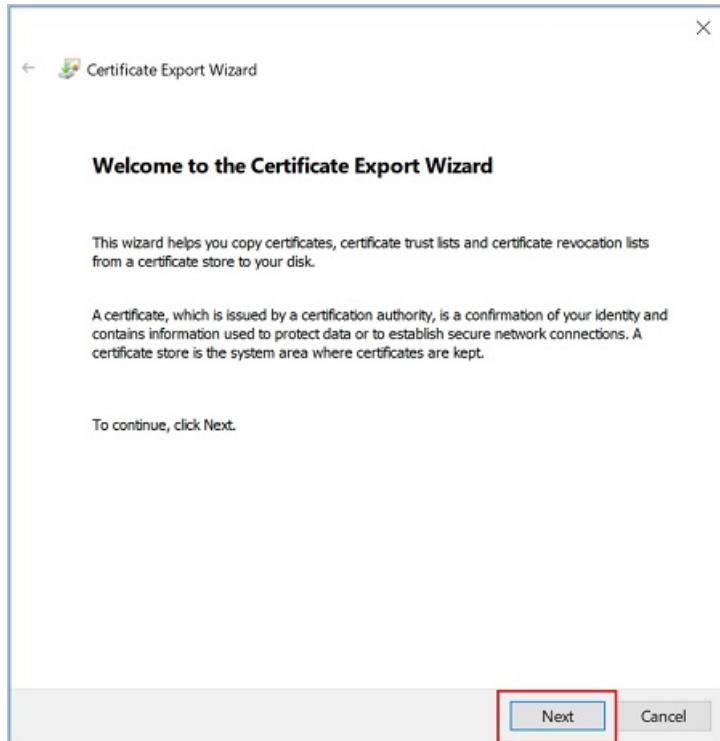
4. Export the client certificate

When you generate a client certificate, it's automatically installed on the computer that you used to generate it. If you want to install the client certificate on another client computer, you need to export the client certificate that you generated.

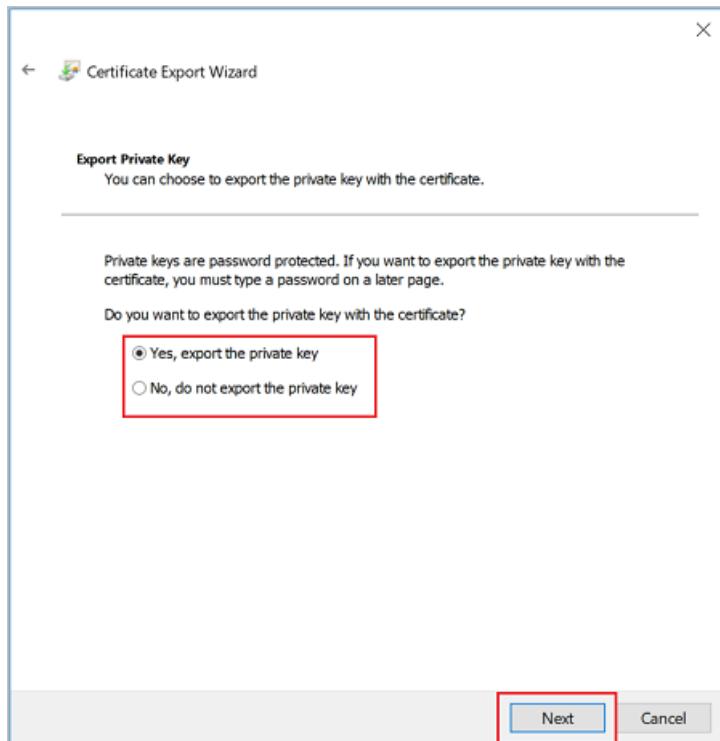
- To export a client certificate, open **Manage user certificates**. The client certificates that you generated are, by default, located in 'Certificates - Current User\Personal\Certificates'. Right-click the client certificate that you want to export, click **all tasks**, and then click **Export** to open the **Certificate Export Wizard**.



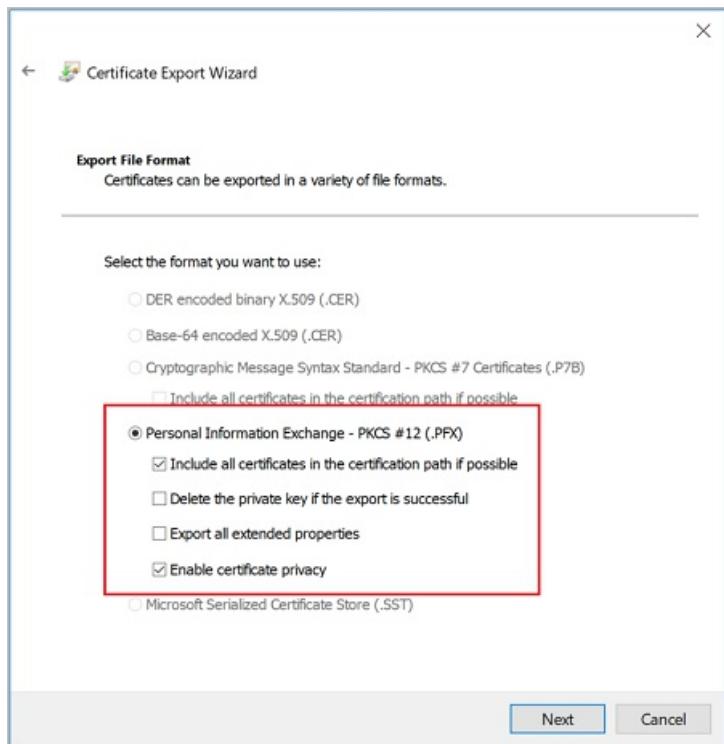
2. In the Certificate Export Wizard, click **Next** to continue.



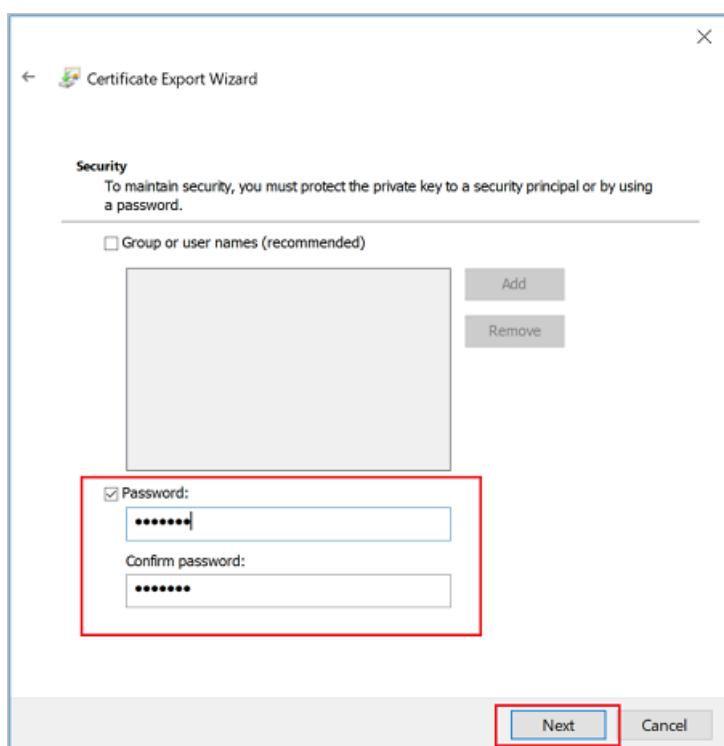
3. Select **Yes, export the private key**, and then click **Next**.



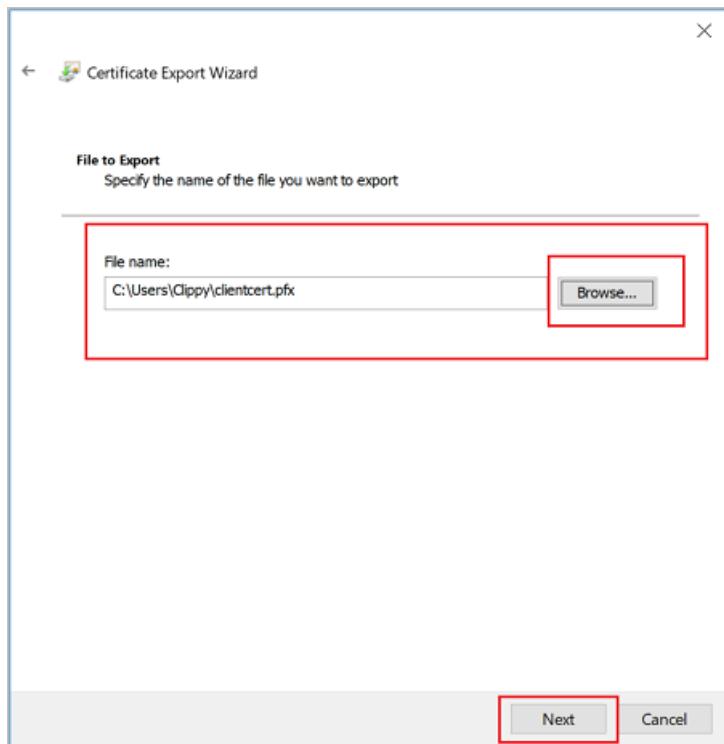
4. On the **Export File Format** page, leave the defaults selected. Make sure that **Include all certificates in the certification path if possible** is selected. This setting additionally exports the root certificate information that is required for successful client authentication. Without it, client authentication fails because the client doesn't have the trusted root certificate. Then, click **Next**.



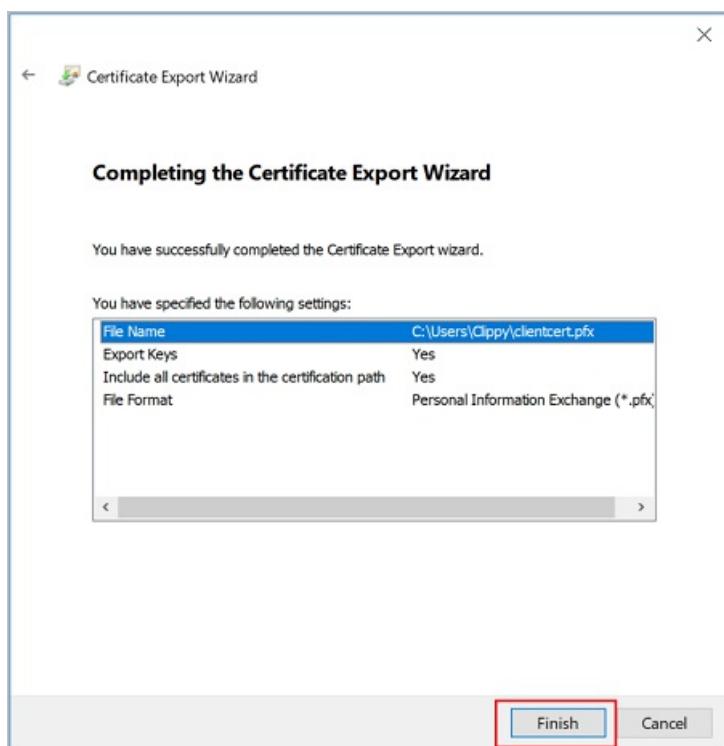
5. On the **Security** page, you must protect the private key. If you select to use a password, make sure to record or remember the password that you set for this certificate. Then, click **Next**.



6. On the **File to Export**, **Browse** to the location to which you want to export the certificate. For **File name**, name the certificate file. Then, click **Next**.



7. Click **Finish** to export the certificate.



5. Install an exported client certificate

Each client that connects to the VNet over a P2S connection requires a client certificate to be installed locally.

To install a client certificate, see [Install a client certificate for Point-to-Site connections](#).

6. Continue with the P2S configuration steps

Continue with your Point-to-Site configuration.

- For **Resource Manager** deployment model steps, see [Configure P2S using native Azure certificate authentication](#).

- For **classic** deployment model steps, see [Configure a Point-to-Site VPN connection to a VNet \(classic\)](#).
- For P2S troubleshooting information, see [Troubleshooting Azure point-to-site connections](#).

Generate and export certificates for Point-to-Site connections using MakeCert

9/10/2018 • 6 minutes to read • [Edit Online](#)

Point-to-Site connections use certificates to authenticate. This article shows you how to create a self-signed root certificate and generate client certificates using MakeCert. If you are looking for different certificate instructions, see [Certificates - PowerShell](#) or [Certificates - Linux](#).

While we recommend using the [Windows 10 PowerShell steps](#) to create your certificates, we provide these MakeCert instructions as an optional method. The certificates that you generate using either method can be installed on [any supported client operating system](#). However, MakeCert has the following limitation:

- MakeCert is deprecated. This means that this tool could be removed at any point. Any certificates that you already generated using MakeCert won't be affected when MakeCert is no longer available. MakeCert is only used to generate the certificates, not as a validating mechanism.

Create a self-signed root certificate

The following steps show you how to create a self-signed certificate using MakeCert. These steps are not deployment-model specific. They are valid for both Resource Manager and classic.

1. Download and install [MakeCert](#).
2. After installation, you can typically find the makecert.exe utility under this path: 'C:\Program Files (x86)\Windows Kits\10\bin<arch>'. Although, it's possible that it was installed to another location. Open a command prompt as administrator and navigate to the location of the MakeCert utility. You can use the following example, adjusting for the proper location:

```
cd C:\Program Files (x86)\Windows Kits\10\bin\x64
```

3. Create and install a certificate in the Personal certificate store on your computer. The following example creates a corresponding .cer file that you upload to Azure when configuring P2S. Replace 'P2SRootCert' and 'P2SRootCert.cer' with the name that you want to use for the certificate. The certificate is located in your 'Certificates - Current User\Personal\Certificates'.

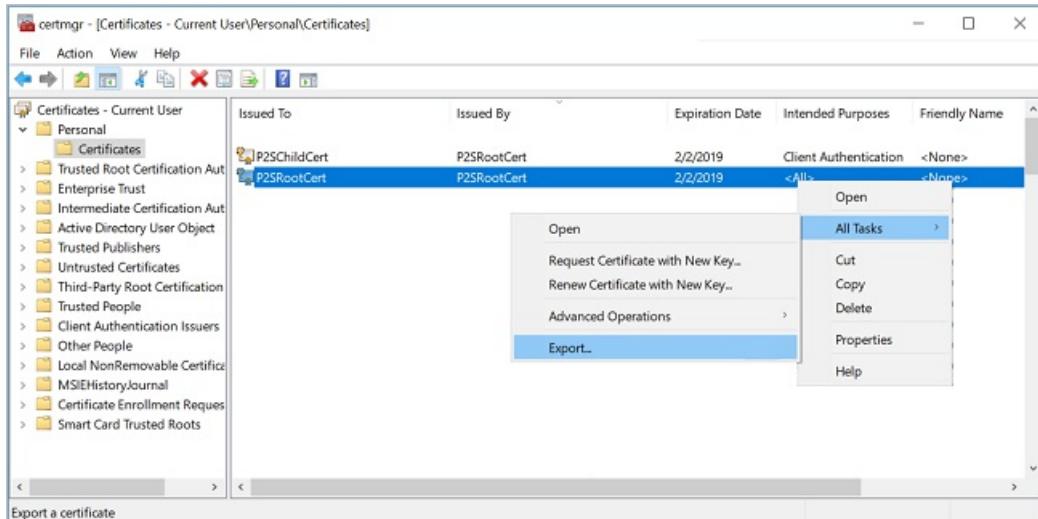
```
makecert -sky exchange -r -n "CN=P2SRootCert" -pe -a sha256 -len 2048 -ss My
```

Export the public key (.cer)

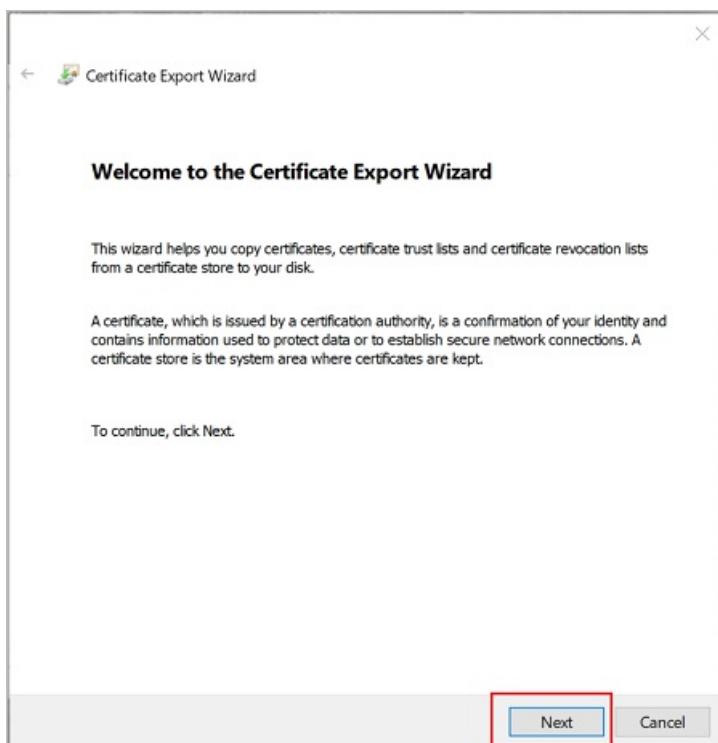
After creating a self-signed root certificate, export the root certificate public key .cer file (not the private key). You will later upload this file to Azure. The following steps help you export the .cer file for your self-signed root certificate:

1. To obtain a .cer file from the certificate, open **Manage user certificates**. Locate the self-signed root certificate, typically in 'Certificates - Current User\Personal\Certificates', and right-click. Click **All Tasks**, and then click **Export**. This opens the **Certificate Export Wizard**. If you can't find the certificate under Current User\Personal\Certificates it could be that you opened Certificates Manager for the local computer certificates (title will be "Certificates - Local Computer" as opposed to "Certificates - Current User"). To open Certificates Manager in current user scope start it from the same PowerShell where the

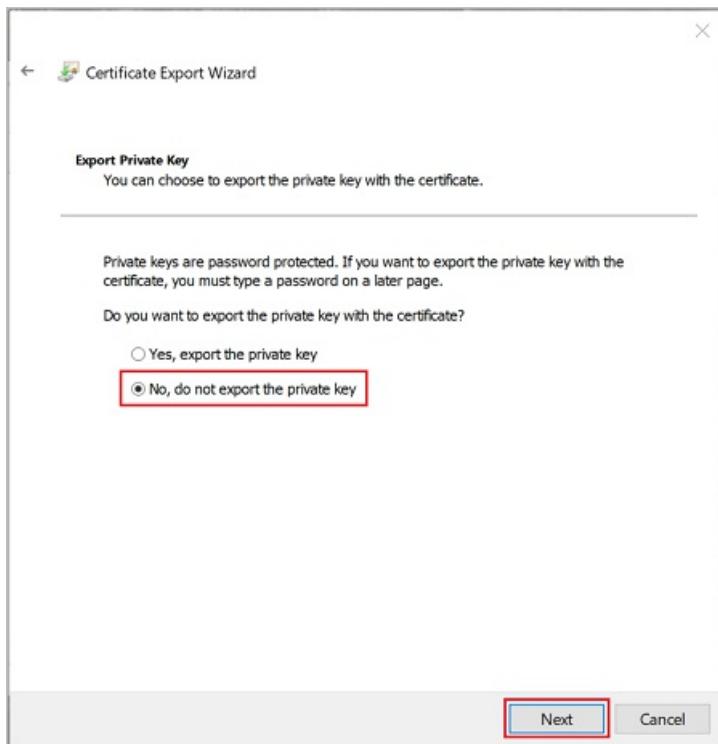
certificates were created by typing `certmgr`.



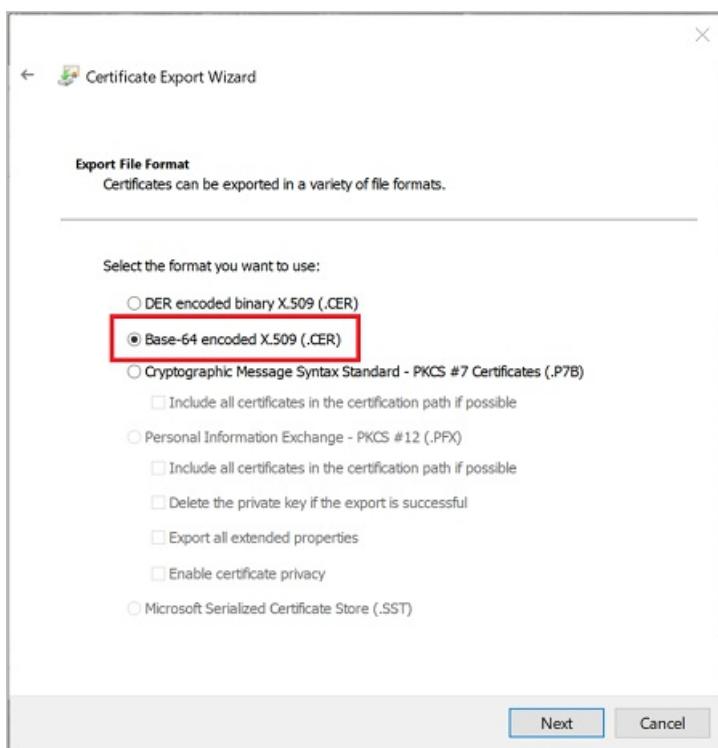
2. In the Wizard, click **Next**.



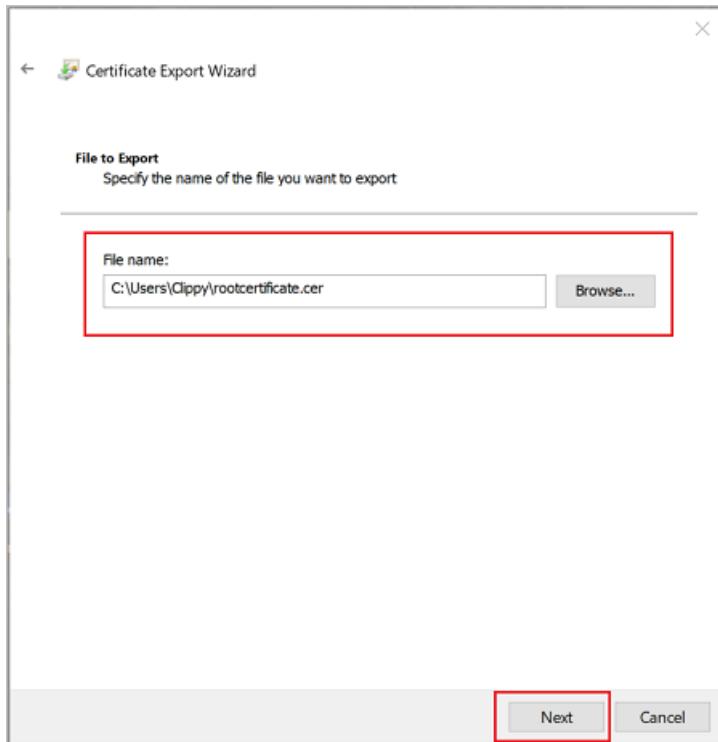
3. Select **No, do not export the private key**, and then click **Next**.



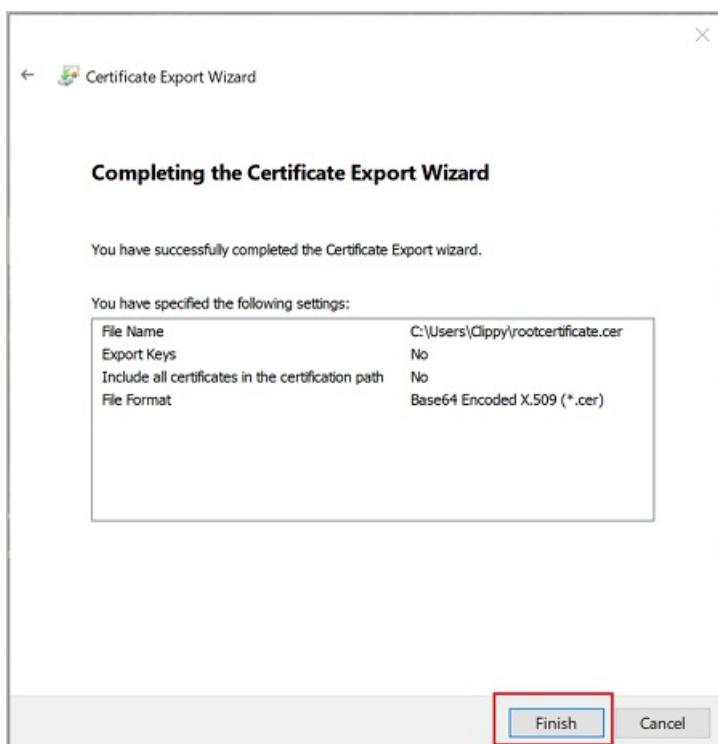
4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**, and then click **Next**.



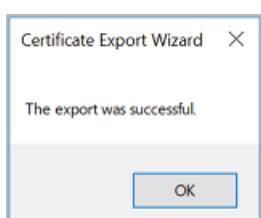
5. For **File to Export**, **Browse** to the location to which you want to export the certificate. For **File name**, name the certificate file. Then, click **Next**.



6. Click **Finish** to export the certificate.



7. Your certificate is successfully exported.



8. The exported certificate looks similar to this:



9. If you open the exported certificate using Notepad, you see something similar to this example. The section in blue contains the information that is uploaded to Azure. If you open your certificate with Notepad and it does not look similar to this, typically this means you did not export it using the Base-64 encoded X.509(.CER) format. Additionally, if you want to use a different text editor, understand that some editors can introduce unintended formatting in the background. This can create problems when uploaded the text from this certificate to Azure.

```
-----BEGIN CERTIFICATE-----
MIIC5zCCAc+gAwIBAgIQRtNVXFGwtIhK4R7RaOak0jANBgkqhkiG9w0BAQsFADAW
MRQwEgYDVQQDDAtQM1NSb290Q2VydDAeFw0xODAyMDIxOTM5MThaFw0xOTAyMDIx
OTU5MThaMBYxFDASBgvNBAMMC1AyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEAvrvt56dNhYGF01/NUS2GcCCQ9mSKQeMCsoMZgyEC8nP
1ZioGeFwd23Thb9+k00k08sPbM4Jm42riyNsmtVNyCviP5pC/V2NyQr/F61+k5X
0GurFxSm/mv6wfOxf/FHvu5PojX7Z5/oEbeYB11GVVPgq6QWSrx33lW1zmD2FeuA
QE46dMPSPHFhnc6P2hfthzs3+tv1R4dg02wr5drVNvr1cVOHqoSbf/dqjV2thzz
ZSSN5kew2G/3H7Mc2ScZD+AYXWRzuiIrKrJSZIuIfJxLpqJaLQTByEU+wT8R8Rnq
GKmyKuUCPoXGYY3TRPmBXgA0800RsKFrXPWp5SiuuQIDAQABozEwlzAOBgNVHQ8B
Af8EBAMCAgQwHQYDVROOBYEFcK6GjsuuTSfxMNz4ATy77DEbyCwMA0GCSqGSIb3
DQEBCwUA4IBAQBzZQCC4SEXqDgR2BL3uj17XD0scR/52U9rVxLorWlZ4Wu4kRA0
EA4IppBNmQep9eaCCqNfc6sbXf4QWjkBv1TBja20rFzt5cTAkwYG6YOaWT1L//fw
u9goi2RihBs6IeBwc621u1Lo0Lw5htCoV0XPSS01mAm5R6//IqyHZRcAK/TffitC
EIYTFcKdavxe9CgY/TtzEMCS7gLARDpHh/nDrxtIeKxlvvUfnOoeXeaSsQwHtumq
GFH3+BgzxEGB8v4oLlQzzbvj+xAb1WKpYqXFbp/ulhd6Ao2qn9sIVuKRkJjBgJ78
o45M2omAFZZaAVQrUa/fprKr3es/6IYrPT8J
-----END CERTIFICATE-----
```

The exported.cer file must be uploaded to Azure. For instructions, see [Configure a Point-to-Site connection](#). To add an additional trusted root certificate, see [this section](#) of the article.

Export the self-signed certificate and private key to store it (optional)

You may want to export the self-signed root certificate and store it safely. If need be, you can later install it on another computer and generate more client certificates, or export another .cer file. To export the self-signed root certificate as a .pfx, select the root certificate and use the same steps as described in [Export a client certificate](#).

Create and install client certificates

You don't install the self-signed certificate directly on the client computer. You need to generate a client certificate from the self-signed certificate. You then export and install the client certificate to the client computer. The following steps are not deployment-model specific. They are valid for both Resource Manager and classic.

Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. You

generate a client certificate from the self-signed root certificate, and then export and install the client certificate. If the client certificate is not installed, authentication fails.

The following steps walk you through generating a client certificate from a self-signed root certificate. You may generate multiple client certificates from the same root certificate. When you generate client certificates using the steps below, the client certificate is automatically installed on the computer that you used to generate the certificate. If you want to install a client certificate on another client computer, you can export the certificate.

1. On the same computer that you used to create the self-signed certificate, open a command prompt as administrator.
2. Modify and run the sample to generate a client certificate.
 - Change "P2SRootCert" to the name of the self-signed root that you are generating the client certificate from. Make sure you are using the name of the root certificate, which is whatever the 'CN=' value was that you specified when you created the self-signed root.
 - Change P2SChildCert to the name you want to generate a client certificate to be.

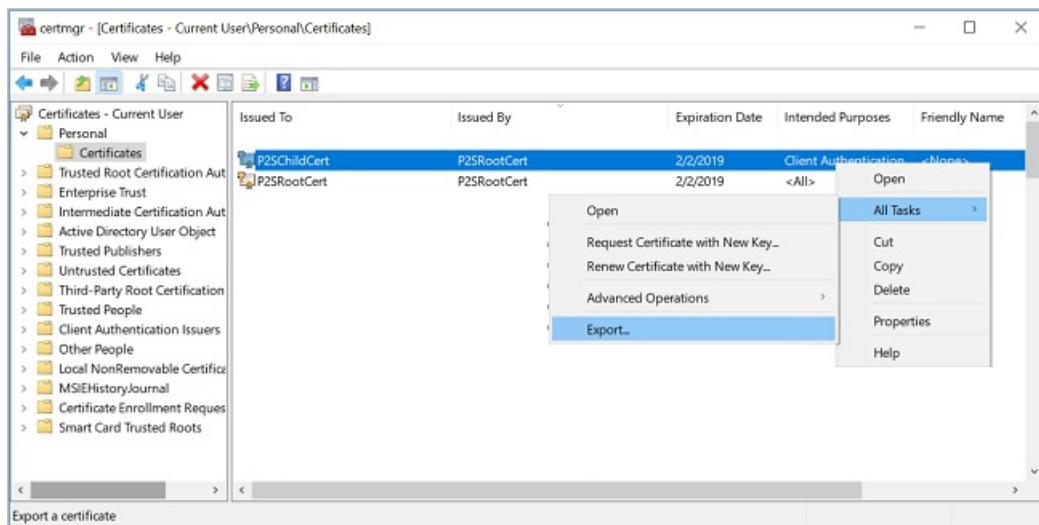
If you run the following example without modifying it, the result is a client certificate named P2SChildcert in your Personal certificate store that was generated from root certificate P2SRootCert.

```
makecert.exe -n "CN=P2SChildCert" -pe -sky exchange -m 96 -ss My -in "P2SRootCert" -is my -a sha256
```

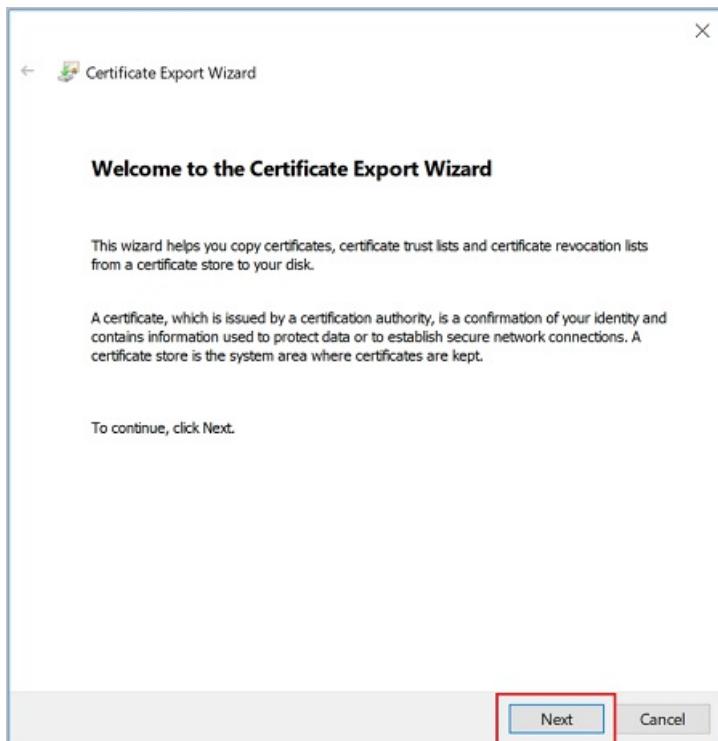
Export a client certificate

When you generate a client certificate, it's automatically installed on the computer that you used to generate it. If you want to install the client certificate on another client computer, you need to export the client certificate that you generated.

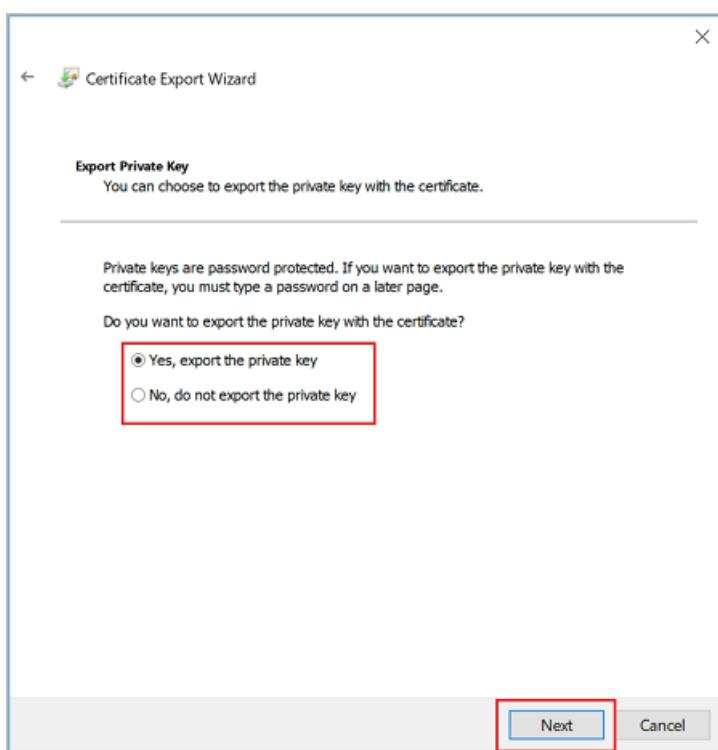
1. To export a client certificate, open **Manage user certificates**. The client certificates that you generated are, by default, located in 'Certificates - Current User\Personal\Certificates'. Right-click the client certificate that you want to export, click **all tasks**, and then click **Export** to open the **Certificate Export Wizard**.



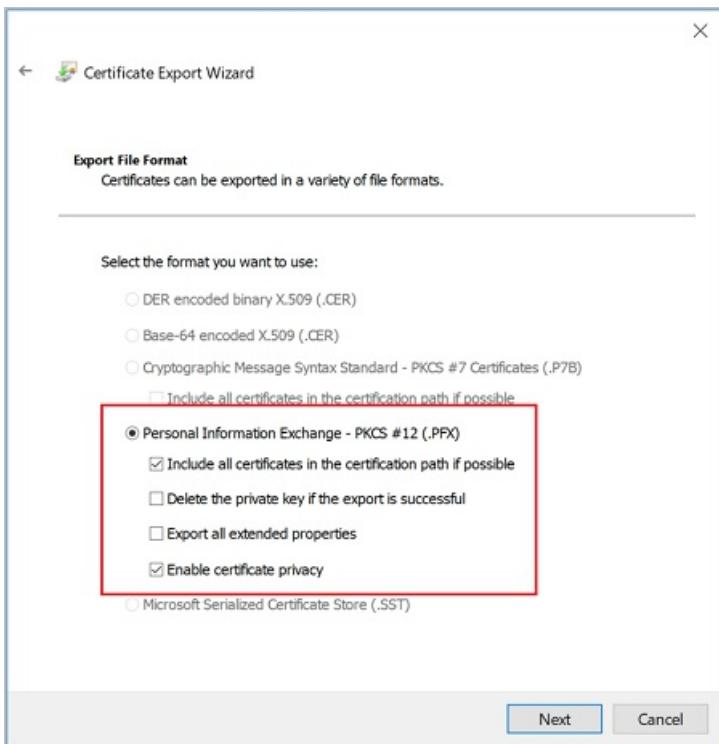
2. In the Certificate Export Wizard, click **Next** to continue.



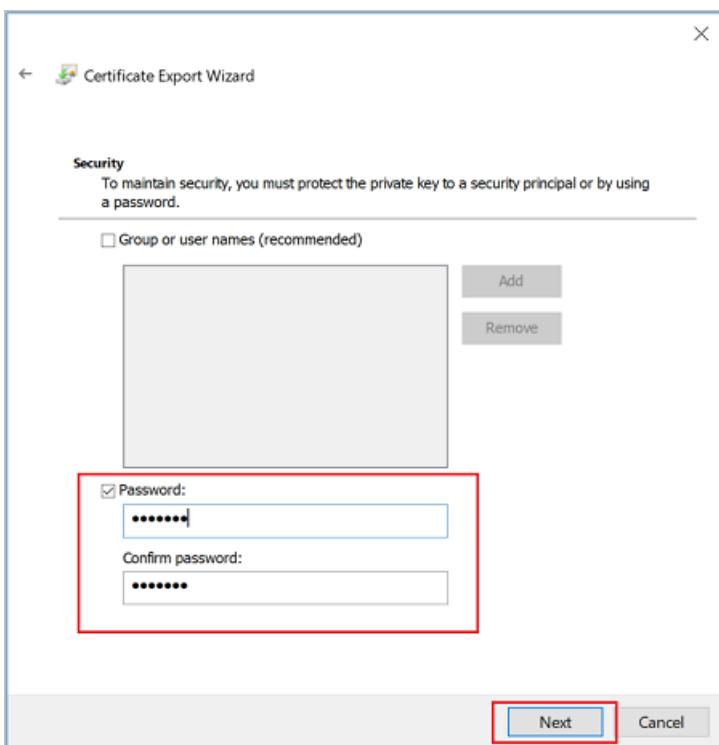
3. Select **Yes, export the private key**, and then click **Next**.



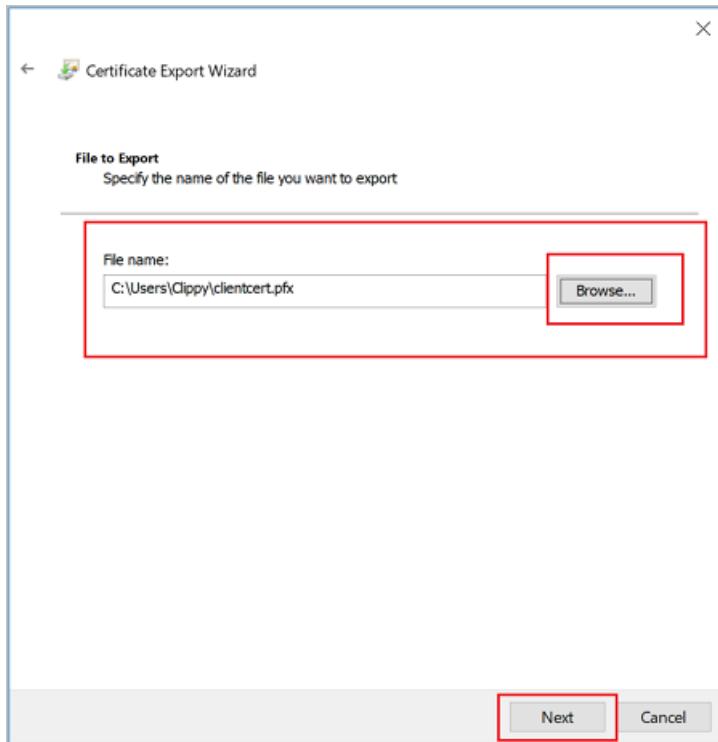
4. On the **Export File Format** page, leave the defaults selected. Make sure that **Include all certificates in the certification path if possible** is selected. This setting additionally exports the root certificate information that is required for successful client authentication. Without it, client authentication fails because the client doesn't have the trusted root certificate. Then, click **Next**.



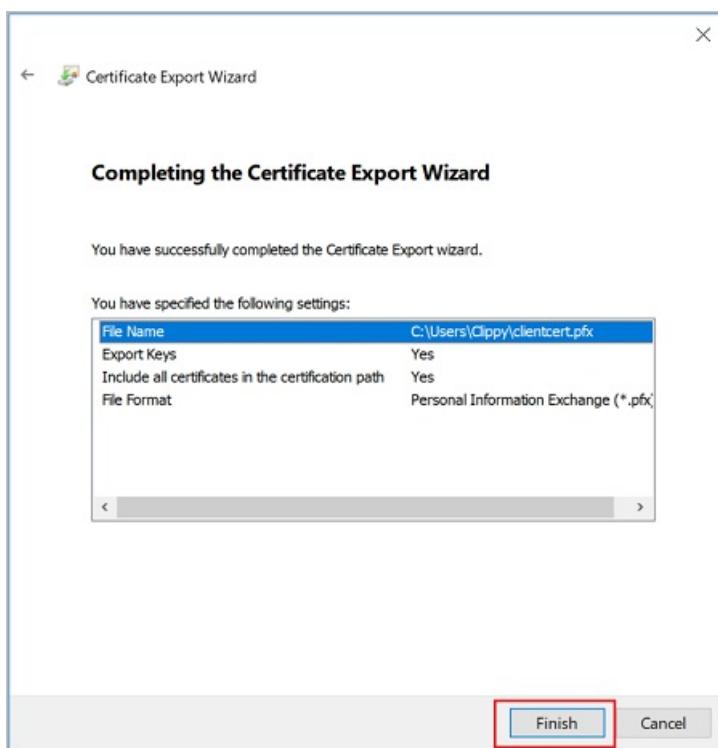
5. On the **Security** page, you must protect the private key. If you select to use a password, make sure to record or remember the password that you set for this certificate. Then, click **Next**.



6. On the **File to Export**, **Browse** to the location to which you want to export the certificate. For **File name**, name the certificate file. Then, click **Next**.



7. Click **Finish** to export the certificate.



Install an exported client certificate

To install a client certificate, see [Install a client certificate](#).

Next steps

Continue with your Point-to-Site configuration.

- For **Resource Manager** deployment model steps, see [Configure P2S using native Azure certificate authentication](#).
- For **classic** deployment model steps, see [Configure a Point-to-Site VPN connection to a VNet \(classic\)](#).

For P2S troubleshooting information, [Troubleshooting Azure point-to-site connections](#).

Generate and export certificates for Point-to-Site using Linux strongSwan CLI

9/10/2018 • 2 minutes to read • [Edit Online](#)

Point-to-Site connections use certificates to authenticate. This article shows you how to create a self-signed root certificate and generate client certificates using the Linux CLI and strongSwan. If you are looking for different certificate instructions, see the [PowerShell](#) or [MakeCert](#) articles.

NOTE

The steps in this article require strongSwan.

The computer configuration used for the steps for this article was the following:

Computer	Ubuntu Server 16.04 ID_LIKE=debian PRETTY_NAME="Ubuntu 16.04.4 LTS" VERSION_ID="16.04"
Dependencies	apt-get install strongswan-ikev2 strongswan-plugin-eap-tls apt-get install libstrongswan-standard-plugins

Install strongSwan

1. `apt-get install strongswan-ikev2 strongswan-plugin-eap-tls`
2. `apt-get install libstrongswan-standard-plugins`

For information about how to install strongSwan using the GUI, see the steps in the [Client configuration](#) article.

Generate keys and certificate

1. Generate the CA certificate.

```
ipsec pki --gen --outform pem > caKey.pem
ipsec pki --self --in caKey.pem --dn "CN=VPN CA" --ca --outform pem > caCert.pem
```

2. Print the CA certificate in base64 format. This is the format that is supported by Azure. You will later upload this to Azure as part of your P2S configuration.

```
openssl x509 -in caCert.pem -outform der | base64 -w0 ; echo
```

3. Generate the user certificate.

```
export PASSWORD="password"
export USERNAME="client"

ipsec pki --gen --outform pem > "${USERNAME}Key.pem"
ipsec pki --pub --in "${USERNAME}Key.pem" | ipsec pki --issue --cacert caCert.pem --cakey caKey.pem --
dn "CN=${USERNAME}" --san "${USERNAME}" --flag clientAuth --outform pem > "${USERNAME}Cert.pem"
```

4. Generate a p12 bundle containing the user certificate. This bundle will be used in the next steps when working with the [Client configuration files](#).

```
openssl pkcs12 -in "${USERNAME}Cert.pem" -inkey "${USERNAME}Key.pem" -certfile caCert.pem -export -out
"${USERNAME}.p12" -password "pass:${PASSWORD}"
```

Next steps

Continue with your Point-to-Site configuration to [Create and install VPN client configuration files](#).

Install client certificates for P2S certificate authentication connections

9/10/2018 • 2 minutes to read • [Edit Online](#)

All clients that connect to a virtual network using Point-to-Site Azure certificate authentication require a client certificate. This article helps you install a client certificate that is used for authentication when connecting to a VNet using P2S.

Acquire a client certificate

No matter what client operating system you want to connect from, you must always have a client certificate. You can generate a client certificate from either a root certificate that was generated using an Enterprise CA solution, or a self-signed root certificate. See the [PowerShell](#), [MakeCert](#), or [Linux](#) instructions for steps to generate a client certificate.

Windows

If you want to create a P2S connection from a client computer other than the one you used to generate the client certificates, you need to install a client certificate. When installing a client certificate, you need the password that was created when the client certificate was exported.

1. Locate and copy the .pfx file to the client computer. On the client computer, double-click the .pfx file to install. Leave the **Store Location** as **Current User**, and then click **Next**.
2. On the **File** to import page, don't make any changes. Click **Next**.
3. On the **Private key protection** page, input the password for the certificate, or verify that the security principal is correct, then click **Next**.
4. On the **Certificate Store** page, leave the default location, and then click **Next**.
5. Click **Finish**. On the **Security Warning** for the certificate installation, click **Yes**. You can feel comfortable clicking 'Yes' because you generated the certificate. The certificate is now successfully imported.

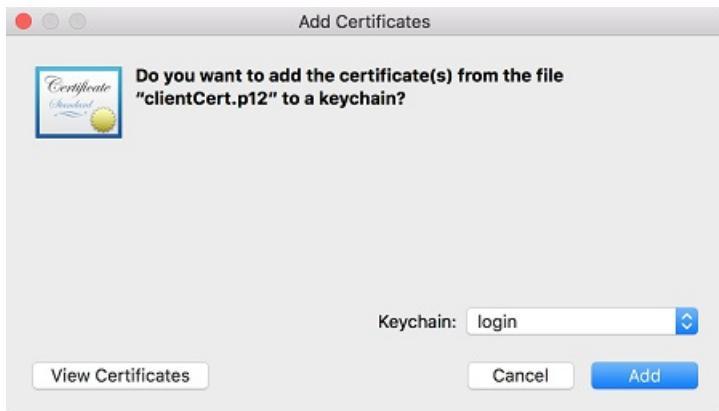
Mac

NOTE

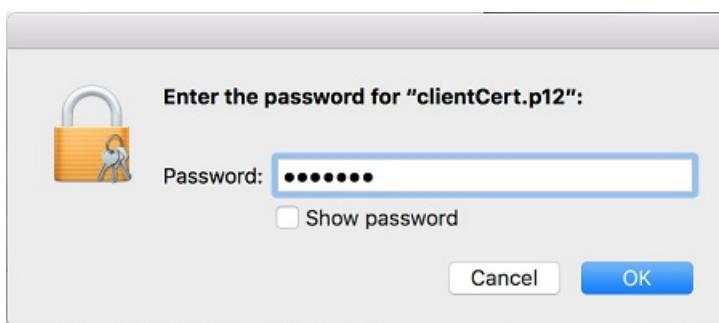
Mac VPN clients are supported for the Resource Manager deployment model only. They are not supported for the classic deployment model.

When installing a client certificate, you need the password that was created when the client certificate was exported.

1. Locate the .pfx certificate file and copy it to your Mac. You can get the certificate to the Mac in several ways, for example, you can email the certificate file.
2. After the certificate copied to the Mac, double-click the certificate to open the **Add Certificates** box, then click **Add** to begin the install.



3. Enter the password that you created when the client certificate was exported. The password protects the private key of the certificate. Click **OK** to complete the installation.



Linux

The Linux client certificate is installed on the client as part of the client configuration. See [Client configuration - Linux](#) for instructions.

Next steps

Continue with the Point-to-Site configuration steps to [Create and install VPN client configuration files](#).

Create and install VPN client configuration files for native Azure certificate authentication P2S configurations

9/10/2018 • 8 minutes to read • [Edit Online](#)

VPN client configuration files are contained in a zip file. Configuration files provide the settings required for a native Windows, Mac IKEv2 VPN, or Linux clients to connect to a VNet over Point-to-Site connections that use native Azure certificate authentication. For more information about Point-to-Site connections, see [About Point-to-Site VPN](#).

IMPORTANT

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. Only point-to-site connections are impacted; site-to-site connections will not be affected. If you're using TLS for point-to-site VPNs on Windows 10 clients, you don't need to take any action. If you are using TLS for point-to-site connections on Windows 7 and Windows 8 clients, see the [VPN Gateway FAQ](#) for update instructions.

NOTE

Client configuration files are specific to the VPN configuration for the VNet. If there are any changes to the Point-to-Site VPN configuration after you generate the VPN client configuration files, such as the VPN protocol type or authentication type, be sure to generate new VPN client configuration files for your user devices.

Generate VPN client configuration files

Before you begin, make sure that all connecting users have a valid certificate installed on the user's device. For more information about installing a client certificate, see [Install a client certificate](#).

You can generate client configuration files using PowerShell, or by using the Azure portal. Either method returns the same zip file. Unzip the file to view the following folders:

- **WindowsAmd64** and **WindowsX86**, which contain the Windows 32-bit and 64-bit installer packages, respectively. The **WindowsAmd64** installer package is for all supported 64-bit Windows clients, not just Amd.
- **Generic**, which contains general information used to create your own VPN client configuration. The Generic folder is provided if IKEv2 or SSTP+IKEv2 was configured on the gateway. If only SSTP is configured, then the Generic folder is not present.

Generate files using the Azure portal

1. In the Azure portal, navigate to the virtual network gateway for the virtual network that you want to connect to.
2. On the virtual network gateway page, click **Point-to-site configuration**.
3. At the top of the Point-to-site configuration page, click **Download VPN client**. It takes a few minutes for the client configuration package to generate.
4. Your browser indicates that a client configuration zip file is available. It is named the same name as your gateway. Unzip the file to view the folders.

Generate files using PowerShell

1. When generating VPN client configuration files, the value for '-AuthenticationMethod' is 'EapTls'. Generate

the VPN client configuration files using the following command:

```
$profile=New-AzureRmVpnClientConfiguration -ResourceGroupName "TestRG" -Name "VNet1GW" -  
AuthenticationMethod "EapTls"  
  
$profile.VPNProfileSASUrl
```

2. Copy the URL to your browser to download the zip file, then unzip the file to view the folders.

Windows

You can use the same VPN client configuration package on each Windows client computer, as long as the version matches the architecture for the client. For the list of client operating systems that are supported, see the Point-to-Site section of the [VPN Gateway FAQ](#).

NOTE

You must have Administrator rights on the Windows client computer from which you want to connect.

Use the following steps to configure the native Windows VPN client for certificate authentication:

1. Select the VPN client configuration files that correspond to the architecture of the Windows computer. For a 64-bit processor architecture, choose the 'VpnClientSetupAmd64' installer package. For a 32-bit processor architecture, choose the 'VpnClientSetupX86' installer package.
2. Double-click the package to install it. If you see a SmartScreen popup, click **More info**, then **Run anyway**.
3. On the client computer, navigate to **Network Settings** and click **VPN**. The VPN connection shows the name of the virtual network that it connects to.
4. Before you attempt to connect, verify that you have installed a client certificate on the client computer. A client certificate is required for authentication when using the native Azure certificate authentication type. For more information about generating certificates, see [Generate Certificates](#). For information about how to install a client certificate, see [Install a client certificate](#).

Mac (OS X)

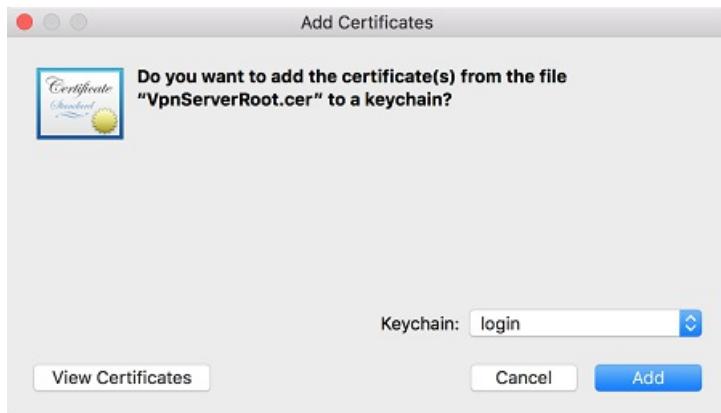
You have to manually configure the native IKEv2 VPN client on every Mac that will connect to Azure. Azure does not provide mobileconfig file for native Azure certificate authentication. The **Generic** contains all of the information that you need for configuration. If you don't see the Generic folder in your download, it's likely that IKEv2 was not selected as a tunnel type. Once IKEv2 is selected, generate the zip file again to retrieve the Generic folder.

The Generic folder contains the following files:

- **VpnSettings.xml**, which contains important settings like server address and tunnel type.
- **VpnServerRoot.cer**, which contains the root certificate required to validate the Azure VPN Gateway during P2S connection setup.

Use the following steps to configure the native VPN client on Mac for certificate authentication. You have to complete these steps on every Mac that will connect to Azure:

1. Import the **VpnServerRoot** root certificate to your Mac. This can be done by copying the file over to your Mac and double-clicking on it.
Click **Add** to import.

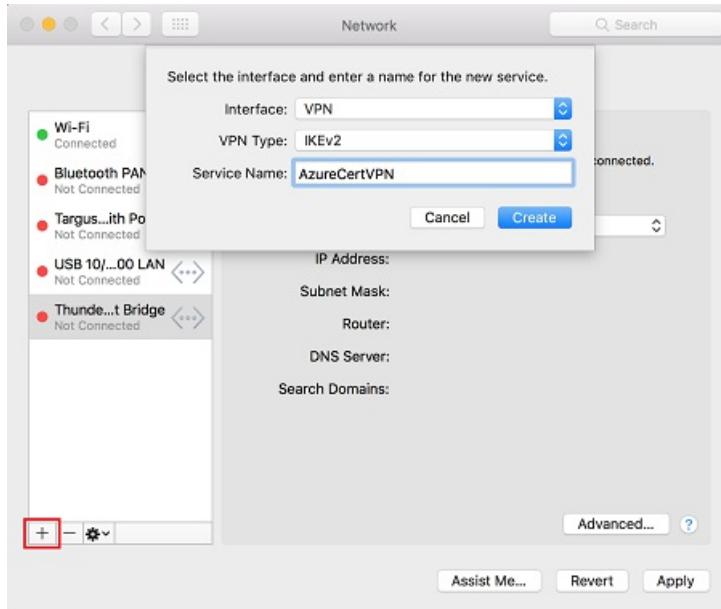


NOTE

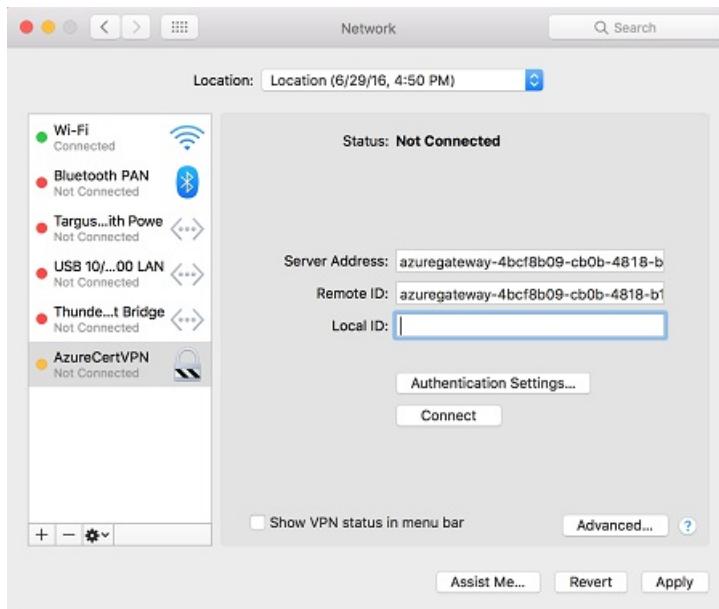
Double-clicking on the certificate may not display the **Add** dialog, but the certificate is installed in the correct store. You can check for the certificate in the login keychain under the certificates category.

2. Verify that you have installed a client certificate that was issued by the root certificate that you uploaded to Azure when you configured your P2S settings. This is different from the VPNServerRoot that you installed in the previous step. The client certificate is used for authentication and is required. For more information about generating certificates, see [Generate Certificates](#). For information about how to install a client certificate, see [Install a client certificate](#).
3. Open the **Network** dialog under **Network Preferences** and click '+' to create a new VPN client connection profile for a P2S connection to the Azure VNet.

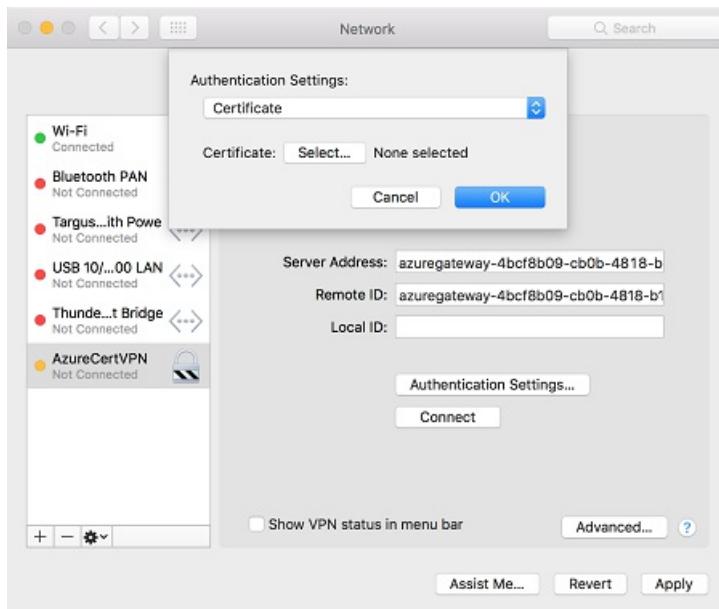
The **Interface** value is 'VPN' and **VPN Type** value is 'IKEv2'. Specify a name for the profile in the **Service Name** field, then click **Create** to create the VPN client connection profile.



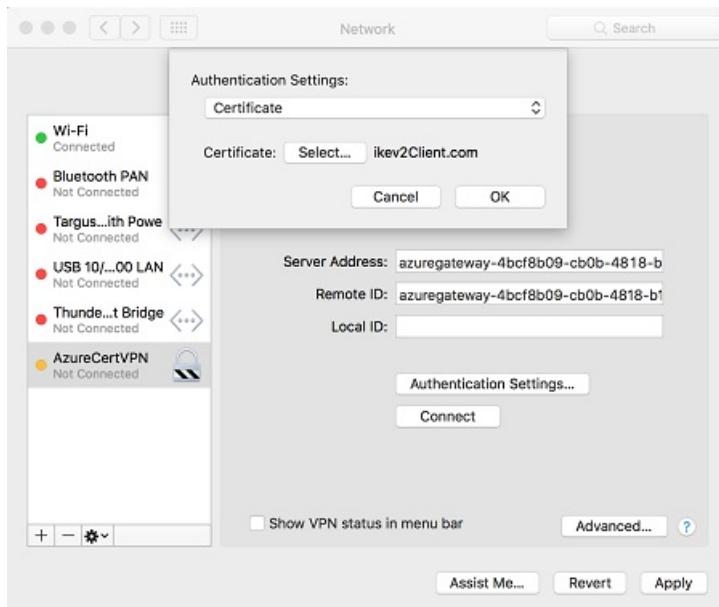
4. In the **Generic** folder, from the **VpnSettings.xml** file, copy the **VpnServer** tag value. Paste this value in the **Server Address** and **Remote ID** fields of the profile.



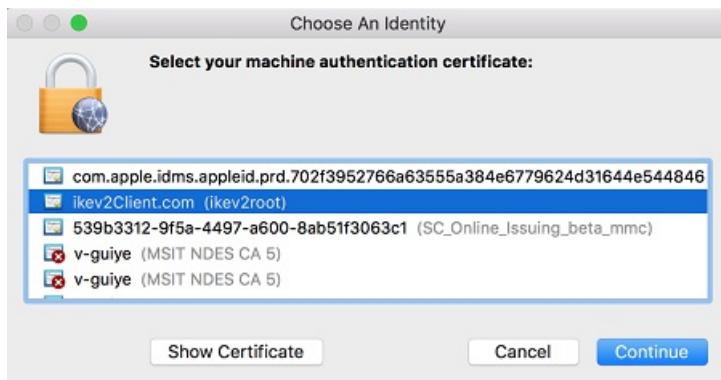
5. Click **Authentication Settings** and select **Certificate**.



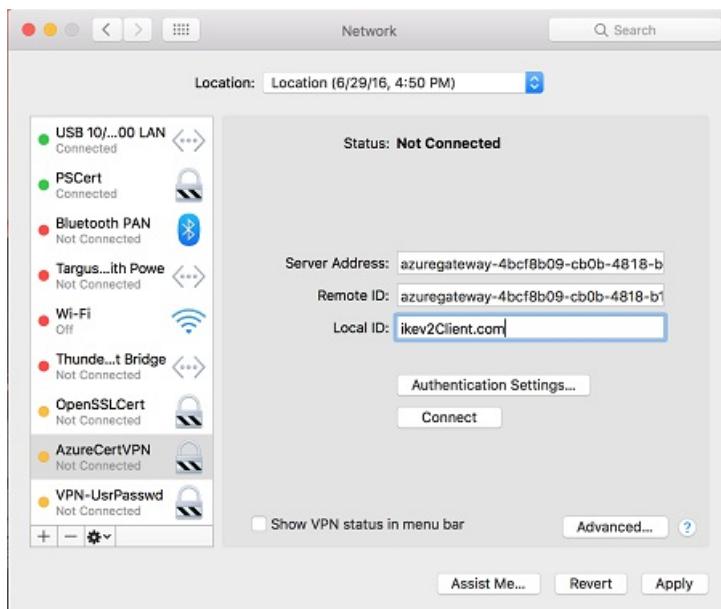
6. Click **Select...** to choose the client certificate that you want to use for authentication. This is the certificate that you installed in Step 2.



7. **Choose An Identity** displays a list of certificates for you to choose from. Select the proper certificate, then click **Continue**.



8. In the **Local ID** field, specify the name of the certificate (from Step 6). In this example, it is "ikev2Client.com". Then, click **Apply** button to save the changes.



9. On the **Network** dialog, click **Apply** to save all changes. Then, click **Connect** to start the P2S connection to the Azure VNet.

Linux (strongSwan GUI)

Extract the key and certificate

For strongSwan, you need to extract the key and the cert from the client certificate (.pfx file) and save them to individual .pem files. Follow the steps below:

1. Download and install OpenSSL from [OpenSSL](#).
2. Open a command-line window and change to the directory where you installed OpenSSL, for example, 'c:\OpenSSL-Win64\bin'.
3. Run the following command to extract the private key and save it to a new file called 'privatekey.pem' from your client certificate:

```
C:\ OpenSSL-Win64\bin> openssl pkcs12 -in clientcert.pfx -nocerts -out privatekey.pem -nodes
```

4. Now run the following command to extract the public cert and save it to a new file:

```
C:\ OpenSSL-Win64\bin> openssl pkcs12 -in clientcert.pfx -nokeys -out publiccert.pem -nodes
```

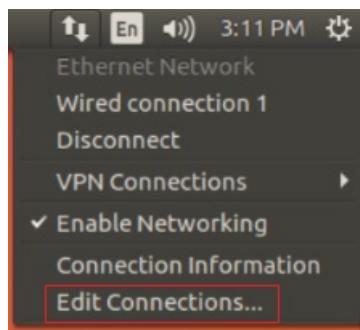
Install and configure

The following instructions were created through strongSwan 5.5.1 on Ubuntu 17.0.4. Ubuntu 16.0.10 does not support strongSwan GUI. If you want to use Ubuntu 16.0.10, you will have to use the [command line](#). The examples below may not match screens that you see, depending on your version of Linux and strongSwan.

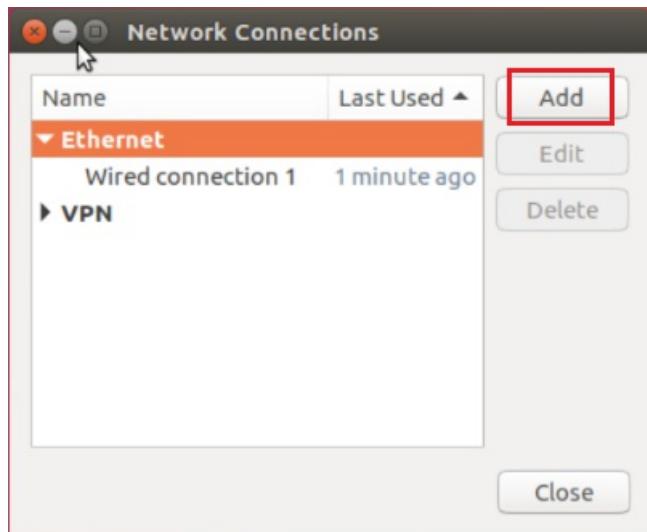
1. Open the **Terminal** to install **strongSwan** and its Network Manager by running the command in the example. If you receive an error that's related to *libcharon-extra-plugins*, replace it with 'strongswan-plugin-eap-mschapv2'.

```
sudo apt-get install strongswan libcharon-extra-plugins moreutils iptables-persistent network-manager-strongswan
```

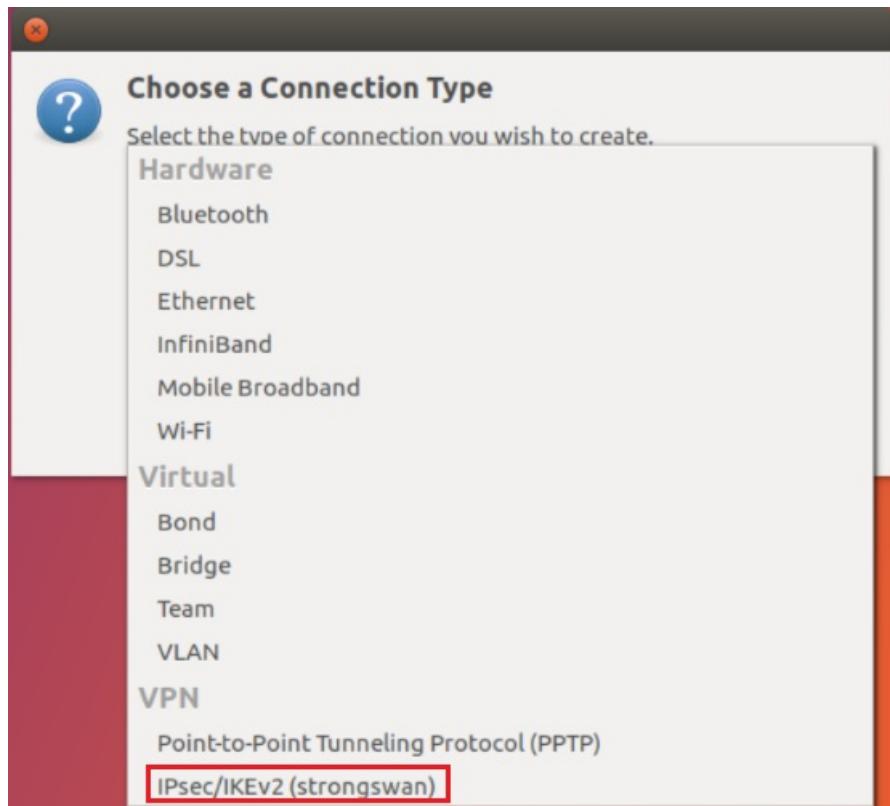
2. Select the **Network Manager** icon (up-arrow/down-arrow), then select **Edit Connections**.



3. Click the **Add** button to create a new connection.



4. Select **IPsec/IKEv2 (strongswan)** from the drop-down menu, and then click **Create**. You can rename your connection in this step.



5. Open the **VpnSettings.xml** file from the **Generic** folder contained in the downloaded client configuration files. Find the tag called **VpnServer** and copy the name, beginning with 'azuregateway' and ending with '.cloudapp.net'.

```
<VpnProfile>
  <VpnServer>azuregateway-<UUID>.cloudapp.net</VpnServer>
  <VpnType>IkeV2,SSTP</VpnType>
<snip>|
```

6. Paste this name into the **Address** field of your new VPN connection in the **Gateway** section. Next, select the folder icon at the end of the **Certificate** field, browse to the **Generic** folder, and select the **VpnServerRoot** file.
7. In the **Client** section of the connection, for **Authentication**, select **Certificate/private key**. For **Certificate** and **Private key**, choose the certificate and the private key that were created earlier. In **Options**, select **Request an inner IP address**. Then, click **Add**.

Cancel **VPN 1 VPN** Apply

Details Identity IPv4 IPv6

Name **VPN 1**

Gateway

Address **azuregateway-8dc85903-7a76-4f97-a2ec-833f20**

Certificate **VpnServerRoot.cer**

Client

Authentication **Certificate/private key**

Certificate **publiccert.pem**

Private key **privatekey.pem**

Username

Password

Options

Request an inner IP address
 Enforce UDP encapsulation
 Use IP compression

Cipher proposals

Enable custom proposals

IKE

ESP

- Click the **Network Manager** icon (up-arrow/down-arrow) and hover over **VPN Connections**. You see the VPN connection that you created. Click to initiate the connection.

Linux (strongSwan CLI)

Install strongSwan

You can use the following CLI commands, or use the strongSwan steps in the [GUI](#) to install strongSwan.

- `apt-get install strongswan-ikev2 strongswan-plugin-eap-tls`
- `apt-get install libstrongswan-standard-plugins`

Install and configure

- Download the VPNClient package from Azure portal.
- Extract the File.
- From the **Generic** folder, copy or move the VpnServerRoot.cer to /etc/ipsec.d/cacerts.
- From the **Generic** folder, copy or move cp client.p12 to /etc/ipsec.d/private/.
- Open VpnSettings.xml file and copy the value. You will use this value in the next step.
- Adjust the values in the example below, then add the example to the /etc/ipsec.conf configuration.

```
conn azure
keyexchange=ikev2
type=tunnel
leftfirewall=yes
left=%any
leftauth=eap-tls
leftid=%client # use the DNS alternative name prefixed with the %
right= Enter the VPN Server value here# Azure VPN gateway address
rightid=%Enter the VPN Server value here# Azure VPN gateway address, prefixed with %
rightsubnet=0.0.0.0/0
leftsourceip=%config
auto=add
```

7. Add the following to `/etc/ipsec.secrets`.

```
: P12 client.p12 'password' # key filename inside /etc/ipsec.d/private directory
```

8. Run the following commands:

```
# ipsec restart
# ipsec up azure
```

Next steps

Return to the article to [complete your P2S configuration](#).

To troubleshoot P2S connections, see the following articles:

- [Troubleshooting Azure point-to-site connections](#)
- [Troubleshoot VPN connections from Mac OS X VPN clients](#)

Configure a Point-to-Site connection to a VNet using RADIUS authentication: PowerShell

6/11/2018 • 18 minutes to read • [Edit Online](#)

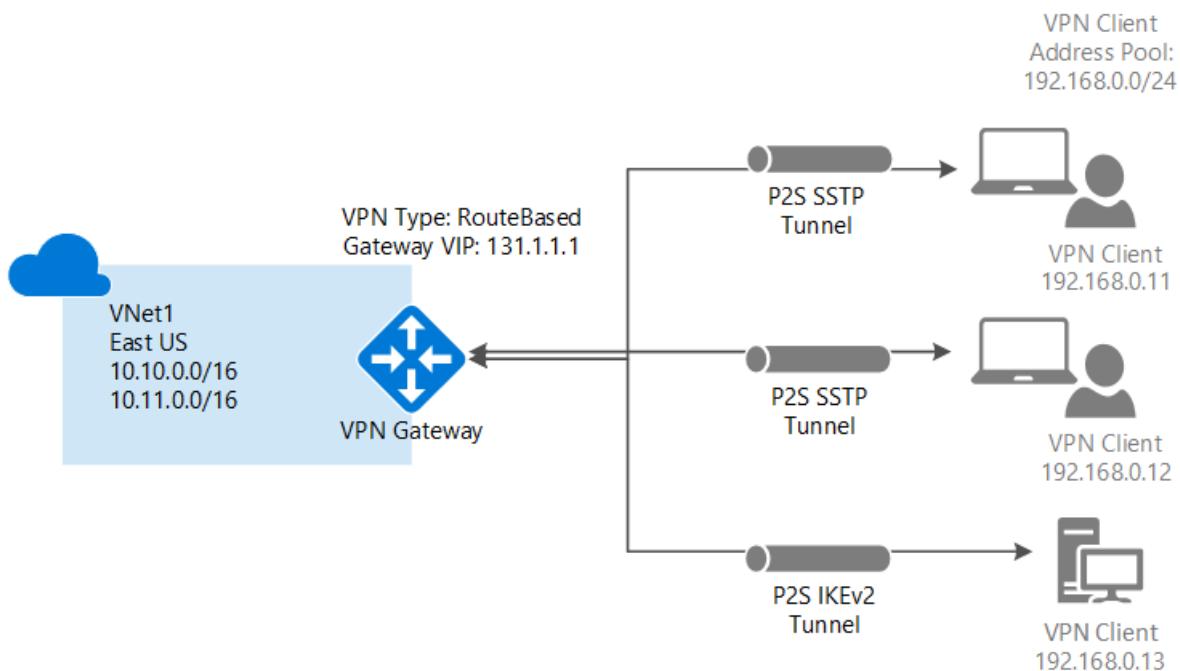
This article shows you how to create a VNet with a Point-to-Site connection that uses RADIUS authentication. This configuration is only available for the Resource Manager deployment model.

A Point-to-Site (P2S) VPN gateway lets you create a secure connection to your virtual network from an individual client computer. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location, such as when you are telecommuting from home or a conference. A P2S VPN is also a useful solution to use instead of a Site-to-Site VPN when you have only a few clients that need to connect to a VNet.

A P2S VPN connection is started from Windows and Mac devices. Connecting clients can use the following authentication methods:

- RADIUS server
- VPN Gateway native certificate authentication

This article helps you configure a P2S configuration with authentication using RADIUS server. If you want to authenticate using generated certificates and VPN gateway native certificate authentication instead, see [Configure a Point-to-Site connection to a VNet using VPN gateway native certificate authentication](#).



Point-to-Site connections do not require a VPN device or a public-facing IP address. P2S creates the VPN connection over either SSTP (Secure Socket Tunneling Protocol), or IKEv2.

- SSTP is an SSL-based VPN tunnel that is supported only on Windows client platforms. It can penetrate firewalls, which makes it an ideal option to connect to Azure from anywhere. On the server side, we support SSTP versions 1.0, 1.1, and 1.2. The client decides which version to use. For Windows 8.1 and above, SSTP uses 1.2 by default.
- IKEv2 VPN, a standards-based IPsec VPN solution. IKEv2 VPN can be used to connect from Mac devices (OSX versions 10.11 and above).

P2S connections require the following:

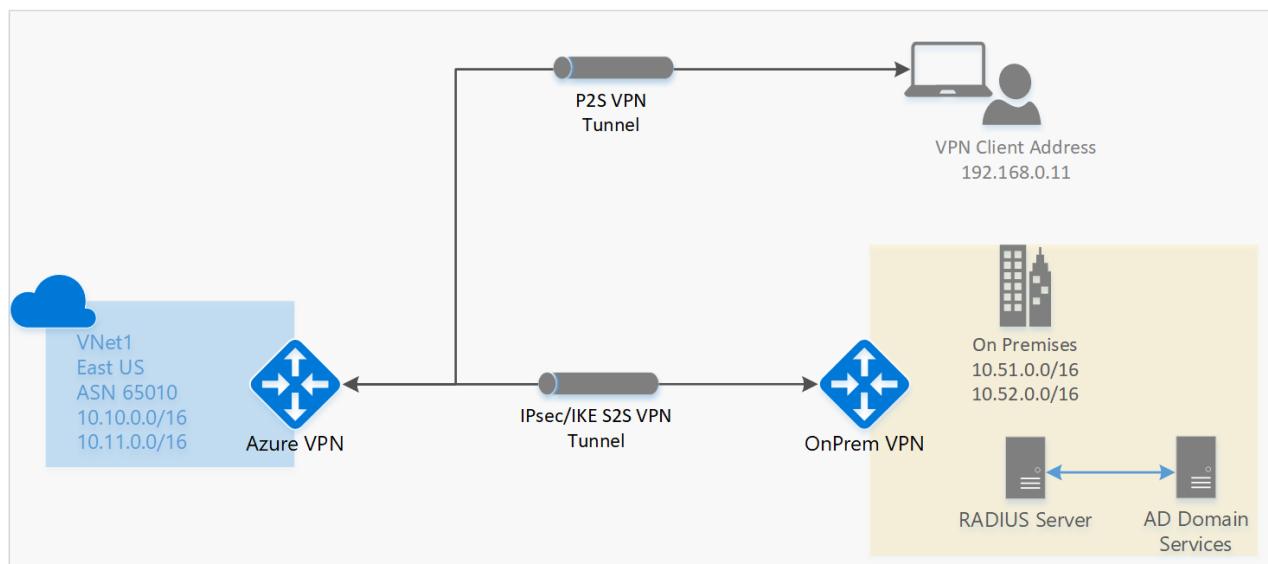
- A RouteBased VPN gateway.
- A RADIUS server to handle user authentication. The RADIUS server can be deployed on-premises, or in the Azure VNet.
- A VPN client configuration package for the Windows devices that will connect to the VNet. A VPN client configuration package provides the settings required for a VPN client to connect over P2S.

About Active Directory (AD) Domain Authentication for P2S VPNs

AD Domain authentication allows users to log in to Azure using their organization domain credentials. It requires a RADIUS server that integrates with the AD server. Organizations can also leverage their existing RADIUS deployment.

The RADIUS server can reside on-premises, or in your Azure VNet. During authentication, the VPN gateway acts as a pass-through and forwards authentication messages back and forth between the RADIUS server and the connecting device. It's important for the VPN gateway to be able to reach the RADIUS server. If the RADIUS server is located on-premises, then a VPN Site-to-Site connection from Azure to the on-premises site is required.

Apart from Active Directory, a RADIUS server can also integrate with other external identity systems. This opens up plenty of authentication options for Point-to-Site VPNs, including MFA options. Check your RADIUS server vendor documentation to get the list of identity systems it integrates with.



IMPORTANT

Only a VPN Site-to-Site connection can be used for connecting to a RADIUS server on-premises. An ExpressRoute connection cannot be used.

Before beginning

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- Install the latest version of the Azure Resource Manager PowerShell cmdlets. For more information about installing PowerShell cmdlets, see [How to install and configure Azure PowerShell](#).

Log in

Before beginning this configuration, you must sign in to your Azure account. The cmdlet prompts you for the sign-in credentials for your Azure account. After signing in, it downloads your account settings so they are available to

Azure PowerShell. For more information, see [Using Windows PowerShell with Resource Manager](#).

To sign in, open your PowerShell console with elevated privileges, and connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Example values

You can use the example values to create a test environment, or refer to these values to better understand the examples in this article. You can either use the steps as a walk-through and use the values without changing them, or change them to reflect your environment.

- **Name: VNet1**
- **Address space: 192.168.0.0/16 and 10.254.0.0/16**

For this example, we use more than one address space to illustrate that this configuration works with multiple address spaces. However, multiple address spaces are not required for this configuration.
- **Subnet name: FrontEnd**
 - **Subnet address range: 192.168.1.0/24**
- **Subnet name: BackEnd**
 - **Subnet address range: 10.254.1.0/24**
- **Subnet name: GatewaySubnet**

The Subnet name *GatewaySubnet* is mandatory for the VPN gateway to work.

 - **GatewaySubnet address range: 192.168.200.0/24**
- **VPN client address pool: 172.16.201.0/24**

VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the VPN client address pool.
- **Subscription:** If you have more than one subscription, verify that you are using the correct one.
- **Resource Group: TestRG**
- **Location: East US**
- **DNS Server: IP address** of the DNS server that you want to use for name resolution for your VNet.
(optional)
- **GW Name: Vnet1GW**
- **Public IP name: VNet1GWPiP**
- **VpnType: RouteBased**

1. Create the resource group, VNet, and Public IP address

The following steps create a resource group and a virtual network in the resource group with three subnets. 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;

1. Create a resource group.

```
New-AzureRmResourceGroup -Name "TestRG" -Location "East US"
```

2. Create the subnet configurations for the virtual network, naming them *FrontEnd*, *BackEnd*, and *GatewaySubnet*. These prefixes must be part of the VNet address space that you declared.

```
$fesub = New-AzureRmVirtualNetworkSubnetConfig -Name "FrontEnd" -AddressPrefix "192.168.1.0/24"  
$besub = New-AzureRmVirtualNetworkSubnetConfig -Name "Backend" -AddressPrefix "10.254.1.0/24"  
$gwsu = New-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix "192.168.200.0/24"
```

3. Create the virtual network.

In this example, the `-DnsServer` server parameter is optional. Specifying a value does not create a new DNS server. The DNS server IP address that you specify should be a DNS server that can resolve the names for the resources you are connecting to from your VNet. For this example, we used a private IP address, but it is likely that this is not the IP address of your DNS server. Be sure to use your own values. The value you specify is used by the resources that you deploy to the VNet, not by the P2S connection.

```
New-AzureRmVirtualNetwork -Name "VNet1" -ResourceGroupName "TestRG" -Location "East US" -AddressPrefix  
"192.168.0.0/16", "10.254.0.0/16" -Subnet $fesub, $besub, $gwsu -DnsServer 10.2.1.3
```

4. A VPN gateway must have a Public IP address. You first request the IP address resource, and then refer to it when creating your virtual network gateway. The IP address is dynamically assigned to the resource when the VPN gateway is created. VPN Gateway currently only supports *Dynamic* Public IP address allocation. You cannot request a Static Public IP address assignment. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.

Specify the variables to request a dynamically assigned Public IP address.

```
$vnet = Get-AzureRmVirtualNetwork -Name "VNet1" -ResourceGroupName "TestRG"  
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet  
$pip = New-AzureRmPublicIpAddress -Name "VNet1GWPIP" -ResourceGroupName "TestRG" -Location "East US" -  
AllocationMethod Dynamic  
$ipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name "gwipconf" -Subnet $subnet -PublicIpAddress  
$pip
```

2. Set up your RADIUS server

Before creating and configuring the virtual network gateway, your RADIUS server should be configured correctly for authentication.

1. If you don't have a RADIUS server deployed, deploy one. For deployment steps, refer to the setup guide provided by your RADIUS vendor.
2. Configure the VPN gateway as a RADIUS client on the RADIUS. When adding this RADIUS client, specify the virtual network *GatewaySubnet* that you created.
3. Once the RADIUS server is set up, get the RADIUS server's IP address and the shared secret that RADIUS clients should use to talk to the RADIUS server. If the RADIUS server is in the Azure VNet, use the CA IP of the RADIUS server VM.

The [Network Policy Server \(NPS\)](#) article provides guidance about configuring a Windows RADIUS server (NPS) for AD domain authentication.

3. Create the VPN gateway

Configure and create the VPN gateway for your VNet.

- The -GatewayType must be 'Vpn' and the -VpnType must be 'RouteBased'.
- A VPN gateway can take up to 45 minutes to complete, depending on the [gateway SKU](#) you select.

```
New-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG `  
-Location $Location -IpConfigurations $ipconf -GatewayType Vpn `  
-VpnType RouteBased -EnableBgp $false -GatewaySku VpnGw1
```

4. Add the RADIUS server and client address pool

- The -RadiusServer can be specified by name or by IP address. If you specify the name and the server resides on-premises, then the VPN gateway may not be able to resolve the name. If that's the case, then it's better to specify the IP address of the server.
- The -RadiusSecret should match what is configured on your RADIUS server.
- The -VpnClientAddressPool is the range from which the connecting VPN clients receive an IP address. Use a private IP address range that does not overlap with the on-premises location that you will connect from, or with the VNet that you want to connect to. Ensure that you have a large enough address pool configured.

1. Create a secure string for the RADIUS secret.

```
$Secure_Secret=Read-Host -AsSecureString -Prompt "RadiusSecret"
```

2. You are prompted to enter the RADIUS secret. The characters that you enter will not be displayed and instead will be replaced by the "*" character.

```
RadiusSecret:***
```

3. Add the VPN client address pool and the RADIUS server information.

For SSTP configurations:

```
$Gateway = Get-AzureRmVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol "SSTP" `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

For IKEv2 configurations:

```
$Gateway = Get-AzureRmVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol "IKEv2" `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

For SSTP + IKEv2

```
$Gateway = Get-AzureRmVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol @("SSTP", "IKEv2") `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

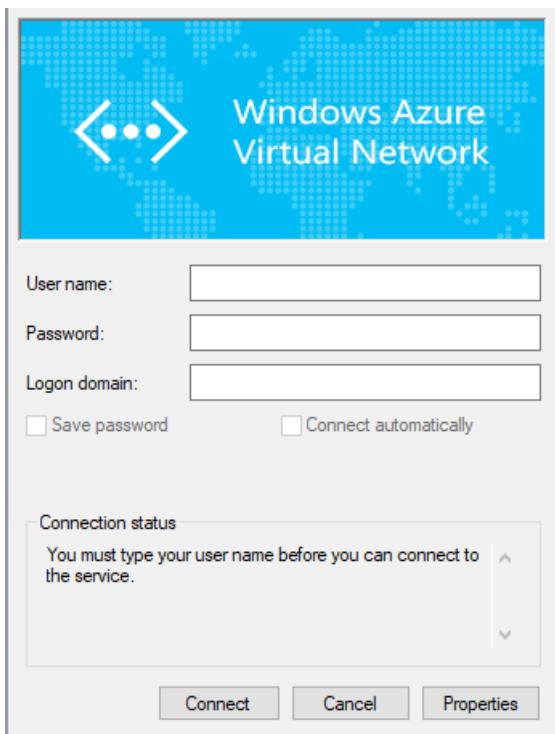
5. Download the VPN client configuration package and set up the VPN client

The VPN client configuration lets devices connect to a VNet over a P2S connection. To generate a VPN client configuration package and set up the VPN client, see [Create a VPN Client Configuration for RADIUS authentication](#).

6. Connect to Azure

To connect from a Windows VPN client

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Enter your domain credentials and click 'Connect'. A pop-up message requesting elevated rights appears. Accept it and enter the credentials.

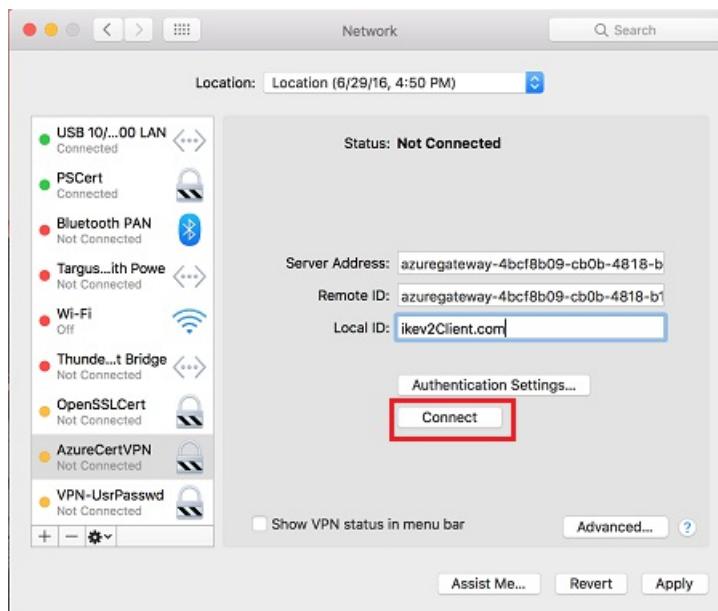


2. Your connection is established.



Connect from a Mac VPN client

From the Network dialog box, locate the client profile that you want to use, then click **Connect**.



To verify your connection

1. To verify that your VPN connection is active, open an elevated command prompt, and run `ipconfig/all`.
2. View the results. Notice that the IP address you received is one of the addresses within the Point-to-Site VPN Client Address Pool that you specified in your configuration. The results are similar to this example:

```
PPP adapter VNet1:  
  Connection-specific DNS Suffix .:  
  Description.....: VNet1  
  Physical Address.....:  
  DHCP Enabled.....: No  
  Autoconfiguration Enabled.....: Yes  
  IPv4 Address.....: 172.16.201.3(Preferred)  
  Subnet Mask.....: 255.255.255.255  
  Default Gateway.....:  
  NetBIOS over Tcpip.....: Enabled
```

To troubleshoot a P2S connection, see [Troubleshooting Azure point-to-site connections](#).

To 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-AzureRmVM
$Nics = Get-AzureRmNetworkInterface | 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.

To troubleshoot an RDP connection to a VM

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.
- Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VPNCClientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
- 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](#).
- Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.
- For more information about RDP connections, see [Troubleshoot Remote Desktop connections to a VM](#).

FAQ

This FAQ applies to P2S using RADIUS authentication

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)

- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing

VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2. Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\
IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Is RADIUS authentication supported on all Azure VPN Gateway SKUs?

RADIUS authentication is supported for VpnGw1, VpnGw2, and VpnGw3 SKUs. If you are using legacy SKUs, RADIUS authentication is supported on Standard and High Performance SKUs. It is not supported on the Basic Gateway SKU.

Is RADIUS authentication supported for the classic deployment model?

No. RADIUS authentication is not supported for the classic deployment model.

Are 3rd-party RADIUS servers supported?

Yes, 3rd-party RADIUS servers are supported.

What are the connectivity requirements to ensure that the Azure gateway is able to reach an on-premises RADIUS server?

A VPN Site-to-Site connection to the on-premises site, with the proper routes configured, is required.

Can traffic to an on-premises RADIUS server (from the Azure VPN gateway) be routed over an ExpressRoute connection?

No. It can only be routed over a Site-to-Site connection.

Is there a change in the number of SSTP connections supported with RADIUS authentication? What is the maximum number of SSTP and IKEv2 connections supported?

There is no change in the maximum number of SSTP connections supported on a gateway with RADIUS authentication. It remains 128. The maximum number of connections supported is 128, irrespective of whether the gateway is configured for SSTP, IKEv2, or both.

What is the difference between doing certificate authentication using a RADIUS server vs. using Azure native certificate authentication (by uploading a trusted certificate to Azure)?

In RADIUS certificate authentication, the authentication request is forwarded to a RADIUS server that handles the actual certificate validation. This option is useful if you want to integrate with a certificate authentication infrastructure that you already have through RADIUS.

When using Azure for certificate authentication, the Azure VPN gateway performs the validation of the certificate. You need to upload your certificate public key to the gateway. You can also specify list of revoked certificates that shouldn't be allowed to connect.

Does RADIUS authentication work with both IKEv2, and SSTP VPN?

Yes, RADIUS authentication is supported for both IKEv2, and SSTP VPN.

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#). To understand more about networking and virtual machines, see [Azure and Linux VM network overview](#).

Create and install VPN client configuration files for P2S RADIUS authentication

6/8/2018 • 11 minutes to read • [Edit Online](#)

To connect to a virtual network over point-to-site (P2S), you need to configure the client device that you'll connect from. You can create P2S VPN connections from Windows, Mac OS X, and Linux client devices.

When you're using RADIUS authentication, there are multiple authentication options: username/password authentication, certificate authentication, and other authentication types. The VPN client configuration is different for each type of authentication. To configure the VPN client, you use client configuration files that contain the required settings. This article helps you create and install the VPN client configuration for the RADIUS authentication type that you want to use.

IMPORTANT

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. Only point-to-site connections are impacted; site-to-site connections will not be affected. If you're using TLS for point-to-site VPNs on Windows 10 clients, you don't need to take any action. If you are using TLS for point-to-site connections on Windows 7 and Windows 8 clients, see the [VPN Gateway FAQ](#) for update instructions.

The configuration workflow for P2S RADIUS authentication is as follows:

1. [Set up the Azure VPN gateway for P2S connectivity](#).
2. [Set up your RADIUS server for authentication](#).
3. **Obtain the VPN client configuration for the authentication option of your choice and use it to set up the VPN client** (this article).
4. [Complete your P2S configuration and connect](#).

IMPORTANT

If there are any changes to the point-to-site VPN configuration after you generate the VPN client configuration profile, such as the VPN protocol type or authentication type, you must generate and install a new VPN client configuration on your users' devices.

To use the sections in this article, first decide which type of authentication you want to use: username/password, certificate, or other types of authentication. Each section has steps for Windows, Mac OS X, and Linux (limited steps available at this time).

Username/password authentication

You can configure username/password authentication to either use Active Directory or not use Active Directory. With either scenario, make sure that all connecting users have username/password credentials that can be authenticated through RADIUS.

When you configure username/password authentication, you can only create a configuration for the EAP-MSCHAPv2 username/password authentication protocol. In the commands, `-AuthenticationMethod` is `EapMSChapv2`.

1. Generate VPN client configuration files

Generate VPN client configuration files for use with username/password authentication. You can generate the VPN client configuration files by using the following command:

```
New-AzureRmVpnClientConfiguration -ResourceGroupName "TestRG" -Name "VNet1GW" -AuthenticationMethod "EapMSChapv2"
```

Running the command returns a link. Copy and paste the link to a web browser to download

VpnClientConfiguration.zip. Unzip the file to view the following folders:

- **WindowsAmd64** and **WindowsX86**: These folders contain the Windows 64-bit and 32-bit installer packages, respectively.
- **Generic**: This folder contains general information that you use to create your own VPN client configuration. You don't need this folder for username/password authentication configurations.
- **Mac**: If you configured IKEv2 when you created the virtual network gateway, you see a folder named **Mac** that contains a **mobileconfig** file. You use this file to configure Mac clients.

If you already created client configuration files, you can retrieve them by using the

```
Get-AzureRmVpnClientConfiguration
```

 cmdlet. But if you make any changes to your P2S VPN configuration, such as the VPN protocol type or authentication type, the configuration isn't updated automatically. You must run the

```
New-AzureRmVpnClientConfiguration
```

 cmdlet to create a new configuration download.

To retrieve previously generated client configuration files, use the following command:

```
Get-AzureRmVpnClientConfiguration -ResourceGroupName "TestRG" -Name "VNet1GW"
```

2. Configure VPN clients

You can configure the following VPN clients:

- [Windows](#)
- [Mac \(OS X\)](#)
- [Linux using strongSwan](#)

Windows VPN client setup

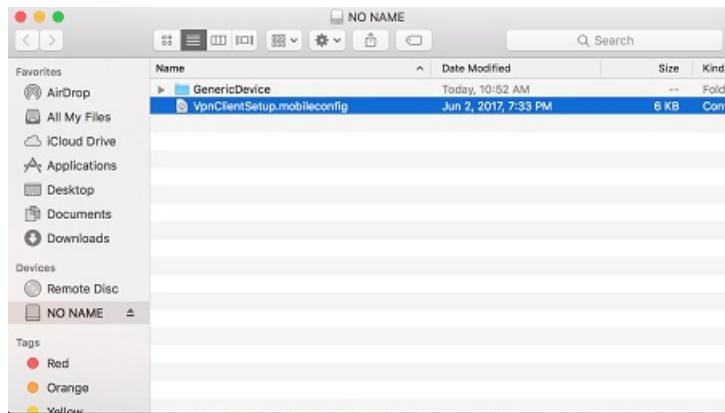
You can use the same VPN client configuration package on each Windows client computer, as long as the version matches the architecture for the client. For the list of client operating systems that are supported, see the [FAQ](#).

Use the following steps to configure the native Windows VPN client for certificate authentication:

1. Select the VPN client configuration files that correspond to the architecture of the Windows computer. For a 64-bit processor architecture, choose the **VpnClientSetupAmd64** installer package. For a 32-bit processor architecture, choose the **VpnClientSetupX86** installer package.
2. To install the package, double-click it. If you see a SmartScreen pop-up, select **More info > Run anyway**.
3. On the client computer, browse to **Network Settings** and select **VPN**. The VPN connection shows the name of the virtual network that it connects to.

Mac (OS X) VPN client setup

1. Select the **VpnClientSetup mobileconfig** file and send it to each of the users. You can use email or another method.
2. Locate the **mobileconfig** file on the Mac.



3. Double-click the profile to install it, and select **Continue**. The profile name is the same as the name of your virtual network.



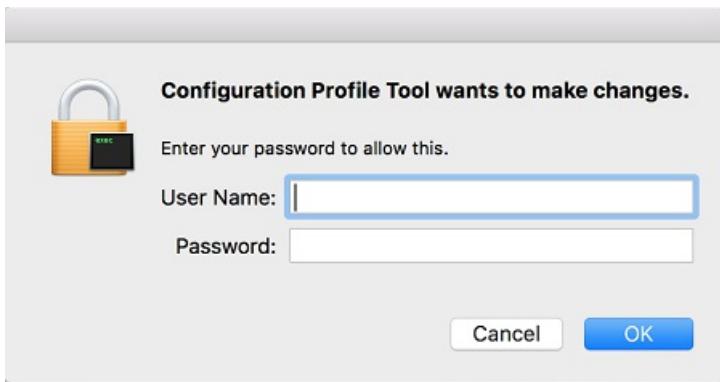
4. Select **Continue** to trust the sender of the profile and proceed with the installation.



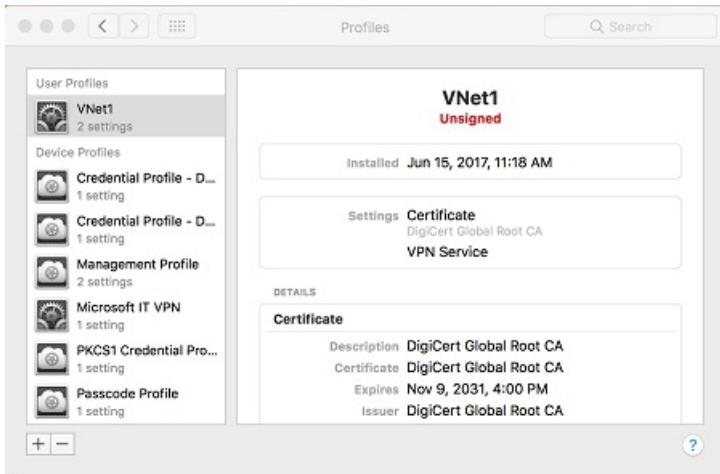
5. During profile installation, you have the option to specify the username and password for VPN authentication. It's not mandatory to enter this information. If you do, the information is saved and automatically used when you initiate a connection. Select **Install** to proceed.



6. Enter a username and password for the privileges that are required to install the profile on your computer. Select **OK**.



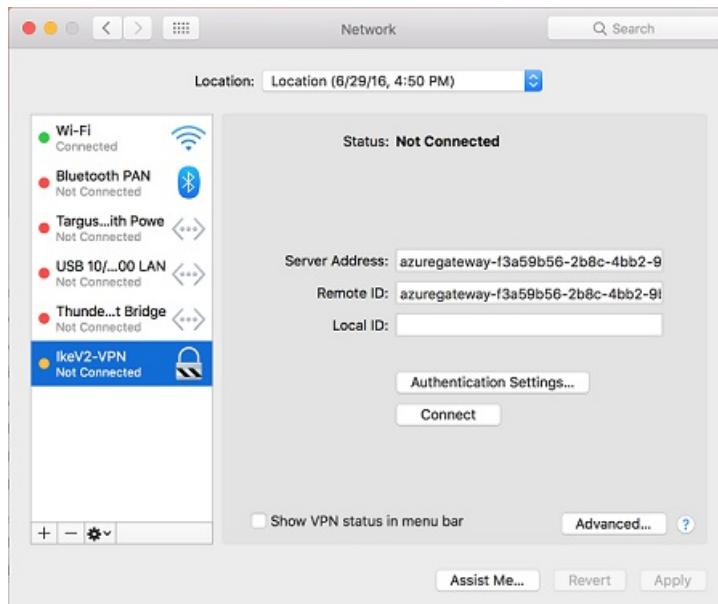
- After the profile is installed, it's visible in the **Profiles** dialog box. You can also open this dialog box later from **System Preferences**.



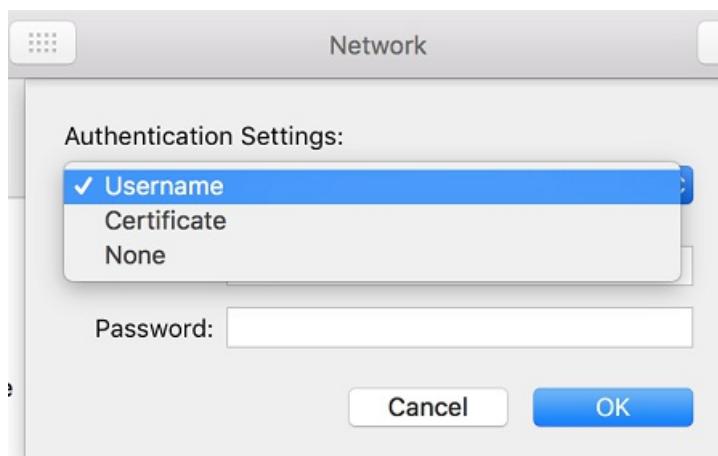
- To access the VPN connection, open the **Network** dialog box from **System Preferences**.



- The VPN connection appears as **IkeV2-VPN**. You can change the name by updating the **mobileconfig** file.



10. Select **Authentication Settings**. Select **Username** in the list and enter your credentials. If you entered the credentials earlier, then **Username** is automatically chosen in the list and the username and password are prepopulated. Select **OK** to save the settings.



11. Back in the **Network** dialog box, select **Apply** to save the changes. To initiate the connection, select **Connect**.

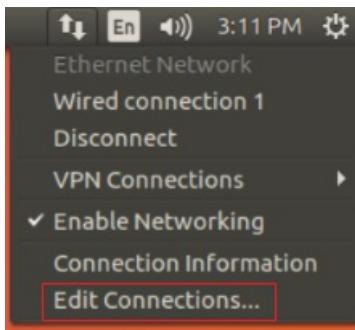
Linux VPN client setup through strongSwan

The following instructions were created through strongSwan 5.5.1 on Ubuntu 17.0.4. Actual screens might be different, depending on your version of Linux and strongSwan.

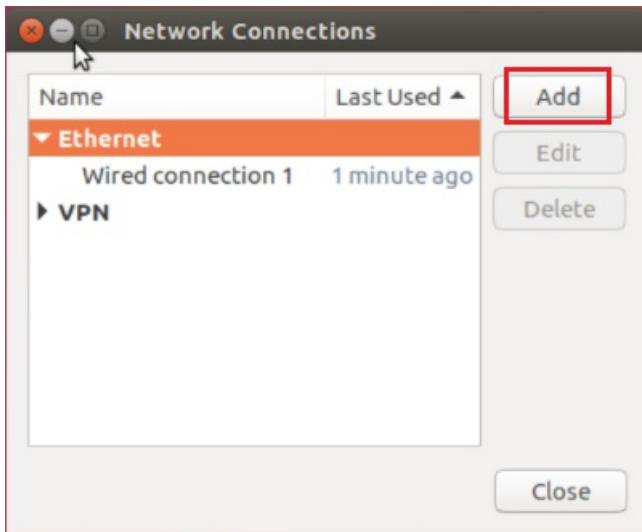
1. Open the **Terminal** to install **strongSwan** and its Network Manager by running the command in the example. If you receive an error that's related to `libcharon-extra-plugins`, replace it with `strongswan-plugin-eap-mschapv2`.

```
sudo apt-get install strongswan libcharon-extra-plugins moreutils iptables-persistent network-manager-strongswan
```

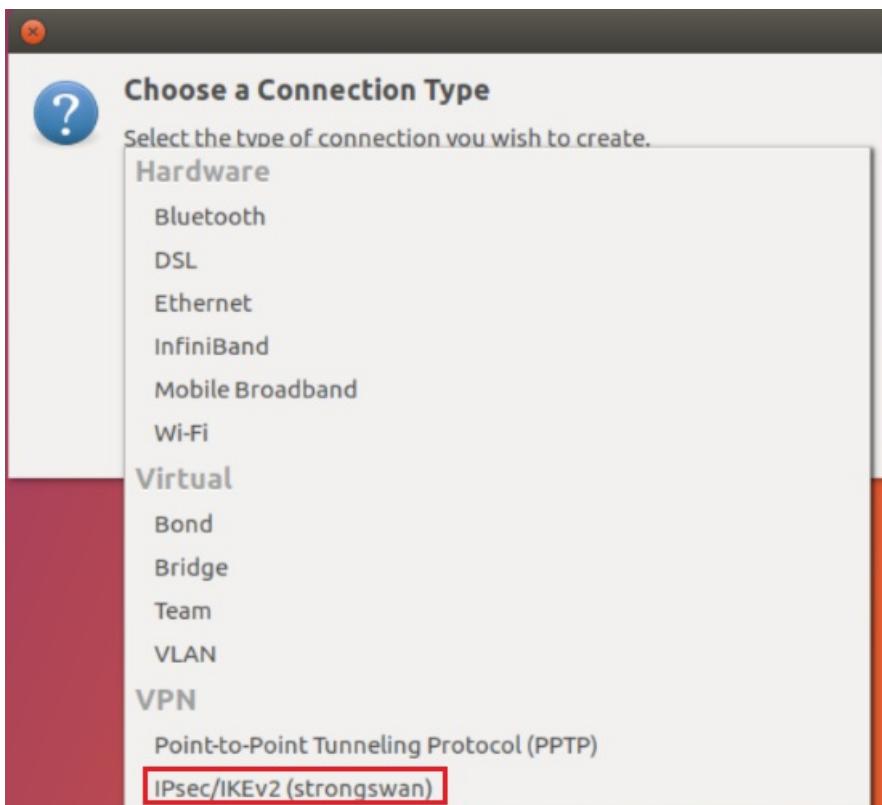
2. Select the **Network Manager** icon (up-arrow/down-arrow), and select **Edit Connections**.



3. Select the **Add** button to create a new connection.



4. Select **IPsec/IKEv2 (strongswan)** from the drop-down menu, and then select **Create**. You can rename your connection in this step.



5. Open the **VpnSettings.xml** file from the **Generic** folder of the downloaded client configuration files. Find the tag called **VpnServer** and copy the name, beginning with **azuregateway** and ending with **.cloudapp.net**.

```

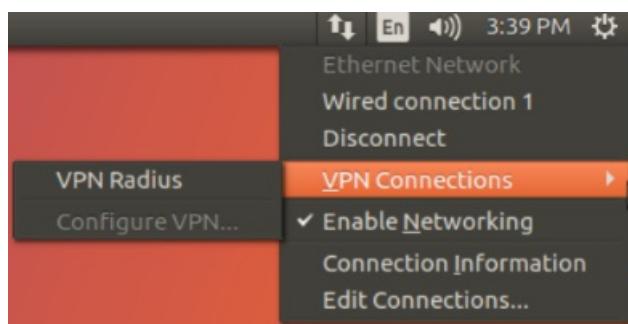
<VpnProfile>
  <VpnServer>azuregateway-<UUID>.cloudapp.net</VpnServer>
  <VpnType>IkeV2,SSTP</VpnType>
<snip>

```

6. Paste this name into the **Address** field of your new VPN connection in the **Gateway** section. Next, select the folder icon at the end of the **Certificate** field, browse to the **Generic** folder, and select the **VpnServerRoot** file.
7. In the **Client** section of the connection, select **EAP** for **Authentication**, and enter your username and password. You might have to select the lock icon on the right to save this information. Then, select **Save**.



8. Select the **Network Manager** icon (up-arrow/down-arrow) and hover over **VPN Connections**. You see the VPN connection that you created. To initiate the connection, select it.



Certificate authentication

You can create VPN client configuration files for RADIUS certificate authentication that uses the EAP-TLS protocol. Typically, an enterprise-issued certificate is used to authenticate a user for VPN. Make sure that all connecting users have a certificate installed on their devices, and that your RADIUS server can validate the certificate.

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. Only point-to-site connections are impacted; site-to-site connections will not be affected. If you're using TLS for point-to-site VPNs on Windows 10 clients, you don't need to take any action. If you are using TLS for point-to-site connections on Windows 7 and Windows 8 clients, see the [VPN Gateway FAQ](#) for update instructions.

In the commands, `-AuthenticationMethod` is `EapTls`. During certificate authentication, the client validates the RADIUS server by validating its certificate. `-RadiusRootCert` is the .cer file that contains the root certificate that's used to validate the RADIUS server.

Each VPN client device requires an installed client certificate. Sometimes a Windows device has multiple client certificates. During authentication, this can result in a pop-up dialog box that lists all the certificates. The user must then choose the certificate to use. The correct certificate can be filtered out by specifying the root certificate that the client certificate should chain to.

`-ClientRootCert` is the .cer file that contains the root certificate. It's an optional parameter. If the device that you want to connect from has only one client certificate, you don't have to specify this parameter.

1. Generate VPN client configuration files

Generate VPN client configuration files for use with certificate authentication. You can generate the VPN client configuration files by using the following command:

```
New-AzureRmVpnClientConfiguration -ResourceGroupName "TestRG" -Name "VNet1GW" -AuthenticationMethod "EapTls" -RadiusRootCert <full path name of .cer file containing the RADIUS root> -ClientRootCert <full path name of .cer file containing the client root> | fl
```

Running the command returns a link. Copy and paste the link to a web browser to download VpnClientConfiguration.zip. Unzip the file to view the following folders:

- **WindowsAmd64** and **WindowsX86**: These folders contain the Windows 64-bit and 32-bit installer packages, respectively.
- **GenericDevice**: This folder contains general information that's used to create your own VPN client configuration.

If you already created client configuration files, you can retrieve them by using the `Get-AzureRmVpnClientConfiguration` cmdlet. But if you make any changes to your P2S VPN configuration, such as the VPN protocol type or authentication type, the configuration isn't updated automatically. You must run the `New-AzureRmVpnClientConfiguration` cmdlet to create a new configuration download.

To retrieve previously generated client configuration files, use the following command:

```
Get-AzureRmVpnClientConfiguration -ResourceGroupName "TestRG" -Name "VNet1GW" | fl
```

2. Configure VPN clients

You can configure the following VPN clients:

- [Windows](#)
- [Mac \(OS X\)](#)
- Linux (supported, no article steps yet)

Windows VPN client setup

1. Select a configuration package and install it on the client device. For a 64-bit processor architecture, choose the **VpnClientSetupAmd64** installer package. For a 32-bit processor architecture, choose the

VpnClientSetupX86 installer package. If you see a SmartScreen pop-up, select **More info > Run anyway**. You can also save the package to install on other client computers.

2. Each client requires a client certificate for authentication. Install the client certificate. For information about client certificates, see [Client certificates for point-to-site](#). To install a certificate that was generated, see [Install a certificate on Windows clients](#).
3. On the client computer, browse to **Network Settings** and select **VPN**. The VPN connection shows the name of the virtual network that it connects to.

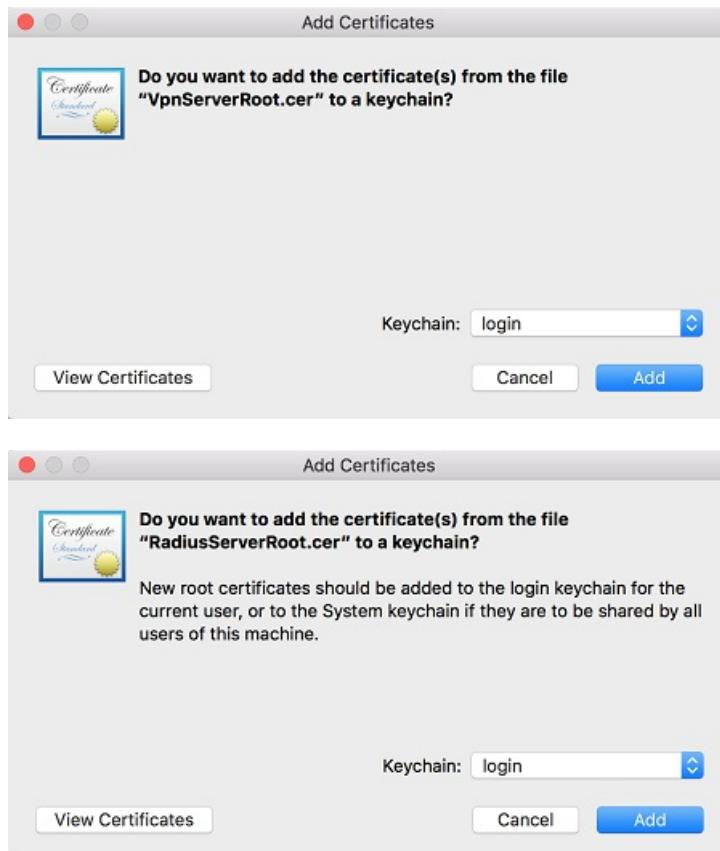
Mac (OS X) VPN client setup

You must create a separate profile for every Mac device that connects to the Azure virtual network. This is because these devices require the user certificate for authentication to be specified in the profile. The **Generic** folder has all the information that's required to create a profile:

- **VpnSettings.xml** contains important settings such as server address and tunnel type.
- **VpnServerRoot.cer** contains the root certificate that's required to validate the VPN gateway during P2S connection setup.
- **RadiusServerRoot.cer** contains the root certificate that's required to validate the RADIUS server during authentication.

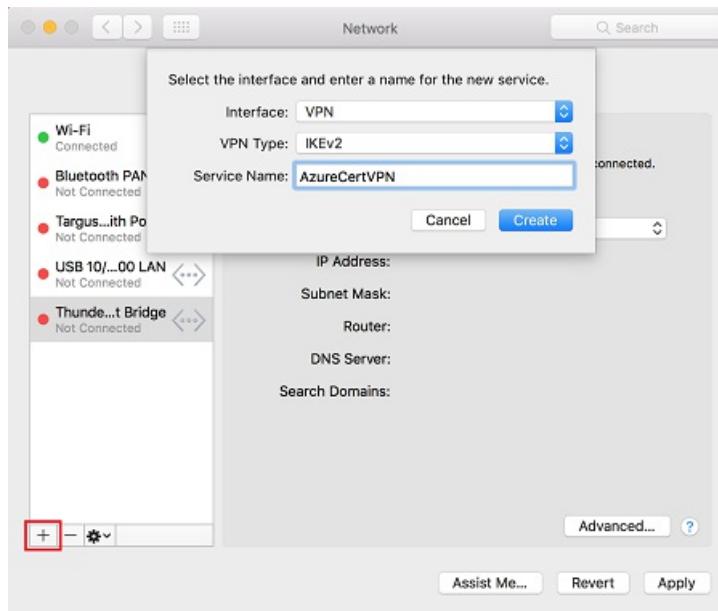
Use the following steps to configure the native VPN client on a Mac for certificate authentication:

1. Import the **VpnServerRoot** and **RadiusServerRoot** root certificates to your Mac. Copy each file to your Mac, double-click it, and then select **Add**.

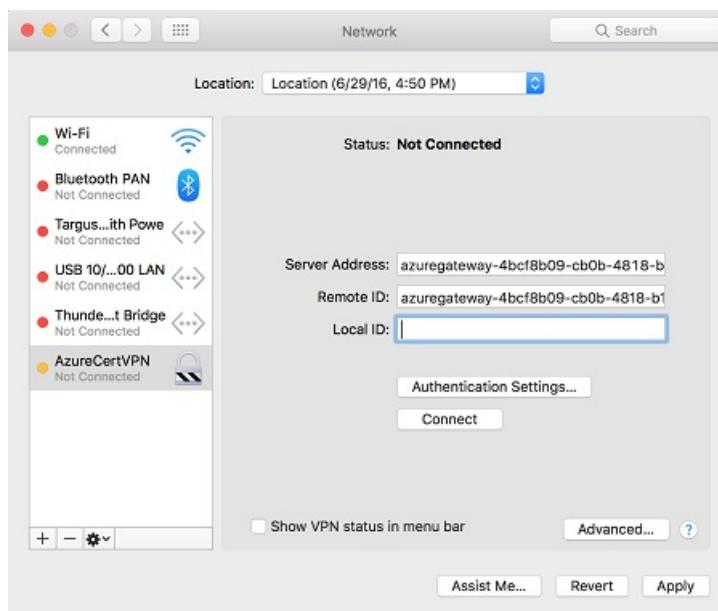


2. Each client requires a client certificate for authentication. Install the client certificate on the client device.
3. Open the **Network** dialog box under **Network Preferences**. Select + to create a new VPN client connection profile for a P2S connection to the Azure virtual network.

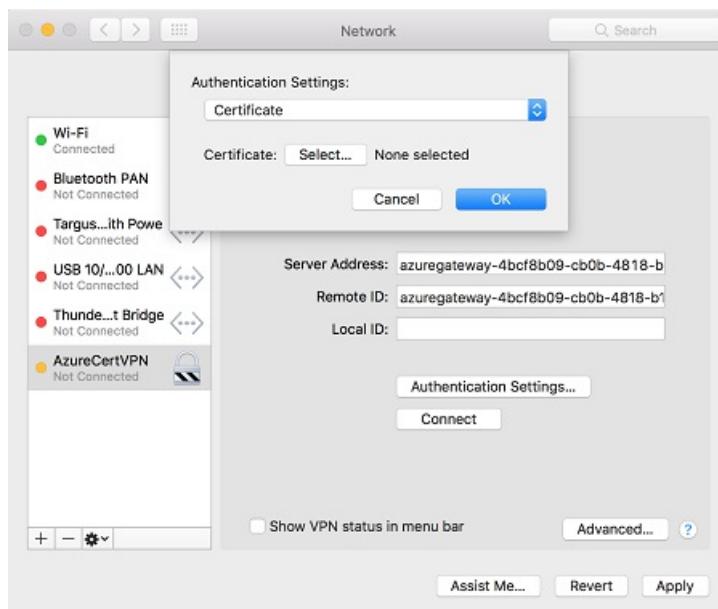
The **Interface** value is **VPN**, and the **VPN Type** value is **IKEv2**. Specify a name for the profile in the **Service Name** box, and then select **Create** to create the VPN client connection profile.



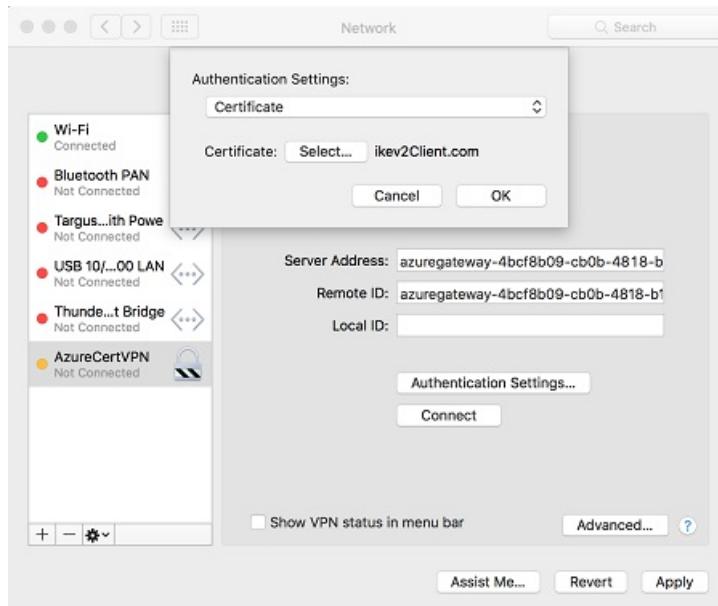
4. In the **Generic** folder, from the **VpnSettings.xml** file, copy the **VpnServer** tag value. Paste this value in the **Server Address** and **Remote ID** boxes of the profile. Leave the **Local ID** box blank.



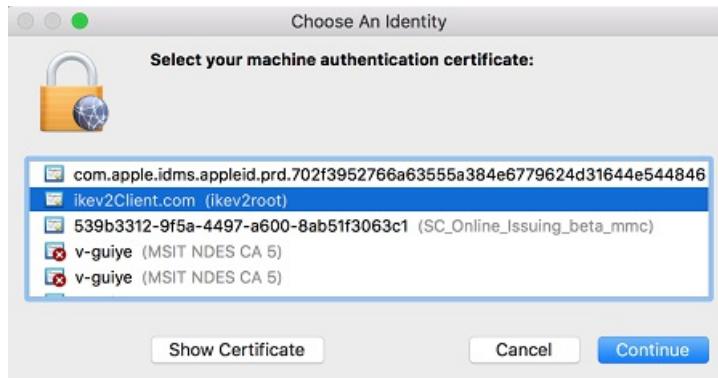
5. Select **Authentication Settings**, and select **Certificate**.



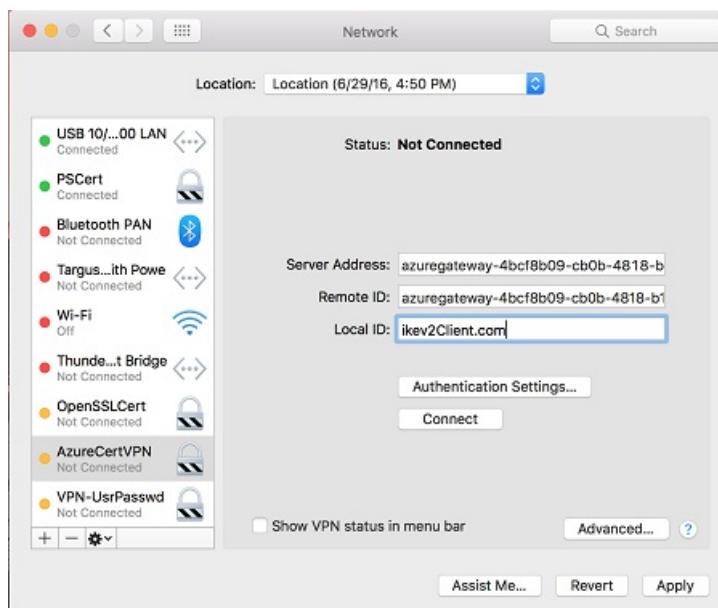
6. Click **Select** to choose the certificate that you want to use for authentication.



7. **Choose An Identity** displays a list of certificates for you to choose from. Select the proper certificate, and then select **Continue**.



8. In the **Local ID** box, specify the name of the certificate (from Step 6). In this example, it's **ikev2Client.com**. Then, select the **Apply** button to save the changes.



9. In the **Network** dialog box, select **Apply** to save all changes. Then, select **Connect** to start the P2S connection to the Azure virtual network.

Working with other authentication types or protocols

To use a different authentication type (for example, OTP), or to use a different authentication protocol (such as PEAP-MSCHAPv2 instead of EAP-MSCHAPv2), you must create your own VPN client configuration profile. To create the profile, you need information such as the virtual network gateway IP address, tunnel type, and split-tunnel routes. You can get this information by using the following steps:

1. Use the `Get-AzureRmVpnClientConfiguration` cmdlet to generate the VPN client configuration for EapMSChapv2. For instructions, see [this section](#) of the article.
2. Unzip the `VpnClientConfiguration.zip` file and look for the **GenenericDevice** folder. Ignore the folders that contain the Windows installers for 64-bit and 32-bit architectures.
3. The **GenenericDevice** folder contains an XML file called **VpnSettings**. This file contains all the required information:
 - **VpnServer**: FQDN of the Azure VPN gateway. This is the address that the client connects to.
 - **VpnType**: Tunnel type that you use to connect.
 - **Routes**: Routes that you have to configure in your profile so that only traffic that's bound for the Azure virtual network is sent over the P2S tunnel.

The **GenenericDevice** folder also contains a .cer file called **VpnServerRoot**. This file contains the root certificate that's required to validate the Azure VPN gateway during P2S connection setup. Install the certificate on all devices that will connect to the Azure virtual network.

Next steps

Return to the article to [complete your P2S configuration](#).

For P2S troubleshooting information, see [Troubleshooting Azure point-to-site connections](#).

Integrate Azure VPN gateway RADIUS authentication with NPS server for Multi-Factor Authentication

7/13/2018 • 2 minutes to read • [Edit Online](#)

The article describes how to integrate Network Policy Server (NPS) with Azure VPN gateway RADIUS authentication to deliver Multi-Factor Authentication (MFA) for point-to-site VPN connections.

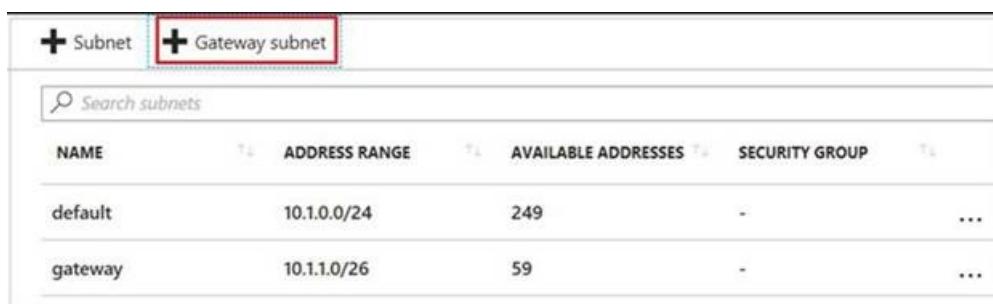
Prerequisite

To enable MFA, the users must be in Azure Active Directory (Azure AD), which must be synced from either the on-premises or cloud environment. Also, the user must have already completed the auto-enrollment process for MFA. For more information, see [Set up my account for two-step verification](#)

Detailed steps

Step 1: Create a virtual network gateway

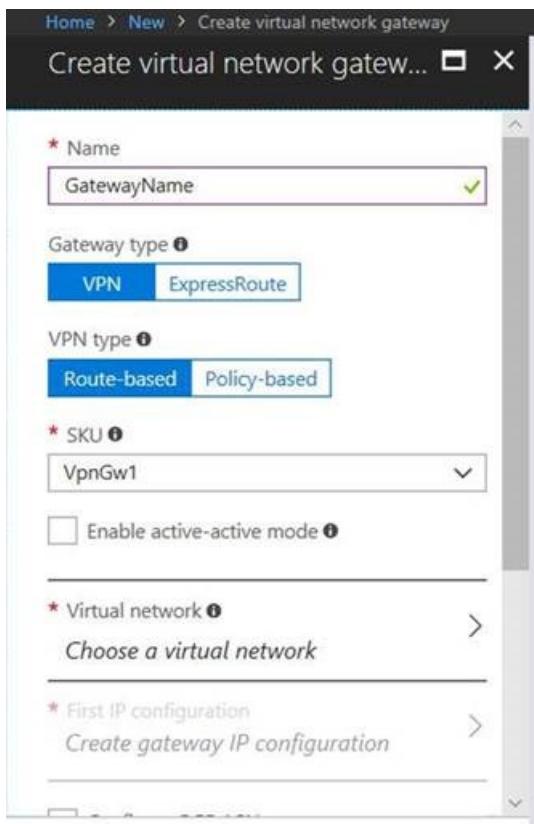
1. Log on to the [Azure portal](#).
2. In the virtual network that will host the virtual network gateway, select **Subnets**, and then select **Gateway subnet** to create a subnet.



Subnets					
NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP	...	
default	10.1.0.0/24	249	-	...	
gateway	10.1.1.0/26	59	-	...	

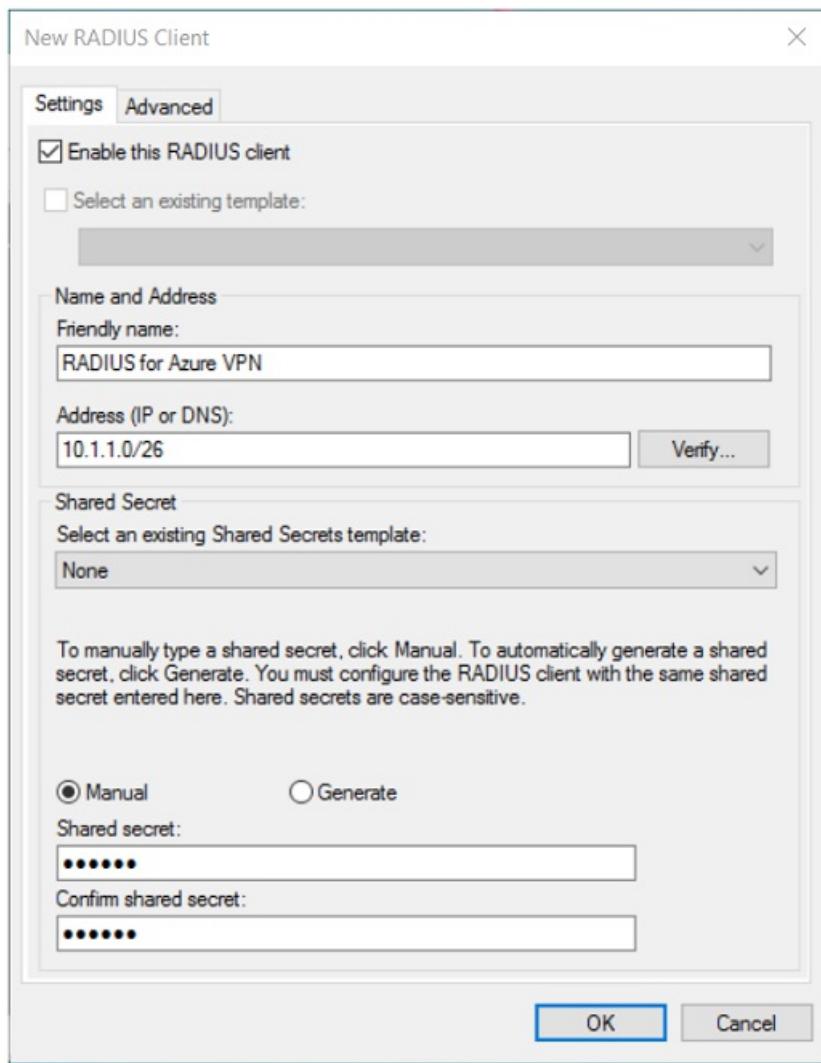
3. Create a virtual network gateway by specifying the following settings:

- **Gateway type:** Select **VPN**.
- **VPN type:** Select **Route-based**.
- **SKU:** Select a SKU type based on your requirements.
- **Virtual network:** Select the virtual network in which you created the gateway subnet.

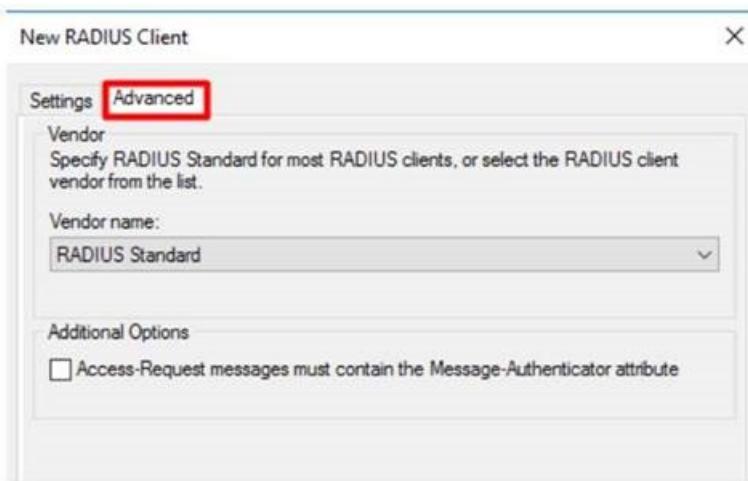


Step 2 Configure the NPS for Azure MFA

1. On the NPS server, [install the NPS extension for Azure MFA](#).
2. Open the NSP console, right-click **RADIUS Clients**, and then select **New**. Create the RADIUS client by specifying the following settings:
 - **Friendly Name:** Type any name.
 - **Address (IP or DNS):** Type the gateway subnet that you created in the Step 1.
 - **Shared secret:** type any secret key, and remember it for later use.



3. On the **Advanced** tab, set the vendor name to **RADIUS Standard** and make sure that the **Additional Options** check box is not selected.



4. Go to **Policies > Network Policies**, double-click **Connections to Microsoft Routing and Remote Access server** policy, select **Grant access**, and then click **OK**.

Step 3 Configure the virtual network gateway

1. Log on to [Azure portal](#).
2. Open the virtual network gateway that you created. Make sure that the gateway type is set to **VPN** and that the VPN type is **route-based**.
3. Click **Point to site configuration > Configure now**, and then specify the following settings:

- **Address pool:** Type the gateway subnet you created in the step 1.
- **Authentication type:** Select **RADIUS authentication**.
- **Server IP address:** Type the IP address of the NPS server.

Save Discard Download VPN client

Connection health

Connections	0
Ingress (bytes)	133797
Egress (bytes)	177284

Address pool
10.1.1.0/26

Tunnel type
SSL VPN (SSTP)
IKEv2 VPN

Authentication type
 Azure certificate RADIUS authentication

RADIUS authentication

* Server IP address
10.0.0.7

* Server secret
123123

Next steps

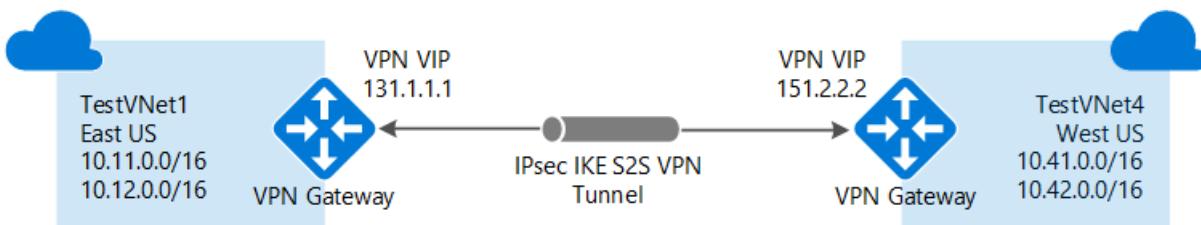
- [Azure Multi-Factor Authentication](#)
- [Integrate your existing NPS infrastructure with Azure Multi-Factor Authentication](#)

Configure a VNet-to-VNet VPN gateway connection using the Azure portal

4/18/2018 • 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 the Azure portal. 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. When you use the Site-to-Site IPsec 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 that. 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:

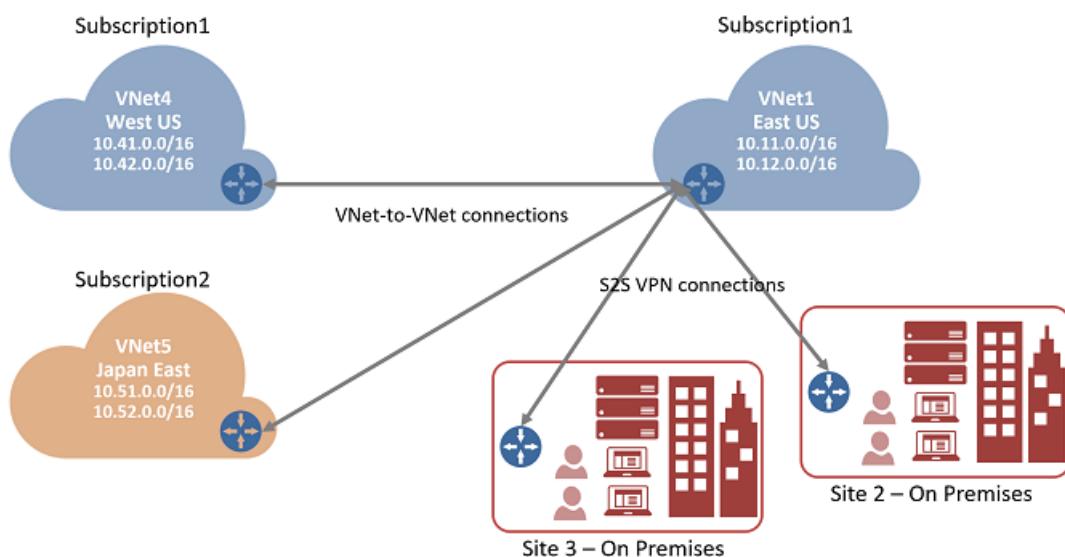
- **Cross region geo-redundancy and geo-presence**

- 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, as shown in the following diagram:



This article helps you connect VNets using the VNet-to-VNet connection type. When using these steps as an exercise, you can use the example settings values. In the example, the virtual networks are in the same subscription, but in different resource groups. If your VNets are in different subscriptions, you can't create the connection in the portal. You can use [PowerShell](#) or [CLI](#). For more information about VNet-to-VNet connections, see the [VNet-to-VNet FAQ](#) at the end of this article.

Example settings

Values for TestVNet1:

- VNet Name: TestVNet1
- Address space: 10.11.0.0/16
- Subscription: Select the subscription you want to use
- Resource Group: TestRG1
- Location: East US
- Subnet Name: FrontEnd
- Subnet Address range: 10.11.0.0/24
- Gateway Subnet name: GatewaySubnet (this will auto-fill in the portal)
- Gateway Subnet address range: 10.11.255.0/27

- DNS Server: Use the IP address of your DNS Server
- Virtual Network Gateway Name: TestVNet1GW
- Gateway Type: VPN
- VPN type: Route-based
- SKU: Select the Gateway SKU you want to use
- Public IP address name: TestVNet1GWIP
- Connection Name: TestVNet1toTestVNet4
- Shared key: You can create the shared key yourself. For this example, we'll use abc123. The important thing is that when you create the connection between the VNets, the value must match.

Values for TestVNet4:

- VNet Name: TestVNet4
- Address space: 10.41.0.0/16
- Subscription: Select the subscription you want to use
- Resource Group: TestRG4
- Location: West US
- Subnet Name: FrontEnd
- Subnet Address range: 10.41.0.0/24
- GatewaySubnet name: GatewaySubnet (this will auto-fill in the portal)
- GatewaySubnet address range: 10.41.255.0/27
- DNS Server: Use the IP address of your DNS Server
- Virtual Network Gateway Name: TestVNet4GW
- Gateway Type: VPN
- VPN type: Route-based
- SKU: Select the Gateway SKU you want to use
- Public IP address name: TestVNet4GWIP
- Connection Name: TestVNet4toTestVNet1
- Shared key: You can create the shared key yourself. For this example, we'll use abc123. The important thing is that when you create the connection between the VNets, the value must match.

1. Create and configure TestVNet1

If you already have a VNet, verify that the settings are compatible with your VPN gateway design. Pay particular attention to any subnets that may overlap with other networks. If you have overlapping subnets, your connection won't work properly. If your VNet is configured with the correct settings, you can begin the steps in the [Specify a DNS server](#) section.

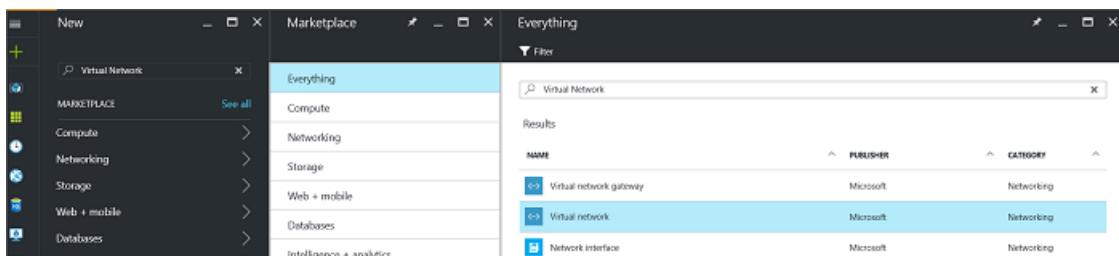
To create a virtual network

To create a VNet in the Resource Manager deployment model by using the Azure portal, follow the steps below. The screenshots are provided as examples. Be sure to replace the values with your own. For more information about working with virtual networks, see the [Virtual Network Overview](#).

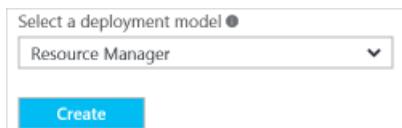
NOTE

In order for this VNet to connect to an on-premises location you need 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 does not route the way you may expect it to. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Take care to plan your network configuration accordingly.

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **+**. In the **Search the marketplace** field, type "Virtual Network". Locate **Virtual Network** from the returned list and click to open the **Virtual Network** page.



3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, select **Resource Manager**, and then click **Create**.



4. On the **Create virtual network** page, configure the VNet settings. When you fill in the fields, the red exclamation mark becomes a green check mark when the characters entered in the field are valid. There may be values that are auto-filled. If so, replace the values with your own. The **Create virtual network** page looks similar to the following example:

Name:

Address space:

Subscription:

Resource group: Create new Use existing

Location:

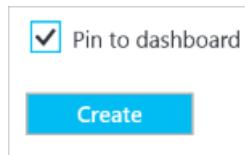
Subnet:

Name:

Address range:

5. **Name:** Enter the name for your virtual network.
6. **Address space:** Enter the address space. If you have multiple address spaces to add, add your first address space. You can add additional address spaces later, after creating the VNet.
7. **Subscription:** Verify that the Subscription listed is the correct one. You can change subscriptions by using the drop-down.
8. **Resource group:** Select an existing resource group, or create a new one by typing a name for your new resource group. If you are creating a new group, name the resource group according to your planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](#).

9. **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
10. **Subnet:** Add the subnet name and subnet address range. You can add additional subnets later, after creating the VNet.
11. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



2. Add additional address space and create subnets

You can add additional address space and create subnets once your VNet has been created.

To add additional address space

1. To add additional address space, under the **Settings** section on your virtual network page, click **Address space** to open the Address space page.
2. Add the additional address space, and then click **Save** at the top of the page.

To create additional subnets

1. To create subnets, in the **Settings** section of your virtual network page, click **Subnets** to open the **Subnets** page.
2. On the Subnets page, click **+Subnet** to open the **Add subnet** page. Name your new subnet and specify the address range.

3. To save your changes, click **OK** at the bottom of the page.

OK

3. Create a gateway subnet

Before creating a virtual network gateway for your virtual network, you first need to create the gateway subnet. The gateway subnet contains the IP addresses that are used by the virtual network gateway. If possible, it's best to create a gateway subnet using a CIDR block of /28 or /27 in order to provide enough IP addresses to accommodate additional future configuration requirements.

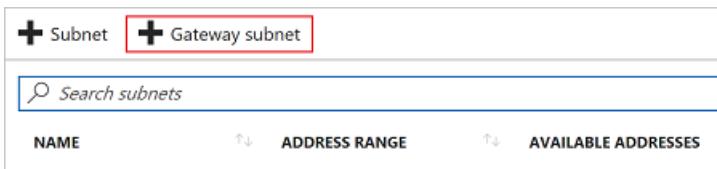
If you are creating this configuration as an exercise, refer to these [Example settings](#) when creating your gateway subnet.

IMPORTANT

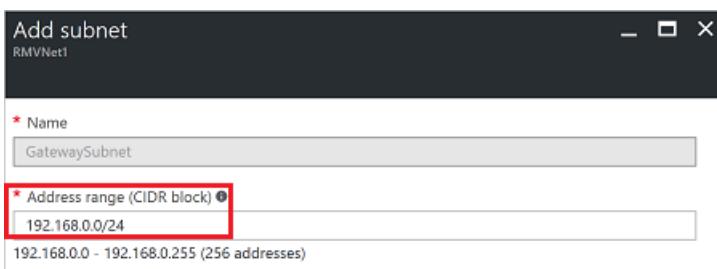
When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

To create a gateway subnet

1. In the [portal](#), navigate to the Resource Manager virtual network for which you want to create a virtual network gateway.
2. In the **Settings** section of your VNet page, click **Subnets** to expand the Subnets page.
3. On the **Subnets** page, click **+Gateway subnet** to open the **Add subnet** page.



4. The **Name** for your subnet is automatically filled in with the value 'GatewaySubnet'. This value is required in order for Azure to recognize the subnet as the gateway subnet. Adjust the auto-filled **Address range** values to match your configuration requirements, then click **OK** at the bottom of the page to create the subnet.



4. Specify a DNS server (optional)

DNS is not required for VNet-to-VNet connections. However, if you want to have name resolution for resources that are deployed to your virtual network, you should specify a DNS server. This setting lets you specify the DNS server that you want to use for name resolution for this virtual network. It does not create a DNS server.

1. On the **Settings** page for your virtual network, navigate to **DNS Servers** and click to open the **DNS servers** page.

Save Discard

⚠ Virtual machines within this virtual network must be restarted to utilize the updated DNS server settings.

DNS servers ⓘ

Default (Azure-provided)

Custom

Add DNS server ...

- **DNS Servers:** Select **Custom**.
 - **Add DNS server:** Enter the IP address of the DNS server that you want to use for name resolution.
2. When you are done adding DNS servers, click **Save** at the top of the page.

5. 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. If you are creating this configuration as an exercise, you can refer to the [Example settings](#).

To create a virtual network gateway

1. In the portal, on the left side, click **+** and type 'virtual network gateway' in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create** at the bottom of the page to open the **Create virtual network gateway** page.
2. On the **Create virtual network gateway** page, fill in the values for your virtual network gateway.

Create virtual network gateway

* Name

Gateway type [?](#)
 VPN ExpressRoute

VPN type [?](#)
 Route-based Policy-based

* SKU [?](#)
 [▼](#)

Enable active-active mode [?](#)

* Virtual network [?](#) >
Choose a virtual network

* First IP configuration [?](#) >
Create gateway IP configuration

Configure BGP ASN

* Subscription
 [▼](#)

Resource group [?](#)
-

* Location [?](#)
 [▼](#)

Pin to dashboard

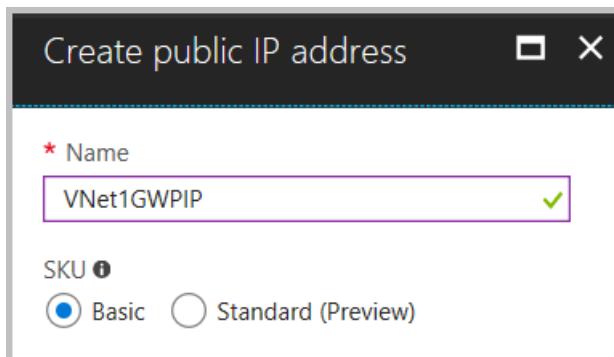
Create [Automation options](#)

Provisioning a virtual network gateway may take up to 45 minutes.

3. On the **Create virtual network gateway** page, specify the values for your virtual network gateway.

- **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **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](#).
- **Location:** You may need to scroll to see Location. Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, when you select a virtual network in the next step, it will not appear in the drop-down list.

- **Virtual network:** Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the 'Choose a virtual network' page. Select the VNet. If you don't see your VNet, make sure the Location field is pointing to the region in which your virtual network is located.
- **Gateway subnet address range:** You will only see this setting if you did not previously create a gateway subnet for your virtual network. If you previously created a valid gateway subnet, this setting will not appear.
- **First IP configuration:** The 'Choose public IP address' page creates a 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. VPN Gateway currently only supports *Dynamic* Public IP address allocation. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.
 - First, click **Create gateway IP configuration** to open the 'Choose public IP address' page, then click **+Create new** to open the 'Create public IP address' page.
 - Next, input a **Name** for your public IP address. Leave the SKU as **Basic** unless there is a specific reason to change it to something else, then click **OK** at the bottom of this page to save your changes.



4. Verify the settings. You can select **Pin to dashboard** at the bottom of the page if you want your gateway to appear on the dashboard.
5. Click **Create** to begin creating the VPN gateway. The settings are validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.

After the gateway is created, 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. You can click the connected device (your virtual network gateway) to view more information.

6. Create and configure TestVNet4

Once you've configured TestVNet1, create TestVNet4 by repeating the previous steps, replacing the values with those of TestVNet4. You don't need to wait until the virtual network gateway for TestVNet1 has finished creating before configuring TestVNet4. If you are using your own values, make sure that the address spaces don't overlap with any of the VNets that you want to connect to.

7. Configure the TestVNet1 gateway connection

When the virtual network gateways for both TestVNet1 and TestVNet4 have completed, you can create your virtual network gateway connections. In this section, you create a connection from VNet1 to VNet4. These steps work only for VNets in the same subscription. If your VNets are in different subscriptions, you must use PowerShell to make the connection. See the [PowerShell](#) article. However, if your VNets are in different resource

groups in the same subscription, you can connect them using the portal.

1. In **All resources**, navigate to the virtual network gateway for your VNet. For example, **TestVNet1GW**.

Click **TestVNet1GW** to open the virtual network gateway page.

NAME	STATUS	CONNECTION TYPE	PEER
No results			

2. Click **+Add** to open the **Add connection** page.

NAME	STATUS	CONNECTION TYPE
TestVNet4GW		VNet-to-VNet
TestVNet1GW		VNet-to-VNet
TestVNet3GW		VNet-to-VNet

3. On the **Add connection** page, in the name field, type a name for your connection. For example, **TestVNet1toTestVNet4**.
4. For **Connection type**, select **VNet-to-VNet** from the dropdown.
5. The **First virtual network gateway** field value is automatically filled in because you are creating this connection from the specified virtual network gateway.
6. The **Second virtual network gateway** field is the virtual network gateway of the VNet that you want to create a connection to. Click **Choose another virtual network gateway** to open the **Choose virtual network gateway** page.
7. View the virtual network gateways that are listed on this page. Notice that only virtual network gateways that are in your subscription are listed. If you want to connect to a virtual network gateway that is not in your subscription, please use the [PowerShell article](#).
8. Click the virtual network gateway that you want to connect to.
9. In the **Shared key** field, type a shared key for your connection. You can generate or create this key yourself. In

a site-to-site connection, the key you use would be exactly the same for your on-premises device and your virtual network gateway connection. The concept is similar here, except that rather than connecting to a VPN device, you are connecting to another virtual network gateway.

10. Click **OK** at the bottom of the page to save your changes.

8. Configure the TestVNet4 gateway connection

Next, create a connection from TestVNet4 to TestVNet1. In the portal, locate the virtual network gateway associated with TestVNet4. Follow the steps from the previous section, replacing the values to create a connection from TestVNet4 to TestVNet1. Make sure that you use the same shared key.

9. Verify your connections

Locate the virtual network gateway in the portal. On the virtual network gateway page, click **Connections** to view the connections page for the virtual network gateway. Once the connection is established, you see the Status values change to **Succeeded** and **Connected**. You can double-click a connection to open the **Essentials** page and view more information.

NAME	STATUS	CONNECTION TYPE	PEER
TestVNet1toTestVNet4	Connected	VNet-to-VNet	TestVNet1GW
TestVNet4toTestVNet1	Connected	VNet-to-VNet	TestVNet1GW

When data begins flowing, you see values for Data in and Data out.

Resource group (change) TestRG1	Data in 1.66 KiB
Status Connected	Data out 1.66 KiB
Location East US	Virtual network TestVNet1, TestVNet4
Subscription name (change) Windows Azure Internal Consumption	Virtual network gateway 1 TestVNet1GW
Subscription ID	Virtual network gateway 2 TestVNet4GW

To add additional connections

If you want to add additional connections, navigate to the virtual network gateway that you want to create the connection from, then click **Connections**. You can create another VNet-to-VNet connection, or create an IPsec Site-to-Site connection to an on-premises location. Be sure to adjust the **Connection type** to match the type of connection you want to create. Before creating additional connections, verify that the address space for your virtual network does not overlap with any of the address spaces that you want to connect to. For steps to create a Site-to-Site connection, see [Create a Site-to-Site connection](#).

VNet-to-VNet FAQ

View the FAQ details for additional information about VNet-to-VNet connections.

The VNet-to-VNet FAQ applies to VPN Gateway connections. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud services that are not 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 are connected together.

Can I used 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 be using 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

See [Network Security](#) for information about how you can limit network traffic to resources in a virtual network.

See [Virtual network traffic routing](#) for information about how Azure routes traffic between Azure, on-premises, and Internet resources.

Configure a VNet-to-VNet VPN gateway connection using PowerShell

9/6/2018 • 17 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.

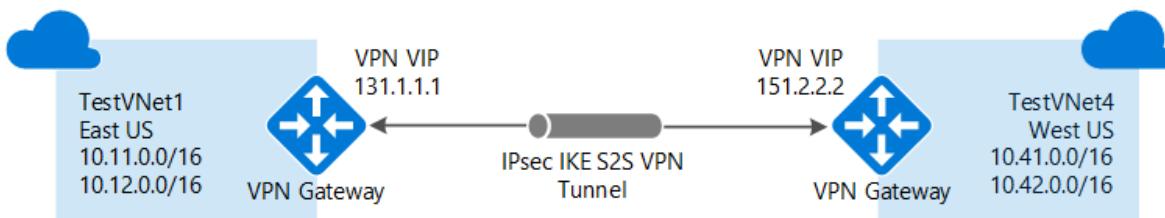
VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

Which VNet-to-VNet steps should I use?

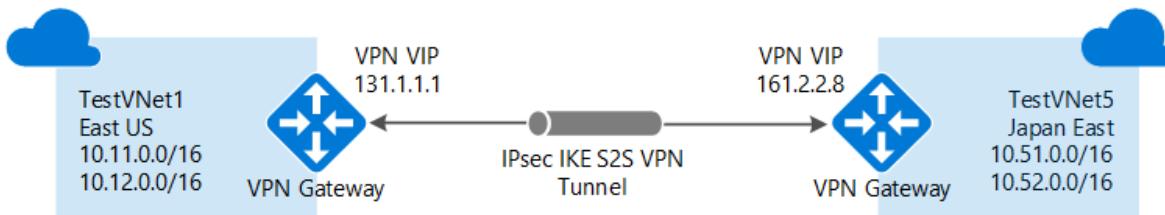
In this article, you see two different sets of steps. One set of steps for [VNets that reside in the same subscription](#) and one for [VNets that reside in different subscriptions](#). The key difference between the sets is that you must use separate PowerShell sessions when configuring the connections 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

Before beginning, you need to install the latest version of the Azure Resource Manager PowerShell cmdlets, at least 4.0 or later. For more information about installing the PowerShell cmdlets, 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. 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.

```
$Sub1 = "Replace_With_Your_Subscription_Name"
$RG1 = "TestRG1"
$Location1 = "East US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$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"
```

2. Connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName $Sub1
```

3. Create a new resource group.

```
New-AzureRmResourceGroup -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.

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-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1  
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1  
$gwsu1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1
```

5. Create TestVNet1.

```
New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 `  
-Location $Location1 -AddressPrefix $VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsu1
```

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-AzureRmPublicIpAddress -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-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1  
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1  
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -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-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 `  
-Location $Location1 -IpConfigurations $gwpipconf1 -GatewayType Vpn `  
-VpnType RouteBased -GatewaySku VpnGw1
```

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. This step can be done within the same PowerShell session because it is in the same subscription.

1. 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"  
$GWSubName4 = "GatewaySubnet"  
$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 = "gwpipconf4"  
$Connection41 = "VNet4toVNet1"
```

2. Create a new resource group.

```
New-AzureRmResourceGroup -Name $RG4 -Location $Location4
```

3. Create the subnet configurations for TestVNet4.

```
$fesub4 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName4 -AddressPrefix $FESubPrefix4  
$besub4 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName4 -AddressPrefix $BESubPrefix4  
$gwsb4 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName4 -AddressPrefix $GWSubPrefix4
```

4. Create TestVNet4.

```
New-AzureRmVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4 `  
-Location $Location4 -AddressPrefix $VnetPrefix41,$VnetPrefix42 -Subnet $fesub4,$besub4,$gwsb4
```

5. Request a public IP address.

```
$gwpip4 = New-AzureRmPublicIpAddress -Name $GWIPName4 -ResourceGroupName $RG4 `  
-Location $Location4 -AllocationMethod Dynamic
```

6. Create the gateway configuration.

```
$vnet4 = Get-AzureRmVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4  
$subnet4 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet4  
$gwipconf4 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName4 -Subnet $subnet4 -  
PublicIpAddress $gwip4
```

7. Create the TestVNet4 gateway. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

```
New-AzureRmVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4 `  
-Location $Location4 -IpConfigurations $gwipconf4 -GatewayType Vpn `  
-VpnType RouteBased -GatewaySku VpnGw1
```

Step 4 - Create the connections

1. Get both virtual network gateways. If both of the gateways are in the same subscription, as they are in the example, you can complete this step in the same PowerShell session.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$vnet4gw = Get-AzureRmVirtualNetworkGateway -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-AzureRmVirtualNetworkGatewayConnection -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-AzureRmVirtualNetworkGatewayConnection -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 a different subscription. 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.

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-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName $Sub5
```

3. Create a new resource group.

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5
```

4. Create the subnet configurations for TestVNet5.

```
$fesub5 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName5 -AddressPrefix $FESubPrefix5  
$besub5 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName5 -AddressPrefix $BESubPrefix5  
$gwsb5 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSBName5 -AddressPrefix $GWSBPrefix5
```

5. Create TestVNet5.

```
New-AzureRmVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5 -Location $Location5 `  
-AddressPrefix $VnetPrefix51,$VnetPrefix52 -Subnet $fesub5,$besub5,$gwsb5
```

6. Request a public IP address.

```
$gwpip5 = New-AzureRmPublicIpAddress -Name $GWIPName5 -ResourceGroupName $RG5 `  
-Location $Location5 -AllocationMethod Dynamic
```

7. Create the gateway configuration.

```
$vnet5 = Get-AzureRmVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5  
$subnet5 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet5  
$gwipconf5 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName5 -Subnet $subnet5 -  
PublicIpAddress $gwpip5
```

8. Create the TestVNet5 gateway.

```
New-AzureRmVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5 -Location $Location5 `  
-IpConfigurations $gwipconf5 -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. Log in and connect to Subscription 1 before running the following example:

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -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. Log in and connect to Subscription 5 before running the following example:

```
$vnet5gw = Get-AzureRmVirtualNetworkGateway -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-AzureRmVirtualNetworkGatewayConnection -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-AzureRmVirtualNetworkGatewayConnection -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 VPN 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-AzureRmVirtualNetworkGatewayConnection' 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-AzureRmVirtualNetworkGatewayConnection -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. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud services that are not 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 are connected together.

Can I used 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 be using 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](#).

Configure a VNet-to-VNet VPN gateway connection using Azure CLI

2/16/2018 • 16 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 Azure CLI. 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 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.

Connecting VNets using Site-to-Site (IPsec) steps

If you are working with a complicated network configuration, you may prefer to connect your VNets using the [Site-to-Site](#) steps, instead of 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 manually 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.

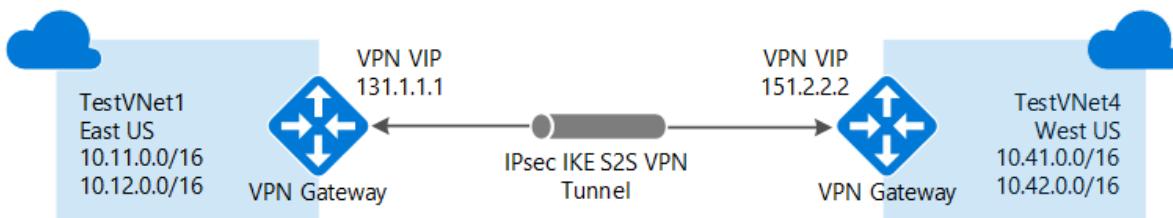
VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

Which VNet-to-VNet steps should I use?

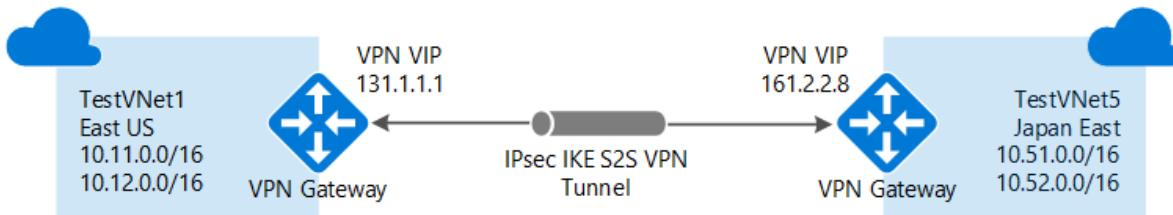
In this article, you see two different sets of VNet-to-VNet connection steps. One set of steps for [VNets that reside in the same subscription](#) and one for [VNets that reside in different subscriptions](#).

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.



Connect VNets that are in the same subscription

Before you begin

Before beginning, install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install Azure CLI 2.0](#).

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)

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

Step 1 - Connect to your subscription

1. Sign in to your Azure subscription with the `az login` command and follow the on-screen directions. For more information about signing in, see [Get Started with Azure CLI 2.0](#).

```
az login
```

2. If you have more than one Azure subscription, list the subscriptions for the account.

```
az account list --all
```

3. Specify the subscription that you want to use.

```
az account set --subscription <replace_with_your_subscription_id>
```

Step 2 - Create and configure TestVNet1

1. Create a resource group.

```
az group create -n TestRG1 -l eastus
```

2. Create TestVNet1 and the subnets for TestVNet1. This example creates a virtual network named TestVNet1 and a subnet named FrontEnd.

```
az network vnet create -n TestVNet1 -g TestRG1 --address-prefix 10.11.0.0/16 -l eastus --subnet-name FrontEnd --subnet-prefix 10.11.0.0/24
```

3. Create an additional address space for the backend subnet. Notice that in this step, we specify both the address space that we created earlier, and the additional address space that we want to add. This is because

the [az network vnet update](#) command overwrites the previous settings. Make sure to specify all of the address prefixes when using this command.

```
az network vnet update -n TestVNet1 --address-prefixes 10.11.0.0/16 10.12.0.0/16 -g TestRG1
```

4. Create the backend subnet.

```
az network vnet subnet create --vnet-name TestVNet1 -n BackEnd -g TestRG1 --address-prefix 10.12.0.0/24
```

5. Create the gateway subnet. Notice that the gateway subnet is named 'GatewaySubnet'. This name is required. 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.

```
az network vnet subnet create --vnet-name TestVNet1 -n GatewaySubnet -g TestRG1 --address-prefix 10.12.255.0/27
```

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.

```
az network public-ip create -n VNet1GWIP -g TestRG1 --allocation-method Dynamic
```

7. Create the virtual network gateway for TestVNet1. VNet-to-VNet configurations require a RouteBased VpnType. If you run this command using the '--no-wait' parameter, you don't see any feedback or output. The '--no-wait' parameter allows the gateway to create in the background. It does not mean that the VPN gateway finishes creating immediately. Creating a gateway can often take 45 minutes or more, depending on the gateway SKU that you use.

```
az network vnet-gateway create -n VNet1GW -l eastus --public-ip-address VNet1GWIP -g TestRG1 --vnet TestVNet1 --gateway-type Vpn --sku VpnGw1 --vpn-type RouteBased --no-wait
```

Step 3 - Create and configure TestVNet4

1. Create a resource group.

```
az group create -n TestRG4 -l westus
```

2. Create TestVNet4.

```
az network vnet create -n TestVNet4 -g TestRG4 --address-prefix 10.41.0.0/16 -l westus --subnet-name FrontEnd --subnet-prefix 10.41.0.0/24
```

3. Create additional subnets for TestVNet4.

```
az network vnet update -n TestVNet4 --address-prefixes 10.41.0.0/16 10.42.0.0/16 -g TestRG4  
az network vnet subnet create --vnet-name TestVNet4 -n BackEnd -g TestRG4 --address-prefix 10.42.0.0/24
```

4. Create the gateway subnet.

```
az network vnet subnet create --vnet-name TestVNet4 -n GatewaySubnet -g TestRG4 --address-prefix  
10.42.255.0/27
```

5. Request a Public IP address.

```
az network public-ip create -n VNet4GWIP -g TestRG4 --allocation-method Dynamic
```

6. Create the TestVNet4 virtual network gateway.

```
az network vnet-gateway create -n VNet4GW -l westus --public-ip-address VNet4GWIP -g TestRG4 --vnet  
TestVNet4 --gateway-type Vpn --sku VpnGw1 --vpn-type RouteBased --no-wait
```

Step 4 - Create the connections

You now have two VNets with VPN gateways. The next step is to create VPN gateway connections between the virtual network gateways. If you used the examples above, your VNet gateways are in different resource groups. When gateways are in different resource groups, you need to identify and specify the resource IDs for each gateway when making a connection. If your VNets are in the same resource group, you can use the [second set of instructions](#) because you don't need to specify the resource IDs.

To connect VNets that reside in different resource groups

1. Get the Resource ID of VNet1GW from the output of the following command:

```
az network vnet-gateway show -n VNet1GW -g TestRG1
```

In the output, find the "id:" line. The values within the quotes are needed to create the connection in the next section. Copy these values to a text editor, such as Notepad, so that you can easily paste them when creating your connection.

Example output:

```
"activeActive": false,  
"bgpSettings": {  
    "asn": 65515,  
    "bgpPeeringAddress": "10.12.255.30",  
    "peerWeight": 0  
},  
"enableBgp": false,  
"etag": "W\"ecb42bc5-c176-44e1-802f-b0ce2962ac04\"",  
"gatewayDefaultSite": null,  
"gatewayType": "Vpn",  
"id": "/subscriptions/d6ff83d6-713d-41f6-a025-  
5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW",  
"ipConfigurations":
```

Copy the values after "**id**:" within the quotes.

```
"id": "/subscriptions/d6ff83d6-713d-41f6-a025-  
5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW"
```

2. Get the Resource ID of VNet4GW and copy the values to a text editor.

```
az network vnet-gateway show -n VNet4GW -g TestRG4
```

3. Create the TestVNet1 to TestVNet4 connection. In this step, you create the connection from TestVNet1 to TestVNet4. There is 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 takes a short while to complete.

```
az network vpn-connection create -n VNet1ToVNet4 -g TestRG1 --vnet-gateway1 /subscriptions/d6ff83d6-713d-41f6-a025-5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW -l eastus --shared-key "aabbcc" --vnet-gateway2 /subscriptions/d6ff83d6-713d-41f6-a025-5eb76334fda9/resourceGroups/TestRG4/providers/Microsoft.Network/virtualNetworkGateways/VNet4GW
```

4. 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. It takes a few minutes to establish the connection.

```
az network vpn-connection create -n VNet4ToVNet1 -g TestRG4 --vnet-gateway1 /subscriptions/d6ff83d6-713d-41f6-a025-5eb76334fda9/resourceGroups/TestRG4/providers/Microsoft.Network/virtualNetworkGateways/VNet4GW -l westus --shared-key "aabbcc" --vnet-gateway2 /subscriptions/d6ff83d6-713d-41f6-a025-5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1G
```

5. Verify your connections. See [Verify your connection](#).

To connect VNets that reside in the same resource group

1. Create the TestVNet1 to TestVNet4 connection. In this step, you create the connection from TestVNet1 to TestVNet4. Notice the resource groups are the same in the examples. You also see a shared key referenced in the examples. You can use your own values for the shared key, however, the shared key must match for both connections. Creating a connection takes a short while to complete.

```
az network vpn-connection create -n VNet1ToVNet4 -g TestRG1 --vnet-gateway1 VNet1GW -l eastus --shared-key "eeffgg" --vnet-gateway2 VNet4GW
```

2. 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. It takes a few minutes to establish the connection.

```
az network vpn-connection create -n VNet4ToVNet1 -g TestRG1 --vnet-gateway1 VNet4GW -l eastus --shared-key "eeffgg" --vnet-gateway2 VNet1GW
```

3. Verify your connections. See [Verify your connection](#).

Connect VNets that are in different subscriptions

In this scenario, you connect TestVNet1 and TestVNet5. The VNets reside different subscriptions. The subscriptions do not need to be associated with the same Active Directory tenant. The steps for this configuration add an additional VNet-to-VNet connection in order to connect TestVNet1 to TestVNet5.

Step 5 - Create and configure TestVNet1

These instructions continue from the steps in the preceding sections. You must complete [Step 1](#) and [Step 2](#) 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 (below).

Step 6 - Verify the IP address ranges

When creating additional connections, it's important to verify that the IP address space of the new virtual network does not overlap with any of your other VNet ranges or local network gateway ranges. 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, Subscription 5. This part may be performed by the administrator in a different organization that owns the subscription. To switch between subscriptions use 'az account list --all' to list the subscriptions available to your account, then use 'az account set --subscription' to switch to the subscription that you want to use.

1. Make sure you are connected to Subscription 5, then create a resource group.

```
az group create -n TestRG5 -l japaneast
```

2. Create TestVNet5.

```
az network vnet create -n TestVNet5 -g TestRG5 --address-prefix 10.51.0.0/16 -l japaneast --subnet-name  
FrontEnd --subnet-prefix 10.51.0.0/24
```

3. Add subnets.

```
az network vnet update -n TestVNet5 --address-prefixes 10.51.0.0/16 10.52.0.0/16 -g TestRG5  
az network vnet subnet create --vnet-name TestVNet5 -n BackEnd -g TestRG5 --address-prefix 10.52.0.0/24
```

4. Add the gateway subnet.

```
az network vnet subnet create --vnet-name TestVNet5 -n GatewaySubnet -g TestRG5 --address-prefix  
10.52.255.0/27
```

5. Request a public IP address.

```
az network public-ip create -n VNet5GWIP -g TestRG5 --allocation-method Dynamic
```

6. Create the TestVNet5 gateway

```
az network vnet-gateway create -n VNet5GW -l japaneast --public-ip-address VNet5GWIP -g TestRG5 --vnet TestVNet5 --gateway-type Vpn --sku VpnGw1 --vpn-type RouteBased --no-wait
```

Step 8 - Create the connections

This step is split into two CLI sessions marked as **[Subscription 1]**, and **[Subscription 5]** because the gateways are in the different subscriptions. To switch between subscriptions use 'az account list --all' to list the subscriptions available to your account, then use 'az account set --subscription' to switch to the subscription that you want to use.

1. **[Subscription 1]** Log in and connect to Subscription 1. Run the following command to get the name and ID of the Gateway from the output:

```
az network vnet-gateway show -n VNet1GW -g TestRG1
```

Copy the output for "id:". Send the ID and the name of the VNet gateway (VNet1GW) to the administrator of Subscription 5 via email or another method.

Example output:

```
"id": "/subscriptions/d6ff83d6-713d-41f6-a025-  
5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW"
```

2. **[Subscription 5]** Log in and connect to Subscription 5. Run the following command to get the name and ID of the Gateway from the output:

```
az network vnet-gateway show -n VNet5GW -g TestRG5
```

Copy the output for "id:". Send the ID and the name of the VNet gateway (VNet5GW) to the administrator of Subscription 1 via email or another method.

3. **[Subscription 1]** In this step, you create the connection from TestVNet1 to TestVNet5. You can use your own values for the shared key, however, the shared key must match for both connections. Creating a connection can take a short while to complete. Make sure you connect to Subscription 1.

```
az network vpn-connection create -n VNet1ToVNet5 -g TestRG1 --vnet-gateway1 /subscriptions/d6ff83d6-  
713d-41f6-a025-  
5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW -l eastus  
--shared-key "eeffggg" --vnet-gateway2 /subscriptions/e7e33b39-fe28-4822-b65c-  
a4db8bbff7cb/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW
```

4. **[Subscription 5]** This step is similar to the one above, except you are creating the connection from TestVNet5 to TestVNet1. Make sure that the shared keys match and that you connect to Subscription 5.

```
az network vpn-connection create -n VNet5ToVNet1 -g TestRG5 --vnet-gateway1 /subscriptions/e7e33b39-  
fe28-4822-b65c-  
a4db8bbff7cb/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW -l  
japaneast --shared-key "eeffggg" --vnet-gateway2 /subscriptions/d6ff83d6-713d-41f6-a025-  
5eb76334fda9/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW
```

Verify the connections

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 VPN 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 `az network vpn-connection show` command. In the example,'--name'refers to the name of the connection that you want to test. When the connection is in the process of being established, its connection status shows 'Connecting'. Once the connection is established, the status changes to 'Connected'.

```
az network vpn-connection show --name VNet1toSite2 --resource-group TestRG1
```

VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN Gateway connections. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud services that are not 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 are connected together.

Can I used 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 be using 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. For more information, see the [Virtual Machines documentation](#).
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).

Connect virtual networks from different deployment models using the portal

4/18/2018 • 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

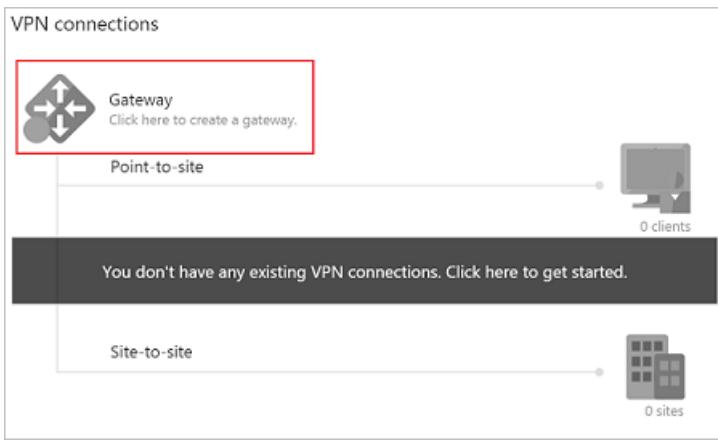
If you don't have a classic VNet and are running these steps as an exercise, you can create a VNet by using [this article](#) and the [Example](#) settings values from above.

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

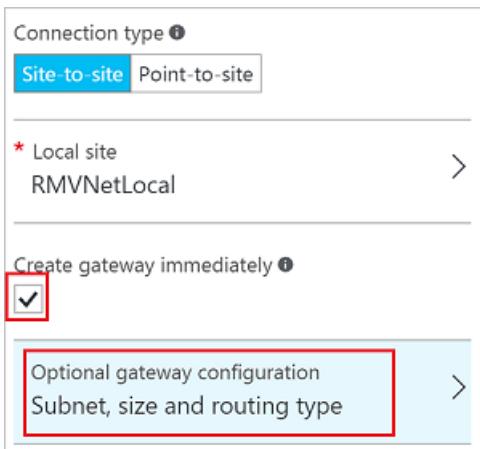
1. Navigate to **All resources** and locate the **ClassicVNet** in the list.
2. On the **Overview** page, in the **VPN connections** section, click **Gateway** to create a gateway.



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

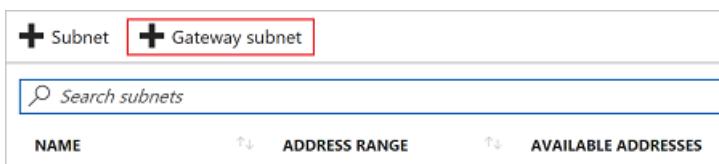
If you don't have a Resource Manager VNet and are running these steps as an exercise, create a virtual network with the steps in [Create a virtual network](#), using the example values.

2. Create a gateway subnet

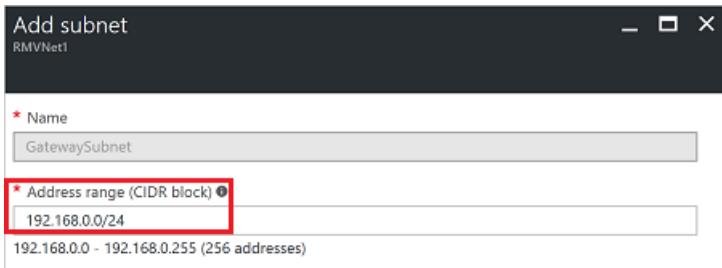
Example value: GatewaySubnet = 192.168.0.0/26

Before creating a virtual network gateway, you first need to create the gateway subnet. Create a gateway subnet with CIDR count of /28 or larger (/27, /26, etc.). If you are creating this as part of an exercise, you can use the Example values.

1. In the [portal](#), navigate to the Resource Manager virtual network for which you want to create a virtual network gateway.
2. In the **Settings** section of your VNet page, click **Subnets** to expand the Subnets page.
3. On the **Subnets** page, click **+Gateway subnet** to open the **Add subnet** page.



4. The **Name** for your subnet is automatically filled in with the value 'GatewaySubnet'. This value is required in order for Azure to recognize the subnet as the gateway subnet. Adjust the auto-filled **Address range** values to match your configuration requirements, then click **OK** at the bottom of the page to create the subnet.



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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

3. Create a virtual network gateway

Example values:

- Virtual network gateway name = RMGateway
- Gateway type = VPN
- VPN type = Route-based
- SKU = VpnGw1
- Location = East US
- Virtual network = RMVNet
- First IP configuration = rmgwpip

1. In the portal, on the left side, click + and type 'virtual network gateway' in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create** at the bottom of the page to open the **Create virtual network gateway** page.
2. On the **Create virtual network gateway** page, fill in the values for your virtual network gateway.

Create virtual network gateway

* Name

Gateway type [?](#)

VPN ExpressRoute

VPN type [?](#)

Route-based Policy-based

* SKU [?](#)

VpnGw1

Enable active-active mode [?](#)

* Virtual network [?](#) >

Choose a virtual network

* First IP configuration [?](#) >

Create gateway IP configuration

Configure BGP ASN

* Subscription

Windows Azure Internal Consumption

Resource group [?](#)

-

* Location [?](#)

West US

Pin to dashboard

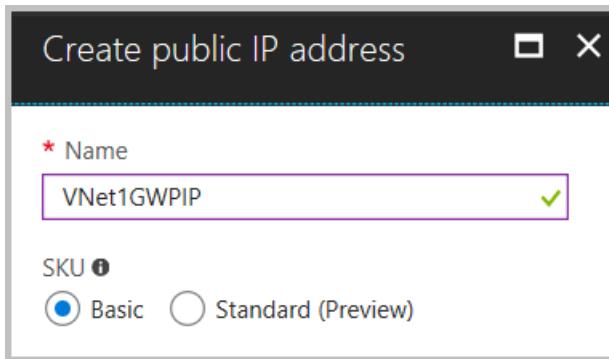
Create [Automation options](#)

Provisioning a virtual network gateway may take up to 45 minutes.

3. On the **Create virtual network gateway** page, specify the values for your virtual network gateway.

- **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **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](#).
- **Location:** You may need to scroll to see Location. Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, when you select a virtual network in the next step, it will not appear in the drop-down list.

- **Virtual network:** Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the 'Choose a virtual network' page. Select the VNet. If you don't see your VNet, make sure the Location field is pointing to the region in which your virtual network is located.
- **Gateway subnet address range:** You will only see this setting if you did not previously create a gateway subnet for your virtual network. If you previously created a valid gateway subnet, this setting will not appear.
- **First IP configuration:** The 'Choose public IP address' page creates a 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. VPN Gateway currently only supports *Dynamic* Public IP address allocation. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. 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.
 - First, click **Create gateway IP configuration** to open the 'Choose public IP address' page, then click **+Create new** to open the 'Create public IP address' page.
 - Next, input a **Name** for your public IP address. Leave the SKU as **Basic** unless there is a specific reason to change it to something else, then click **OK** at the bottom of this page to save your changes.



4. Verify the settings. You can select **Pin to dashboard** at the bottom of the page if you want your gateway to appear on the dashboard.
5. Click **Create** to begin creating the VPN gateway. The settings are validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.

After the gateway is created, 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. You can click the connected device (your virtual network gateway) to view more information.

4. 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
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. In the portal, from **All resources**, click **+Add**.
2. In the **Everything** page search box, type **Local network gateway**, then click to return a list of resources. Click **Local network gateway** to open the page, then click **Create** to open the **Create local network gateway** page.

The screenshot shows the 'Create local network gateway' dialog box. It contains the following fields:

- Name**: A required field with a red asterisk.
- IP address**: A required field with a red asterisk.
- Address space**: A section with a button to "Add additional address range".
- Configure BGP settings**: An optional checkbox.
- Subscription**: Set to "Windows Azure Internal Consumption".
- Resource group**: Options to "Create new" or "Use existing".
- Location**: Set to "East US".
- Pin to dashboard**: An optional checkbox.
- Create**: A blue button at the bottom left.
- Automation options**: A link at the bottom right.

3. On the **Create local network gateway page**, specify the values for your local network gateway.
 - **Name**: Specify a name for your local network gateway object. If possible, use something intuitive, such as **ClassicVNetLocal** or **TestVNet1Local**. This makes it easier for you to identify the local network gateway in the portal.
 - **IP address**: Specify a valid Public **IP address** for the VPN device or virtual network gateway to which you want to connect.
 - **If this local network represents an on-premises location**: Specify the Public IP address of the VPN device that you want to connect to. It cannot be behind NAT and has to be reachable by

Azure.

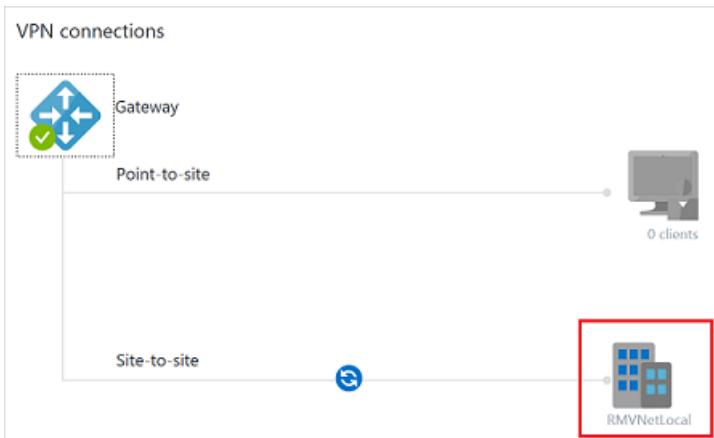
- **If this local network represents another VNet:** Specify the Public IP address that was assigned to the virtual network gateway for that VNet.
- **If you don't yet have the IP address:** You can make up a valid placeholder IP address, and then come back and modify this setting before connecting.
- **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 to which you connect.
- **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:** 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.

4. Click **Create** to create the local network gateway.

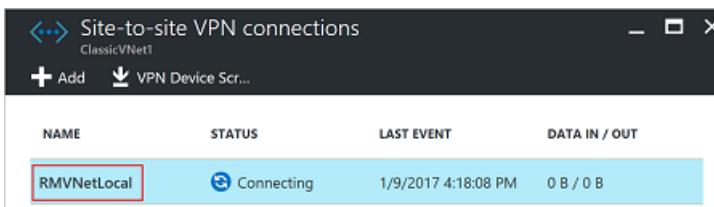
Section 3 - Modify the classic VNet local site settings

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.

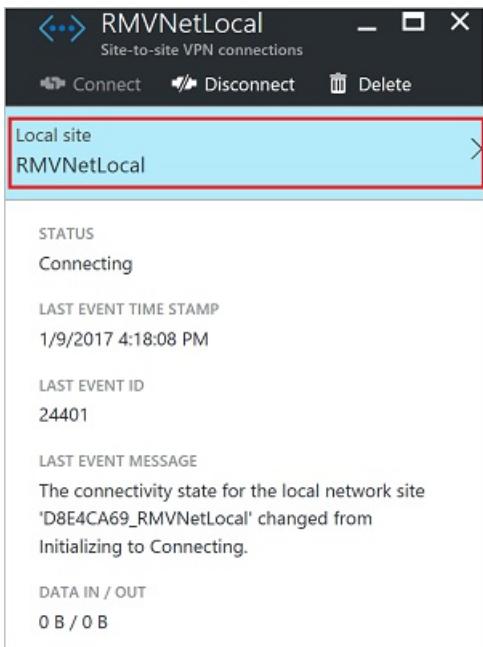
1. In the Azure portal, navigate to the classic virtual network.
2. On the page for your virtual network, click **Overview**.
3. In the **VPN connections** section, click the name of your local site in the graphic.



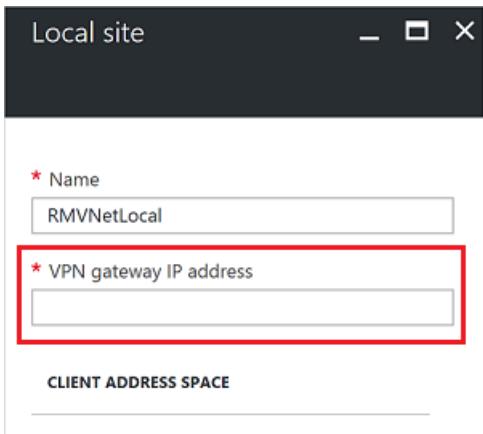
4. On the **Site-to-site VPN connections** page, click the name of the site.



5. On the connection page for your local site, click the name of the local site to open the **Local site** page.



- On the **Local site** page, replace the **VPN gateway IP address** with the IP address of the Resource Manager gateway.



- 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.
- On the **Add connection** page, name the connection. For example, 'RMtoClassic'.
- Site-to-Site** is already selected on this page.
- Select the virtual network gateway that you want to associate with this site.
- 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.
- 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-AzureRmAccount
```

Get a list of your Azure subscriptions.

```
Get-AzureRmSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzureRmSubscription -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
```

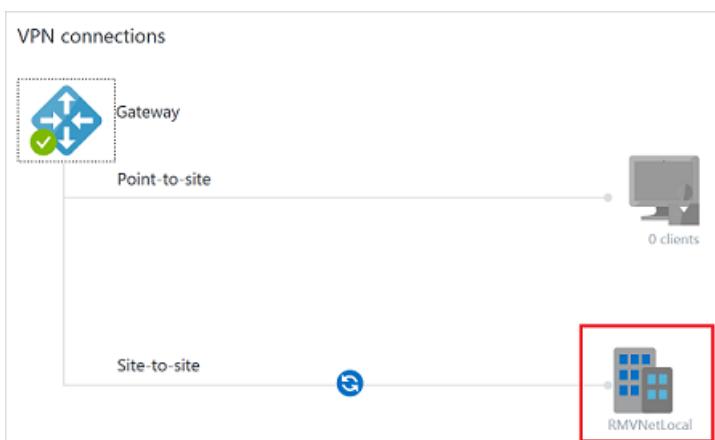
Section 6 - Verify your connections

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.

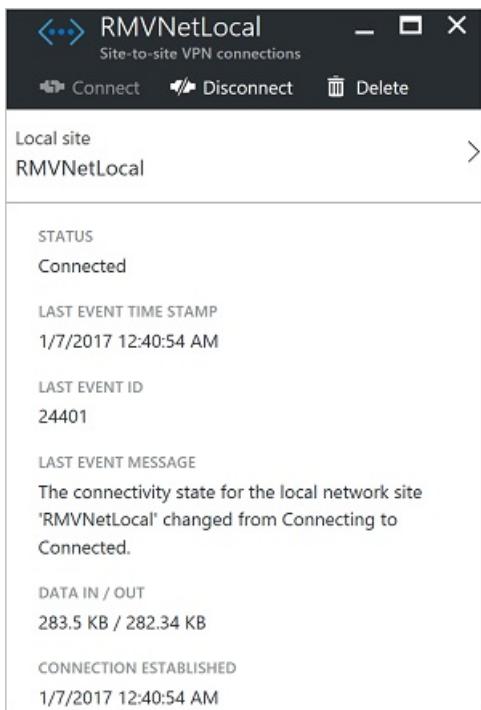
1. In the [Azure portal](#), click **All resources** and navigate to your classic virtual network.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.
3. On the VPN connections graphic, click the site.



4. On the **Site-to-site VPN connections** blade, view the information about your site.

NAME	STATUS	LAST EVENT	DATA IN / OUT
RMVNetLocal	Connected	1/7/2017 12:40:54 AM	177.97 KB / 177.72 KB

5. To view more information about the connection, click the name of the connection to open the **Site-to-site VPN Connection** blade.



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.

1. In the [Azure portal](#), click **All resources** and navigate to your virtual network gateway.
2. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
3. 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.

Essentials ^	
Resource group	Data in 2.35 KB
Status Connected	Data out 3.14 KB
Location East US	Virtual network
Subscription name	Virtual network gateway
Subscription ID	Local network gateway

VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN Gateway connections. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud services that are not 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 are connected together.

Can I used 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 be using 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.

Connect virtual networks from different deployment models using PowerShell

4/18/2018 • 14 minutes to read • [Edit Online](#)

This article helps you 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 use PowerShell, but you can also create this configuration using the Azure portal 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

The following steps walk you through the settings necessary to configure a dynamic or route-based gateway for each VNet and create a VPN connection between the gateways. This configuration does not support static or policy-based gateways.

Prerequisites

- Both VNets have already been created.
- 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.
- You have installed the latest PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information. Make sure you install both the Service Management (SM) and the Resource Manager (RM) cmdlets.

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 settings

VNet Name = ClassicVNet

Location = West US

Virtual Network Address Spaces = 10.0.0.0/24

Subnet-1 = 10.0.0.0/27

GatewaySubnet = 10.0.0.32/29

Local Network Name = RMVNetLocal

GatewayType = DynamicRouting

Resource Manager VNet settings

VNet Name = RMVNet

Resource Group = RG1

Virtual Network IP Address Spaces = 192.168.0.0/16
Subnet-1 = 192.168.1.0/24
GatewaySubnet = 192.168.0.0/26
Location = East US
Gateway public IP name = gwipip
Local Network Gateway = ClassicVNetLocal
Virtual Network Gateway name = RMGateway
Gateway IP addressing configuration = gwipconfig

Section 1 - Configure the classic VNet

1. Download your network configuration file

1. Log in to your Azure account in the PowerShell console with elevated rights. The following cmdlet prompts you for the login credentials for your Azure Account. After logging in, it downloads your account settings so that they are available to Azure PowerShell. The classic Service Management (SM) Azure PowerShell cmdlets are used in this section.

```
Add-AzureAccount
```

Get your Azure subscription.

```
Get-AzureSubscription
```

If you have more than one subscription, select the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionName "Name of subscription"
```

2. Export your Azure network configuration file by running the following command. You can change the location of the file to export to a different location if necessary.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

3. Open the .xml file that you downloaded to edit it. For an example of the network configuration file, see the [Network Configuration Schema](#).

2. Verify the gateway subnet

In the **VirtualNetworkSites** element, add a gateway subnet to your VNet if one has not already been created. When working with the network configuration file, the gateway subnet MUST be named "GatewaySubnet" or Azure cannot recognize and use it as a gateway subnet.

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 VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

Example:

```

<VirtualNetworkSites>
  <VirtualNetworkSite name="ClassicVNet" Location="West US">
    <AddressSpace>
      <AddressPrefix>10.0.0.0/24</AddressPrefix>
    </AddressSpace>
    <Subnets>
      <Subnet name="Subnet-1">
        <AddressPrefix>10.0.0.0/27</AddressPrefix>
      </Subnet>
      <Subnet name="GatewaySubnet">
        <AddressPrefix>10.0.0.32/29</AddressPrefix>
      </Subnet>
    </Subnets>
  </VirtualNetworkSite>
</VirtualNetworkSites>

```

3. Add the local network site

The local network site you add represents the RM VNet to which you want to connect. Add a **LocalNetworkSites** element to the file if one doesn't already exist. At this point in the configuration, the VPNGatewayAddress can be any valid public IP address because we haven't yet created the gateway for the Resource Manager VNet. Once we create the gateway, we replace this placeholder IP address with the correct public IP address that has been assigned to the RM gateway.

```

<LocalNetworkSites>
  <LocalNetworkSite name="RMVNetLocal">
    <AddressSpace>
      <AddressPrefix>192.168.0.0/16</AddressPrefix>
    </AddressSpace>
    <VPNGatewayAddress>13.68.210.16</VPNGatewayAddress>
  </LocalNetworkSite>
</LocalNetworkSites>

```

4. Associate the VNet with the local network site

In this section, we specify the local network site that you want to connect the VNet to. In this case, it is the Resource Manager VNet that you referenced earlier. Make sure the names match. This step does not create a gateway. It specifies the local network that the gateway will connect to.

```

<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="RMVNetLocal">
      <Connection type="IPsec" />
    </LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>

```

5. Save the file and upload

Save the file, then import it to Azure by running the following command. Make sure you change the file path as necessary for your environment.

```
Set-AzureVNetConfig -ConfigurationPath C:\AzureNet\NetworkConfig.xml
```

You will see a similar result showing that the import succeeded.

OperationDescription	OperationId	OperationStatus
Set-AzureVNetConfig	e0ee6e66-9167-cfa7-a746-7casb9	Succeeded

6. Create the gateway

Before running this example, refer to the network configuration file that you downloaded for the exact names that Azure expects to see. The network configuration file contains the values for your classic virtual networks.

Sometimes the names for classic VNets are changed in the network configuration file when creating classic VNet settings in the Azure portal due to the differences in the deployment models. For example, if you used the Azure portal to create a classic VNet named 'Classic VNet' and created it in a resource group named 'ClassicRG', the name that is contained in the network configuration file is converted to 'Group ClassicRG Classic VNet'. When specifying the name of a VNet that contains spaces, use quotation marks around the value.

Use the following example to create a dynamic routing gateway:

```
New-AzureVNetGateway -VNetName ClassicVNet -GatewayType DynamicRouting
```

You can check the status of the gateway by using the **Get-AzureVNetGateway** cmdlet.

Section 2 - Configure the RM VNet gateway

To create a VPN gateway for the RM VNet, follow the following instructions. Don't start the steps until after you have retrieved the public IP address for the classic VNet's gateway.

1. Log in to your Azure account in the PowerShell console. The following cmdlet prompts you for the login credentials for your Azure Account. After logging in, your account settings are downloaded so that they are available to Azure PowerShell.

```
Connect-AzureRmAccount
```

Get a list of your Azure subscriptions.

```
Get-AzureRmSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Name of subscription"
```

2. Create a local network gateway. In a virtual network, the local network gateway typically refers to your on-premises location. In this case, the local network gateway refers to your Classic VNet. Give it a name by which Azure can refer to it, and also specify the address space prefix. Azure uses the IP address prefix you specify to identify which traffic to send to your on-premises location. If you need to adjust the information here later, before creating your gateway, you can modify the values and run the sample again.

-Name is the name you want to assign to refer to the local network gateway.

-AddressPrefix is the Address Space for your classic VNet.

-GatewayIpAddress is the public IP address of the classic VNet's gateway. Be sure to change the following sample to reflect the correct IP address.

```
New-AzureRmLocalNetworkGateway -Name ClassicVNetLocal `  
-Location "West US" -AddressPrefix "10.0.0.0/24" `  
-GatewayIpAddress "n.n.n.n" -ResourceGroupName RG1
```

3. Request a public IP address to be allocated to the virtual network gateway for the Resource Manager VNet. You can't specify the IP address that you want to use. The IP address is dynamically allocated to the virtual network gateway. However, this does not mean the IP address changes. The only time the virtual network gateway IP address changes is when the gateway is deleted and recreated. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of the gateway.

In this step, we also set a variable that is used in a later step.

```
$ipaddress = New-AzureRmPublicIpAddress -Name gwpip `  
-ResourceGroupName RG1 -Location 'EastUS' `  
-AllocationMethod Dynamic
```

4. Verify that your virtual network has a gateway subnet. If no gateway subnet exists, add one. Make sure the gateway subnet is named *GatewaySubnet*.
5. Retrieve the subnet used for the gateway by running the following command. In this step, we also set a variable to be used in the next step.

-Name is the name of your Resource Manager VNet.

-ResourceGroupName is the resource group that the VNet is associated with. The gateway subnet must already exist for this VNet and must be named *GatewaySubnet* to work properly.

```
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet `  
-VirtualNetwork (Get-AzureRmVirtualNetwork -Name RMVNet -ResourceGroupName RG1)
```

6. Create the gateway IP addressing configuration. The gateway configuration defines the subnet and the public IP address to use. Use the following sample to create your gateway configuration.

In this step, the **-SubnetId** and **-PublicIpAddressId** parameters must be passed the id property from the subnet, and IP address objects, respectively. You can't use a simple string. These variables are set in the step to request a public IP and the step to retrieve the subnet.

```
$gwpipconfig = New-AzureRmVirtualNetworkGatewayIpConfig `  
-Name gwpipconfig -SubnetId $subnet.id `  
-PublicIpAddressId $ipaddress.id
```

7. Create the Resource Manager virtual network gateway by running the following command. The **-VpnType** must be *RouteBased*. It can take 45 minutes or more for the gateway to create.

```
New-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName RG1 `  
-Location "EastUS" -GatewaySKU Standard -GatewayType Vpn `  
-IpConfigurations $gwpipconfig `  
-EnableBgp $false -VpnType RouteBased
```

8. Copy the public IP address once the VPN gateway has been created. You use it when you configure the local network settings for your Classic VNet. You can use the following cmdlet to retrieve the public IP address. The public IP address is listed in the return as *IpAddress*.

```
Get-AzureRmPublicIpAddress -Name gwpip -ResourceGroupName RG1
```

Section 3 - Modify the classic VNet local site settings

In this section, you work with the classic VNet. You replace the placeholder IP address that you used when specifying the local site settings that will be used to connect to the Resource Manager VNet gateway.

1. Export the network configuration file.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

2. Using a text editor, modify the value for VPNGatewayAddress. Replace the placeholder IP address with the public IP address of the Resource Manager gateway and then save the changes.

```
<VPNGatewayAddress>13.68.210.16</VPNGatewayAddress>
```

3. Import the modified network configuration file to Azure.

```
Set-AzureVNetConfig -ConfigurationPath C:\AzureNet\NetworkConfig.xml
```

Section 4 - Create a connection between the gateways

Creating a connection between the gateways requires PowerShell. You may need to add your Azure Account to use the classic version of the PowerShell cmdlets. To do so, use **Add-AzureAccount**.

1. In the PowerShell console, set your shared key. Before running the cmdlets, refer to the network configuration file that you downloaded for the exact names that Azure expects to see. When specifying the name of a VNet that contains spaces, use single quotation marks around the value.

In following example, **-VNetName** is the name of the classic VNet and **-LocalNetworkSiteName** is the name you specified for the local network site. The **-SharedKey** is a value that you generate and specify. In the example, we used 'abc123', but you can generate and use something more complex. The important thing is that the value you specify here must be the same value that you specify in the next step when you create your connection. The return should show **Status: Successful**.

```
Set-AzureVNetGatewayKey -VNetName ClassicVNet `  
-LocalNetworkSiteName RMVNetLocal -SharedKey abc123
```

2. Create the VPN connection by running the following commands:

Set the variables.

```
$vnet01gateway = Get-AzureRMLocalNetworkGateway -Name ClassicVNetLocal -ResourceGroupName RG1  
$vnet02gateway = Get-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName RG1
```

Create the connection. Notice that the **-ConnectionType** is IPsec, not Vnet2Vnet.

```
New-AzureRmVirtualNetworkGatewayConnection -Name RM-Classic -ResourceGroupName RG1 `  
-Location "East US" -VirtualNetworkGateway1 `  
$vnet02gateway -LocalNetworkGateway2 `  
$vnet01gateway -ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

Section 5 - Verify your connections

To verify the connection from your classic VNet to your Resource Manager VNet

PowerShell

You can verify that your connection succeeded by using the 'Get-AzureVNetConnection' cmdlet.

1. Use the following cmdlet example, configuring the values to match your own. The name of the virtual network must be in quotes if it contains spaces.

```
Get-AzureVNetConnection "Group ClassicRG ClassicVNet"
```

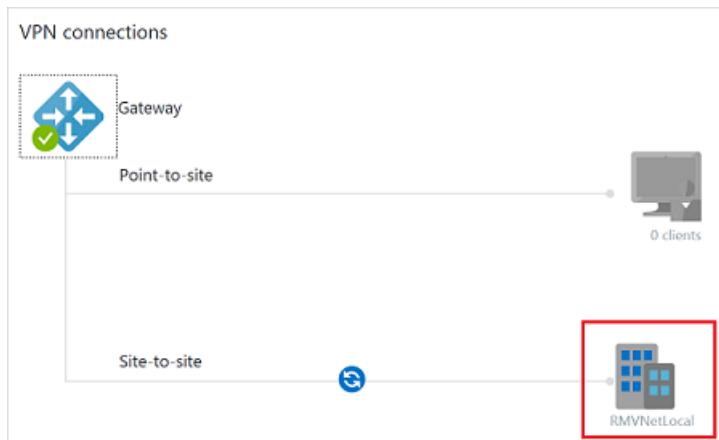
2. After the cmdlet has finished, view the values. In the example below, the Connectivity State shows as 'Connected' and you can see ingress and egress bytes.

```
ConnectivityState      : Connected
EgressBytesTransferred : 181664
IngressBytesTransferred : 182080
LastConnectionEstablished : 1/7/2016 12:40:54 AM
LastEventID           : 24401
LastEventMessage       : The connectivity state for the local network site 'RMVNetLocal' changed
from Connecting to
                           Connected.
LastEventTimeStamp     : 1/7/2016 12:40:54 AM
LocalNetworkSiteName   : RMVNetLocal
```

Azure portal

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.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.
3. On the VPN connections graphic, click the site.

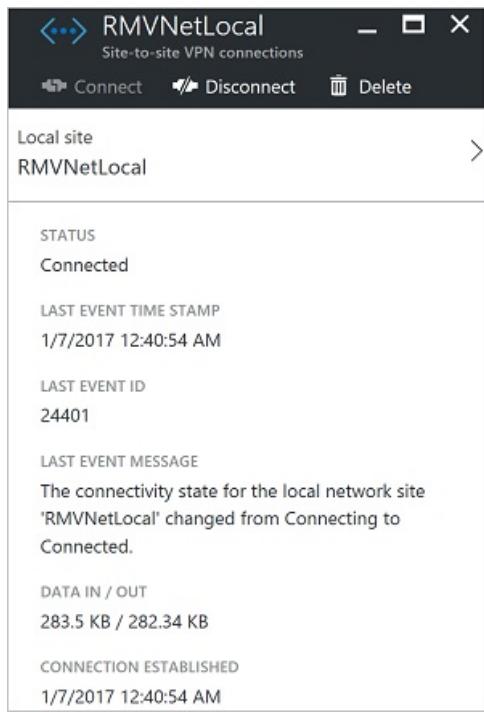


4. On the **Site-to-site VPN connections** blade, view the information about your site.

The screenshot shows the 'Site-to-site VPN connections' blade for the 'ClassicVNet' resource group. The blade title is 'Site-to-site VPN connections' with 'ClassicVNet' underneath. It includes 'Add' and 'VPN Device Scr...' buttons. The main table lists a single connection:

NAME	STATUS	LAST EVENT	DATA IN / OUT
RMVNetLocal	Connected	1/7/2017 12:40:54 AM	177.97 KB / 177.72 KB

5. To view more information about the connection, click the name of the connection to open the **Site-to-site VPN Connection** blade.



To verify the connection from your Resource Manager VNet to your classic VNet

PowerShell

You can verify that your connection succeeded by using the 'Get-AzureRmVirtualNetworkGatewayConnection' 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-AzureRmVirtualNetworkGatewayConnection -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}
```

Azure portal

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](#), click **All resources** and navigate to your virtual network gateway.
2. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
3. 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.

Essentials ^	
Resource group	Data in 2.35 KB
Status Connected	Data out 3.14 KB
Location East US	Virtual network
Subscription name	Virtual network gateway
Subscription ID	Local network gateway

VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN Gateway connections. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud

services that are not 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 are connected together.

Can I used 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 be using 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.

Configure ExpressRoute and Site-to-Site coexisting connections using PowerShell

9/7/2018 • 8 minutes to read • [Edit Online](#)

Configuring Site-to-Site VPN and ExpressRoute coexisting connections has several advantages:

- You can configure a Site-to-Site VPN as a secure failover path for ExpressRoute.
- Alternatively, you can use Site-to-Site VPNs to connect to sites that are not connected through ExpressRoute.

The steps to configure both scenarios are covered in this article. This article applies to the Resource Manager deployment model and uses PowerShell. You can also configure these scenarios using the Azure Portal, although documentation is not yet available.

NOTE

If you want to create a Site-to-Site VPN over an ExpressRoute circuit, please see [this article](#).

Limits and limitations

- **Transit routing is not supported.** You cannot route (via Azure) between your local network connected via Site-to-Site VPN and your local network connected via ExpressRoute.
- **Basic SKU gateway is not supported.** You must use a non-Basic SKU gateway for both the [ExpressRoute gateway](#) and the [VPN gateway](#).
- **Only route-based VPN gateway is supported.** You must use a route-based [VPN Gateway](#).
- **Static route should be configured for your VPN gateway.** If your local network is connected to both ExpressRoute and a Site-to-Site VPN, you must have a static route configured in your local network to route the Site-to-Site VPN connection to the public Internet.

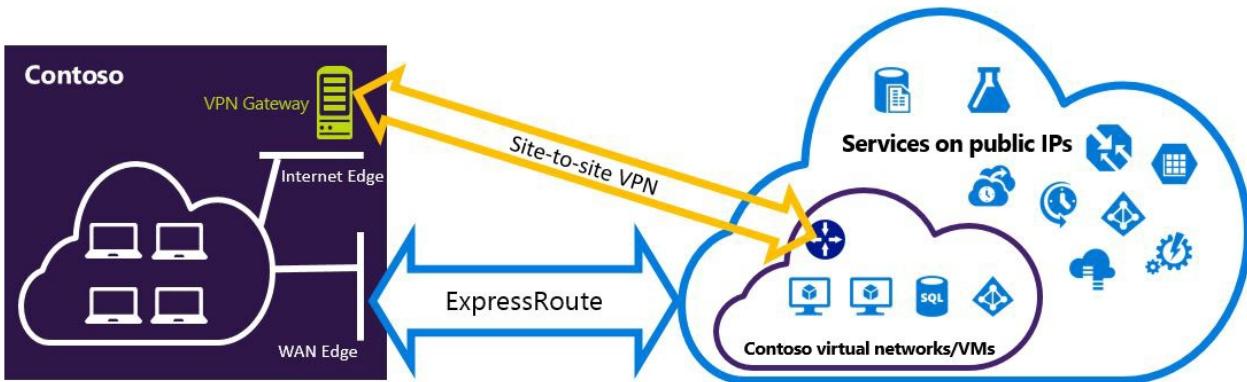
Configuration designs

Configure a Site-to-Site VPN as a failover path for ExpressRoute

You can configure a Site-to-Site VPN connection as a backup for ExpressRoute. This connection applies only to virtual networks linked to the Azure private peering path. There is no VPN-based failover solution for services accessible through Azure Microsoft peering. The ExpressRoute circuit is always the primary link. Data flows through the Site-to-Site VPN path only if the ExpressRoute circuit fails. To avoid asymmetrical routing, your local network configuration should also prefer the ExpressRoute circuit over the Site-to-Site VPN. You can prefer the ExpressRoute path by setting higher local preference for the routes received from the ExpressRoute.

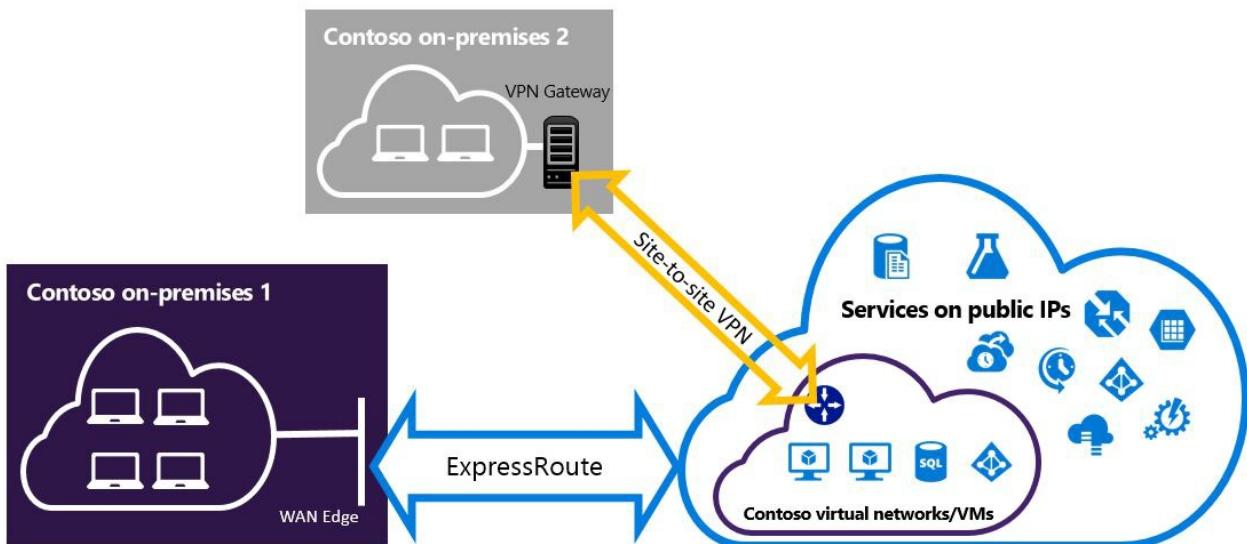
NOTE

While ExpressRoute circuit is preferred over Site-to-Site VPN when both routes are the same, Azure will use the longest prefix match to choose the route towards the packet's destination.



Configure a Site-to-Site VPN to connect to sites not connected through ExpressRoute

You can configure your network where some sites connect directly to Azure over Site-to-Site VPN, and some sites connect through ExpressRoute.



NOTE

You cannot configure a virtual network as a transit router.

Selecting the steps to use

There are two different sets of procedures to choose from. The configuration procedure that you select depends on whether you have an existing virtual network that you want to connect to, or you want to create a new virtual network.

- I don't have a VNet and need to create one.

If you don't already have a virtual network, this procedure walks you through creating a new virtual network using Resource Manager deployment model and creating new ExpressRoute and Site-to-Site VPN connections. To configure a virtual network, follow the steps in [To create a new virtual network and coexisting connections](#).

- I already have a Resource Manager deployment model VNet.

You may already have a virtual network in place with an existing Site-to-Site VPN connection or ExpressRoute connection. In this scenario if the gateway subnet mask is /28 or smaller (/28, /29, etc.), you

have to delete the existing gateway. The [To configure coexisting connections for an already existing VNet](#) section walks you through deleting the gateway, and then creating new ExpressRoute and Site-to-Site VPN connections.

If you delete and recreate your gateway, you will have downtime for your cross-premises connections. However, your VMs and services will still be able to communicate out through the load balancer while you configure your gateway if they are configured to do so.

To create a new virtual network and coexisting connections

This procedure walks you through creating a VNet and Site-to-Site and ExpressRoute connections that will coexist.

1. Install the latest version of the Azure PowerShell cmdlets. For information about installing the cmdlets, see [How to install and configure Azure PowerShell](#). The cmdlets that you use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Log in to your account and set up the environment.

```
Connect-AzureRmAccount  
Select-AzureRmSubscription -SubscriptionName 'yoursubscription'  
$location = "Central US"  
$resgrp = New-AzureRmResourceGroup -Name "ErVpnCoex" -Location $location  
$VNetASN = 65515
```

3. Create a virtual network including Gateway Subnet. For more information about creating a virtual network, see [Create a virtual network](#). For more information about creating subnets, see [Create a subnet](#)

IMPORTANT

The Gateway Subnet must be /27 or a shorter prefix (such as /26 or /25).

Create a new VNet.

```
$vnet = New-AzureRmVirtualNetwork -Name "CoexVnet" -ResourceGroupName $resgrp.ResourceGroupName -  
Location $location -AddressPrefix "10.200.0.0/16"
```

Add subnets.

```
Add-AzureRmVirtualNetworkSubnetConfig -Name "App" -VirtualNetwork $vnet -AddressPrefix "10.200.1.0/24"  
Add-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix  
"10.200.255.0/24"
```

Save the VNet configuration.

```
$vnet = Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

4. Next, create your Site-to-Site VPN gateway. For more information about the VPN gateway configuration, see [Configure a VNet with a Site-to-Site connection](#). The *GatewaySku* is only supported for *VpnGw1*, *VpnGw2*, *VpnGw3*, *Standard*, and *HighPerformance* VPN gateways. ExpressRoute-VPN Gateway coexist configurations are not supported on the Basic SKU. The *VpnType* must be *RouteBased*.

```

$gwSubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzureRmPublicIpAddress -Name "VPNGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -AllocationMethod Dynamic
$gwConfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "VPNGatewayIpConfig" -SubnetId $gwSubnet.Id -
PublicIpAddressId $gwIP.Id
New-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -IpConfigurations $gwConfig -GatewayType "Vpn" -VpnType "RouteBased" -GatewaySku
"VpnGw1"

```

Azure VPN gateway supports BGP routing protocol. You can specify ASN (AS Number) for that Virtual Network by adding the -Asn switch in the following command. Not specifying that parameter will default to AS number 65515.

```

$azureVpn = New-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -IpConfigurations $gwConfig -GatewayType "Vpn" -VpnType
"RouteBased" -GatewaySku "VpnGw1" -Asn $VNetASN

```

You can find the BGP peering IP and the AS number that Azure uses for the VPN gateway in \$azureVpn.BgpSettings.BgpPeeringAddress and \$azureVpn.BgpSettings.Asn. For more information, see [Configure BGP for Azure VPN gateway](#).

5. Create a local site VPN gateway entity. This command doesn't configure your on-premises VPN gateway. Rather, it allows you to provide the local gateway settings, such as the public IP and the on-premises address space, so that the Azure VPN gateway can connect to it.

If your local VPN device only supports static routing, you can configure the static routes in the following way:

```

$MyLocalNetworkAddress = @("10.100.0.0/16", "10.101.0.0/16", "10.102.0.0/16")
$localVpn = New-AzureRmLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -GatewayIpAddress *<Public IP>* -AddressPrefix
$MyLocalNetworkAddress

```

If your local VPN device supports the BGP and you want to enable dynamic routing, you need to know the BGP peering IP and the AS number that your local VPN device uses.

```

$localVPNPublicIP = "<Public IP>"
$localBGPPeeringIP = "<Private IP for the BGP session>"
$localBGPASN = "<ASN>"
$localAddressPrefix = $localBGPPeeringIP + "/32"
$localVpn = New-AzureRmLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -GatewayIpAddress $localVPNPublicIP -AddressPrefix
$localAddressPrefix -BgpPeeringAddress $localBGPPeeringIP -Asn $localBGPASN

```

6. Configure your local VPN device to connect to the new Azure VPN gateway. For more information about VPN device configuration, see [VPN Device Configuration](#).
7. Link the Site-to-Site VPN gateway on Azure to the local gateway.

```

$azureVpn = Get-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName
New-AzureRmVirtualNetworkGatewayConnection -Name "VPNConnection" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -VirtualNetworkGateway1 $azureVpn -LocalNetworkGateway2
$localVpn -ConnectionType IPsec -SharedKey <yourkey>

```

8. If you are connecting to an existing ExpressRoute circuit, skip steps 8 & 9 and, jump to step 10. Configure

ExpressRoute circuits. For more information about configuring ExpressRoute circuit, see [create an ExpressRoute circuit](#).

9. Configure Azure private peering over the ExpressRoute circuit. For more information about configuring Azure private peering over the ExpressRoute circuit, see [configure peering](#)
10. Create an ExpressRoute gateway. For more information about the ExpressRoute gateway configuration, see [ExpressRoute gateway configuration](#). The GatewaySKU must be *Standard*, *HighPerformance*, or *UltraPerformance*.

```
$gwSubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzureRmPublicIpAddress -Name "ERGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -AllocationMethod Dynamic
$gwConfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "ERGatewayIpConfig" -SubnetId $gwSubnet.Id -
PublicIpAddressId $gwIP.Id
$gw = New-AzureRmVirtualNetworkGateway -Name "ERGateway" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -IpConfigurations $gwConfig -GatewayType "ExpressRoute" -GatewaySku Standard
```

11. Link the ExpressRoute gateway to the ExpressRoute circuit. After this step has been completed, the connection between your on-premises network and Azure, through ExpressRoute, is established. For more information about the link operation, see [Link VNets to ExpressRoute](#).

```
$ckt = Get-AzureRmExpressRouteCircuit -Name "YourCircuit" -ResourceGroupName "YourCircuitResourceGroup"
New-AzureRmVirtualNetworkGatewayConnection -Name "ERConnection" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -VirtualNetworkGateway1 $gw -PeerId $ckt.Id -
ConnectionType ExpressRoute
```

To configure coexisting connections for an already existing VNet

If you have a virtual network that has only one virtual network gateway (let's say, Site-to-Site VPN gateway) and you want to add another gateway of a different type (let's say, ExpressRoute gateway), check the gateway subnet size. If the gateway subnet is /27 or larger, you can skip the steps below and follow the steps in the previous section to add either a Site-to-Site VPN gateway or an ExpressRoute gateway. If the gateway subnet is /28 or /29, you have to first delete the virtual network gateway and increase the gateway subnet size. The steps in this section show you how to do that.

1. You'll need to install the latest version of the Azure PowerShell cmdlets. For more information about installing cmdlets, see [How to install and configure Azure PowerShell](#). The cmdlets that you use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Delete the existing ExpressRoute or Site-to-Site VPN gateway.

```
Remove-AzureRmVirtualNetworkGateway -Name <yourgatewayname> -ResourceGroupName <yourresourcegroup>
```

3. Delete Gateway Subnet.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup> Remove-
AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet
```

4. Add a Gateway Subnet that is /27 or larger.

NOTE

If you don't have enough IP addresses left in your virtual network to increase the gateway subnet size, you need to add more IP address space.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Add-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix
"10.200.255.0/24"
```

Save the VNet configuration.

```
$vnet = Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

5. At this point, you have a virtual network with no gateways. To create new gateways and set up the connections, follow the steps in the previous section.

To add point-to-site configuration to the VPN gateway

You can follow the steps below to add Point-to-Site configuration to your VPN gateway in a co-existence setup.

1. Add VPN Client address pool.

```
$azureVpn = Get-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName
Set-AzureRmVirtualNetworkGatewayVpnClientConfig -VirtualNetworkGateway $azureVpn -VpnClientAddressPool
"10.251.251.0/24"
```

2. Upload the VPN root certificate to Azure for your VPN gateway. In this example, it's assumed that the root certificate is stored in the local machine where the following PowerShell cmdlets are run.

```
$p2sCertFullName = "RootErVpnCoexP2S.cer"
$p2sCertMatchName = "RootErVpnCoexP2S"
$p2sCertToUpload=get-childitem Cert:\CurrentUser\My | Where-Object {$_.Subject -match $p2sCertMatchName}
if ($p2sCertToUpload.count -eq 1){write-host "cert found"} else {write-host "cert not found" exit}
$p2sCertData = [System.Convert]::ToBase64String($p2sCertToUpload.RawData) Add-
AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $p2sCertFullName -
VirtualNetworkGatewayname $azureVpn.Name -ResourceGroupName $resgrp.ResourceGroupName -PublicCertData
$p2sCertData
```

For more information on Point-to-Site VPN, see [Configure a Point-to-Site connection](#).

Next steps

For more information about ExpressRoute, see the [ExpressRoute FAQ](#).

Add a Site-to-Site connection to a VNet with an existing VPN gateway connection

2/15/2018 • 3 minutes to read • [Edit Online](#)

This article helps you add Site-to-Site (S2S) connections to a VPN gateway that has an existing connection by using the Azure portal. This type of connection is often referred to as a "multi-site" configuration. You can add a S2S connection to a VNet that already has a S2S connection, Point-to-Site connection, or VNet-to-VNet connection. There are some limitations when adding connections. Check the [Before you begin](#) section in this article to verify before you start your configuration.

This article applies to Resource Manager VNets that have a RouteBased VPN gateway. These steps do not apply to ExpressRoute/Site-to-Site coexisting connection configurations. See [ExpressRoute/S2S coexisting connections](#) for information about coexisting connections.

Deployment models and methods

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

We update this table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from this table.

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Article	Supported
Classic	Not Supported	Article

Before you begin

Verify the following items:

- You are not creating an ExpressRoute/S2S coexisting connection.
- You have a virtual network that was created using the Resource Manager deployment model with an existing connection.
- The virtual network gateway for your VNet is RouteBased. If you have a PolicyBased VPN gateway, you must delete the virtual network gateway and create a new VPN gateway as RouteBased.
- None of the address ranges overlap for any of the VNets that this VNet is connecting to.
- You have compatible VPN device and someone who is able to configure it. See [About VPN Devices](#). If you aren't familiar with configuring your VPN device, or 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.
- You have an externally facing public IP address for your VPN device. This IP address cannot be located behind a NAT.

Part 1 - Configure a connection

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.

2. Click **All resources** and locate your **virtual network gateway** from the list of resources and click it.
3. On the **Virtual network gateway** page, click **Connections**.

The screenshot shows the 'RMGateway - Connections' page for a virtual network gateway. The left sidebar contains a search bar and links to Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. Below these are sections for SETTINGS (Connections, Point-to-site configuration, Properties, Locks, Automation script) and SUPPORT + TROUBLESHOOTING (New support request). The 'Connections' link under SETTINGS is highlighted with a blue background.

4. On the **Connections** page, click **+Add**.

The screenshot shows the 'Add' page for creating a new connection. At the top is a large 'Add' button with a plus sign. Below it is a search bar labeled 'Search connections'. A table lists the connection details:

NAME	STATUS	CONNECTION TYPE	PEER	...
Site1	Succeeded	Site-to-site (IPsec)	Site1	...

5. On the **Add connection** page, fill out the following fields:

- **Name:** The name you want to give to the site you are creating the connection to.
- **Connection type:** Select **Site-to-site (IPsec)**.

 Add connection — X

RMGateway

* Name
Site2 ✓

Connection type ⓘ
Site-to-site (IPsec) ▾

* Virtual network gateway ⓘ 
RMGateway

* Local network gateway ⓘ >
Choose a local network gateway

* Shared key (PSK) ⓘ

Subscription ⓘ
Windows Azure Internal Consumption ▾

Resource group ⓘ 
RG1
Create new

Location ⓘ
East US ▾

Part 2 - Add a local network gateway

1. Click **Local network gateway *Choose a local network gateway***. This will open the **Choose local network gateway** page.

The screenshot shows two adjacent windows. The left window is titled 'Add connection' and has 'RMGateway' in its top right corner. It contains fields for 'Name' (Site2), 'Connection type' (Site-to-site (IPsec)), 'Virtual network gateway' (RMGateway), and 'Local network gateway' (with a link to 'Choose a local network gateway'). The right window is titled 'Choose local network...' and lists 'Create new' and 'ClassicVNetLocal RG1'.

2. Click **Create new** to open the **Create local network gateway** page.

The screenshot shows two adjacent windows. The left window is titled 'Choose local network...' and has 'Create new' highlighted. The right window is titled 'Create local network g...' and contains fields for 'Name', 'IP address', and 'Address space' (with a link to 'Add additional address range').

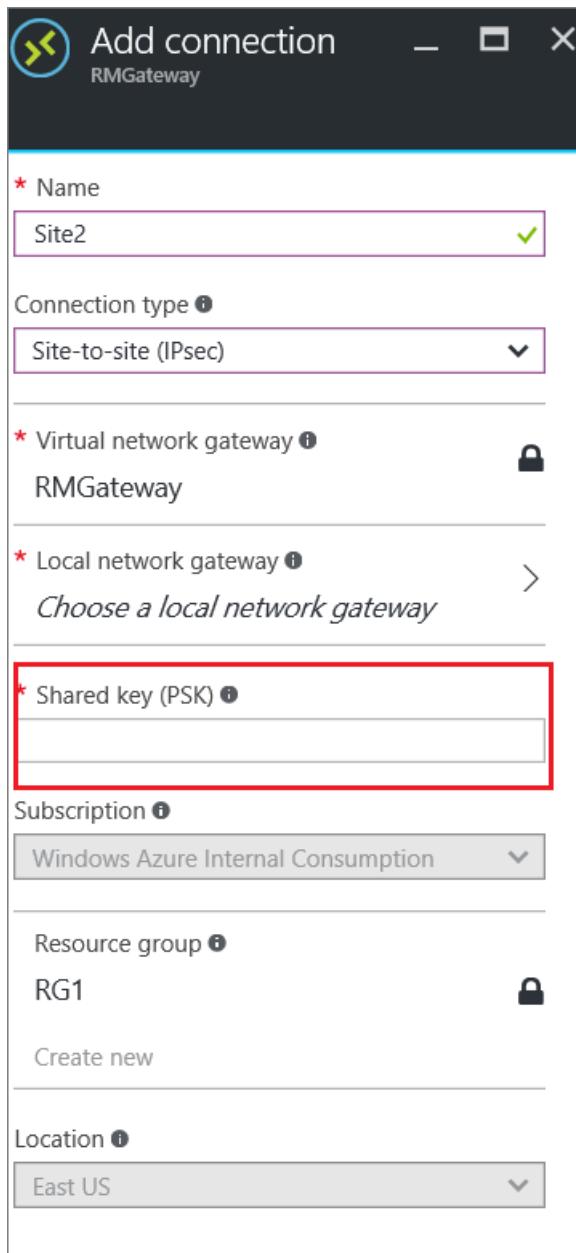
3. On the **Create local network gateway** page, fill out the following fields:

- **Name:** The name you want to give to the local network gateway resource.
- **IP address:** The public IP address of the VPN device on the site that you want to connect to.
- **Address space:** The address space that you want to be routed to the new local network site.

4. Click **OK** on the **Create local network gateway** page to save the changes.

Part 3 - Add the shared key and create the connection

1. On the **Add connection** page, add the shared key that you want to use to create your connection. You can either get the shared key from your VPN device, or make one up here and then configure your VPN device to use the same shared key. The important thing is that the keys are exactly the same.



- At the bottom of the page, click **OK** to create the connection.

Part 4 - Verify the VPN connection

You can verify that your connection succeeded by using the 'Get-AzureRmVirtualNetworkGatewayConnection' cmdlet, with or without '-Debug'.

- 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-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

- 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
```

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See the virtual machines [learning path](#) for more information.

Connect Azure VPN gateways to multiple on-premises policy-based VPN devices using PowerShell

4/18/2018 • 6 minutes to read • [Edit Online](#)

This article helps you configure an Azure route-based VPN gateway to connect to multiple on-premises policy-based VPN devices leveraging custom IPsec/IKE policies on S2S VPN connections.

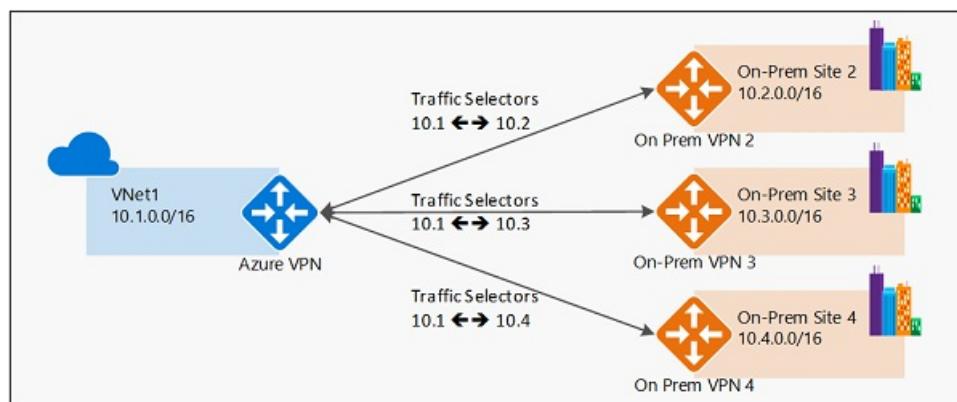
About policy-based and route-based VPN gateways

Policy- vs. route-based VPN devices differ in how the IPsec traffic selectors are set on a connection:

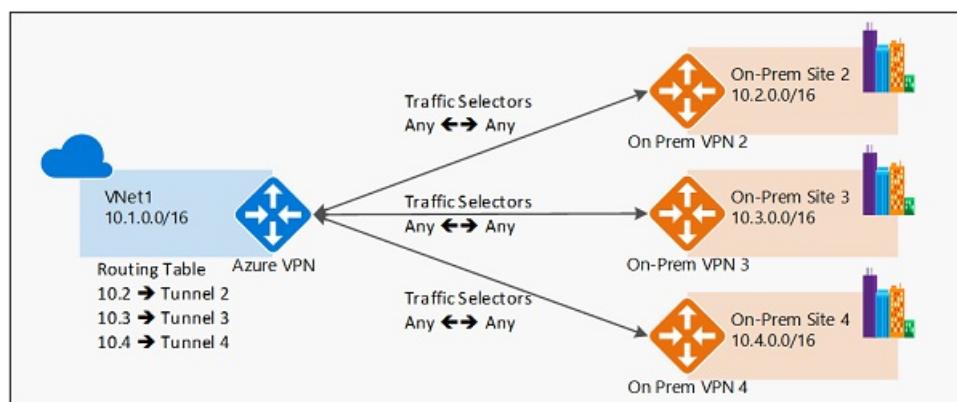
- **Policy-based** VPN devices use the combinations of prefixes from both networks to define how traffic is encrypted/decrypted through IPsec tunnels. It is typically built on firewall devices that perform packet filtering. IPsec tunnel encryption and decryption are added to the packet filtering and processing engine.
- **Route-based** VPN devices use any-to-any (wildcard) traffic selectors, and let routing/forwarding tables direct traffic to different IPsec tunnels. It is typically built on router platforms where each IPsec tunnel is modeled as a network interface or VTI (virtual tunnel interface).

The following diagrams highlight the two models:

Policy-based VPN example



Route-based VPN example



Azure support for policy-based VPN

Currently, Azure supports both modes of VPN gateways: route-based VPN gateways and policy-based VPN gateways. They are built on different internal platforms, which result in different specifications:

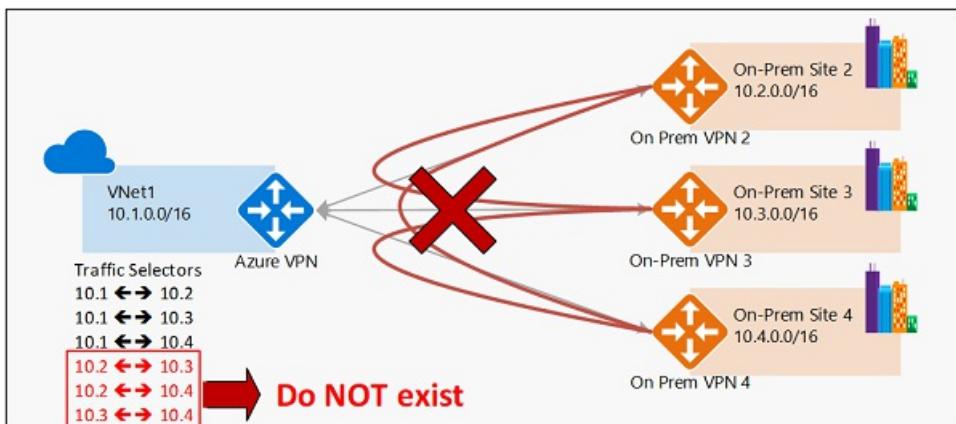
	POLICYBASED VPN GATEWAY	ROUTEBASED VPN GATEWAY
Azure Gateway SKU	Basic	Basic, Standard, HighPerformance, VpnGw1, VpnGw2, VpnGw3
IKE version	IKEv1	IKEv2
Max. S2S connections	1	Basic/Standard: 10 HighPerformance: 30

With the custom IPsec/IKE policy, you can now configure Azure route-based VPN gateways to use prefix-based traffic selectors with option "**PolicyBasedTrafficSelectors**", to connect to on-premises policy-based VPN devices. This capability allows you to connect from an Azure virtual network and VPN gateway to multiple on-premises policy-based VPN/firewall devices, removing the single connection limit from the current Azure policy-based VPN gateways.

IMPORTANT

1. To enable this connectivity, your on-premises policy-based VPN devices must support **IKEv2** to connect to the Azure route-based VPN gateways. Check your VPN device specifications.
2. The on-premises networks connecting through policy-based VPN devices with this mechanism can only connect to the Azure virtual network; **they cannot transit to other on-premises networks or virtual networks via the same Azure VPN gateway**.
3. The configuration option is part of the custom IPsec/IKE connection policy. If you enable the policy-based traffic selector option, you must specify the complete policy (IPsec/IKE encryption and integrity algorithms, key strengths, and SA lifetimes).

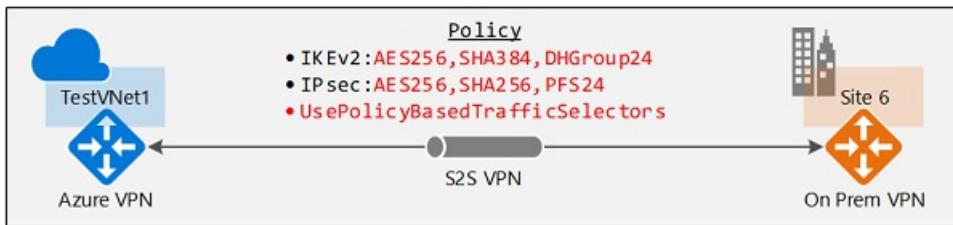
The following diagram shows why transit routing via Azure VPN gateway doesn't work with the policy-based option:



As shown in the diagram, the Azure VPN gateway has traffic selectors from the virtual network to each of the on-premises network prefixes, but not the cross-connection prefixes. For example, on-premises site 2, site 3, and site 4 can each communicate to VNet1 respectively, but cannot connect via the Azure VPN gateway to each other. The diagram shows the cross-connect traffic selectors that are not available in the Azure VPN gateway under this configuration.

Configure policy-based traffic selectors on a connection

The instructions in this article follow the same example as described in [Configure IPsec/IKE policy for S2S or VNet-to-VNet connections](#) to establish a S2S VPN connection. This is shown in the following diagram:



The workflow to enable this connectivity:

1. Create the virtual network, VPN gateway, and local network gateway for your cross-premises connection
2. Create an IPsec/IKE policy
3. Apply the policy when you create a S2S or VNet-to-VNet connection, and **enable the policy-based traffic selectors** on the connection
4. If the connection is already created, you can apply or update the policy to an existing connection

Enable policy-based traffic selectors on a connection

Make sure you have completed [Part 3 of the Configure IPsec/IKE policy article](#) for this section. The following example uses the same parameters and steps:

Step 1 - Create the virtual network, VPN gateway, and local network gateway

1. Declare your variables & connect to your subscription

For this exercise, we start by declaring our variables. Be sure to replace the values with your own when configuring for production.

```
$Sub1      = "<YourSubscriptionName>"  
$RG1       = "TestPolicyRG1"  
$Location1 = "East US 2"  
$VNetName1 = "TestVNet1"  
$FESubName1 = "FrontEnd"  
$BESubName1 = "Backend"  
$GWSubName1 = "GatewaySubnet"  
$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"  
$DNS1       = "8.8.8.8"  
$GWName1    = "VNet1GW"  
$GW1IPName1 = "VNet1GWIP1"  
$GW1IPconf1 = "gw1ipconf1"  
$Connection16 = "VNet1toSite6"  
  
$LNGName6   = "Site6"  
$LNGPrefix61 = "10.61.0.0/16"  
$LNGPrefix62 = "10.62.0.0/16"  
$LNGIP6     = "131.107.72.22"
```

To use the Resource Manager cmdlets, make sure you switch to PowerShell mode. For more information, see [Using Windows PowerShell with Resource Manager](#).

Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzureRmAccount  
Select-AzureRmSubscription -SubscriptionName $Sub1  
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

2. Create the virtual network, VPN gateway, and local network gateway

The following example creates the virtual network, TestVNet1 with three subnets, and the VPN gateway. 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.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1

$gw1pip1    = New-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$vnet1      = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1    = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gw1ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress
$gw1pip1

New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -
IpConfigurations $gw1ipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance

New-AzureRmLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1 -Location $Location1 -
GatewayIpAddress $LNGIP6 -AddressPrefix $LNGPrefix61,$LNGPrefix62
```

Step 2 - Create a S2S VPN connection with an IPsec/IKE policy

1. Create an IPsec/IKE policy

IMPORTANT

You need to create an IPsec/IKE policy in order to enable "UsePolicyBasedTrafficSelectors" option on the connection.

The following example creates an IPsec/IKE policy with these algorithms and parameters:

- IKEv2: AES256, SHA384, DHGroup24
- IPsec: AES256, SHA256, PFS24, SA Lifetime 3600 seconds & 2048KB

```
$ipsecpolicy6 = New-AzureRmIpsecPolicy -IkeEncryption AES256 -IkeIntegrity SHA384 -DhGroup DHGroup24 -
IpsecEncryption AES256 -IpsecIntegrity SHA256 -PfsGroup PFS24 -SALifeTimeSeconds 3600 -SADataSizeKilobytes
2048
```

2. Create the S2S VPN connection with policy-based traffic selectors and IPsec/IKE policy

Create an S2S VPN connection and apply the IPsec/IKE policy created in the previous step. Be aware of the additional parameter "-UsePolicyBasedTrafficSelectors \$True" which enables policy-based traffic selectors on the connection.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng6 = Get-AzureRmLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1 -
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng6 -Location $Location1 -ConnectionType IPsec -
UsePolicyBasedTrafficSelectors $True -IpsecPolicies $ipsecpolicy6 -SharedKey 'AzureA1b2C3'
```

After completing the steps, the S2S VPN connection will use the IPsec/IKE policy defined, and enable policy-based traffic selectors on the connection. You can repeat the same steps to add more connections to additional on-premises policy-based VPN devices from the same Azure VPN gateway.

Update policy-based traffic selectors for a connection

The last section shows you how to update the policy-based traffic selectors option for an existing S2S VPN connection.

1. Get the connection

Get the connection resource.

```
$RG1      = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6 = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1
```

2. Check the policy-based traffic selectors option

The following line shows whether the policy-based traffic selectors are used for the connection:

```
$connection6.UsePolicyBasedTrafficSelectors
```

If the line returns "**True**", then policy-based traffic selectors are configured on the connection; otherwise it returns "**False**".

3. Update the policy-based traffic selectors on a connection

Once you obtain the connection resource, you can enable or disable the option.

Disable UsePolicyBasedTrafficSelectors

The following example disables the policy-based traffic selectors option, but leaves the IPsec/IKE policy unchanged:

```
$RG1      = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6 = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -
    UsePolicyBasedTrafficSelectors $False
```

Enable UsePolicyBasedTrafficSelectors

The following example enables the policy-based traffic selectors option, but leaves the IPsec/IKE policy unchanged:

```
$RG1      = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6 = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -
    UsePolicyBasedTrafficSelectors $True
```

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Also review [Configure IPsec/IKE policy for S2S VPN or VNet-to-VNet connections](#) for more details on custom IPsec/IKE policies.

Configure IPsec/IKE policy for S2S VPN or VNet-to-VNet connections

4/18/2018 • 12 minutes to read • [Edit Online](#)

This article walks you through the steps to configure IPsec/IKE policy for Site-to-Site VPN or VNet-to-VNet connections using the Resource Manager deployment model and PowerShell.

About IPsec and IKE policy parameters for Azure VPN gateways

IPsec and IKE protocol standard supports a wide range of cryptographic algorithms in various combinations. Refer to [About cryptographic requirements and Azure VPN gateways](#) to see how this can help ensuring cross-premises and VNet-to-VNet connectivity satisfy your compliance or security requirements.

This article provides instructions to create and configure an IPsec/IKE policy and apply to a new or existing connection:

- [Part 1 - Workflow to create and set IPsec/IKE policy](#)
- [Part 2 - Supported cryptographic algorithms and key strengths](#)
- [Part 3 - Create a new S2S VPN connection with IPsec/IKE policy](#)
- [Part 4 - Create a new VNet-to-VNet connection with IPsec/IKE policy](#)
- [Part 5 - Manage \(create, add, remove\) IPsec/IKE policy for a connection](#)

IMPORTANT

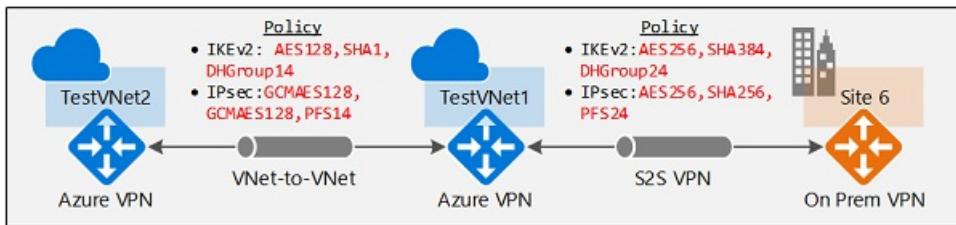
1. Note that IPsec/IKE policy only works on the following gateway SKUs:
 - **VpnGw1, VpnGw2, VpnGw3** (route-based)
 - **Standard** and **HighPerformance** (route-based)
2. You can only specify **one** policy combination for a given connection.
3. You must specify all algorithms and parameters for both IKE (Main Mode) and IPsec (Quick Mode). Partial policy specification is not allowed.
4. Consult with your VPN device vendor specifications to ensure the policy is supported on your on-premises VPN devices. S2S or VNet-to-VNet connections cannot establish if the policies are incompatible.

Part 1 - Workflow to create and set IPsec/IKE policy

This section outlines the workflow to create and update IPsec/IKE policy on a S2S VPN or VNet-to-VNet connection:

1. Create a virtual network and a VPN gateway
2. Create a local network gateway for cross premises connection, or another virtual network and gateway for VNet-to-VNet connection
3. Create an IPsec/IKE policy with selected algorithms and parameters
4. Create a connection (IPsec or VNet2VNet) with the IPsec/IKE policy
5. Add/update/remove an IPsec/IKE policy for an existing connection

The instructions in this article helps you set up and configure IPsec/IKE policies as shown in the diagram:



Part 2 - Supported cryptographic algorithms & key strengths

The following table lists the supported cryptographic algorithms and key strengths configurable by the customers:

IPSEC/IKEV2	OPTIONS
IKEv2 Encryption	AES256, AES192, AES128, DES3, DES
IKEv2 Integrity	SHA384, SHA256, SHA1, MD5
DH Group	DHGroup24, ECP384, ECP256, DHGroup14, DHGroup2048, DHGroup2, DHGroup1, None
IPsec Encryption	GCMAES256, GCMAES192, GCMAES128, AES256, AES192, AES128, DES3, DES, None
IPsec Integrity	GCMAES256, GCMAES192, GCMAES128, SHA256, SHA1, MD5
PFS Group	PFS24, ECP384, ECP256, PFS2048, PFS2, PFS1, None
QM SA Lifetime	(Optional: default values are used if not specified) Seconds (integer; min. 300/default 27000 seconds) KBytes (integer; min. 1024/default 102400000 KBytes)
Traffic Selector	UsePolicyBasedTrafficSelectors** (\$True/\$False; Optional , default \$False if not specified)

IMPORTANT

1. Your on-premises VPN device configuration must match or contain the following algorithms and parameters that you specify on the Azure IPsec/IKE policy:
 - IKE encryption algorithm (Main Mode / Phase 1)
 - IKE integrity algorithm (Main Mode / Phase 1)
 - DH Group (Main Mode / Phase 1)
 - IPsec encryption algorithm (Quick Mode / Phase 2)
 - IPsec integrity algorithm (Quick Mode / Phase 2)
 - PFS Group (Quick Mode / Phase 2)
 - Traffic Selector (if UsePolicyBasedTrafficSelectors is used)
 - The SA lifetimes are local specifications only, do not need to match.
2. If GCMAES is used as for IPsec Encryption algorithm, you must select the same GCMAES algorithm and key length for IPsec Integrity; for example, using GCMAES128 for both
3. In the table above:
 - IKEv2 corresponds to Main Mode or Phase 1
 - IPsec corresponds to Quick Mode or Phase 2
 - DH Group specifies the Diffie-Hellmen Group used in Main Mode or Phase 1
 - PFS Group specified the Diffie-Hellmen Group used in Quick Mode or Phase 2
4. IKEv2 Main Mode SA lifetime is fixed at 28,800 seconds on the Azure VPN gateways
5. Setting "UsePolicyBasedTrafficSelectors" to \$True on a connection will configure the Azure VPN gateway to connect to policy-based VPN firewall on premises. If you enable PolicyBasedTrafficSelectors, you need to ensure your VPN device has the matching traffic selectors defined with all combinations of your on-premises network (local network gateway) prefixes to/from the Azure virtual network prefixes, instead of any-to-any. For example, if your on-premises network prefixes are 10.1.0.0/16 and 10.2.0.0/16, and your virtual network prefixes are 192.168.0.0/16 and 172.16.0.0/16, you need to specify the following traffic selectors:
 - 10.1.0.0/16 <=====> 192.168.0.0/16
 - 10.1.0.0/16 <=====> 172.16.0.0/16
 - 10.2.0.0/16 <=====> 192.168.0.0/16
 - 10.2.0.0/16 <=====> 172.16.0.0/16

For more information regarding policy-based traffic selectors, see [Connect multiple on-premises policy-based VPN devices](#).

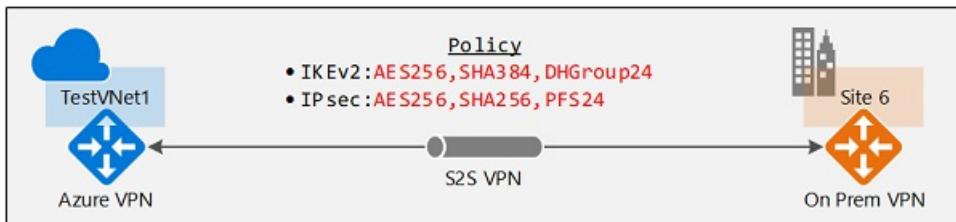
The following table lists the corresponding Diffie-Hellman Groups supported by the custom policy:

DIFFIE-HELLMAN GROUP	DHGROUP	PFSGROUP	KEY LENGTH
1	DHGroup1	PFS1	768-bit MODP
2	DHGroup2	PFS2	1024-bit MODP
14	DHGroup14 DHGroup2048	PFS2048	2048-bit MODP
19	ECP256	ECP256	256-bit ECP
20	ECP384	ECP284	384-bit ECP
24	DHGroup24	PFS24	2048-bit MODP

Refer to [RFC3526](#) and [RFC5114](#) for more details.

Part 3 - Create a new S2S VPN connection with IPsec/IKE policy

This section walks you through the steps of creating a S2S VPN connection with an IPsec/IKE policy. The following steps create the connection as shown in the diagram:



See [Create a S2S VPN connection](#) for more detailed step-by-step instructions for creating a S2S VPN connection.

Before you begin

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- Install the Azure Resource Manager PowerShell cmdlets. See [Overview of Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Step 1 - Create the virtual network, VPN gateway, and local network gateway

1. Declare your variables

For this exercise, we start by declaring our variables. Be sure to replace the values with your own when configuring for production.

```
$Sub1      = "<YourSubscriptionName>"  
$RG1       = "TestPolicyRG1"  
$Location1 = "East US 2"  
$VNetName1 = "TestVNet1"  
$FESubName1 = "FrontEnd"  
$BESubName1 = "Backend"  
$GWSubName1 = "GatewaySubnet"  
$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"  
$DNS1       = "8.8.8.8"  
$GWName1    = "VNet1GW"  
$GW1IPName1 = "VNet1GWIP1"  
$GW1IPconf1 = "gw1ipconf1"  
$Connection16 = "VNet1toSite6"  
  
$LNGName6   = "Site6"  
$LNGPrefix61 = "10.61.0.0/16"  
$LNGPrefix62 = "10.62.0.0/16"  
$LNGIP6     = "131.107.72.22"
```

2. Connect to your subscription and create a new resource group

Make sure you switch to PowerShell mode to use the Resource Manager cmdlets. For more information, see [Using Windows PowerShell with Resource Manager](#).

Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzureRmAccount  
Select-AzureRmSubscription -SubscriptionName $Sub1  
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

3. Create the virtual network, VPN gateway, and local network gateway

The following sample creates the virtual network, TestVNet1, with three subnets, and the VPN gateway. 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.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1

$gw1pip1      = New-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$vnet1        = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1      = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gw1ipconf1   = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress
$gw1pip1

New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -
IpConfigurations $gw1ipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1

New-AzureRmLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1 -Location $Location1 -
GatewayIpAddress $LNGIP6 -AddressPrefix $LNGPrefix61,$LNGPrefix62
```

Step 2 - Create a S2S VPN connection with an IPsec/IKE policy

1. Create an IPsec/IKE policy

The following sample script creates an IPsec/IKE policy with the following algorithms and parameters:

- IKEv2: AES256, SHA384, DHGroup24
- IPsec: AES256, SHA256, PFS None, SA Lifetime 14400 seconds & 102400000KB

```
$ipsecpolicy6 = New-AzureRmIpsecPolicy -IkeEncryption AES256 -IkeIntegrity SHA384 -DhGroup DHGroup24 -
IpsecEncryption AES256 -IpsecIntegrity SHA256 -PfsGroup None -SALifeTimeSeconds 14400 -SADataSizeKilobytes
102400000
```

If you use GCMAES for IPsec, you must use the same GCMAES algorithm and key length for both IPsec encryption and integrity. For example above, the corresponding parameters will be "-IpsecEncryption GCMAES256 -IpsecIntegrity GCMAES256" when using GCMAES256.

2. Create the S2S VPN connection with the IPsec/IKE policy

Create an S2S VPN connection and apply the IPsec/IKE policy created earlier.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng6 = Get-AzureRmLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1 -
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng6 -Location $Location1 -ConnectionType IPsec -
IpsecPolicies $ipsecpolicy6 -SharedKey 'AzureA1b2C3'
```

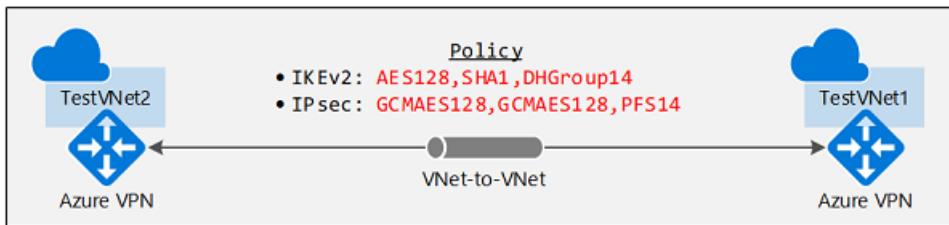
You can optionally add "-UsePolicyBasedTrafficSelectors \$True" to the create connection cmdlet to enable Azure VPN gateway to connect to policy-based VPN devices on premises, as described above.

IMPORTANT

Once an IPsec/IKE policy is specified on a connection, the Azure VPN gateway will only send or accept the IPsec/IKE proposal with specified cryptographic algorithms and key strengths on that particular connection. Make sure your on-premises VPN device for the connection uses or accepts the exact policy combination, otherwise the S2S VPN tunnel will not establish.

Part 4 - Create a new VNet-to-VNet connection with IPsec/IKE policy

The steps of creating a VNet-to-VNet connection with an IPsec/IKE policy are similar to that of a S2S VPN connection. The following sample scripts create the connection as shown in the diagram:



See [Create a VNet-to-VNet connection](#) for more detailed steps for creating a VNet-to-VNet connection. You must complete [Part 3](#) to create and configure TestVNet1 and the VPN Gateway.

Step 1 - Create the second virtual network and VPN gateway

1. Declare your variables

Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG2      = "TestPolicyRG2"
$Location2 = "East US 2"
$VNetName2 = "TestVNet2"
$FESubName2 = "FrontEnd"
$BESubName2 = "Backend"
$GWSubName2 = "GatewaySubnet"
$VNetPrefix21 = "10.21.0.0/16"
$VNetPrefix22 = "10.22.0.0/16"
$FESubPrefix2 = "10.21.0.0/24"
$BESubPrefix2 = "10.22.0.0/24"
$GWSubPrefix2 = "10.22.255.0/27"
$DNS2      = "8.8.8.8"
$GWName2   = "VNet2GW"
$GW2IPName1 = "VNet2GWIP1"
$GW2IPconf1 = "gw2ipconf1"
$Connection21 = "VNet2toVNet1"
$Connection12 = "VNet1toVNet2"
```

2. Create the second virtual network and VPN gateway in the new resource group

```

New-AzureRmResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$besub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsb2 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix
$VNetPrefix21,$VNetPrefix22 -Subnet $fesub2,$besub2,$gwsb2

$gw2pip1 = New-AzureRmPublicIpAddress -Name $GW2IPName1 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic
$vnet2 = Get-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gw2ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW2IPconf1 -Subnet $subnet2 -PublicIpAddress
$gw2pip1

New-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -
IpConfigurations $gw2ipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance

```

Step 2 - Create a VNet-toVNet connection with the IPsec/IKE policy

Similar to the S2S VPN connection, create an IPsec/IKE policy then apply to policy to the new connection.

1. Create an IPsec/IKE policy

The following sample script creates a different IPsec/IKE policy with the following algorithms and parameters:

- IKEv2: AES128, SHA1, DHGroup14
- IPsec: GCMAES128, GCMAES128, PFS14, SA Lifetime 14400 seconds & 102400000KB

```

$ipsecpolicy2 = New-AzureRmIpsecPolicy -IkeEncryption AES128 -IkeIntegrity SHA1 -DhGroup DHGroup14 -
IpsecEncryption GCMAES128 -IpsecIntegrity GCMAES128 -PfsGroup PFS14 -SALifeTimeSeconds 14400 -
SADataSizeKilobytes 102400000

```

2. Create VNet-to-VNet connections with the IPsec/IKE policy

Create a VNet-to-VNet connection and apply the IPsec/IKE policy you created. In this example, both gateways are in the same subscription. So it is possible to create and configure both connections with the same IPsec/IKE policy in the same PowerShell session.

```

$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet2gw = Get-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -
VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType
Vnet2Vnet -IpsecPolicies $ipsecpolicy2 -SharedKey 'AzureA1b2C3'

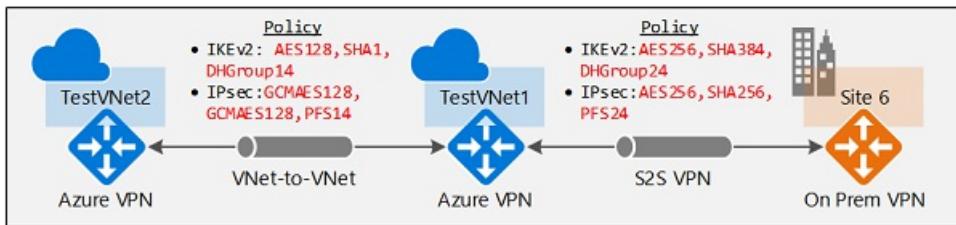
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -
VirtualNetworkGateway1 $vnet2gw -VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType
Vnet2Vnet -IpsecPolicies $ipsecpolicy2 -SharedKey 'AzureA1b2C3'

```

IMPORTANT

Once an IPsec/IKE policy is specified on a connection, the Azure VPN gateway will only send or accept the IPsec/IKE proposal with specified cryptographic algorithms and key strengths on that particular connection. Make sure the IPsec policies for both connections are the same, otherwise the VNet-to-VNet connection will not establish.

After completing these steps, the connection is established in a few minutes, and you will have the following network topology as shown in the beginning:



Part 5 - Update IPsec/IKE policy for a connection

The last section shows you how to manage IPsec/IKE policy for an existing S2S or VNet-to-VNet connection. The exercise below walks you through the following operations on a connection:

1. Show the IPsec/IKE policy of a connection
2. Add or update the IPsec/IKE policy to a connection
3. Remove the IPsec/IKE policy from a connection

The same steps apply to both S2S and VNet-to-VNet connections.

IMPORTANT

IPsec/IKE policy is supported on *Standard* and *HighPerformance* route-based VPN gateways only. It does not work on the Basic gateway SKU or the policy-based VPN gateway.

1. Show the IPsec/IKE policy of a connection

The following example shows how to get the IPsec/IKE policy configured on a connection. The scripts also continue from the exercises above.

```
$RG1      = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6 = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1
$connection6.IpsecPolicies
```

The last command lists the current IPsec/IKE policy configured on the connection, if there is any. The following is a sample output for the connection:

```
SALifeTimeSeconds : 14400
SDAContentSizeKilobytes : 102400000
IpsecEncryption : AES256
IpsecIntegrity : SHA256
IkeEncryption : AES256
IkeIntegrity : SHA384
DhGroup : DHGroup24
PfsGroup : PFS24
```

If there is no IPsec/IKE policy configured, the command (PS > \$connection6.policy) gets an empty return. It does not mean IPsec/IKE is not configured on the connection, but that there is no custom IPsec/IKE policy. The actual connection uses the default policy negotiated between your on-premises VPN device and the Azure VPN gateway.

2. Add or update an IPsec/IKE policy for a connection

The steps to add a new policy or update an existing policy on a connection are the same: create a new policy then apply the new policy to the connection.

```

$RG1          = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6  = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

$newpolicy6   = New-AzureRmIpsecPolicy -IkeEncryption AES128 -IkeIntegrity SHA1 -DhGroup DHGroup14 -
IpsecEncryption AES256 -IpsecIntegrity SHA256 -PfsGroup None -SALifeTimeSeconds 14400 -SADataSizeKilobytes
102400000

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -IpsecPolicies
$newpolicy6

```

To enable "UsePolicyBasedTrafficSelectors" when connecting to an on-premises policy-based VPN device, add the "-UsePolicyBaseTrafficSelectors" parameter to the cmdlet, or set it to \$False to disable the option:

```

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -IpsecPolicies
$newpolicy6 -UsePolicyBasedTrafficSelectors $True

```

You can get the connection again to check if the policy is updated.

```

$connection6  = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1
$connection6.IpsecPolicies

```

You should see the output from the last line, as shown in the following example:

```

SALifeTimeSeconds : 14400
SADataSizeKilobytes : 102400000
IpsecEncryption : AES256
IpsecIntegrity : SHA256
IkeEncryption : AES128
IkeIntegrity : SHA1
DhGroup : DHGroup14
PfsGroup : None

```

3. Remove an IPsec/IKE policy from a connection

Once you remove the custom policy from a connection, the Azure VPN gateway reverts back to the [default list of IPsec/IKE proposals](#) and renegotiates again with your on-premises VPN device.

```

$RG1          = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6  = Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

$currentpolicy = $connection6.IpsecPolicies[0]
$connection6.IpsecPolicies.Remove($currentpolicy)

Set-AzureRmVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6

```

You can use the same script to check if the policy has been removed from the connection.

Next steps

See [Connect multiple on-premises policy-based VPN devices](#) for more details regarding policy-based traffic selectors.

Once your connection is complete, you can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Configure active-active S2S VPN connections with Azure VPN Gateways

7/25/2018 • 14 minutes to read • [Edit Online](#)

This article walks you through the steps to create active-active cross-premises and VNet-to-VNet connections using the Resource Manager deployment model and PowerShell.

About highly available cross-premises connections

To achieve high availability for cross-premises and VNet-to-VNet connectivity, you should deploy multiple VPN gateways and establish multiple parallel connections between your networks and Azure. See [Highly Available Cross-Premises and VNet-to-VNet Connectivity](#) for an overview of connectivity options and topology.

This article provides the instructions to set up an active-active cross-premises VPN connection, and active-active connection between two virtual networks.

- [Part 1 - Create and configure your Azure VPN gateway in active-active mode](#)
- [Part 2 - Establish active-active cross-premises connections](#)
- [Part 3 - Establish active-active VNet-to-VNet connections](#)

If you already have a VPN gateway, you can:

- [Update an existing VPN gateway from active-standby to active-active, or vice versa](#)

You can combine these together to build a more complex, highly available network topology that meets your needs.

IMPORTANT

The active-active mode uses only the following SKUs:

- VpnGw1, VpnGw2, VpnGw3
- HighPerformance (for old legacy SKUs)

Part 1 - Create and configure active-active VPN gateways

The following steps will configure your Azure VPN gateway in active-active modes. The key differences between the active-active and active-standby gateways:

- You need to create two Gateway IP configurations with two public IP addresses
- You need set the EnableActiveActiveFeature flag
- The gateway SKU must be VpnGw1, VpnGw2, VpnGw3, or HighPerformance (legacy SKU).

The other properties are the same as the non-active-active gateways.

Before you begin

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- You'll need to install the Azure Resource Manager PowerShell cmdlets. See [Overview of Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Step 1 - Create and configure VNet1

1. Declare your variables

For this exercise, we'll start by declaring our variables. The example below declares the variables using the values for this exercise. Be sure to replace the values with your own when configuring for production. You can use these variables if you are running through the steps to become familiar with this type of configuration. Modify the variables, and then copy and paste into your PowerShell console.

```
$Sub1 = "Ross"
$RG1 = "TestAARG1"
$Location1 = "West US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$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"
$VNet1ASN = 65010
$DNS1 = "8.8.8.8"
$GWName1 = "VNet1GW"
$GW1IPName1 = "VNet1GWIP1"
$GW1IPName2 = "VNet1GWIP2"
$GW1IPconf1 = "gw1ipconf1"
$GW1IPconf2 = "gw1ipconf2"
$Connection12 = "VNet1toVNet2"
$Connection151 = "VNet1toSite5_1"
$Connection152 = "VNet1toSite5_2"
```

2. Connect to your subscription and create a new resource group

Make sure you switch to PowerShell mode to use the Resource Manager cmdlets. For more information, see [Using Windows PowerShell with Resource Manager](#).

Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $Sub1
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

3. Create TestVNet1

The sample below 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.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1
```

Step 2 - Create the VPN gateway for TestVNet1 with active-active mode

1. Create the public IP addresses and gateway IP configurations

Request two public IP addresses to be allocated to the gateway you will create for your VNet. You'll also define the subnet and IP configurations required.

```

$gw1pip1 = New-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$gw1pip2 = New-AzureRmPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic

$vnet1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gw1ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress
$gw1pip1
$gw1ipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf2 -Subnet $subnet1 -PublicIpAddress
$gw1pip2

```

2. Create the VPN gateway with active-active configuration

Create the virtual network gateway for TestVNet1. Note that there are two GatewayIpConfig entries, and the EnableActiveActiveFeature flag is set. Creating a gateway can take a while (45 minutes or more to complete).

```

New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -IpConfigurations
$gw1ipconf1,$gw1ipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -Asn $VNet1ASN -
EnableActiveActiveFeature -Debug

```

3. Obtain the gateway public IP addresses and the BGP Peer IP address

Once the gateway is created, you will need to obtain the BGP Peer IP address on the Azure VPN Gateway. This address is needed to configure the Azure VPN Gateway as a BGP Peer for your on-premises VPN devices.

```

$gw1pip1 = Get-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1
$gw1pip2 = Get-AzureRmPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1

```

Use the following cmdlets to show the two public IP addresses allocated for your VPN gateway, and their corresponding BGP Peer IP addresses for each gateway instance:

```

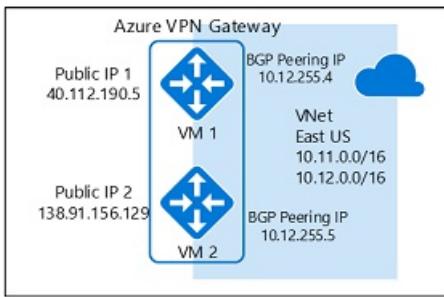
PS D:\> $gw1pip1.IpAddress
40.112.190.5

PS D:\> $gw1pip2.IpAddress
138.91.156.129

PS D:\> $vnet1gw.BgpSettingsText
{
  "Asn": 65010,
  "BgpPeeringAddress": "10.12.255.4,10.12.255.5",
  "PeerWeight": 0
}

```

The order of the public IP addresses for the gateway instances and the corresponding BGP Peering Addresses are the same. In this example, the gateway VM with public IP of 40.112.190.5 will use 10.12.255.4 as its BGP Peering Address, and the gateway with 138.91.156.129 will use 10.12.255.5. This information is needed when you set up your on premises VPN devices connecting to the active-active gateway. The gateway is shown in the diagram below with all addresses:



Once the gateway is created, you can use this gateway to establish active-active cross-premises or VNet-to-VNet connection. The following sections walk through the steps to complete the exercise.

Part 2 - Establish an active-active cross-premises connection

To establish a cross-premises connection, you need to create a Local Network Gateway to represent your on-premises VPN device, and a Connection to connect the Azure VPN gateway with the local network gateway. In this example, the Azure VPN gateway is in active-active mode. As a result, even though there is only one on-premises VPN device (local network gateway) and one connection resource, both Azure VPN gateway instances will establish S2S VPN tunnels with the on-premises device.

Before proceeding, please make sure you have completed [Part 1](#) of this exercise.

Step 1 - Create and configure the local network gateway

1. Declare your variables

This exercise will continue to build the configuration shown in the diagram. Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG5 = "TestAARG5"
$Location5 = "West US"
$LNGName51 = "Site5_1"
$LNGPrefix51 = "10.52.255.253/32"
$LNGIP51 = "131.107.72.22"
$LNGASN5 = 65050
$BGPPeerIP51 = "10.52.255.253"
```

A couple of things to note regarding the local network gateway parameters:

- The local network gateway can be in the same or different location and resource group as the VPN gateway. This example shows them in different resource groups but in the same Azure location.
- If there is only one on-premises VPN device as shown above, the active-active connection can work with or without BGP protocol. This example uses BGP for the cross-premises connection.
- If BGP is enabled, the prefix you need to declare for the local network gateway is the host address of your BGP Peer IP address on your VPN device. In this case, it's a /32 prefix of "10.52.255.253/32".
- As a reminder, you must use different BGP ASNs between your on-premises networks and Azure VNet. If they are the same, you need to change your VNet ASN if your on-premises VPN device already uses the ASN to peer with other BGP neighbors.

2. Create the local network gateway for Site5

Before you continue, please make sure you are still connected to Subscription 1. Create the resource group if it is not yet created.

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5
New-AzureRmLocalNetworkGateway -Name $LNGName51 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress
$LNGIP51 -AddressPrefix $LNGPrefix51 -Asn $LNGASN5 -BgpPeeringAddress $BGPPeerIP51
```

Step 2 - Connect the VNet gateway and local network gateway

1. Get the two gateways

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$lng5gw1 = Get-AzureRmLocalNetworkGateway -Name $LNGName51 -ResourceGroupName $RG5
```

2. Create the TestVNet1 to Site5 connection

In this step, you create the connection from TestVNet1 to Site5_1 with "EnableBGP" set to \$True.

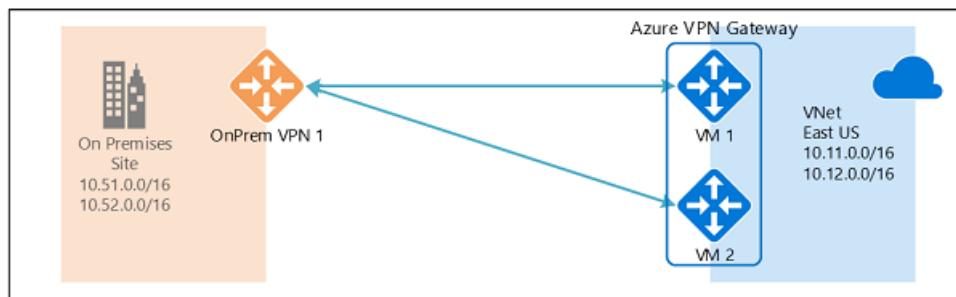
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection151 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw1 -Location $Location1 -ConnectionType IPsec -  
SharedKey 'AzureA1b2C3' -EnableBGP $True
```

3. VPN and BGP parameters for your on-premises VPN device

The example below lists the parameters you will enter into the BGP configuration section on your on-premises VPN device for this exercise:

```
- Site5 ASN : 65050  
- Site5 BGP IP : 10.52.255.253  
- Prefixes to announce : (for example) 10.51.0.0/16 and 10.52.0.0/16  
- Azure VNet ASN : 65010  
- Azure VNet BGP IP 1 : 10.12.255.4 for tunnel to 40.112.190.5  
- Azure VNet BGP IP 2 : 10.12.255.5 for tunnel to 138.91.156.129  
- Static routes : Destination 10.12.255.4/32, nexthop the VPN tunnel interface to 40.112.190.5  
                  Destination 10.12.255.5/32, nexthop the VPN tunnel interface to 138.91.156.129  
- eBGP Multihop : Ensure the "multihop" option for eBGP is enabled on your device if needed
```

The connection should be established after a few minutes, and the BGP peering session will start once the IPsec connection is established. This example so far has configured only one on-premises VPN device, resulting in the diagram shown below:



Step 3 - Connect two on-premises VPN devices to the active-active VPN gateway

If you have two VPN devices at the same on-premises network, you can achieve dual redundancy by connecting the Azure VPN gateway to the second VPN device.

1. Create the second local network gateway for Site5

The gateway IP address, address prefix, and BGP peering address for the second local network gateway must not overlap with the previous local network gateway for the same on-premises network.

```
$LNGName52 = "Site5_2"  
$LNGPrefix52 = "10.52.255.254/32"  
$LNGIP52 = "131.107.72.23"  
$BGPPeerIP52 = "10.52.255.254"
```

```
New-AzureRmLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress  
$LNGIP52 -AddressPrefix $LNGPrefix52 -Asn $LNGASN5 -BgpPeeringAddress $BGPPeerIP52
```

2. Connect the VNet gateway and the second local network gateway

Create the connection from TestVNet1 to Site5_2 with "EnableBGP" set to \$True

```
$lng5gw2 = Get-AzureRmLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5
```

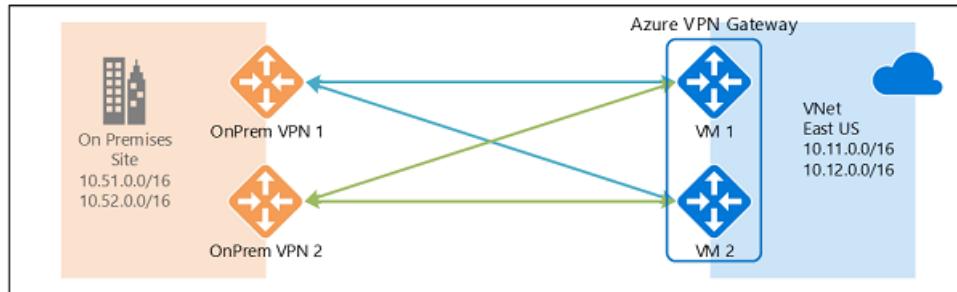
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection152 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw2 -Location $Location1 -ConnectionType IPsec -  
SharedKey 'AzureA1b2C3' -EnableBGP $True
```

3. VPN and BGP parameters for your second on-premises VPN device

Similarly, below lists the parameters you will enter into the second VPN device:

```
- Site5 ASN : 65050  
- Site5 BGP IP : 10.52.255.254  
- Prefixes to announce : (for example) 10.51.0.0/16 and 10.52.0.0/16  
- Azure VNet ASN : 65010  
- Azure VNet BGP IP 1 : 10.12.255.4 for tunnel to 40.112.190.5  
- Azure VNet BGP IP 2 : 10.12.255.5 for tunnel to 138.91.156.129  
- Static routes : Destination 10.12.255.4/32, nexthop the VPN tunnel interface to 40.112.190.5  
                  Destination 10.12.255.5/32, nexthop the VPN tunnel interface to 138.91.156.129  
- eBGP Multihop : Ensure the "multihop" option for eBGP is enabled on your device if needed
```

Once the connection (tunnels) are established, you will have dual redundant VPN devices and tunnels connecting your on-premises network and Azure:



Part 3 - Establish an active-active VNet-to-VNet connection

This section creates an active-active VNet-to-VNet connection with BGP.

The instructions below continue from the previous steps listed above. You must complete [Part 1](#) to create and configure TestVNet1 and the VPN Gateway with BGP.

Step 1 - Create TestVNet2 and the VPN gateway

It is important to make sure that the IP address space of the new virtual network, TestVNet2, does not overlap with any of your VNet ranges.

In this example, the virtual networks belong to the same subscription. You can set up VNet-to-VNet connections between different subscriptions; please refer to [Configure a VNet-to-VNet connection](#) to learn more details. Make sure you add the "-EnableBgp \$True" when creating the connections to enable BGP.

1. Declare your variables

Be sure to replace the values with the ones that you want to use for your configuration.

```

$RG2 = "TestAARG2"
$Location2 = "East US"
$VNetName2 = "TestVNet2"
$FESubName2 = "FrontEnd"
$BESubName2 = "Backend"
$GWSubName2 = "GatewaySubnet"
$VNetPrefix21 = "10.21.0.0/16"
$VNetPrefix22 = "10.22.0.0/16"
$FESubPrefix2 = "10.21.0.0/24"
$BESubPrefix2 = "10.22.0.0/24"
$GWSubPrefix2 = "10.22.255.0/27"
$VNet2ASN = 65020
$DNS2 = "8.8.8.8"
$GWName2 = "VNet2GW"
$GW2IPName1 = "VNet2GWIP1"
$GW2IPconf1 = "gw2ipconf1"
$GW2IPName2 = "VNet2GWIP2"
$GW2IPconf2 = "gw2ipconf2"
$Connection21 = "VNet2toVNet1"
$Connection12 = "VNet1toVNet2"

```

2. Create TestVNet2 in the new resource group

```

New-AzureRmResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$besub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix
$VNetPrefix21,$VNetPrefix22 -Subnet $fesub2,$besub2,$gwsub2

```

3. Create the active-active VPN gateway for TestVNet2

Request two public IP addresses to be allocated to the gateway you will create for your VNet. You'll also define the subnet and IP configurations required.

```

$gw2pip1 = New-AzureRmPublicIpAddress -Name $GW2IPName1 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic
$gw2pip2 = New-AzureRmPublicIpAddress -Name $GW2IPName2 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic

$vnet2 = Get-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gw2ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW2IPconf1 -Subnet $subnet2 -PublicIpAddress
$gw2pip1
$gw2ipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW2IPconf2 -Subnet $subnet2 -PublicIpAddress
$gw2pip2

```

Create the VPN gateway with the AS number and the "EnableActiveActiveFeature" flag. Note that you must override the default ASN on your Azure VPN gateways. The ASNs for the connected VNets must be different to enable BGP and transit routing.

```

New-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -IpConfigurations
$gw2ipconf1,$gw2ipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -Asn $VNet2ASN -
EnableActiveActiveFeature

```

Step 2 - Connect the TestVNet1 and TestVNet2 gateways

In this example, both gateways are in the same subscription. You can complete this step in the same PowerShell session.

1. Get both gateways

Make sure you log in and connect to Subscription 1.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$vnet2gw = Get-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2
```

2. Create both connections

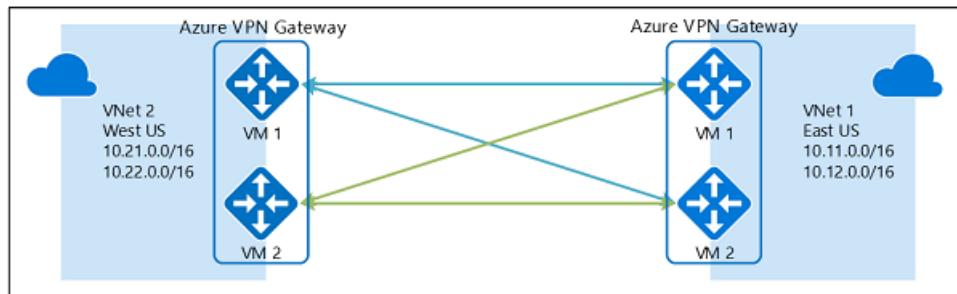
In this step, you will create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1.

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1  
$vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey  
'AzureA1b2C3' -EnableBgp $True  
  
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -VirtualNetworkGateway1  
$vnet2gw -VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType Vnet2Vnet -SharedKey  
'AzureA1b2C3' -EnableBgp $True
```

IMPORTANT

Be sure to enable BGP for BOTH connections.

After completing these steps, the connection will be established in a few minutes, and the BGP peering session will be up once the VNet-to-VNet connection is completed with dual redundancy:



Update an existing VPN gateway

This section helps you change an existing Azure VPN gateway from active-standby to active-active mode, or vice versa.

Change an active-standby gateway to an active-active gateway

The following example converts an active-standby gateway into an active-active gateway. When you change an active-standby gateway to active-active, you create another public IP address, then add a second Gateway IP configuration.

1. Declare your variables

Replace the following parameters used for the examples with the settings that you require for your own configuration, then declare these variables.

```
$GWName = "TestVNetAA1GW"  
$VNetName = "TestVNetAA1"  
$RG = "TestVPNAActiveActive01"  
$GWIPName2 = "gwpip2"  
$GWIPconf2 = "gw1ipconf2"
```

After declaring the variables, you can copy and paste this example to your PowerShell console.

```
$vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gw = Get-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
$location = $gw.Location
```

2. Create the public IP address, then add the second gateway IP configuration

```
$gwpip2 = New-AzureRmPublicIpAddress -Name $GWIPName2 -ResourceGroupName $RG -Location $location -
AllocationMethod Dynamic
Add-AzureRmVirtualNetworkGatewayIpConfig -VirtualNetworkGateway $gw -Name $GWIPconf2 -Subnet $subnet -
PublicIpAddress $gwpip2
```

3. Enable active-active mode and update the gateway

In this step, you enable active-active mode and update the gateway. In the example, the VPN gateway is currently using a legacy Standard SKU. However, active-active does not support the Standard SKU. To resize the legacy SKU to one that is supported (in this case, HighPerformance), you simply specify the supported legacy SKU that you want to use.

- You can't change a legacy SKU to one of the new SKUs using this step. You can only resize a legacy SKU to another supported legacy SKU. For example, you can't change the SKU from Standard to VpnGw1 (even though VpnGw1 is supported for active-active) because Standard is a legacy SKU and VpnGw1 is a current SKU. For more information about resizing and migrating SKUs, see [Gateway SKUs](#).
- If you want to resize a current SKU, for example VpnGw1 to VpnGw3, you can do so using this step because the SKUs are in the same SKU family. To do so, you would use the value: `-GatewaySku VpnGw3`

When you are using this in your environment, if you don't need to resize the gateway, you won't need to specify the `-GatewaySku`. Notice that in this step, you must set the gateway object in PowerShell to trigger the actual update. This update can take 30 to 45 minutes, even if you are not resizing your gateway.

```
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -EnableActiveActiveFeature -GatewaySku
HighPerformance
```

Change an active-active gateway to an active-standby gateway

1. Declare your variables

Replace the following parameters used for the examples with the settings that you require for your own configuration, then declare these variables.

```
$GWName = "TestVNetAA1GW"
$RG = "TestVPNActiveActive01"
```

After declaring the variables, get the name of the IP configuration you want to remove.

```
$gw = Get-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
$ipconfname = $gw.IpConfigurations[1].Name
```

2. Remove the gateway IP configuration and disable the active-active mode

Use this example to remove the gateway IP configuration and disable active-active mode. Notice that you must set the gateway object in PowerShell to trigger the actual update.

```
Remove-AzureRmVirtualNetworkGatewayIpConfig -Name $ipconfname -VirtualNetworkGateway $gw
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -DisableActiveActiveFeature
```

This update can take up to 30 to 45 minutes.

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Create a zone-redundant virtual network gateway in Azure Availability Zones - Preview

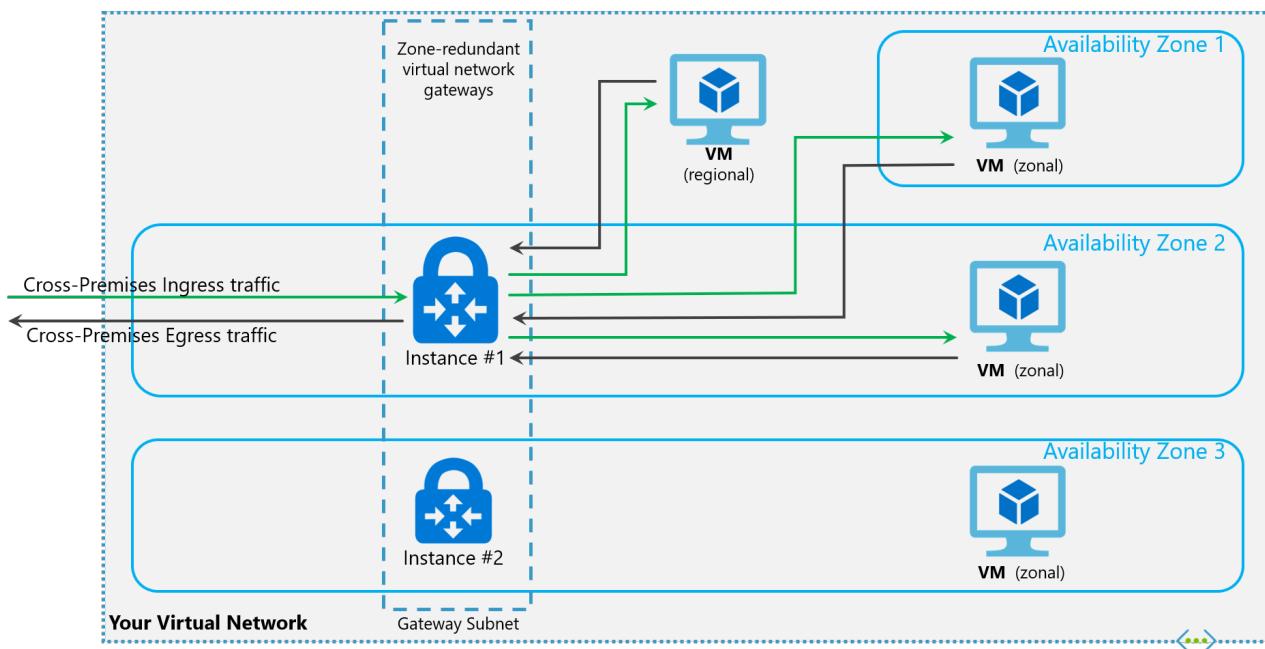
7/10/2018 • 8 minutes to read • [Edit Online](#)

You can deploy VPN and ExpressRoute gateways in [Azure Availability Zones](#). This brings resiliency, scalability, and higher availability to virtual network gateways. Deploying gateways in Azure Availability Zones physically and logically separates gateways within a region, while protecting your on-premises network connectivity to Azure from zone-level failures.

Zonal and zone-redundant gateways have fundamental performance improvements over regular virtual network gateways. Additionally, creating a zone-redundant or zonal virtual network gateway is faster than creating other gateways. Rather than taking 45 minutes, create times take approximately 15 minutes for an ExpressRoute gateway, and 19 minutes for a VPN gateway.

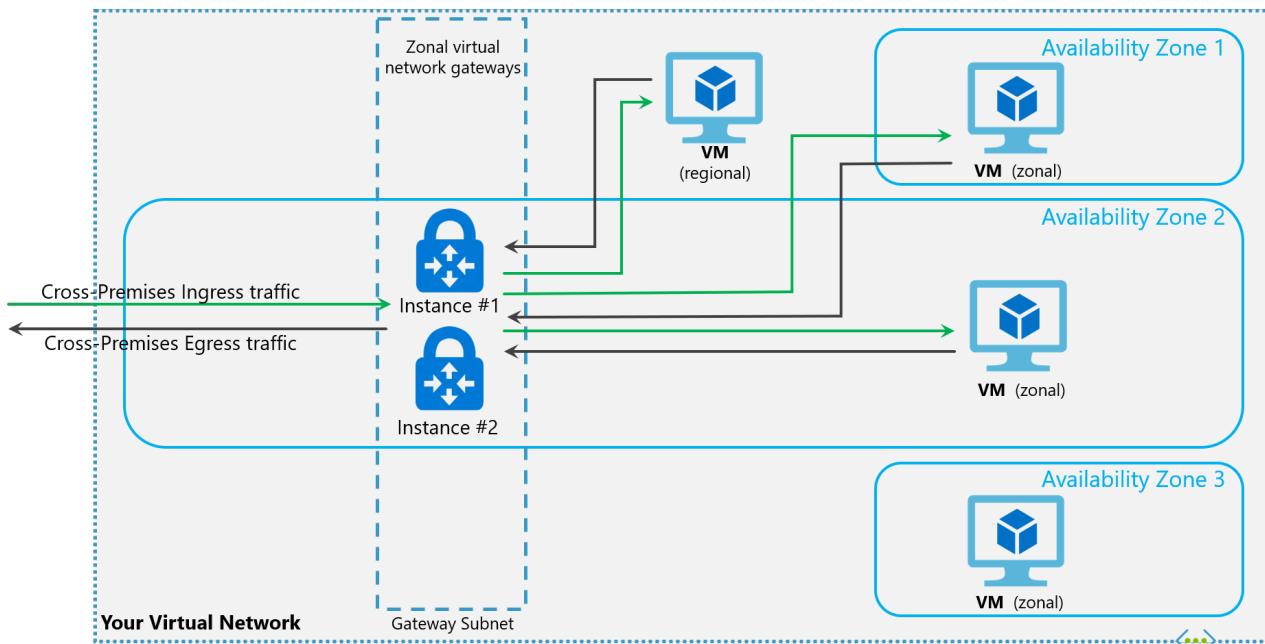
Zone-redundant gateways

To automatically deploy your virtual network gateways across availability zones, you can use zone-redundant virtual network gateways. With zone-redundant gateways, you can harness the 99.99% uptime SLA at GA to access your mission-critical, scalable services on Azure.



Zonal gateways

To deploy gateways in a specific zone, you use zonal gateways. When you deploy a zonal gateway, both instances of the gateway are deployed in the same Availability Zone.



Gateway SKUs

Zone-redundant and zonal gateways must use the new gateway SKUs. Once you [self-enroll in the Preview](#), you will see the new virtual network gateway SKUs in all of the Azure AZ regions. These SKUs are similar to the corresponding SKUs for ExpressRoute and VPN Gateway, except that they are specific to zone-redundant and zonal gateways.

The new gateway SKUs are:

VPN Gateway

- VpnGw1AZ
- VpnGw2AZ
- VpnGw3AZ

ExpressRoute

- ErGw1AZ
- ErGw2AZ
- ErGw3AZ

Public IP SKUs

Zone-redundant gateways and zonal gateways both rely on the Azure public IP resource *Standard* SKU. The configuration of the Azure public IP resource determines whether the gateway that you deploy is zone-redundant, or zonal. If you create a public IP resource with a *Basic* SKU, the gateway will not have any zone redundancy, and the gateway resources will be regional.

Zone-redundant gateways

When you create a public IP address using the **Standard** public IP SKU without specifying a zone, the behavior differs depending on whether the gateway is a VPN gateway, or an ExpressRoute gateway.

- For a VPN gateway, the two gateway instances will be deployed in any 2 out of these three zones to provide zone-redundancy.
- For an ExpressRoute gateway, since there can be more than two instances, the gateway can span across all the three zones.

Zonal gateways

When you create a public IP address using the **Standard** public IP SKU and specify the Zone (1, 2, or 3), all the gateway instances will be deployed in the same zone.

Regional gateways

When you create a public IP address using the **Basic** public IP SKU, the gateway is deployed as a regional gateway and does not have any zone-redundancy built into the gateway.

Before you begin

You can use either PowerShell installed locally on your computer, or the Azure Cloud Shell. If you choose to install and use the PowerShell locally, this feature requires the latest version of the PowerShell module.

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

Click Try It in the upper right corner of a code block.	
Open Cloud Shell in your browser.	
Click the Cloud Shell button on the menu in the upper right of the Azure portal.	

To use PowerShell locally

If you are using PowerShell locally on your computer, rather than using Cloud Shell, you must install PowerShell module 6.1.1 or higher. To check the version of PowerShell that you have installed, use the following command:

```
Get-Module AzureRM -ListAvailable | Select-Object -Property Name,Version,Path
```

If you need to upgrade, see [Install Azure PowerShell module](#).

Before beginning this configuration, you must sign in to your Azure account. The cmdlet prompts you for the sign-in credentials for your Azure account. After signing in, it downloads your account settings so they are available to Azure PowerShell. For more information, see [Using Windows PowerShell with Resource Manager](#).

To sign in, open your PowerShell console with elevated privileges, and connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

1. Enroll in the Preview

Before you can configure a zone-redundant or zonal gateway, you must first self-enroll your subscription in the Preview. Once your subscription has been provisioned, you will start to see the new gateway SKUs in all of the Azure AZ regions.

Make sure that you are signed into your Azure account and are using the subscription that you want to whitelist for this Preview. Use the following example to enroll:

```
Register-AzureRmProviderFeature -FeatureName AllowVMSSVirtualNetworkGateway -ProviderNamespace Microsoft.Network  
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.Network
```

Use the following command to verify that the 'AllowVMSSVirtualNetworkGateway' feature is registered with your subscription:

```
Get-AzureRmProviderFeature -ProviderNamespace Microsoft.Network
```

The result will look similar to this example:

```
PS C:\> Get-AzureRmProviderFeature -ProviderNamespace Microsoft.Network  


| FeatureName                                | ProviderName      | RegistrationState |
|--------------------------------------------|-------------------|-------------------|
| AllowRegionalGatewayManager                | Microsoft.Network | Registered        |
| AllowRegionalGatewayManagerForBrooklyn     | Microsoft.Network | Registered        |
| AllowRegionalGatewayManagerForExpressRoute | Microsoft.Network | Registered        |
| AllowVmssHealthProbe                       | Microsoft.Network | Registered        |
| AllowVMSSVirtualNetworkGateway             | Microsoft.Network | Registered        |


```

2. Declare your variables

The values used for the example steps are listed below. Additionally, some of the examples use declared variables within the steps. If you are using these steps in your own environment, be sure to replace these values with your own. When specifying location, verify that the region you specify is supported. For more information, see the [FAQ](#).

```
$RG1      = "TestRG1"  
$VNet1    = "VNet1"  
$Location1 = "CentralUS"  
$FESubnet1 = "FrontEnd"  
$BESubnet1 = "Backend"  
$GwSubnet1 = "GatewaySubnet"  
$VNet1Prefix = "10.1.0.0/16"  
$FEPrefix1  = "10.1.0.0/24"  
$BEPrefix1  = "10.1.1.0/24"  
$GwPrefix1  = "10.1.255.0/27"  
$Gw1       = "VNet1GW"  
$GwIP1     = "VNet1GWIP"  
$GwIPConf1 = "gwipconf1"
```

3. Create the virtual network

Create a resource group.

```
New-AzureRmResourceGroup -ResourceGroupName $RG1 -Location $Location1
```

Create a virtual network.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubnet1 -AddressPrefix $FEPPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubnet1 -AddressPrefix $BEPrefix1
$vnet = New-AzureRmVirtualNetwork -Name $VNet1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNet1Prefix -Subnet $fesub1,$besub1
```

4. Add the gateway subnet

The gateway subnet contains the reserved IP addresses that the virtual network gateway services use. Use the following examples to add and set a gateway subnet:

Add the gateway subnet.

```
$getvnet = Get-AzureRmVirtualNetwork -ResourceGroupName $RG1 -Name VNet1
Add-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27 -VirtualNetwork
$getvnet
```

Set the gateway subnet configuration for the virtual network.

```
$getvnet | Set-AzureRmVirtualNetwork
```

5. Request a public IP address

In this step, choose the instructions that apply to the gateway that you want to create. The selection of zones for deploying the gateways depends on the zones specified for the public IP address.

For zone-redundant gateways

Request a public IP address with a **Standard** PublicIpAddress SKU and do not specify any zone. In this case, the Standard public IP address created will be a zone-redundant public IP.

```
$pip1 = New-AzureRmPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod
Static -Sku Standard
```

For zonal gateways

Request a public IP address with a **Standard** PublicIpAddress SKU. Specify the zone (1, 2 or 3). All gateway instances will be deployed in this zone.

```
$pip1 = New-AzureRmPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod
Static -Sku Standard -Zone 1
```

For regional gateways

Request a public IP address with a **Basic** PublicIpAddress SKU. In this case, the gateway is deployed as a regional gateway and does not have any zone-redundancy built into the gateway. The gateway instances are created in any zones, respectively.

```
$pip1 = New-AzureRmPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod
Dynamic -Sku Basic
```

6. Create the IP configuration

```
$getvnet = Get-AzureRmVirtualNetwork -ResourceGroupName $RG1 -Name $VNet1
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name $GwSubnet1 -VirtualNetwork $getvnet
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GwIPConf1 -Subnet $subnet -PublicIpAddress $pip1
```

7. Create the gateway

Create the virtual network gateway.

For ExpressRoute

```
New-AzureRmVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations
$GwIPConf1 -GatewayType ExpressRoute
```

For VPN Gateway

```
New-AzureRmVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations
$GwIPConf1 -GatewayType Vpn -VpnType RouteBased
```

How to provide feedback

We would appreciate your feedback. Send an email to aznetworkgateways@microsoft.com to report any issues or provide feedback (positive or negative) for the zone-redundant and zonal VPN and Express Route gateways. Include your company name in “[]” in the subject line. Also include your subscription ID if you are reporting an issue.

FAQ

How do I sign up for the Preview?

You can [self-enroll](#) using the PowerShell commands in this article.

What will change when I enroll?

From your perspective, during Preview, you can deploy your gateways with zone-redundancy. This means that all instances of the gateways will be deployed across Azure Availability Zones, and each Availability Zone is a different fault and update domain. This makes your gateways more reliable, available, and resilient to zone failures.

Can I use the Azure portal?

Yes, you can use the Azure portal for the Preview. However, you still need to enroll using PowerShell or you won't be able to use the portal during Preview.

What regions are available for the Preview?

Zone-redundant and zonal gateways are available in production/Azure Public regions.

Will I be billed for participating in this Preview?

You will not be billed for your gateways during Preview. However, there is no SLA attached with your deployment. We are very interested in hearing your feedback.

NOTE

For ExpressRoute gateways, the gateway is not billed/charged. However, the circuit itself (not the gateway) will be billed.

What regions are available for me to try this in?

The public preview is available in Central US and France Central regions (Azure regions that have Availability

Zones generally available). Going forward, we will make the Zone-Redundant Gateways available to you in other Azure Public Regions.

Can I change my existing virtual network gateways to zone-redundant or zonal gateways?

Migrating your existing virtual network gateways to zone-redundant or zonal gateways is currently not supported. You can, however, delete your existing gateway and re-create a zone-redundant or zonal gateway.

Can I deploy both VPN and Express Route gateways in same virtual network?

Co-existence of both VPN and Express Route gateways in the same virtual network is supported during the Public Preview. However, be aware of the following requirements and limitations:

- Reserve a /27 IP address range for the gateway subnet.
- Zone-redundant/zonal Express Route gateways can only co-exist with zone-redundant/zonal VPN gateways.
- Deploy the zone-redundant/zonal Express Route gateway before deploying the zone-redundant/zonal VPN gateway.
- A zone-redundant/zonal Express Route gateway can be connected to, at most, 4 circuits.

Next steps

We would appreciate your feedback. Send an email to aznetworkgateways@microsoft.com to report any issues or provide feedback (positive or negative) for the zone-redundant and zonal VPN and Express Route gateways. Include your company name in “[]” in the subject line. Also include your subscription ID if you are reporting an issue.

How to configure BGP on Azure VPN Gateways using PowerShell

4/18/2018 • 9 minutes to read • [Edit Online](#)

This article walks you through the steps to enable BGP on a cross-premises Site-to-Site (S2S) VPN connection and a VNet-to-VNet connection using the Resource Manager deployment model and PowerShell.

About BGP

BGP is the standard routing protocol commonly used in the Internet to exchange routing and reachability information between two or more networks. BGP enables the Azure VPN Gateways and your on-premises VPN devices, called BGP peers or neighbors, to 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 a BGP gateway learns from one BGP peer to all other BGP peers.

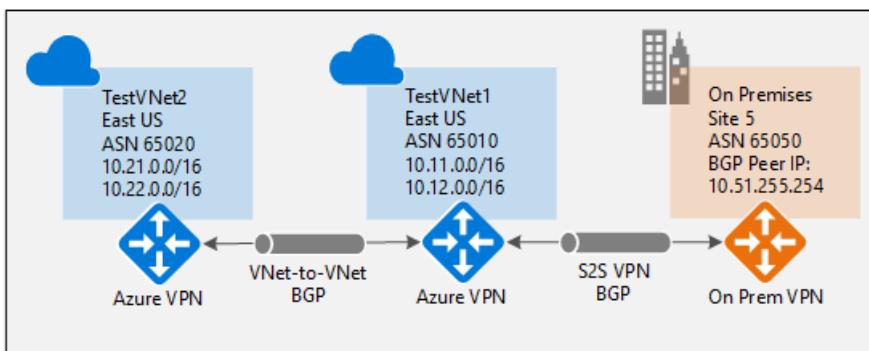
See [Overview of BGP with Azure VPN Gateways](#) for more discussion on benefits of BGP and to understand the technical requirements and considerations of using BGP.

Getting started with BGP on Azure VPN gateways

This article walks you through the steps to do the following tasks:

- [Part 1 - Enable BGP on your Azure VPN gateway](#)
- [Part 2 - Establish a cross-premises connection with BGP](#)
- [Part 3 - Establish a VNet-to-VNet connection with BGP](#)

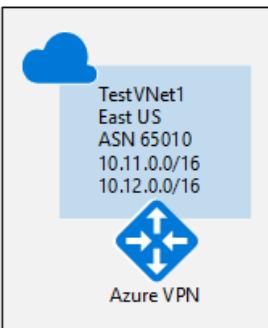
Each part of the instructions forms a basic building block for enabling BGP in your network connectivity. If you complete all three parts, you build the topology as shown in the following diagram:



You can combine parts together to build a more complex, multi-hop, transit network that meets your needs.

Part 1 - Configure BGP on the Azure VPN Gateway

The configuration steps set up the BGP parameters of the Azure VPN gateway as shown in the following diagram:



Before you begin

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- Install the Azure Resource Manager PowerShell cmdlets. For more information about installing the PowerShell cmdlets, see [How to install and configure Azure PowerShell](#).

Step 1 - Create and configure VNet1

1. Declare your variables

For this exercise, we start by declaring our variables. The following example declares the variables using the values for this exercise. Be sure to replace the values with your own when configuring for production. You can use these variables if you are running through the steps to become familiar with this type of configuration. Modify the variables, and then copy and paste into your PowerShell console.

```
$Sub1 = "Replace_With_Your_Subscription_Name"
$RG1 = "TestBGP RG1"
$Location1 = "East US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$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"
$VNet1ASN = 65010
$DNS1 = "8.8.8.8"
$GWName1 = "VNet1GW"
$GWIPName1 = "VNet1GWIP"
$GWIPconfName1 = "gwipconf1"
$Connection12 = "VNet1toVNet2"
$Connection15 = "VNet1toSite5"
```

2. Connect to your subscription and create a new resource group

To use the Resource Manager cmdlets, Make sure you switch to PowerShell mode. For more information, see [Using Windows PowerShell with Resource Manager](#).

Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $Sub1
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

3. Create TestVNet1

The following sample 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.

```

$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1 $besub1 = New-
AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsu1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSuName1 -AddressPrefix $GWSuPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsu1

```

Step 2 - Create the VPN Gateway for TestVNet1 with BGP parameters

1. Create the IP and subnet configurations

Request a public IP address to be allocated to the gateway you will create for your VNet. You'll also define the required subnet and IP configurations.

```

$gwpip1 = New-AzureRmPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic

$vnet1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwpipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 -Subnet $subnet1 -PublicIpAddress
$gwpip1

```

2. Create the VPN gateway with the AS number

Create the virtual network gateway for TestVNet1. BGP requires a Route-Based VPN gateway, and also the addition parameter, -Asn, to set the ASN (AS Number) for TestVNet1. If you do not set the ASN parameter, ASN 65515 is assigned. Creating a gateway can take a while (30 minutes or more to complete).

```

New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -
IpConfigurations $gwpipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance -Asn $VNet1ASN

```

3. Obtain the Azure BGP Peer IP address

Once the gateway is created, you need to obtain the BGP Peer IP address on the Azure VPN Gateway. This address is needed to configure the Azure VPN Gateway as a BGP Peer for your on-premises VPN devices.

```

$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet1gw.BgpSettingsText

```

The last command shows the corresponding BGP configurations on the Azure VPN Gateway; for example:

```

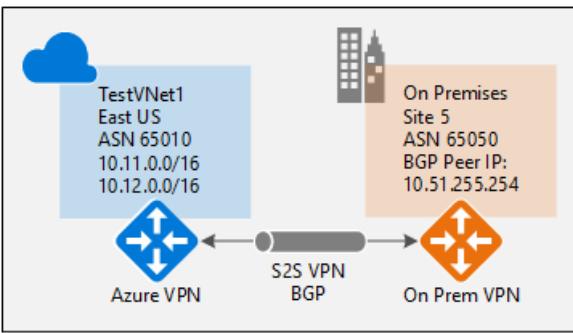
$vnet1gw.BgpSettingsText
{
    "Asn": 65010,
    "BgpPeeringAddress": "10.12.255.30",
    "PeerWeight": 0
}

```

Once the gateway is created, you can use this gateway to establish cross-premises connection or VNet-to-VNet connection with BGP. The following sections walk through the steps to complete the exercise.

Part 2 - Establish a cross-premises connection with BGP

To establish a cross-premises connection, you need to create a Local Network Gateway to represent your on-premises VPN device, and a Connection to connect the VPN gateway with the local network gateway. While there are articles that walk you through these steps, this article contains the additional properties required to specify the BGP configuration parameters.



Before proceeding, make sure you have completed [Part 1](#) of this exercise.

Step 1 - Create and configure the local network gateway

1. Declare your variables

This exercise continues to build the configuration shown in the diagram. Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG5 = "TestBGPRG5"
$Location5 = "East US 2"
$LNGName5 = "Site5"
$LNGPrefix50 = "10.52.255.254/32"
$LNGIP5 = "Your_VPN_Device_IP"
$LNGASN5 = 65050
$BGPPeerIP5 = "10.52.255.254"
```

A couple of things to note regarding the local network gateway parameters:

- The local network gateway can be in the same or different location and resource group as the VPN gateway. This example shows them in different resource groups in different locations.
- The minimum prefix you need to declare for the local network gateway is the host address of your BGP Peer IP address on your VPN device. In this case, it's a /32 prefix of "10.52.255.254/32".
- As a reminder, you must use different BGP ASNs between your on-premises networks and Azure VNet. If they are the same, you need to change your VNet ASN if your on-premises VPN device already uses the ASN to peer with other BGP neighbors.

Before you continue, make sure you are still connected to Subscription 1.

2. Create the local network gateway for Site5

Be sure to create the resource group if it is not created, before you create the local network gateway. Notice the two additional parameters for the local network gateway: Asn and BgpPeerAddress.

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5

New-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress
$LNGIP5 -AddressPrefix $LNGPrefix50 -Asn $LNGASN5 -BgpPeeringAddress $BGPPeerIP5
```

Step 2 - Connect the VNet gateway and local network gateway

1. Get the two gateways

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG5
```

2. Create the TestVNet1 to Site5 connection

In this step, you create the connection from TestVNet1 to Site5. You must specify "-EnableBGP \$True" to enable BGP for this connection. As discussed earlier, it is possible to have both BGP and non-BGP connections for the same Azure VPN Gateway. Unless BGP is enabled in the connection property, Azure will not enable BGP for this

connection even though BGP parameters are already configured on both gateways.

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lwgw -Location $Location1 -ConnectionType IPsec -  
SharedKey 'AzureA1b2C3' -EnableBGP $True
```

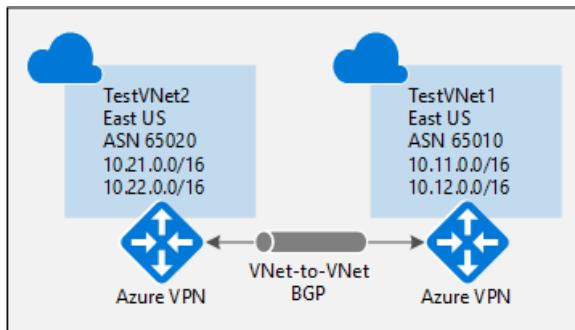
The following example lists the parameters you enter into the BGP configuration section on your on-premises VPN device for this exercise:

```
- Site5 ASN : 65050  
- Site5 BGP IP : 10.52.255.254  
- Prefixes to announce : (for example) 10.51.0.0/16 and 10.52.0.0/16  
- Azure VNet ASN : 65010  
- Azure VNet BGP IP : 10.12.255.30  
- Static route : Add a route for 10.12.255.30/32, with nexthop being the VPN tunnel interface on your device  
- eBGP Multihop : Ensure the "multihop" option for eBGP is enabled on your device if needed
```

The connection is established after a few minutes, and the BGP peering session starts once the IPsec connection is established.

Part 3 - Establish a VNet-to-VNet connection with BGP

This section adds a VNet-to-VNet connection with BGP, as shown in the following diagram:



The following instructions continue from the previous steps. You must complete [Part 1](#) to create and configure TestVNet1 and the VPN Gateway with BGP.

Step 1 - Create TestVNet2 and the VPN gateway

It is important to make sure that the IP address space of the new virtual network, TestVNet2, does not overlap with any of your VNet ranges.

In this example, the virtual networks belong to the same subscription. You can set up VNet-to-VNet connections between different subscriptions. For more information, see [Configure a VNet-to-VNet connection](#). Make sure you add the "-EnableBgp \$True" when creating the connections to enable BGP.

1. Declare your variables

Be sure to replace the values with the ones that you want to use for your configuration.

```

$RG2 = "TestBGPRG2"
$Location2 = "West US"
$VNetName2 = "TestVNet2"
$FESubName2 = "FrontEnd"
$BESubName2 = "Backend"
$GWSubName2 = "GatewaySubnet"
$VNetPrefix21 = "10.21.0.0/16"
$VNetPrefix22 = "10.22.0.0/16"
$FESubPrefix2 = "10.21.0.0/24"
$BESubPrefix2 = "10.22.0.0/24"
$GWSubPrefix2 = "10.22.255.0/27"
$VNet2ASN = 65020
$DNS2 = "8.8.8.8"
$GWName2 = "VNet2GW"
$GWIPName2 = "VNet2GWIP"
$GWIPconfName2 = "gwipconf2"
$Connection21 = "VNet2toVNet1"
$Connection12 = "VNet1toVNet2"

```

2. Create TestVNet2 in the new resource group

```

New-AzureRmResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FEsubPrefix2
$besub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsb2 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix
$VNetPrefix21,$VNetPrefix22 -Subnet $fesub2,$besub2,$gwsb2

```

3. Create the VPN gateway for TestVNet2 with BGP parameters

Request a public IP address to be allocated to the gateway you will create for your VNet and define the required subnet and IP configurations.

```

$gwip2      = New-AzureRmPublicIpAddress -Name $GWIPName2 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic

$vnet2      = Get-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2    = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gwipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName2 -Subnet $subnet2 -PublicIpAddress
$gwip2

```

Create the VPN gateway with the AS number. You must override the default ASN on your Azure VPN gateways. The ASNs for the connected VNets must be different to enable BGP and transit routing.

```

New-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -
IpConfigurations $gwipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku Standard -Asn $VNet2ASN

```

Step 2 - Connect the TestVNet1 and TestVNet2 gateways

In this example, both gateways are in the same subscription. You can complete this step in the same PowerShell session.

1. Get both gateways

Make sure you log in and connect to Subscription 1.

```

$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet2gw = Get-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2

```

2. Create both connections

In this step, you create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1.

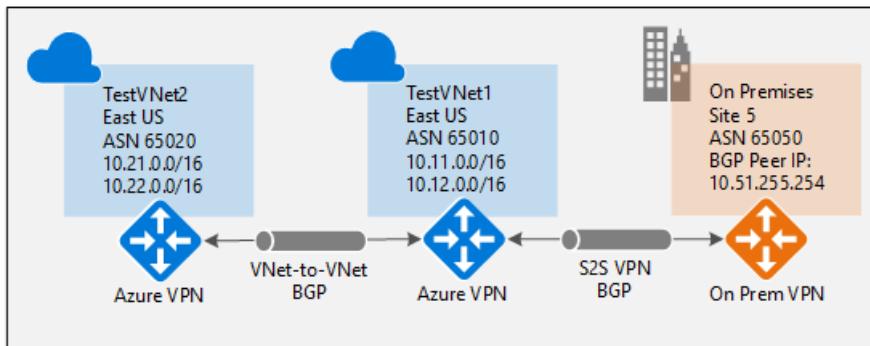
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType  
Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True  
  
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -  
VirtualNetworkGateway1 $vnet2gw -VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType  
Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True
```

IMPORTANT

Be sure to enable BGP for BOTH connections.

After completing these steps, the connection is established after a few minutes. The BGP peering session is up once the VNet-to-VNet connection is completed.

If you completed all three parts of this exercise, you have established the following network topology:



Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

How to configure BGP on an Azure VPN gateway by using CLI

10/5/2017 • 10 minutes to read • [Edit Online](#)

This article helps you enable BGP on a cross-premises Site-to-Site (S2S) VPN connection and a VNet-to-VNet connection (that is, a connection between virtual networks) by using the Azure Resource Manager deployment model and Azure CLI.

About BGP

BGP is the standard routing protocol commonly used on the internet to exchange routing and reachability information between two or more networks. BGP enables the VPN gateways and your on-premises VPN devices, called BGP peers or neighbors, to exchange routes. The routes inform both gateways about the availability and reachability for prefixes to go through the gateways or routers involved. BGP can also enable transit routing among multiple networks by propagating the routes that a BGP gateway learns from one BGP peer, to all other BGP peers.

For more information on the benefits of BGP, and to understand the technical requirements and considerations of using BGP, see [Overview of BGP with Azure VPN gateways](#).

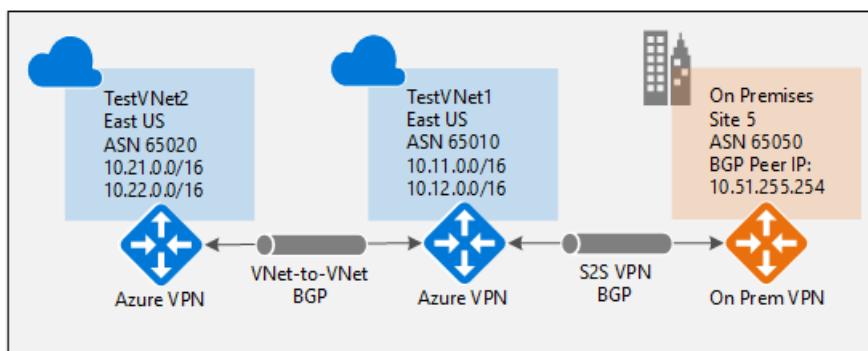
This article helps you with the following tasks:

- [Enable BGP for your VPN gateway](#) (required)

You can then complete either of the following sections, or both:

- [Establish a cross-premises connection with BGP](#)
- [Establish a VNet-to-VNet connection with BGP](#)

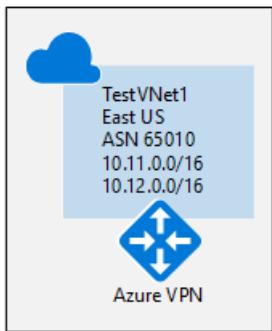
Each of these three sections forms a basic building block for enabling BGP in your network connectivity. If you complete all three sections, you build the topology as shown in the following diagram:



You can combine these sections to build a more complex multihop transit network that meets your needs.

Enable BGP for your VPN gateway

This section is required before you perform any of the steps in the other two configuration sections. The following configuration steps set up the BGP parameters of the Azure VPN gateway as shown in the following diagram:



Before you begin

Install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install Azure CLI 2.0](#) and [Get Started with Azure CLI 2.0](#).

Step 1: Create and configure TestVNet1

1. Connect to your subscription

Sign in to your Azure subscription with the `az login` command and follow the on-screen directions. For more information about signing in, see [Get Started with Azure CLI 2.0](#).

```
az login
```

If you have more than one Azure subscription, list the subscriptions for the account.

```
az account list --all
```

Specify the subscription that you want to use.

```
az account set --subscription <replace_with_your_subscription_id>
```

2. Create a resource group

The following example creates a resource group named TestRG1 in the "eastus" location. If you already have a resource group in the region where you want to create your virtual network, you can use that one instead.

```
az group create --name TestBGPRG1 --location eastus
```

3. Create TestVNet1

The following example creates a virtual network named TestVNet1 and three subnets: GatewaySubnet, FrontEnd, and BackEnd. When you're substituting values, it's important that you always name your gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation fails.

The first command creates the front-end address space and the FrontEnd subnet. The second command creates an additional address space for the BackEnd subnet. The third and fourth commands create the BackEnd subnet and GatewaySubnet.

```
az network vnet create -n TestVNet1 -g TestBGPRG1 --address-prefix 10.11.0.0/16 -l eastus --subnet-name FrontEnd --subnet-prefix 10.11.0.0/24

az network vnet update -n TestVNet1 --address-prefixes 10.11.0.0/16 10.12.0.0/16 -g TestBGPRG1

az network vnet subnet create --vnet-name TestVNet1 -n BackEnd -g TestBGPRG1 --address-prefix 10.12.0.0/24

az network vnet subnet create --vnet-name TestVNet1 -n GatewaySubnet -g TestBGPRG1 --address-prefix 10.12.255.0/27
```

Step 2: Create the VPN gateway for TestVNet1 with BGP parameters

1. Create the public IP address

Request a public IP address. The public IP address will be allocated to the VPN gateway that you create for your virtual network.

```
az network public-ip create -n GWPubIP -g TestBGPRG1 --allocation-method Dynamic
```

2. Create the VPN gateway with the AS number

Create the virtual network gateway for TestVNet1. BGP requires a Route-Based VPN gateway. You also need the additional parameter `-Asn` to set the autonomous system number (ASN) for TestVNet1. Creating a gateway can take a while (45 minutes or more) to complete.

If you run this command by using the `--no-wait` parameter, you don't see any feedback or output. The `--no-wait` parameter allows the gateway to be created in the background. It does not mean that the VPN gateway is created immediately.

```
az network vnet-gateway create -n VNet1GW -l eastus --public-ip-address GWPubIP -g TestBGPRG1 --vnet TestVNet1 --gateway-type Vpn --sku HighPerformance --vpn-type RouteBased --asn 65010 --no-wait
```

3. Obtain the Azure BGP peer IP address

After the gateway is created, you need to obtain the BGP peer IP address on the Azure VPN gateway. This address is needed to configure the VPN gateway as a BGP peer for your on-premises VPN devices.

Run the following command and check the `bgpSettings` section at the top of the output:

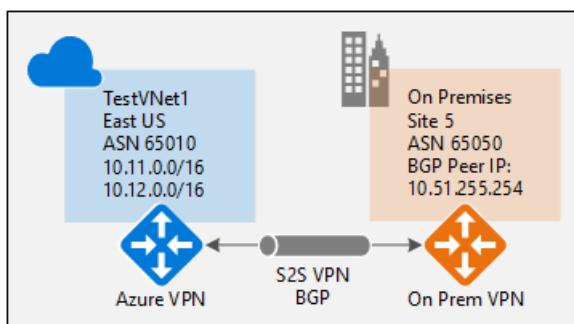
```
az network vnet-gateway list -g TestBGPRG1
```

```
"bgpSettings": {  
    "asn": 65010,  
    "bgpPeeringAddress": "10.12.255.30",  
    "peerWeight": 0  
}
```

After the gateway is created, you can use this gateway to establish a cross-premises connection or a VNet-to-VNet connection with BGP.

Establish a cross-premises connection with BGP

To establish a cross-premises connection, you need to create a local network gateway to represent your on-premises VPN device. Then you connect the Azure VPN gateway with the local network gateway. Although these steps are similar to creating other connections, they include the additional properties required to specify the BGP configuration parameters.



Step 1: Create and configure the local network gateway

This exercise continues to build the configuration shown in the diagram. Be sure to replace the values with the ones that you want to use for your configuration. When you're working with local network gateways, keep in mind the following things:

- The local network gateway can be in the same location and resource group as the VPN gateway, or it can be in a different location and resource group. This example shows the gateways in different resource groups in different locations.
- The minimum prefix that you need to declare for the local network gateway is the host address of your BGP peer IP address on your VPN device. In this case, it's a /32 prefix of 10.52.255.254/32.
- As a reminder, you must use different BGP ASNs between your on-premises networks and the Azure virtual network. If they are the same, you need to change your VNet ASN if your on-premises VPN devices already use the ASN to peer with other BGP neighbors.

Before you proceed, make sure that you've completed the [Enable BGP for your VPN gateway](#) section of this exercise and that you're still connected to Subscription 1. Notice that in this example, you create a new resource group. Also, notice the two additional parameters for the local network gateway: `Asn` and `BgpPeerAddress`.

```
az group create -n TestBGPRG5 -l eastus2

az network local-gateway create --gateway-ip-address 23.99.221.164 -n Site5 -g TestBGPRG5 --local-address-prefixes 10.51.255.254/32 --asn 65050 --bgp-peering-address 10.51.255.254
```

Step 2: Connect the VNet gateway and local network gateway

In this step, you create the connection from TestVNet1 to Site5. You must specify the `--enable-bgp` parameter to enable BGP for this connection.

In this example, the virtual network gateway and local network gateway are in different resource groups. When the gateways are in different resource groups, you must specify the entire resource ID of the two gateways to set up a connection between the virtual networks.

1. Get the resource ID of VNet1GW

Use the output from the following command to get the resource ID for VNet1GW:

```
az network vnet-gateway show -n VNet1GW -g TestBGPRG1
```

In the output, find the `"id":` line. You need the values within the quotation marks to create the connection in the next section.

Example output:

```
{
  "activeActive": false,
  "bgpSettings": {
    "asn": 65010,
    "bgpPeeringAddress": "10.12.255.30",
    "peerWeight": 0
  },
  "enableBgp": true,
  "etag": "W/\"<your etag number>\\"",
  "gatewayDefaultSite": null,
  "gatewayType": "Vpn",
  "id": "/subscriptions/<subscription
ID>/resourceGroups/TestBGPRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW",
```

Copy the values after `"id":` to a text editor, such as Notepad, so that you can easily paste them when creating your connection.

```
"id": "/subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW"
```

2. Get the resource ID of Site5

Use the following command to get the resource ID of Site5 from the output:

```
az network local-gateway show -n Site5 -g TestBGPRG5
```

3. Create the TestVNet1-to-Site5 connection

In this step, you create the connection from TestVNet1 to Site5. As discussed earlier, it is possible to have both BGP and non-BGP connections for the same Azure VPN gateway. Unless BGP is enabled in the connection property, Azure will not enable BGP for this connection, even though BGP parameters are already configured on both gateways. Replace the subscription IDs with your own.

```
az network vpn-connection create -n VNet1ToSite5 -g TestBGPRG1 --vnet-gateway1 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW --enable-bgp -l eastus --shared-key "abc123" --local-gateway2 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG5/providers/Microsoft.Network/localNetworkGateways/Site5 --no-wait
```

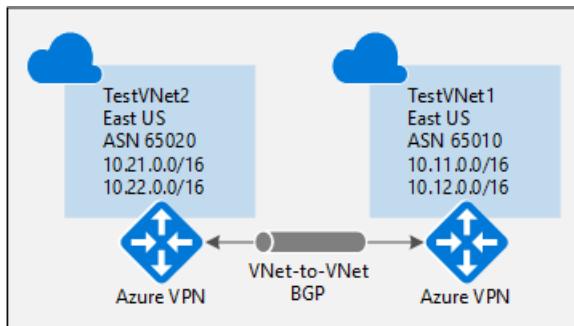
For this exercise, the following example lists the parameters to enter in the BGP configuration section of your on-premises VPN device:

Site5 ASN	:	65050
Site5 BGP IP	:	10.52.255.254
Prefixes to announce	:	(for example) 10.51.0.0/16 and 10.52.0.0/16
Azure VNet ASN	:	65010
Azure VNet BGP IP	:	10.12.255.30
Static route	:	Add a route for 10.12.255.30/32, with nexthop being the VPN tunnel interface on your device
eBGP Multihop	:	Ensure the "multihop" option for eBGP is enabled on your device if needed

The connection should be established after a few minutes. The BGP peering session starts after the IPsec connection is established.

Establish a VNet-to-VNet connection with BGP

This section adds a VNet-to-VNet connection with BGP, as shown in the following diagram:



The following instructions continue from the steps in the preceding sections. To create and configure TestVNet1 and the VPN gateway with BGP, you must complete the [Enable BGP for your VPN gateway](#) section.

Step 1: Create TestVNet2 and the VPN gateway

It's important to make sure that the IP address space of the new virtual network, TestVNet2, does not overlap with any of your VNet ranges.

In this example, the virtual networks belong to the same subscription. You can set up VNet-to-VNet connections between different subscriptions. To learn more, see [Configure a VNet-to-VNet connection](#). Make sure that you add `-EnableBgp $True` when creating the connections to enable BGP.

1. Create a new resource group

```
az group create -n TestBGPRG2 -l westus
```

2. Create TestVNet2 in the new resource group

The first command creates the front-end address space and the FrontEnd subnet. The second command creates an additional address space for the BackEnd subnet. The third and fourth commands create the BackEnd subnet and GatewaySubnet.

```
az network vnet create -n TestVNet2 -g TestBGPRG2 --address-prefix 10.21.0.0/16 -l westus --subnet-name FrontEnd --subnet-prefix 10.21.0.0/24

az network vnet update -n TestVNet2 --address-prefixes 10.21.0.0/16 10.22.0.0/16 -g TestBGPRG2

az network vnet subnet create --vnet-name TestVNet2 -n BackEnd -g TestBGPRG2 --address-prefix 10.22.0.0/24

az network vnet subnet create --vnet-name TestVNet2 -n GatewaySubnet -g TestBGPRG2 --address-prefix 10.22.255.0/27
```

3. Create the public IP address

Request a public IP address. The public IP address will be allocated to the VPN gateway that you create for your virtual network.

```
az network public-ip create -n GWPubIP2 -g TestBGPRG2 --allocation-method Dynamic
```

4. Create the VPN gateway with the AS number

Create the virtual network gateway for TestVNet2. You must override the default ASN on your Azure VPN gateways. The ASNs for the connected virtual networks must be different to enable BGP and transit routing.

```
az network vnet-gateway create -n VNet2GW -l westus --public-ip-address GWPubIP2 -g TestBGPRG2 --vnet TestVNet2 --gateway-type Vpn --sku Standard --vpn-type RouteBased --asn 65020 --no-wait
```

Step 2: Connect the TestVNet1 and TestVNet2 gateways

In this step, you create the connection from TestVNet1 to Site5. To enable BGP for this connection, you must specify the `--enable-bgp` parameter.

In the following example, the virtual network gateway and local network gateway are in different resource groups. When the gateways are in different resource groups, you must specify the entire resource ID of the two gateways to set up a connection between the virtual networks.

1. Get the resource ID of VNet1GW

Get the resource ID of VNet1GW from the output of the following command:

```
az network vnet-gateway show -n VNet1GW -g TestBGPRG1
```

2. Get the resource ID of VNet2GW

Get the resource ID of VNet2GW from the output of the following command:

```
az network vnet-gateway show -n VNet2GW -g TestBGPRG2
```

3. Create the connections

Create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1. Replace the subscription IDs with your own.

```
az network vpn-connection create -n VNet1ToVNet2 -g TestBGPRG1 --vnet-gateway1 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW --enable-bgp -l eastus --shared-key "efg456" --vnet-gateway2 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG2/providers/Microsoft.Network/virtualNetworkGateways/VNet2GW
```

```
az network vpn-connection create -n VNet2ToVNet1 -g TestBGPRG2 --vnet-gateway1 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG2/providers/Microsoft.Network/virtualNetworkGateways/VNet2GW --enable-bgp -l westus --shared-key "efg456" --vnet-gateway2 /subscriptions/<subscription ID>/resourceGroups/TestBGPRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW
```

IMPORTANT

Enable BGP for *both* connections.

After you complete these steps, the connection will be established in a few minutes. The BGP peering session will be up after the VNet-to-VNet connection is completed.

Next steps

After your connection is completed, you can add virtual machines to your virtual networks. For steps, see [Create a virtual machine](#).

Configure forced tunneling using the Azure Resource Manager deployment model

2/1/2018 • 6 minutes to read • [Edit Online](#)

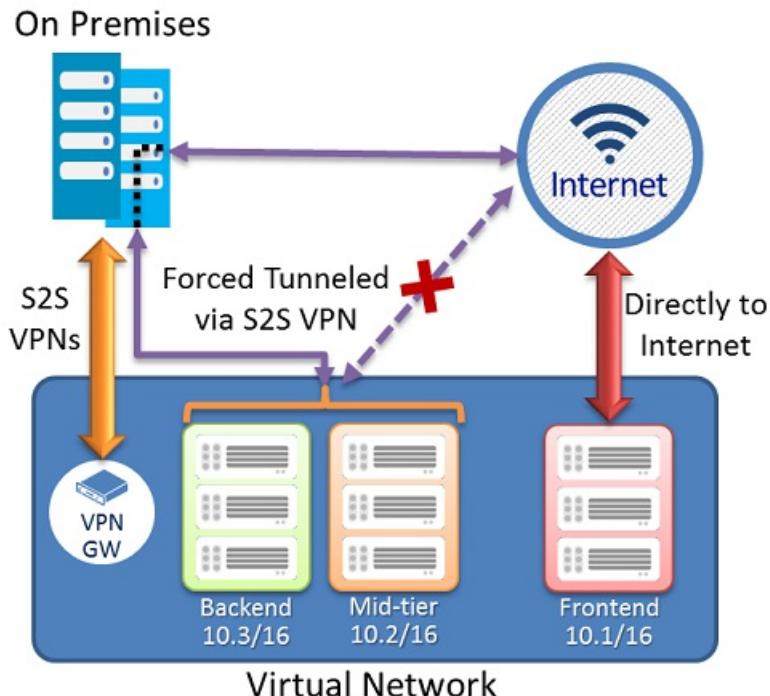
Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies. Without forced tunneling, Internet-bound traffic from your VMs in Azure always traverses from Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic. Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches.

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

This article walks you through configuring forced tunneling for virtual networks created using the Resource Manager deployment model. Forced tunneling can be configured by using PowerShell, not through the portal. If you want to configure forced tunneling for the classic deployment model, select classic article from the following dropdown list:

About forced tunneling

The following diagram illustrates how forced tunneling works.



In the example above, the Frontend subnet is not forced tunneled. The workloads in the Frontend subnet can continue to accept and respond to customer requests from the Internet directly. The Mid-tier and Backend subnets are forced tunneled. Any outbound connections from these two subnets to the Internet will be forced or redirected back to an on-premises site via one of the S2S VPN tunnels.

This allows you to restrict and inspect Internet access from your virtual machines or cloud services in Azure, while

continuing to enable your multi-tier service architecture required. If there are no Internet-facing workloads in your virtual networks, you also can apply forced tunneling to the entire virtual networks.

Requirements and considerations

Forced tunneling in Azure is configured via virtual network user-defined routes. Redirecting traffic to an on-premises site is expressed as a Default Route to the Azure VPN gateway. For more information about user-defined routing and virtual networks, see [User-defined routes and IP forwarding](#).

- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network.
 - **On-premises routes:** To the Azure VPN gateway.
 - **Default route:** Directly to the Internet. Packets destined to the private IP addresses not covered by the previous two routes are dropped.
- This procedure uses user-defined routes (UDR) to create a routing table to add a default route, and then associate the routing table to your VNet subnet(s) to enable forced tunneling on those subnets.
- Forced tunneling must be associated with a VNet that has a route-based VPN gateway. You need to set a "default site" among the cross-premises local sites connected to the virtual network. Also, the on-premises VPN device must be configured using 0.0.0.0/0 as traffic selectors.
- ExpressRoute forced tunneling is not configured via this mechanism, but instead, is enabled by advertising a default route via the ExpressRoute BGP peering sessions. For more information, see the [ExpressRoute Documentation](#).

Configuration overview

The following procedure helps you create a resource group and a VNet. You'll then create a VPN gateway and configure forced tunneling. In this procedure, the virtual network 'MultiTier-VNet' has three subnets: 'Frontend', 'Midtier', and 'Backend', with four cross-premises connections: 'DefaultSiteHQ', and three Branches.

The procedure steps set the 'DefaultSiteHQ' as the default site connection for forced tunneling, and configure the 'Midtier' and 'Backend' subnets to use forced tunneling.

Before you begin

Install the latest version of the Azure Resource Manager PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

IMPORTANT

Installing the latest version of the PowerShell cmdlets is required. Otherwise, you may receive validation errors when running some of the cmdlets.

To log in

Before beginning this configuration, you must sign in to your Azure account. The cmdlet prompts you for the sign-in credentials for your Azure account. After signing in, it downloads your account settings so they are available to Azure PowerShell. For more information, see [Using Windows PowerShell with Resource Manager](#).

To sign in, open your PowerShell console with elevated privileges, and connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Configure forced tunneling

NOTE

You may see warnings saying "The output object type of this cmdlet will be modified in a future release". This is expected behavior and you can safely ignore these warnings.

1. Create a resource group.

```
New-AzureRmResourceGroup -Name 'ForcedTunneling' -Location 'North Europe'
```

2. Create a virtual network and specify subnets.

```
$s1 = New-AzureRmVirtualNetworkSubnetConfig -Name "Frontend" -AddressPrefix "10.1.0.0/24"
$s2 = New-AzureRmVirtualNetworkSubnetConfig -Name "Midtier" -AddressPrefix "10.1.1.0/24"
$s3 = New-AzureRmVirtualNetworkSubnetConfig -Name "Backend" -AddressPrefix "10.1.2.0/24"
$s4 = New-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix "10.1.200.0/28"
$vnet = New-AzureRmVirtualNetwork -Name "MultiTier-VNet" -Location "North Europe" -ResourceGroupName
"ForcedTunneling" -AddressPrefix "10.1.0.0/16" -Subnet $s1,$s2,$s3,$s4
```

3. Create the local network gateways.

```
$lNg1 = New-AzureRmLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -GatewayIpAddress "111.111.111.111" -AddressPrefix "192.168.1.0/24"
$lNg2 = New-AzureRmLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling" -Location
"North Europe" -GatewayIpAddress "111.111.111.112" -AddressPrefix "192.168.2.0/24"
$lNg3 = New-AzureRmLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling" -Location
"North Europe" -GatewayIpAddress "111.111.111.113" -AddressPrefix "192.168.3.0/24"
$lNg4 = New-AzureRmLocalNetworkGateway -Name "Branch3" -ResourceGroupName "ForcedTunneling" -Location
"North Europe" -GatewayIpAddress "111.111.111.114" -AddressPrefix "192.168.4.0/24"
```

4. Create the route table and route rule.

```
New-AzureRmRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling" -Location "North
Europe"
$rt = Get-AzureRmRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling"
Add-AzureRmRouteConfig -Name "DefaultRoute" -AddressPrefix "0.0.0.0/0" -NextHopType
VirtualNetworkGateway -RouteTable $rt
Set-AzureRmRouteTable -RouteTable $rt
```

5. Associate the route table to the Midtier and Backend subnets.

```
$vnet = Get-AzureRmVirtualNetwork -Name "MultiTier-Vnet" -ResourceGroupName "ForcedTunneling"
Set-AzureRmVirtualNetworkSubnetConfig -Name "MidTier" -VirtualNetwork $vnet -AddressPrefix
"10.1.1.0/24" -RouteTable $rt
Set-AzureRmVirtualNetworkSubnetConfig -Name "Backend" -VirtualNetwork $vnet -AddressPrefix
"10.1.2.0/24" -RouteTable $rt
Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

6. Create the virtual network gateway. This step takes some time to complete, sometimes 45 minutes or more, because you are creating and configuring the gateway. If you see ValidateSet errors regarding the **GatewaySKU** value, verify that you have installed the [latest version of the PowerShell cmdlets](#). The latest version of the PowerShell cmdlets contains the new validated values for the latest Gateway SKUs.

```
$pip = New-AzureRmPublicIpAddress -Name "GatewayIP" -ResourceGroupName "ForcedTunneling" -Location
"North Europe" -AllocationMethod Dynamic
$gws subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$ipconfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "gwIpConfig" -SubnetId $gws subnet.Id -
PublicIpAddressId $pip.Id
New-AzureRmVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -IpConfigurations $ipconfig -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -EnableBgp
$false
```

7. Assign a default site to the virtual network gateway. The **-GatewayDefaultSite** is the cmdlet parameter that allows the forced routing configuration to work, so take care to configure this setting properly.

```
$LocalGateway = Get-AzureRmLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName
"ForcedTunneling"
$VirtualGateway = Get-AzureRmVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName
"ForcedTunneling"
Set-AzureRmVirtualNetworkGatewayDefaultSite -GatewayDefaultSite $LocalGateway -VirtualNetworkGateway
$VirtualGateway
```

8. Establish the Site-to-Site VPN connections.

```
$gateway = Get-AzureRmVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling"
$lng1 = Get-AzureRmLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling"
$lng2 = Get-AzureRmLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling"
$lng3 = Get-AzureRmLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling"
$lng4 = Get-AzureRmLocalNetworkGateway -Name "Branch3" -ResourceGroupName "ForcedTunneling"

New-AzureRmVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng1 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection2" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng2 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection3" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng3 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection4" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng4 -ConnectionType
IPsec -SharedKey "preSharedKey"

Get-AzureRmVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling"
```

Configure forced tunneling using the classic deployment model

8/2/2017 • 5 minutes to read • [Edit Online](#)

Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies. Without forced tunneling, Internet-bound traffic from your VMs in Azure will always traverse from Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic.

Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches.

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

This article walks you through configuring forced tunneling for virtual networks created using the classic deployment model. Forced tunneling can be configured by using PowerShell, not through the portal. If you want to configure forced tunneling for the Resource Manager deployment model, select classic article from the following dropdown list:

Requirements and considerations

Forced tunneling in Azure is configured via virtual network user-defined routes (UDR). Redirecting traffic to an on-premises site is expressed as a Default Route to the Azure VPN gateway. The following section lists the current limitation of the routing table and routes for an Azure Virtual Network:

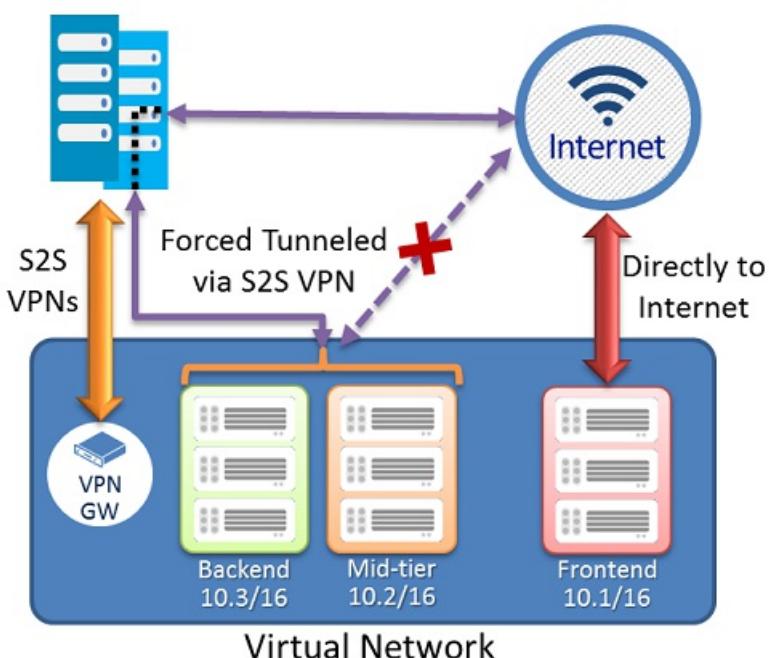
- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network.
 - **On-premises routes:** To the Azure VPN gateway.
 - **Default route:** Directly to the Internet. Packets destined to the private IP addresses not covered by the previous two routes will be dropped.
- With the release of user-defined routes, you can create a routing table to add a default route, and then associate the routing table to your VNet subnet(s) to enable forced tunneling on those subnets.
- You need to set a "default site" among the cross-premises local sites connected to the virtual network.
- Forced tunneling must be associated with a VNet that has a dynamic routing VPN gateway (not a static gateway).
- ExpressRoute forced tunneling is not configured via this mechanism, but instead, is enabled by advertising a default route via the ExpressRoute BGP peering sessions. Please see the [ExpressRoute Documentation](#) for more information.

Configuration overview

In the following example, the Frontend subnet is not forced tunneled. The workloads in the Frontend subnet can continue to accept and respond to customer requests from the Internet directly. The Mid-tier and Backend subnets are forced tunneled. Any outbound connections from these two subnets to the Internet will be forced or redirected back to an on-premises site via one of the S2S VPN tunnels.

This allows you to restrict and inspect Internet access from your virtual machines or cloud services in Azure, while continuing to enable your multi-tier service architecture required. You also can apply forced tunneling to the entire virtual networks if there are no Internet-facing workloads in your virtual networks.

On Premises



Before you begin

Verify that you have the following items before beginning configuration.

- An Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- A configured virtual network.
- The latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Configure forced tunneling

The following procedure will help you specify forced tunneling for a virtual network. The configuration steps correspond to the VNet network configuration file.

```

<VirtualNetworkSite name="MultiTier-VNet" Location="North Europe">
  <AddressSpace>
    <AddressPrefix>10.1.0.0/16</AddressPrefix>
  </AddressSpace>
  <Subnets>
    <Subnet name="Frontend">
      <AddressPrefix>10.1.0.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Midtier">
      <AddressPrefix>10.1.1.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Backend">
      <AddressPrefix>10.1.2.0/23</AddressPrefix>
    </Subnet>
    <Subnet name="GatewaySubnet">
      <AddressPrefix>10.1.200.0/28</AddressPrefix>
    </Subnet>
  </Subnets>
  <Gateway>
    <ConnectionsToLocalNetwork>
      <LocalNetworkSiteRef name="DefaultSiteHQ">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch1">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch2">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch3">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
    </ConnectionsToLocalNetwork>
  </Gateway>
</VirtualNetworkSite>
</VirtualNetworkSite>

```

In this example, the virtual network 'MultiTier-VNet' has three subnets: 'Frontend', 'Midtier', and 'Backend' subnets, with four cross premises connections: 'DefaultSiteHQ', and three Branches.

The steps will set the 'DefaultSiteHQ' as the default site connection for forced tunneling, and configure the Midtier and Backend subnets to use forced tunneling.

1. Create a routing table. Use the following cmdlet to create your route table.

```
New-AzureRouteTable -Name "MyRouteTable" -Label "Routing Table for Forced Tunneling" -Location "North Europe"
```

2. Add a default route to the routing table.

The following example adds a default route to the routing table created in Step 1. Note that the only route supported is the destination prefix of "0.0.0.0/0" to the "VPNGateway" NextHop.

```
Get-AzureRouteTable -Name "MyRouteTable" | Set-AzureRoute -RouteTable "MyRouteTable" -RouteName "DefaultRoute" -AddressPrefix "0.0.0.0/0" -NextHopType VPNGateway
```

3. Associate the routing table to the subnets.

After a routing table is created and a route added, use the following example to add or associate the route table to a VNet subnet. The example adds the route table "MyRouteTable" to the Midtier and Backend subnets of VNet MultiTier-VNet.

```
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Midtier" -RouteTableName "MyRouteTable"
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Backend" -RouteTableName "MyRouteTable"
```

4. Assign a default site for forced tunneling.

In the preceding step, the sample cmdlet scripts created the routing table and associated the route table to two of the VNet subnets. The remaining step is to select a local site among the multi-site connections of the virtual network as the default site or tunnel.

```
$DefaultSite = @("DefaultSiteHQ")
Set-AzureVNetGatewayDefaultSite -VNetName "MultiTier-VNet" -DefaultSite "DefaultSiteHQ"
```

Additional PowerShell cmdlets

To delete a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To list a route table

```
Get-AzureRouteTable [-Name <routeTableName> [-DetailLevel <detailLevel>]]
```

To delete a route from a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To remove a route from a subnet

```
Remove-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To list the route table associated with a subnet

```
Get-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To remove a default site from a VNet VPN gateway

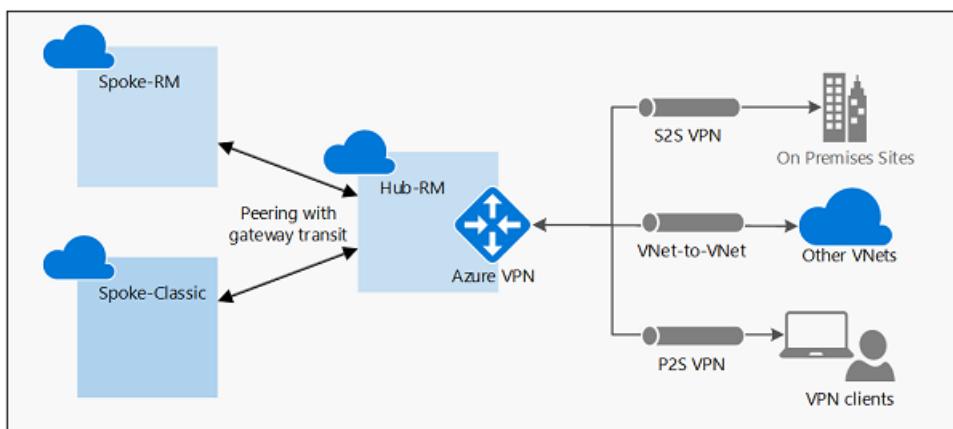
```
Remove-AzureVnetGatewayDefaultSite -VNetName <virtualNetworkName>
```

Configure VPN gateway transit for virtual network peering

4/25/2018 • 5 minutes to read • [Edit Online](#)

This article helps you configure gateway transit for virtual network peering. [Virtual network peering](#) seamlessly connects two Azure virtual networks, merging the two virtual networks into one for connectivity purposes.

[Gateway transit](#) is a peering property that enables one virtual network to utilize the VPN gateway in the peered virtual network for cross-premises or VNet-to-VNet connectivity. The following diagram shows how gateway transit works with virtual network peering.



In the diagram, gateway transit allows the peered virtual networks to use the Azure VPN gateway in Hub-RM. Connectivity available on the VPN gateway, including S2S, P2S, and VNet-to-VNet connections, applies to all three virtual networks. The transit option is available for peering between the same or different deployment models. The constraint is that the VPN gateway can only be in the virtual network using Resource Manager deployment model, as shown in the diagram.

In hub-and-spoke network architecture, gateway transit allows spoke virtual networks to share the VPN gateway in the hub, instead of deploying VPN gateways in every spoke virtual network. Routes to the gateway-connected virtual networks or on-premises networks will propagate to the routing tables for the peered virtual networks using gateway transit. You can disable the automatic route propagation from the VPN gateway. Create a routing table with the "**Disable BGP route propagation**" option, and associate the routing table to the subnets to prevent the route distribution to those subnets. For more information, see [Virtual network routing table](#).

There are two scenarios described in this document:

1. Both virtual networks are using the Resource Manager deployment model
2. The spoke virtual network is classic, and the hub virtual network with gateway is in Resource Manager

IMPORTANT

Gateway transit is currently not supported with global virtual network peering.

Requirements

The example in this document requires the following resources to be created:

1. Hub-RM virtual network with a VPN gateway
2. Spoke-RM virtual network

3. Spoke-Classic virtual network with the classic deployment model
4. The account you use requires the necessary roles and permission. See the [Permissions](#) section of this article for details.

Refer to the following documents for instructions:

1. [Create a VPN gateway in a virtual network](#)
2. [Create virtual network peering with the same deployment model](#)
3. [Create virtual network peering with different deployment models](#)

Permissions

The accounts you use to create a virtual network peering must have the necessary roles or permissions. In the example below, if you were peering two virtual networks named Hub-RM and Spoke-Classic, your account must have the following roles or permissions for each virtual network:

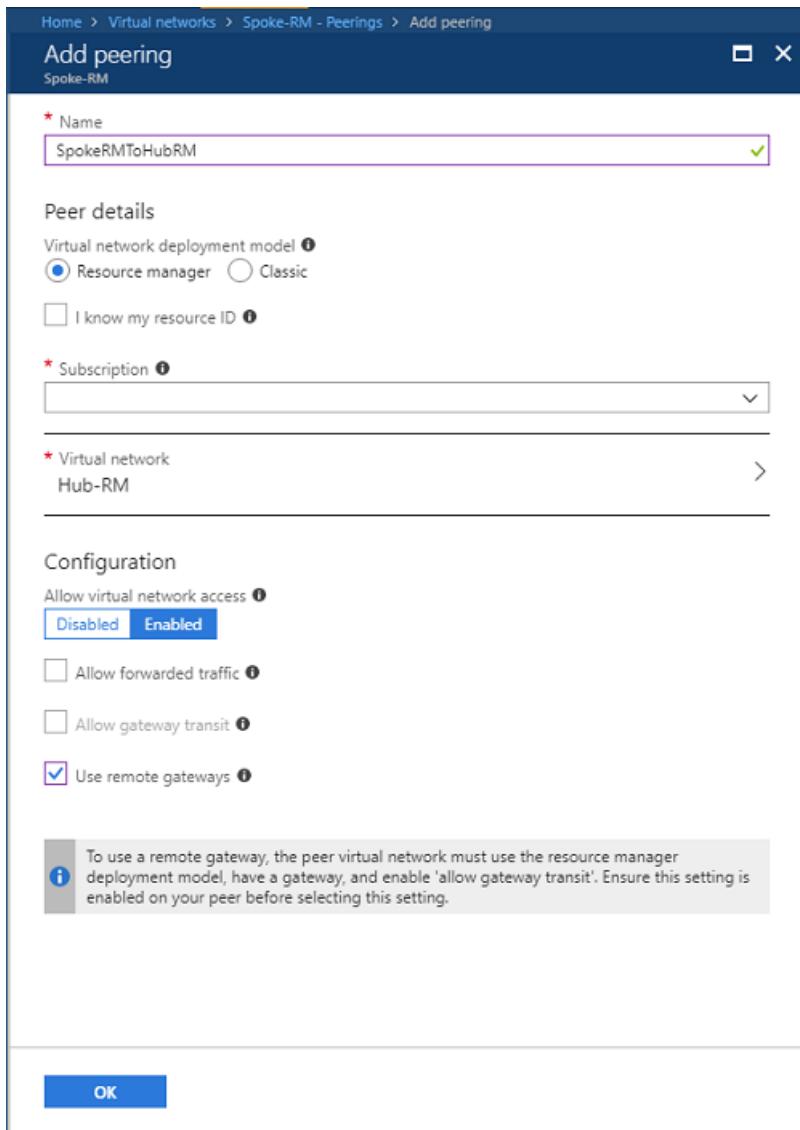
VIRTUAL NETWORK	DEPLOYMENT MODEL	ROLE	PERMISSIONS
Hub-RM	Resource Manager	Network Contributor	Microsoft.Network/virtualNetworks/virtualNetworkPeering/write
	Classic	Classic Network Contributor	N/A
Spoke-Classic	Resource Manager	Network Contributor	Microsoft.Network/virtualNetworks/peer
	Classic	Classic Network Contributor	Microsoft.ClassicNetwork/virtualNetworks/peer

Learn more about [built-in roles](#) and assigning specific permissions to [custom roles](#) (Resource Manager only).

Resource Manager to Resource Manager peering with gateway transit

Follow the instructions to create or update the virtual network peerings to enable gateway transit.

1. Create or update the virtual network peering from Spoke-RM to Hub-RM from the Azure portal. Navigate to the Spoke-RM virtual network resource, click on "Peerings", then "Add":
 - Set the "Resource Manager" option
 - Select the Hub-RM virtual network in the corresponding subscription
 - Make sure "Allow virtual network access" is "Enabled"
 - Set the "**Use remote gateways**" option
 - Click "OK"



2. If the peering is already created, navigate to the peering resource, then enable the "**Use remote gateways**" option similar to the screenshot shown in step (1)
3. Create or update the virtual network peering from Hub-RM to Spoke-RM from the Azure portal. Navigate to the Hub-RM virtual network resource, click on "Peerings", then "Add":
 - Set the "Resource Manager" option
 - Make sure "Allow virtual network access" is "Enabled"
 - Select the "Spoke-RM" virtual network in the corresponding subscription
 - Set the "**Allow gateway transit**" option
 - Click "OK"

Home > Virtual networks > Hub-RM - Peerings > Add peering

Add peering

Hub-RM

* Name
HubRMTOSpokeRM ✓

Peer details

Virtual network deployment model i
 Resource manager Classic

I know my resource ID i

* Subscription i

* Virtual network
Spoke-RM >

Configuration

Allow virtual network access i

Allow forwarded traffic i

Allow gateway transit i

Use remote gateways i

i Virtual network 'Hub-RM' has a gateway; peerings created from this virtual network can't enable 'use remote gateways'.

4. If the peering is already created, navigate to the peering resource, then enable the "**Allow gateway transit**" option similar to the screenshot shown in step (3)
5. Verify the peering status as "**Connected**" on both virtual networks

PowerShell sample

You can also use PowerShell to create or update the peering with the example above. Replace the variables with the names of your virtual networks and resource groups.

```

$SpokeRG = "SpokeRG1"
$SpokeRM = "Spoke-RM"
$HubRG   = "HubRG1"
$HubRM   = "Hub-RM"

$spokermvnet = Get-AzureRmVirtualNetwork -Name $SpokeRM -ResourceGroup $SpokeRG
$hubrmvnet   = Get-AzureRmVirtualNetwork -Name $HubRM -ResourceGroup $HubRG

Add-AzureRmVirtualNetworkPeering ` 
-Name SpokeRMtoHubRM ` 
-VirtualNetwork $spokermvnet ` 
-RemoteVirtualNetworkId $hubrmvnet.Id ` 
-UseRemoteGateways

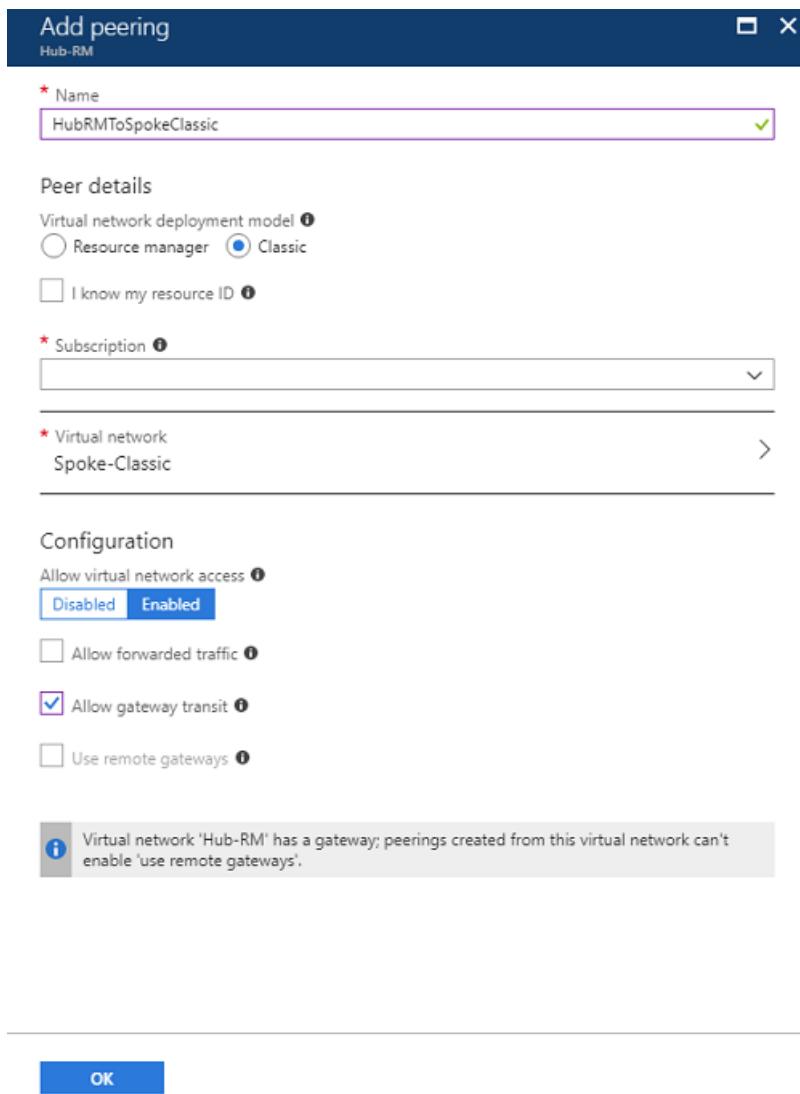
Add-AzureRmVirtualNetworkPeering ` 
-Name HubRMTospokeRM ` 
-VirtualNetwork $hubrmvnet ` 
-RemoteVirtualNetworkId $spokermvnet.Id ` 
-AllowGatewayTransit

```

Classic to Resource Manager peering with gateway transit

The steps are similar to the Resource Manager example, except the operations are applied on the Hub-RM virtual network only.

1. Create or update the virtual network peering from Hub-RM to Spoke-RM from the Azure portal. Navigate to the Hub-RM virtual network resource, click on "Peerings", then "Add":
 - Set the "Classic" option for Virtual network deployment model
 - Select the "Spoke-Classic" virtual network in the corresponding subscription
 - Make sure "Allow virtual network access" is "Enabled"
 - Set the **"Allow gateway transit"** option
 - Click "OK"



2. If the peering is already created, navigate to the peering resource, then enable the "**Allow gateway transit**" option similar to the screenshot shown in step (1)
3. There is no operation on the Spoke-Classic virtual network
4. Verify the peering status as "**Connected**" on the Hub-RM virtual network

Once the status shows "Connected", the spoke virtual networks can start using VNet-to-VNet or cross-premises connectivity through the VPN gateway in the hub virtual network.

PowerShell sample

You can also use PowerShell to create or update the peering with the example above. Replace the variables and subscription ID with the values of your virtual network and resource groups, and subscription. You only need to create virtual network peering on the hub virtual network.

```
$HubRG    = "HubRG1"
$HubRM   = "Hub-RM"

$hubrmvnet = Get-AzureRmVirtualNetwork -Name $HubRM -ResourceGroup $HubRG

Add-AzureRmVirtualNetworkPeering ` 
-Name HubRMTOSpokeRM ` 
-VirtualNetwork $hubrmvnet ` 
-RemoteVirtualNetworkId "/subscriptions/<subscription Id>/resourceGroups/Default-` 
Networking/providers/Microsoft.ClassicNetwork/virtualNetworks/Spoke-Classic" ` 
-AllowGatewayTransit
```

Next steps

- Learn more about [virtual network peering constraints and behaviors](#) and [virtual network peering settings](#) before creating a virtual network peering for production use.
- Learn how to [create a hub and spoke network topology](#) with virtual network peering and gateway transit.

Modify local network gateway settings using the Azure portal

8/15/2017 • 3 minutes to read • [Edit Online](#)

Sometimes the settings for your local network gateway AddressPrefix or GatewayIPAddress change. This article shows you how to modify your local network gateway settings. You can also modify these settings using a different method by selecting a different option from the following list:

Modify IP address prefixes

When you modify IP address prefixes, the steps you follow depend on whether your local network gateway has a connection.

To modify local network gateway IP address prefixes - no gateway connection

To add additional address prefixes:

1. On the Local Network Gateway resource, in the **Settings** section, click **Configuration**.
2. Add the IP address space in the *Add additional address range* box.
3. Click **Save** to save your settings.

To remove address prefixes:

1. On the Local Network Gateway resource, in the **Settings** section, click **Configuration**.
2. Click the '...' on the line containing the prefix you want to remove.
3. Click **Remove**.
4. Click **Save** to save your settings.

To modify local network gateway IP address prefixes - existing gateway connection

If you have a gateway connection and want to add or remove the IP address prefixes contained in your local network gateway, you need to do the following steps, in order. This results in some downtime for your VPN connection. When modifying IP address prefixes, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection.

1. On the Local Network Gateway resource, in the **Settings** section, click **Connections**.
2. Click the ... on the line for each connection, then click **Delete**.
3. Click **Save** to save your settings.

2. Modify the address prefixes.

To add additional address prefixes:

1. On the Local Network Gateway resource, in the **Settings** section, click **Configuration**.
2. Add the IP address space.
3. Click **Save** to save your settings.

To remove address prefixes:

1. On the Local Network Gateway resource, in the **Settings** section, click **Configuration**.
2. Click the ... on the line containing the prefix you want to remove.
3. Click **Remove**.
4. Click **Save** to save your settings.

3. Recreate the connection.

1. Navigate to the Virtual Network Gateway for your VNet. (Not the Local Network Gateway.)
2. On the Virtual Network Gateway, in the **Settings** section, click **Connections**.
3. Click the **+ Add** to open the **Add connection** blade.
4. Recreate your connection.
5. Click **OK** to create the connection.

Modify the gateway IP address

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. When you change the public IP address, the steps you follow depend on whether your local network gateway has a connection.

To modify the local network gateway IP address - no gateway connection

Use the example to modify a local network gateway that does not have a gateway connection. When modifying this value, you can also modify the address prefixes at the same time.

1. On the Local Network Gateway resource, in the **Settings** section, click **Configuration**.
2. In the **IP address** box, modify the IP address.
3. Click **Save** to save the settings.

To modify the local network gateway IP address - existing gateway connection

To modify a local network gateway that has a connection, you need to first remove the connection. After the connection is removed, you can modify the gateway IP address and recreate a new connection. You can also modify the address prefixes at the same time. This results in some downtime for your VPN connection. When modifying the gateway IP address, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection.

1. On the Local Network Gateway resource, in the **Settings** section, click **Connections**.
2. Click the ... on the line for the connection, then click **Delete**.
3. Click **Save** to save your settings.

2. Modify the IP address.

You can also modify the address prefixes at the same time.

1. In the **IP address** box, modify the IP address.
2. Click **Save** to save the settings.

3. Recreate the connection.

1. Navigate to the Virtual Network Gateway for your VNet. (Not the Local Network Gateway.)
2. On the Virtual Network Gateway, in the **Settings** section, click **Connections**.
3. Click the **+ Add** to open the **Add connection** blade.
4. Recreate your connection.
5. Click **OK** to create the connection.

Next steps

You can verify your gateway connection. See [Verify a gateway connection](#).

Modify local network gateway settings using PowerShell

8/15/2017 • 3 minutes to read • [Edit Online](#)

Sometimes the settings for your local network gateway AddressPrefix or GatewayIPAddress change. This article shows you how to modify your local network gateway settings. You can also modify these settings using a different method by selecting a different option from the following list:

Before you begin

Install the latest version of the Azure Resource Manager PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Modify IP address prefixes

To modify local network gateway IP address prefixes - no gateway connection

To add additional address prefixes:

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24', '10.101.1.0/24', '10.101.2.0/24')
```

To remove address prefixes:

Leave out the prefixes that you no longer need. In this example, we no longer need prefix 10.101.2.0/24 (from the previous example), so we update the local network gateway, excluding that prefix.

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24', '10.101.1.0/24')
```

To modify local network gateway IP address prefixes - existing gateway connection

If you have a gateway connection and want to add or remove the IP address prefixes contained in your local network gateway, you need to do the following steps, in order. This results in some downtime for your VPN connection. When modifying IP address prefixes, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. Modify the address prefixes for your local network gateway.

Set the variable for the LocalNetworkGateway.

```
$local = Get-AzureRmLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1
```

Modify the prefixes.

```
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `  
-AddressPrefix @('10.101.0.0/24','10.101.1.0/24')
```

3. Create the connection. In this example, we configure an IPsec connection type. When you recreate your connection, use the connection type that is specified for your configuration. For additional connection types, see the [PowerShell cmdlet](#) page.

Set the variable for the VirtualNetworkGateway.

```
$gateway1 = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection. This example uses the variable \$local that you set in step 2.

```
New-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 `  
-ResourceGroupName TestRG1 -Location 'East US' `  
-VirtualNetworkGateway1 $gateway1 -LocalNetworkGateway2 $local `  
-ConnectionType IPsec `  
-RoutingWeight 10 -SharedKey 'abc123'
```

Modify the gateway IP address

To modify the local network gateway 'GatewayIpAddress' - no gateway connection

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. Use the example to modify a local network gateway that does not have a gateway connection.

When modifying this value, you can also modify the address prefixes at the same time. Be sure to use the existing name of your local network gateway in order to overwrite the current settings. If you use a different name, you create a new local network gateway, instead of overwriting the existing one.

```
New-AzureRmLocalNetworkGateway -Name Site1 `  
-Location "East US" -AddressPrefix @('10.101.0.0/24','10.101.1.0/24') `  
-GatewayIpAddress "5.4.3.2" -ResourceGroupName TestRG1
```

To modify the local network gateway 'GatewayIpAddress' - existing gateway connection

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. If a gateway connection already exists, you first need to remove the connection. After the connection is removed, you can modify the gateway IP address and recreate a new connection. You can also modify the address prefixes at the same time. This results in some downtime for your VPN connection. When modifying the gateway IP address, you don't need to delete the VPN gateway. You only need to remove the connection.

1. Remove the connection. You can find the name of your connection by using the 'Get-AzureRmVirtualNetworkGatewayConnection' cmdlet.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name VNet1toSite1 `  
-ResourceGroupName TestRG1
```

2. Modify the 'GatewayIpAddress' value. You can also modify the address prefixes at the same time. Be sure to use the existing name of your local network gateway to overwrite the current settings. If you don't, you create a new local network gateway, instead of overwriting the existing one.

```
New-AzureRmLocalNetworkGateway -Name Site1 `  
-Location "East US" -AddressPrefix @('10.101.0.0/24','10.101.1.0/24') `  
-GatewayIpAddress "104.40.81.124" -ResourceGroupName TestRG1
```

3. Create the connection. In this example, we configure an IPsec connection type. When you recreate your connection, use the connection type that is specified for your configuration. For additional connection types, see the [PowerShell cmdlet](#) page. To obtain the VirtualNetworkGateway name, you can run the 'Get-AzureRmVirtualNetworkGateway' cmdlet.

Set the variables.

```
$local = Get-AzureRMLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1 `  
$vnetgw = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection.

```
New-AzureRmVirtualNetworkGatewayConnection -Name VNet1Site1 -ResourceGroupName TestRG1 `  
-Location "East US" `  
-VirtualNetworkGateway1 $vnetgw `  
-LocalNetworkGateway2 $local `  
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

Next steps

You can verify your gateway connection. See [Verify a gateway connection](#).

Modify local network gateway settings using the Azure CLI

11/29/2017 • 2 minutes to read • [Edit Online](#)

Sometimes the settings for your local network gateway Address Prefix or Gateway IP Address change. This article shows you how to modify your local network gateway settings. You can also modify these settings using a different method by selecting a different option from the following list:

Before you begin

Install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install Azure CLI 2.0](#).

Sign in to your Azure subscription with the `az login` command and follow the on-screen directions. For more information about signing in, see [Get Started with Azure CLI 2.0](#).

```
az login
```

If you have more than one Azure subscription, list the subscriptions for the account.

```
az account list --all
```

Specify the subscription that you want to use.

```
az account set --subscription <replace_with_your_subscription_id>
```

Modify IP address prefixes

To modify local network gateway IP address prefixes - no gateway connection

If you don't have a gateway connection and you want to add or remove IP address prefixes, you use the same command that you use to create the local network gateway, `az network local-gateway create`. You can also use this command to update the gateway IP address for the VPN device. To overwrite the current settings, use the existing name of your local network gateway. If you use a different name, you create a new local network gateway, instead of overwriting the existing one.

Each time you make a change, the entire list of prefixes must be specified, not just the prefixes that you want to change. Specify only the prefixes that you want to keep. In this case, 10.0.0.0/24 and 20.0.0.0/24

```
az network local-gateway create --gateway-ip-address 23.99.221.164 --name Site2 -g TestRG1 --local-address-prefixes 10.0.0.0/24 20.0.0.0/24
```

To modify local network gateway IP address prefixes - existing gateway connection

If you have a gateway connection and want to add or remove IP address prefixes, you can update the prefixes using `az network local-gateway update`. This results in some downtime for your VPN connection. When modifying the IP address prefixes, you don't need to delete the VPN gateway.

Each time you make a change, the entire list of prefixes must be specified, not just the prefixes that you want to

change. In this example, 10.0.0.0/24 and 20.0.0.0/24 are already present. We add the prefixes 30.0.0.0/24 and 40.0.0.0/24 and specify all 4 of the prefixes when updating.

```
az network local-gateway update --local-address-prefixes 10.0.0.0/24 20.0.0.0/24 30.0.0.0/24 40.0.0.0/24 --name VNet1toSite2 -g TestRG1
```

Modify the gateway IP address

To modify the local network gateway 'gatewayIpAddress'

If the VPN device that you want to connect to has changed its public IP address, you need to modify the local network gateway to reflect that change. The gateway IP address can be changed without removing an existing VPN gateway connection (if you have one). To modify the gateway IP address, replace the values 'Site2' and 'TestRG1' with your own using the [az network local-gateway update](#) command.

```
az network local-gateway update --gateway-ip-address 23.99.222.170 --name Site2 --resource-group TestRG1
```

Verify that the IP address is correct in the output:

```
"gatewayIpAddress": "23.99.222.170",
```

Next steps

You can verify your gateway connection. See [Verify a gateway connection](#).

Create a route-based VPN gateway using the Azure portal

9/11/2018 • 3 minutes to read • [Edit Online](#)

This article helps you quickly create a route-based Azure VPN gateway using the Azure portal. A VPN gateway is used when creating a VPN connection to your on-premises network. You can also use a VPN gateway to connect VNets.

The steps in this article will create a VNet, a subnet, a gateway subnet, and a route-based VPN gateway (virtual network gateway). Once the gateway creation has completed, you can then create connections. These steps require an Azure subscription. If you don't have an Azure subscription, create a [free account](#) before you begin.

Create a virtual network

1. From a browser, navigate to the [Azure portal](#) and sign in with your Azure account.
2. Click **Create a resource**. In the **Search the marketplace** field, type 'virtual network'. Locate **Virtual network** from the returned list and click to open the **Virtual Network** page.
3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, verify that **Resource Manager** is selected from the dropdown, and then click **Create**. This opens the **Create virtual network** page.
4. On the **Create virtual network** page, configure the VNet settings. When you fill in the fields, the red exclamation mark becomes a green check mark when the characters entered in the field are valid. Use the following values:
 - **Name:** TestVNet1
 - **Address space:** 10.1.0.0/16
 - **Subscription:** Verify that the subscription listed is the one you want to use. You can change subscriptions by using the drop-down.
 - **Resource group:** TestRG1
 - **Location:** East US
 - **Subnet:** Frontend
 - **Address range:** 10.1.0.0/24

Create virtual network X

* Name
VNet1 ✓

* Address space ✓
10.1.0.0/16
10.1.0.0 - 10.1.255.255 (65536 addresses)

* Subscription
Windows Azure Internal Consumption ✓

* Resource group
 Create new Use existing
TestRG1 ✓

* Location
East US ✓

Subnet

* Name
Frontend ✓

* Address range ✓
10.1.0.0/24
10.1.0.0 - 10.1.0.255 (256 addresses)

Service endpoints ✓
 Disabled Enabled

Pin to dashboard

Create Automation options

5. After entering the values, select **Pin to dashboard** to make it easy to find your VNet on the dashboard, and then click **Create**. After clicking **Create**, you see a tile on your dashboard that reflects the progress of your VNet. The tile changes as the VNet is being created.

Add a gateway subnet

The gateway subnet contains the reserved IP addresses that the virtual network gateway services use. Create a

gateway subnet.

1. In the portal, navigate to the virtual network for which you want to create a virtual network gateway.
2. On your virtual network page, click **Subnets** to expand **VNet1 - Subnets** page.
3. Click **+Gateway subnet** at the top to open the **Add subnet** page.

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES
Frontend	10.1.0.0/24	251

4. The **Name** for your subnet is automatically filled in with the required value 'GatewaySubnet'. Adjust the auto-filled **Address range** values to match the following values:

Address range (CIDR block): 10.1.255.0/27

Add subnet
VNet1

* Name
GatewaySubnet

* Address range (CIDR block) ⓘ
10.1.255.0/27
10.1.1.0 - 10.1.1.255 (251 + 5 Azure reserved addresses)

Route table
None >

Service endpoints

Services ⓘ
0 selected

5. To create the gateway subnet, click **OK** at the bottom of the page.

Configure gateway settings

1. On the left side of the portal page, click **+ Create a resource** and type 'Virtual Network Gateway' in the search box, then press **Enter**. In **Results**, locate and click **Virtual network gateway**.
2. At the bottom of the 'Virtual network gateway' page, click **Create** to open the **Create virtual network gateway** page.
3. On the **Create virtual network gateway** page, specify the values for your virtual network gateway.
 - **Name:** Vnet1GW
 - **Gateway type:** VPN
 - **VPN type:** Route-based
 - **SKU:** VpnGw1
 - **Location:** East US
 - **Virtual network:** Click **Virtual network/Choose a virtual network** to open the **Choose a virtual**

network page. Select **VNet1**.

Create virtual network gatew...

* Name
VNet1GW

Gateway type

VPN type

* SKU

Enable active-active mode

* Virtual network >

* First IP configuration >

Configure BGP ASN

* Subscription
Windows Azure Internal Consumption

Resource group

* Location <input type="button" value="▼"/>

Pin to dashboard

Provisioning a virtual network gateway may take up to 45 minutes.

Create a public IP address

A VPN gateway must have a dynamically allocated public IP address. When you create a connection to a VPN gateway, this is the IP address that your on-premises device connects to.

1. Select **First IP configuration Create gateway IP configuration** to request a public IP address.

2. On the **Choose public IP page**, click **+ Create new** to open the **Create public IP address** page.

3. Configure the settings with the following values:

- **Name: VNet1GWIP**
- **SKU: Basic**

4. Click **OK** at the bottom of this page to save your changes.

Create the VPN gateway

1. Verify the settings on the **Create virtual network gateway** page. Adjust values if necessary.

Create virtual network gateway

* Name
VNet1GW

Gateway type
 VPN ExpressRoute

VPN type
 Route-based Policy-based

* SKU
VpnGw1

Enable active-active mode

* Virtual network
VNet1

* First IP configuration
VNet1GWIP

Configure BGP ASN

* Subscription
Windows Azure Internal Consumption

Resource group
TestRG1

Pin to dashboard

Create [Automation options](#)

Provisioning a virtual network gateway may take up to 45 minutes.

2. Click **Create** at the bottom of the page.

After you click **Create**, the settings are validated and the **Deploying Virtual network gateway** tile appears on the dashboard. A VPN gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.

View the VPN gateway

- After the gateway is created, navigate to VNet1 in the portal. The VPN gateway appears on the Overview page as a connected device.

The screenshot shows the Azure portal interface for a virtual network named 'VNet1'. On the left, there's a navigation sidebar with options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Address space', and 'Connected devices'. The 'Connected devices' section contains a table with one row, which is highlighted with a red border. The table has columns for 'DEVICE', 'TYPE', 'IP ADDRESS', and 'SUBNET'. The single entry is 'VNet1GW' under 'DEVICE' and 'Virtual network gateway' under 'TYPE'. To the right of the table, there are details about the resource: Resource group 'TestRG1', Location 'East US', Subscription 'Windows Azure Internal Consumption', Address space '10.1.0.0/16', DNS servers 'Azure provided DNS service', and a note about the IP address being 'GatewaySubnet'.

- In the device list, click **VNet1GW** to view more information.

This screenshot shows the detailed view of the 'VNet1GW' virtual network gateway. The left sidebar has options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', and 'Diagnose and solve problems', with 'Overview' selected. The main pane displays various properties of the gateway: Resource group 'TestRG1', Location 'East US', Subscription ID, SKU 'VpnGw1', VPN type 'Route-based', Virtual network 'VNet1', and Public IP address '52.191.12.152 (VNet1GWIP)'. There's also a link 'Click here to add tags'.

Next steps

Once the gateway has finished creating, you can create a connection between your virtual network and another VNet. Or, create a connection between your virtual network and an on-premises location.

[Create a site-to-site connection](#)

[Create a point-to-site connection](#)

[Create a connection to another VNet](#)

Create a route-based VPN gateway using PowerShell

4/18/2018 • 4 minutes to read • [Edit Online](#)

This article helps you quickly create a route-based Azure VPN gateway using PowerShell. A VPN gateway is used when creating a VPN connection to your on-premises network. You can also use a VPN gateway to connect VNets.

The steps in this article will create a VNet, a subnet, a gateway subnet, and a route-based VPN gateway (virtual network gateway). Once the gateway creation has completed, you can then create connections. These steps require an Azure subscription. If you don't have an Azure subscription, create a [free account](#) before you begin.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account. Just click the **Copy** to copy the code, paste it into the Cloud Shell, and then press enter to run it. There are a few ways to launch the Cloud Shell:

Click Try It in the upper right corner of a code block.	
Open Cloud Shell in your browser.	
Click the Cloud Shell button on the menu in the upper right of the Azure portal.	

If you choose to install and use the PowerShell locally, this tutorial requires the Azure PowerShell module version 5.3.0 or later. Run `Get-Module -ListAvailable AzureRM` to find the version. If you need to upgrade, see [Install Azure PowerShell module](#). If you are running PowerShell locally, you also need to run `Connect-AzureRmAccount` to create a connection with Azure.

Create a resource group

Create an Azure resource group with [New-AzureRmResourceGroup](#). A resource group is a logical container into which Azure resources are deployed and managed.

```
New-AzureRmResourceGroup -Name TestRG1 -Location EastUS
```

Create a virtual network

Create a virtual network with [New-AzureRmVirtualNetwork](#). The following example creates a virtual network named **VNet1** in the **EastUS** location:

```
$virtualNetwork = New-AzureRmVirtualNetwork `<br> -ResourceGroupName TestRG1 `<br> -Location EastUS `<br> -Name VNet1 `<br> -AddressPrefix 10.1.0.0/16
```

Create a subnet configuration using the [New-AzureRmVirtualNetworkSubnetConfig](#) cmdlet.

```
$subnetConfig = Add-AzureRmVirtualNetworkSubnetConfig `  
-Name Frontend `  
-AddressPrefix 10.1.0.0/24 `  
-VirtualNetwork $virtualNetwork
```

Set the subnet configuration for the virtual network using the [Set-AzureRmVirtualNetwork](#) cmdlet.

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

Add a gateway subnet

The gateway subnet contains the reserved IP addresses that the virtual network gateway services use. Use the following examples to add a gateway subnet:

Set a variable for your VNet.

```
$vnet = Get-AzureRmVirtualNetwork -ResourceGroupName TestRG1 -Name VNet1
```

Create the gateway subnet using the [Add-AzureRmVirtualNetworkSubnetConfig](#) cmdlet.

```
Add-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27 -VirtualNetwork $vnet
```

Set the subnet configuration for the virtual network using the [Set-AzureRmVirtualNetwork](#) cmdlet.

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

Request a public IP address

A VPN gateway must have a dynamically allocated public IP address. When you create a connection to a VPN gateway, this is the IP address that you specify. Use the following example to request a public IP address:

```
$gwpip= New-AzureRmPublicIpAddress -Name VNet1GWIP -ResourceGroupName TestRG1 -Location 'East US' -  
AllocationMethod Dynamic
```

Create the gateway IP address configuration

The gateway configuration defines the subnet and the public IP address to use. Use the following example to create your gateway configuration:

```
$vnet = Get-AzureRmVirtualNetwork -Name VNet1 -ResourceGroupName TestRG1  
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet  
$gwpipconfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name gwipconfig1 -SubnetId $subnet.Id -  
PublicIpAddressId $gwpip.Id
```

Create the VPN gateway

A VPN gateway can take 45 minutes or more to create. Once the gateway has completed, you can create a connection between your virtual network and another VNet. Or, create a connection between your virtual network

and an on-premises location. Create a VPN gateway using the [New-AzureRmVirtualNetworkGateway](#) cmdlet.

```
New-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1 `  
-Location 'East US' -IpConfigurations $gwpconfig -GatewayType Vpn `  
-VpnType RouteBased -GatewaySku VpnGw1
```

View the VPN gateway

You can view the VPN gateway using the [Get-AzureRmVirtualNetworkGateway](#) cmdlet.

```
Get-AzureRmVirtualNetworkGateway -Name Vnet1GW -ResourceGroup TestRG1
```

The output will look similar to this example:

```
Name : VNet1GW  
ResourceGroupName : TestRG1  
Location : eastus  
Id : /subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW  
Etag : W/"0952d-9da8-4d7d-a8ed-28c8ca0413"  
ResourceGuid : dc6ce1de-2c4494-9d0b-20b03ac595  
ProvisioningState : Succeeded  
Tags :  
IpConfigurations : [  
    {  
        "PrivateIpAllocationMethod": "Dynamic",  
        "Subnet": {  
            "Id": "/subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworks/VNet1/subnets/GatewaySubnet"  
        },  
        "PublicIpAddress": {  
            "Id": "/subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/publicIPAddresses/VNet1GWIP"  
        },  
        "Name": "default",  
        "Etag": "W/"0952d-9da8-4d7d-a8ed-28c8ca0413\"",  
        "Id": "/subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW/ipConfigurations/default"  
    }  
]  
GatewayType : Vpn  
VpnType : RouteBased  
EnableBgp : False  
ActiveActive : False  
GatewayDefaultSite : null  
Sku : {  
    "Capacity": 2,  
    "Name": "VpnGw1",  
    "Tier": "VpnGw1"  
}  
VpnClientConfiguration : null  
BgpSettings : {
```

View the public IP address

To view the public IP address for your VPN gateway, use the [Get-AzureRmPublicIpAddress](#) cmdlet.

```
Get-AzureRmPublicIpAddress -Name VNet1GWIP -ResourceGroupName TestRG1
```

In the example response, the `IpAddress` value is the public IP address.

```
Name          : VNet1GWIP
ResourceGroupName : TestRG1
Location       : eastus
Id            : /subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/publicIPAddresses/VNet1GWIP
Etag          : W/"5001666a-bc2a-484b-bcf5-ad488dabd8ca"
ResourceGuid   : 3c7c481e-9828-4dae-abdc-f95b383
ProvisioningState : Succeeded
Tags          :
PublicIpAllocationMethod : Dynamic
IpAddress      : 13.90.153.3
PublicIpAddressVersion : IPv4
IdleTimeoutInMinutes : 4
IpConfiguration : {
    "Id": "/subscriptions/<subscription ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW/ipConfigurations/default"
}
DnsSettings    : null
Zones          : {}
Sku           : {
    "Name": "Basic"
}
IpTags         : {}
```

Clean up resources

When you no longer need the resources you created, use the [Remove-AzureRmResourceGroup](#) command to delete the resource group. This will delete the resource group and all of the resources it contains.

```
Remove-AzureRmResourceGroup -Name TestRG1
```

Next steps

Once the gateway has finished creating, you can create a connection between your virtual network and another VNet. Or, create a connection between your virtual network and an on-premises location.

[Create a site-to-site connection](#)

[Create a point-to-site connection](#)

[Create a connection to another VNet](#)

Create a route-based VPN gateway using CLI

4/9/2018 • 3 minutes to read • [Edit Online](#)

This article helps you quickly create a route-based Azure VPN gateway using the Azure CLI. A VPN gateway is used when creating a VPN connection to your on-premises network. You can also use a VPN gateway to connect VNets.

The steps in this article will create a VNet, a subnet, a gateway subnet, and a route-based VPN gateway (virtual network gateway). A virtual network gateway can take 45 minutes or more to create. Once the gateway creation has completed, you can then create connections. These steps require an Azure subscription. If you don't have an Azure subscription, create a [free account](#) before you begin.

Open Azure Cloud Shell

Azure Cloud Shell is a free, interactive shell that you can use to run the steps in this article. Common Azure tools are preinstalled and configured in Cloud Shell for you to use with your account. Just select the **Copy** button to copy the code, paste it in Cloud Shell, and then press Enter to run it. There are a few ways to open Cloud Shell:

Select Try It in the upper-right corner of a code block.	
Open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

If you choose to install and use the CLI locally, this article requires that you are running the Azure CLI version 2.0.4 or later. To find the installed version, run `az --version`. If you need to install or upgrade, see [Install Azure CLI 2.0](#).

Create a resource group

Create a resource group using the [az group create](#) command. A resource group is a logical container into which Azure resources are deployed and managed.

```
az group create --name TestRG1 --location eastus
```

Create a virtual network

Create a virtual network using the [az network vnet create](#) command. The following example creates a virtual network named **VNet1** in the **EastUS** location:

```
az network vnet create \
-n VNet1 \
-g TestRG1 \
-l eastus \
--address-prefix 10.1.0.0/16 \
--subnet-name Frontend \
--subnet-prefix 10.1.0.0/24
```

Add a gateway subnet

The gateway subnet contains the reserved IP addresses that the virtual network gateway services use. Use the following examples to add a gateway subnet:

```
az network vnet subnet create \
--vnet-name VNet1 \
-n GatewaySubnet \
-g TestRG1 \
--address-prefix 10.1.255.0/27
```

Request a public IP address

A VPN gateway must have a dynamically allocated public IP address. The public IP address will be allocated to the VPN gateway that you create for your virtual network. Use the following example to request a public IP address:

```
az network public-ip create \
-n VNet1GWIP \
-g TestRG1 \
--allocation-method Dynamic
```

Create the VPN gateway

Create the VPN gateway using the [az network vnet-gateway create](#) command.

If you run this command by using the `--no-wait` parameter, you don't see any feedback or output. The `--no-wait` parameter allows the gateway to be created in the background. It does not mean that the VPN gateway is created immediately.

```
az network vnet-gateway create \
-n VNet1GW \
-l eastus \
--public-ip-address VNet1GWIP \
-g TestRG1 \
--vnet VNet1 \
--gateway-type Vpn \
--sku VpnGw1 \
--vpn-type RouteBased \
--no-wait
```

A VPN gateway can take 45 minutes or more to create.

View the VPN gateway

```
az network vnet-gateway show \
-n VNet1GW \
-g TestRG1
```

The response looks similar to this:

```
{
  "activeActive": false,
  "bgpSettings": null,
  "enableBgp": false,
  "etag": "W/\"6c61f8cb-d90f-4796-8697\"",
  "gatewayDefaultSite": null,
  "gatewayType": "Vpn",
  "id": "/subscriptions/<subscription
ID>/resourceGroups/TestRG11/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW",
  "ipConfigurations": [
    {
      "etag": "W/\"6c61f8cb-d90f-4796-8697"",
      "id": "/subscriptions/<subscription
ID>/resourceGroups/TestRG11/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW/ipConfigurations/vnetGat
ewayConfig0",
      "name": "vnetGatewayConfig0",
      "privateIpAllocationMethod": "Dynamic",
      "provisioningState": "Updating",
      "publicIpAddress": {
        "id": "/subscriptions/<subscription
ID>/resourceGroups/TestRG11/providers/Microsoft.Network/publicIPAddresses/VNet1GWIP",
        "resourceGroup": "TestRG1"
      },
      "resourceGroup": "TestRG1",
      "subnet": {
        "id": "/subscriptions/<subscription
ID>/resourceGroups/TestRG11/providers/Microsoft.Network/virtualNetworks/VNet1/subnets/GatewaySubnet",
        "resourceGroup": "TestRG1"
      }
    ],
    "location": "eastus",
    "name": "VNet1GW",
    "provisioningState": "Updating",
    "resourceGroup": "TestRG1",
    "resourceGuid": "69c269e3-622c-4123-9231",
    "sku": {
      "capacity": 2,
      "name": "VpnGw1",
      "tier": "VpnGw1"
    },
    "tags": null,
    "type": "Microsoft.Network/virtualNetworkGateways",
    "vpnClientConfiguration": null,
    "vpnType": "RouteBased"
  }
}
```

View the public IP address

To view the public IP address assigned to your gateway, use the following example:

```
az network public-ip show \
--name VNet1GWIP \
--resource-group TestRG11
```

The value associated with the **ipAddress** field is the public IP address of your VPN gateway.

Example response:

```
{  
  "dnsSettings": null,  
  "etag": "W/\"a12d4d03-b27a-46cc-b222-8d9364b8166a\"",  
  "id": "/subscriptions/<subscription  
ID>/resourceGroups/TestRG1/providers/Microsoft.Network/publicIPAddresses/VNet1GWIP",  
  "idleTimeoutInMinutes": 4,  
  "ipAddress": "13.90.195.184",  
  "ipConfiguration": {  
    "etag": null,  
    "id": "/subscriptions/<subscription  
ID>/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW/ipConfigurations/vnetGate  
wayConfig0",
```

Clean up resources

When you no longer need the resources you created, use [az group delete](#) to delete the resource group. This will delete the resource group and all of the resources it contains.

```
az group delete --name TestRG1 --yes
```

Next steps

Once the gateway has finished creating, you can create a connection between your virtual network and another VNet. Or, create a connection between your virtual network and an on-premises location.

[Create a site-to-site connection](#)

[Create a point-to-site connection](#)

[Create a connection to another VNet](#)

Verify a VPN Gateway connection

9/13/2018 • 3 minutes to read • [Edit Online](#)

This article shows you how to verify a VPN gateway connection for both the classic and Resource Manager deployment models.

Azure portal

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](#), click **All resources** and navigate to your virtual network gateway.
2. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
3. 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.

Essentials ^	
Resource group	Data in 2.35 KB
Status Connected	Data out 3.14 KB
Location East US	Virtual network
Subscription name	Virtual network gateway
Subscription ID	Local network gateway

PowerShell

To verify a VPN gateway connection for the Resource Manager deployment model using PowerShell, install the latest version of the [Azure Resource Manager PowerShell cmdlets](#).

You can verify that your connection succeeded by using the 'Get-AzureRmVirtualNetworkGatewayConnection' 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-AzureRmVirtualNetworkGatewayConnection -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
```

Azure CLI

To verify a VPN gateway connection for the Resource Manager deployment model using Azure CLI, install the latest version of the [CLI commands](#) (2.0 or later).

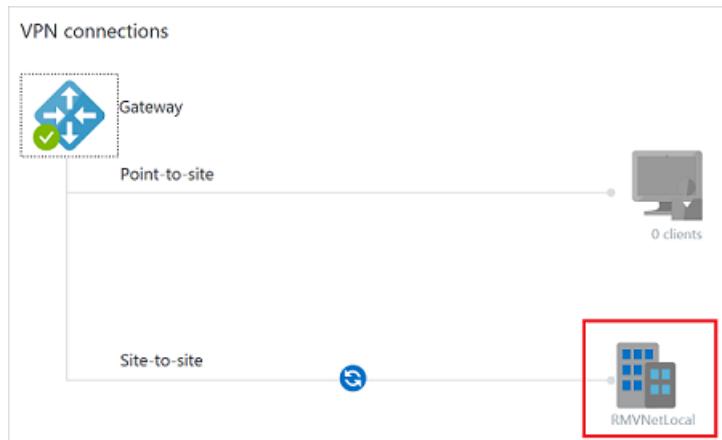
You can verify that your connection succeeded by using the `az network vpn-connection show` command. In the example, '--name' refers to the name of the connection that you want to test. When the connection is in the process of being established, its connection status shows 'Connecting'. Once the connection is established, the status changes to 'Connected'.

```
az network vpn-connection show --name VNet1toSite2 --resource-group TestRG1
```

Azure portal (classic)

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.

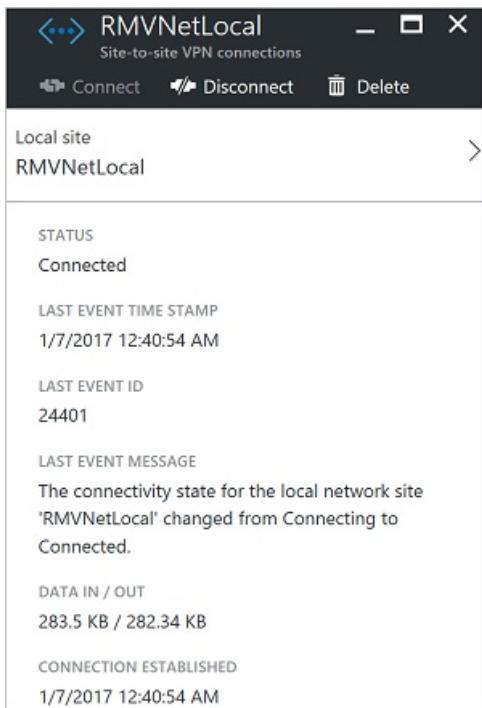
1. In the [Azure portal](#), click **All resources** and navigate to your classic virtual network.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.
3. On the VPN connections graphic, click the site.



4. On the **Site-to-site VPN connections** blade, view the information about your site.

NAME	STATUS	LAST EVENT	DATA IN / OUT
RMVNetLocal	Connected	1/7/2017 12:40:54 AM	177.97 KB / 177.72 KB

5. To view more information about the connection, click the name of the connection to open the **Site-to-site VPN Connection** blade.



PowerShell (classic)

To verify your VPN gateway connection for the classic deployment model using PowerShell, install the latest versions of the Azure PowerShell cmdlets. Be sure to download and install the [Service Management](#) module. Use 'Add-AzureAccount' to log in to the classic deployment model.

You can verify that your connection succeeded by using the 'Get-AzureVNetConnection' cmdlet.

1. Use the following cmdlet example, configuring the values to match your own. The name of the virtual network must be in quotes if it contains spaces.

```
Get-AzureVNetConnection "Group ClassicRG ClassicVNet"
```

2. After the cmdlet has finished, view the values. In the example below, the Connectivity State shows as 'Connected' and you can see ingress and egress bytes.

```
ConnectivityState      : Connected
EgressBytesTransferred : 181664
IngressBytesTransferred : 182080
LastConnectionEstablished : 1/7/2016 12:40:54 AM
LastEventID           : 24401
LastEventMessage       : The connectivity state for the local network site 'RMVNetLocal' changed
from Connecting to
                           Connected.
LastEventTimeStamp     : 1/7/2016 12:40:54 AM
LocalNetworkSiteName   : RMVNetLocal
```

Next steps

- You can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Reset a VPN Gateway

9/13/2018 • 3 minutes to read • [Edit Online](#)

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. This article helps you reset your VPN gateway.

What happens during a reset?

A VPN gateway is composed of two VM instances running in an active-standby configuration. When you reset the gateway, it reboots the gateway, and then reapplies the cross-premises configurations to it. The gateway keeps the public IP address it already has. This means you won't need to update the VPN router configuration with a new public IP address for Azure VPN gateway.

When you issue the command to reset the gateway, the current active instance of the Azure VPN gateway is rebooted immediately. There will be a brief gap during the failover from the active instance (being rebooted), to the standby instance. The gap should be less than one minute.

If the connection is not restored after the first reboot, issue the same command again to reboot the second VM instance (the new active gateway). If the two reboots are requested back to back, there will be a slightly longer period where both VM instances (active and standby) are being rebooted. This will cause a longer gap on the VPN connectivity, up to 2 to 4 minutes for VMs to complete the reboots.

After two reboots, if you are still experiencing cross-premises connectivity problems, please open a support request from the Azure portal.

Before you begin

Before you reset your gateway, verify the key items listed below for each IPsec Site-to-Site (S2S) VPN tunnel. Any mismatch in the items will result in the disconnect of S2S VPN tunnels. Verifying and correcting the configurations for your on-premises and Azure VPN gateways saves you from unnecessary reboots and disruptions for the other working connections on the gateways.

Verify the following items before resetting your gateway:

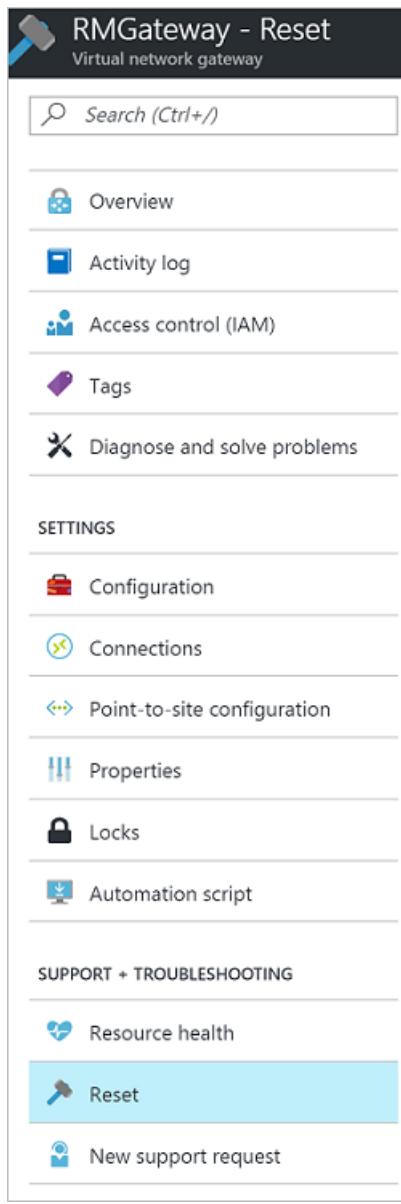
- The Internet IP addresses (VIPs) for both the Azure VPN gateway and the on-premises VPN gateway are configured correctly in both the Azure and the on-premises VPN policies.
- The pre-shared key must be the same on both Azure and on-premises VPN gateways.
- If you apply specific IPsec/IKE configuration, such as encryption, hashing algorithms, and PFS (Perfect Forward Secrecy), ensure both the Azure and on-premises VPN gateways have the same configurations.

Azure portal

You can reset a Resource Manager VPN gateway using the Azure portal. If you want to reset a classic gateway, see the [PowerShell](#) steps.

Resource Manager deployment model

1. Open the [Azure portal](#) and navigate to the Resource Manager virtual network gateway that you want to reset.
2. On the blade for the virtual network gateway, click 'Reset'.



3. On the Reset blade, click the **Reset** button.

PowerShell

Resource Manager deployment model

The cmdlet for resetting a gateway is **Reset-AzureRmVirtualNetworkGateway**. Before performing a reset, make sure you have the latest version of the [Resource Manager PowerShell cmdlets](#). The following example resets a virtual network gateway named VNet1GW in the TestRG1 resource group:

```
$gw = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
Reset-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw
```

Result:

When you receive a return result, you can assume the gateway reset was successful. However, there is nothing in the return result that indicates explicitly that the reset was successful. If you want to look closely at the history to see exactly when the gateway reset occurred, you can view that information in the [Azure portal](#). In the portal, navigate to '**GatewayName**' -> **Resource Health**.

Classic deployment model

The cmdlet for resetting a gateway is **Reset-AzureVNetGateway**. Before performing a reset, make sure you have the latest version of the [Service Management \(SM\) PowerShell cmdlets](#). The following example resets the gateway

for a virtual network named "ContosoVNet":

```
Reset-AzureVNetGateway -VnetName "ContosoVNet"
```

Result:

```
Error      : 
 HttpStatusCode : OK
 Id          : f1600632-c819-4b2f-ac0e-f4126bec1ff8
 Status      : Successful
 RequestId   : 9ca273de2c4d01e986480ce1ffa4d6d9
 StatusCode  : OK
```

Azure CLI

To reset the gateway, use the [az network vnet-gateway reset](#) command. The following example resets a virtual network gateway named VNet5GW in the TestRG5 resource group:

```
az network vnet-gateway reset -n VNet5GW -g TestRG5
```

Result:

When you receive a return result, you can assume the gateway reset was successful. However, there is nothing in the return result that indicates explicitly that the reset was successful. If you want to look closely at the history to see exactly when the gateway reset occurred, you can view that information in the [Azure portal](#). In the portal, navigate to '**GatewayName**' -> **Resource Health**.

Delete a virtual network gateway using the portal

1/4/2018 • 3 minutes to read • [Edit Online](#)

This article provides the instructions for deleting an Azure VPN gateways deployed using the Resource Manager deployment model. There are a couple of different approaches you can take when you want to delete a virtual network gateway for a VPN gateway configuration.

- If you want to delete everything and start over, as in the case of a test environment, you can delete the resource group. When you delete a resource group, it deletes all the resources within the group. This method is only recommended if you don't want to keep any of the resources in the resource group. You can't selectively delete only a few resources using this approach.
- If you want to keep some of the resources in your resource group, deleting a virtual network gateway becomes slightly more complicated. Before you can delete the virtual network gateway, you must first delete any resources that are dependent on the gateway. The steps you follow depend on the type of connections that you created and the dependent resources for each connection.

IMPORTANT

The instructions below describe how to delete Azure VPN gateways deployed using the Resource Manager deployment model. To delete a VPN gateway deployed using the classic deployment model, please use Azure PowerShell as described [here](#).

Delete a VPN gateway

To delete a virtual network gateway, you must first delete each resource that pertains to the virtual network gateway. Resources must be deleted in a certain order due to dependencies.

Step 1: Navigate to the virtual network gateway

1. In the [Azure portal](#), navigate to **All resources**.
2. To open the virtual network gateway page, navigate to the virtual network gateway that you want to delete and click it.

Step 2: Delete connections

1. On the page for your virtual network gateway, click **Connections** to view all connections to the gateway.
2. Click the '...' on the row of the name of the connection, then select **Delete** from the dropdown.
3. Click **Yes** to confirm that you want to delete the connection. If you have multiple connections, delete each connection.

Step 3: Delete the virtual network gateway

Be aware that if you have a P2S configuration to this VNet in addition to your S2S configuration, deleting the virtual network gateway will automatically disconnect all P2S clients without warning.

1. On the virtual network gateway page, click **Overview**.
2. On the **Overview** page, click **Delete** to delete the gateway.

At this point, the virtual network gateway is deleted. The next steps help you delete any resources that are no longer being used.

To delete the local network gateway

1. In **All resources**, locate the local network gateways that were associated with each connection.
2. On the **Overview** blade for the local network gateway, click **Delete**.

To delete the Public IP address resource for the gateway

1. In **All resources**, locate the Public IP address resource that was associated to the gateway. If the virtual network gateway was active-active, you will see two Public IP addresses.
2. On the **Overview** page for the Public IP address, click **Delete**, then **Yes** to confirm.

To delete the gateway subnet

1. In **All resources**, locate the virtual network.
2. On the **Subnets** blade, click the **GatewaySubnet**, then click **Delete**.
3. Click **Yes** to confirm that you want to delete the gateway subnet.

Delete a VPN gateway by deleting the resource group

If you are not concerned about keeping any of your resources in the resource group and you just want to start over, you can delete an entire resource group. This is a quick way to remove everything. The following steps apply only to the Resource Manager deployment model.

1. In **All resources**, locate the resource group and click to open the blade.
2. Click **Delete**. On the Delete blade, view the affected resources. Make sure that you want to delete all of these resources. If not, use the steps in [Delete a VPN gateway](#) at the top of this article.
3. To proceed, type the name of the resource group that you want to delete, then click **Delete**.

Delete a virtual network gateway using PowerShell

4/18/2018 • 8 minutes to read • [Edit Online](#)

There are a couple of different approaches you can take when you want to delete a virtual network gateway for a VPN gateway configuration.

- If you want to delete everything and start over, as in the case of a test environment, you can delete the resource group. When you delete a resource group, it deletes all the resources within the group. This method is only recommended if you don't want to keep any of the resources in the resource group. You can't selectively delete only a few resources using this approach.
- If you want to keep some of the resources in your resource group, deleting a virtual network gateway becomes slightly more complicated. Before you can delete the virtual network gateway, you must first delete any resources that are dependent on the gateway. The steps you follow depend on the type of connections that you created and the dependent resources for each connection.

Before beginning

1. Download the latest Azure Resource Manager PowerShell cmdlets.

Download and install the latest version of the Azure Resource Manager PowerShell cmdlets. For more information about downloading and installing PowerShell cmdlets, see [How to install and configure Azure PowerShell](#).

2. Connect to your Azure account.

Open your PowerShell console and connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Delete a Site-to-Site VPN gateway

To delete a virtual network gateway for a S2S configuration, you must first delete each resource that pertains to the virtual network gateway. Resources must be deleted in a certain order due to dependencies. When working with the examples below, some of the values must be specified, while other values are an output result. We use the following specific values in the examples for demonstration purposes:

VNet name: VNet1

Resource Group name: RG1

Virtual network gateway name: GW1

The following steps apply to the Resource Manager deployment model.

1. Get the virtual network gateway that you want to delete.

```
$GW=Get-AzurermVirtualNetworkGateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Check to see if the virtual network gateway has any connections.

```
get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object {$_ .VirtualNetworkGateway1.Id -eq $GW.Id}  
$Conns=get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object {$_ .VirtualNetworkGateway1.Id -eq $GW.Id}
```

3. Delete all connections.

You may be prompted to confirm the deletion of each of the connections.

```
$Conns | ForEach-Object {Remove-AzureRmVirtualNetworkGatewayConnection -Name $_ .name -ResourceGroupName $_ .ResourceGroupName}
```

4. Delete the virtual network gateway.

You may be prompted to confirm the deletion of the gateway. If you have a P2S configuration to this VNet in addition to your S2S configuration, deleting the virtual network gateway will automatically disconnect all P2S clients without warning.

```
Remove-AzureRmVirtualNetworkGateway -Name "GW1" -ResourceGroupName "RG1"
```

At this point, your virtual network gateway has been deleted. You can use the next steps to delete any resources that are no longer being used.

5 Delete the local network gateways.

Get the list of the corresponding local network gateways.

```
$LNG=Get-AzureRmLocalNetworkGateway -ResourceGroupName "RG1" | where-object {$_ .Id -In  
$Conns.LocalNetworkGateway2.Id}
```

Delete the local network gateways. You may be prompted to confirm the deletion of each of the local network gateway.

```
$LNG | ForEach-Object {Remove-AzureRmLocalNetworkGateway -Name $_ .Name -ResourceGroupName  
$_ .ResourceGroupName}
```

6. Delete the Public IP address resources.

Get the IP configurations of the virtual network gateway.

```
$GWIpcfgs = $Gateway.IpConfigurations
```

Get the list of Public IP address resources used for this virtual network gateway. If the virtual network gateway was active-active, you will see two Public IP addresses.

```
$PubIP=Get-AzureRmPublicIpAddress | where-object {$_ .Id -In $GWIpcfgs.PublicIpAddress.Id}
```

Delete the Public IP resources.

```
$PubIP | foreach-object {remove-azurermpublicIpAddress -Name $_.Name -ResourceGroupName "RG1"}
```

7. Delete the gateway subnet and set the configuration.

```
$GWSUB = Get-AzureRmVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet"  
Set-AzureRmVirtualNetwork -VirtualNetwork $GWSUB
```

Delete a VNet-to-VNet VPN gateway

To delete a virtual network gateway for a V2V configuration, you must first delete each resource that pertains to the virtual network gateway. Resources must be deleted in a certain order due to dependencies. When working with the examples below, some of the values must be specified, while other values are an output result. We use the following specific values in the examples for demonstration purposes:

VNet name: VNet1

Resource Group name: RG1

Virtual network gateway name: GW1

The following steps apply to the Resource Manager deployment model.

1. Get the virtual network gateway that you want to delete.

```
$GW=get-azurermvirtualnetworkgateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Check to see if the virtual network gateway has any connections.

```
get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object  
{$_._VirtualNetworkGateway1.Id -eq $GW.Id}
```

There may be other connections to the virtual network gateway that are part of a different resource group. Check for additional connections in each additional resource group. In this example, we are checking for connections from RG2. Run this for each resource group that you have which may have a connection to the virtual network gateway.

```
get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "RG2" | where-object  
{$_._VirtualNetworkGateway2.Id -eq $GW.Id}
```

3. Get the list of connections in both directions.

Because this is a VNet-to-VNet configuration, you need the list of connections in both directions.

```
$ConnsL=get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object  
{$_._VirtualNetworkGateway1.Id -eq $GW.Id}
```

In this example, we are checking for connections from RG2. Run this for each resource group that you have which may have a connection to the virtual network gateway.

```
$ConnsR=get-azurermvirtualnetworkgatewayconnection -ResourceGroupName "<NameOfResourceGroup2>" | where-object  
{$_._VirtualNetworkGateway2.Id -eq $GW.Id}
```

4. Delete all connections.

You may be prompted to confirm the deletion of each of the connections.

```
$ConnsL | ForEach-Object {Remove-AzureRmVirtualNetworkGatewayConnection -Name $_.name -ResourceGroupName  
$_.ResourceGroupName}  
$ConnsR | ForEach-Object {Remove-AzureRmVirtualNetworkGatewayConnection -Name $_.name -ResourceGroupName  
$_.ResourceGroupName}
```

5. Delete the virtual network gateway.

You may be prompted to confirm the deletion of the virtual network gateway. If you have P2S configurations to your VNets in addition to your V2V configuration, deleting the virtual network gateways will automatically disconnect all P2S clients without warning.

```
Remove-AzureRmVirtualNetworkGateway -Name "GW1" -ResourceGroupName "RG1"
```

At this point, your virtual network gateway has been deleted. You can use the next steps to delete any resources that are no longer being used.

6. Delete the Public IP address resources

Get the IP configurations of the virtual network gateway.

```
$GWIpcfgs = $Gateway.IpConfigurations
```

Get the list of Public IP address resources used for this virtual network gateway. If the virtual network gateway was active-active, you will see two Public IP addresses.

```
$PubIP=Get-AzureRmPublicIpAddress | where-object {$_.Id -In $GWIpcfgs.PublicIpAddress.Id}
```

Delete the Public IP resources. You may be prompted to confirm the deletion of the Public IP.

```
$PubIP | foreach-object {remove-azurermpublicipaddress -Name $_.Name -ResourceGroupName "  
<NameOfResourceGroup1>"}
```

7. Delete the gateway subnet and set the configuration.

```
$GWSub = Get-AzureRmVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-  
AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet"  
Set-AzureRmVirtualNetwork -VirtualNetwork $GWSub
```

Delete a Point-to-Site VPN gateway

To delete a virtual network gateway for a P2S configuration, you must first delete each resource that pertains to the virtual network gateway. Resources must be deleted in a certain order due to dependencies. When working with the examples below, some of the values must be specified, while other values are an output result. We use the following specific values in the examples for demonstration purposes:

VNet name: VNet1

Resource Group name: RG1

Virtual network gateway name: GW1

The following steps apply to the Resource Manager deployment model.

NOTE

When you delete the VPN gateway, all connected clients will be disconnected from the VNet without warning.

1. Get the virtual network gateway that you want to delete.

```
$GW=get-azurermvirtualnetworkgateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Delete the virtual network gateway.

You may be prompted to confirm the deletion of the virtual network gateway.

```
Remove-AzureRmVirtualNetworkGateway -Name "GW1" -ResourceGroupName "RG1"
```

At this point, your virtual network gateway has been deleted. You can use the next steps to delete any resources that are no longer being used.

3. Delete the Public IP address resources

Get the IP configurations of the virtual network gateway.

```
$GWIpcfgs = $Gateway.IpConfigurations
```

Get the list of Public IP addresses used for this virtual network gateway. If the virtual network gateway was active-active, you will see two Public IP addresses.

```
$PubIP=Get-AzureRmPublicIpAddress | where-object {$_.Id -In $GWIpcfgs.PublicIpAddress.Id}
```

Delete the Public IPs. You may be prompted to confirm the deletion of the Public IP.

```
$PubIP | foreach-object {remove-azurermpublicIpAddress -Name $_.Name -ResourceGroupName "<NameOfResourceGroup1>"}
```

4. Delete the gateway subnet and set the configuration.

```
$GWSUB = Get-AzureRmVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet"  
Set-AzureRmVirtualNetwork -VirtualNetwork $GWSUB
```

Delete a VPN gateway by deleting the resource group

If you are not concerned about keeping any of your resources in the resource group and you just want to start over, you can delete an entire resource group. This is a quick way to remove everything. The following steps apply only to the Resource Manager deployment model.

1. Get a list of all the resource groups in your subscription.

```
Get-AzureRmResourceGroup
```

2. Locate the resource group that you want to delete.

Locate the resource group that you want to delete and view the list of resources in that resource group. In the example, the name of the resource group is RG1. Modify the example to retrieve a list of all the resources.

```
Find-AzureRmResource -ResourceGroupNameContains RG1
```

3. Verify the resources in the list.

When the list is returned, review it to verify that you want to delete all the resources in the resource group, as well as the resource group itself. If you want to keep some of the resources in the resource group, use the steps in the earlier sections of this article to delete your gateway.

4. Delete the resource group and resources.

To delete the resource group and all the resource contained in the resource group, modify the example and run.

```
Remove-AzureRmResourceGroup -Name RG1
```

5. Check the status.

It takes some time for Azure to delete all the resources. You can check the status of your resource group by using this cmdlet.

```
Get-AzureRmResourceGroup -ResourceGroupName RG1
```

The result that is returned shows 'Succeeded'.

```
ResourceGroupName : RG1
Location        : eastus
ProvisioningState : Succeeded
```

Working with virtual network gateway SKUs (legacy SKUs)

3/21/2018 • 4 minutes to read • [Edit Online](#)

This article contains information about the legacy (old) virtual network gateway SKUs. The legacy SKUs still work in both deployment models for VPN gateways that have already been created. Classic VPN gateways continue to use the legacy SKUs, both for existing gateways, and for new gateways. When creating new Resource Manager VPN gateways, use the new gateway SKUs. For information about the new SKUs, see [About VPN Gateway](#).

Gateway SKUs

The legacy (old) VPN gateway SKUs are:

- Basic
- Standard
- HighPerformance

VPN Gateway does not use the UltraPerformance gateway SKU. For information about the UltraPerformance SKU, see the [ExpressRoute](#) documentation.

When working with the legacy SKUs, consider the following:

- If you want to use a PolicyBased VPN type, you must use the Basic SKU. PolicyBased VPNs (previously called Static Routing) are not supported on any other SKU.
- BGP is not supported on the Basic SKU.
- ExpressRoute-VPN Gateway coexist configurations are not supported on the Basic SKU.
- Active-active S2S VPN Gateway connections can be configured on the HighPerformance SKU only.

Estimated aggregate throughput by SKU

The following table shows the gateway types and the estimated aggregate throughput by gateway SKU. This table applies to the Resource Manager and classic deployment models.

Pricing differs between gateway SKUs. For more information, see [VPN Gateway Pricing](#).

Note that the UltraPerformance gateway SKU is not represented in this table. For information about the UltraPerformance SKU, see the [ExpressRoute](#) documentation.

	VPN GATEWAY THROUGHPUT (1)	VPN GATEWAY MAX IPSEC TUNNELS (2)	EXPRESSROUTE GATEWAY THROUGHPUT	VPN GATEWAY AND EXPRESSROUTE COEXIST
Basic SKU (3)(5)(6)	100 Mbps	10	500 Mbps (6)	No
Standard SKU (4)(5)	100 Mbps	10	1000 Mbps	Yes
High Performance SKU (4)	200 Mbps	30	2000 Mbps	Yes

(1) The VPN throughput is a rough estimate based on the measurements between VNets in the same Azure region. It is not a guaranteed throughput for cross-premises connections across the Internet. It is the maximum

possible throughput measurement.

(2) The number of tunnels refer to RouteBased VPNs. A PolicyBased VPN can only support one Site-to-Site VPN tunnel.

(3) BGP is not supported for the Basic SKU.

(4) PolicyBased VPNs are not supported for this SKU. They are supported for the Basic SKU only.

(5) Active-active S2S VPN Gateway connections are not supported for this SKU. Active-active is supported on the HighPerformance SKU only.

(6) Basic SKU is deprecated for use with ExpressRoute.

Supported configurations by SKU and VPN type

The following table lists the requirements for PolicyBased and RouteBased VPN gateways. This table applies to both the Resource Manager and classic deployment models. For the classic model, PolicyBased VPN gateways are the same as Static gateways, and Route-based gateways are the same as Dynamic gateways.

	POLICYBASED BASIC VPN GATEWAY	ROUTEBASED BASIC VPN GATEWAY	ROUTEBASED STANDARD VPN GATEWAY	ROUTEBASED HIGH PERFORMANCE VPN GATEWAY
Site-to-Site connectivity (S2S)	PolicyBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration
Point-to-Site connectivity (P2S)	Not supported	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)
Authentication method	Pre-shared key	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity
Maximum number of S2S connections	1	10	10	30
Maximum number of P2S connections	Not supported	128	128	128
Active routing support (BGP)	Not supported	Not supported	Supported	Supported

Resize a gateway

You can resize your gateway to a gateway SKU within the same SKU family. For example, if you have a Standard SKU, you can resize to a HighPerformance SKU. However, you can't resize your VPN gateway between the old SKUs and the new SKU families. For example, you can't go from a Standard SKU to a VpnGw2 SKU, or a Basic SKU to VpnGw1.

To resize a gateway for the classic deployment model, use the following command:

```
Resize-AzureVirtualNetworkGateway -GatewayId <Gateway ID> -GatewaySKU HighPerformance
```

To resize a gateway for the Resource Manager deployment model using PowerShell, use the following command:

```
$gw = Get-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg  
Resize-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -GatewaySku HighPerformance
```

You can also resize a gateway in the Azure portal.

Change to the new gateway SKUs

If you are working with the Resource Manager deployment model, you can change to the new gateway SKUs. When you change from a legacy gateway SKU to a new SKU, you delete the existing VPN gateway and create a new VPN gateway.

Workflow:

1. Remove any connections to the virtual network gateway.
2. Delete the old VPN gateway.
3. Create the new VPN gateway.
4. Update your on-premises VPN devices with the new VPN gateway IP address (for Site-to-Site connections).
5. Update the gateway IP address value for any VNet-to-VNet local network gateways that will connect to this gateway.
6. Download new client VPN configuration packages for P2S clients connecting to the virtual network through this VPN gateway.
7. Recreate the connections to the virtual network gateway.

Considerations:

- To move to the new SKUs, your VPN gateway must be in the Resource Manager deployment model.
- If you have a classic VPN gateway, you must continue using the older legacy SKUs for that gateway, however, you can resize between the legacy SKUs. You cannot change to the new SKUs.
- You will have connectivity downtime when you change from a legacy SKU to a new SKU.
- When changing to a new gateway SKU, the public IP address for your VPN gateway will change. This happens even if you specify the same public IP address object that you used previously.

Next steps

For more information about the new Gateway SKUs, see [Gateway SKUs](#).

For more information about configuration settings, see [About VPN Gateway configuration settings](#).

Overview of partner VPN device configurations

4/18/2018 • 3 minutes to read • [Edit Online](#)

This article provides an overview of configuring on-premises VPN devices for connecting to Azure VPN gateways. A sample Azure virtual network and VPN gateway setup is used to show you how to connect to different on-premises VPN device configurations by using the same parameters.

Device requirements

Azure VPN gateways use standard IPsec/IKE protocol suites for site-to-site (S2S) VPN tunnels. For a list of IPsec/IKE parameters and cryptographic algorithms for Azure VPN gateways, see [About VPN devices](#). You can also specify the exact algorithms and key strengths for a specific connection as described in [About cryptographic requirements](#).

Single VPN tunnel

The first configuration in the sample consists of a single S2S VPN tunnel between an Azure VPN gateway and an on-premises VPN device. You can optionally configure the [Border Gateway Protocol \(BGP\) across the VPN tunnel](#).



For step-by-step instructions to set up a single VPN tunnel, see [Configure a site-to-site connection](#). The following sections specify the connection parameters for the sample configuration and provide a PowerShell script to help you get started.

Connection parameters

This section lists the parameters for the examples that are described in the previous sections.

PARAMETER	VALUE
Virtual network address prefixes	10.11.0.0/16 10.12.0.0/16
Azure VPN gateway IP	Azure VPN Gateway IP
On-premises address prefixes	10.51.0.0/16 10.52.0.0/16
On-premises VPN device IP	On-premises VPN device IP
* Virtual network BGP ASN	65010
* Azure BGP peer IP	10.12.255.30

PARAMETER	VALUE
* On-premises BGP ASN	65050
* On-premises BGP peer IP	10.52.255.254

* Optional parameter for BGP only.

Sample PowerShell script

This section provides a sample script to get you started. For detailed instructions, see [Create an S2S VPN connection by using PowerShell](#).

```
# Declare your variables

$Sub1      = "Replace_With_Your_Subscription_Name"
$RG1       = "TestRG1"
$Location1 = "East US 2"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$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"
$VNet1ASN   = 65010
$DNS1       = "8.8.8.8"
$GWName1    = "VNet1GW"
$GWIPName1  = "VNet1GWIP"
$GWIPconfName1 = "gwipconf1"
$Connection15 = "VNet1toSite5"
$LNGName5   = "Site5"
$LNGPrefix50 = "10.52.255.254/32"
$LNGPrefix51 = "10.51.0.0/16"
$LNGPrefix52 = "10.52.0.0/16"
$LNGIPS     = "Your_VPN_Device_IP"
$LNGASNS   = 65050
$BGPPeerIP5 = "10.52.255.254"

# Connect to your subscription and create a new resource group

Connect-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $Sub1
New-AzureRmResourceGroup -Name $RG1 -Location $Location1

# Create virtual network

$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1

# Create VPN gateway

$gwip1    = New-AzureRmPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$vnet1    = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1  = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 -Subnet $subnet1 -PublicIpAddress
$gwip1

New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -InConfigurations
```

```
$gwpconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -Asn $VNet1ASN

# Create local network gateway

New-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1 -Location $Location1 -GatewayIpAddress $LNGIP5 -AddressPrefix $LNGPrefix51,$LNGPrefix52 -Asn $LNGASNS -BgpPeeringAddress $BGPPeerIP5

# Create the S2S VPN connection

$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -EnableBGP $False
```

(Optional) Use custom IPsec/IKE policy with **UsePolicyBasedTrafficSelectors**

If your VPN devices don't support any-to-any traffic selectors, such as route-based or VTI-based configurations, create a custom IPsec/IKE policy with the [UsePolicyBasedTrafficSelectors](#) option.

IMPORTANT

You must create an IPsec/IKE policy to enable the **UsePolicyBasedTrafficSelectors** option on the connection.

The sample script creates an IPsec/IKE policy with the following algorithms and parameters:

- IKEv2: AES256, SHA384, DHGroup24
- IPsec: AES256, SHA1, PFS24, SA Lifetime 7,200 seconds, and 20,480,000 KB (20 GB)

The script applies the IPsec/IKE policy and enables the **UsePolicyBasedTrafficSelectors** option on the connection.

```
$ipsecpolicy5 = New-AzureRmIpsecPolicy -IkeEncryption AES256 -IkeIntegrity SHA384 -DhGroup DHGroup24 -IpsecEncryption AES256 -IpsecIntegrity SHA1 -PfsGroup PFS24 -SALifeTimeSeconds 7200 -SADataSizeKilobytes 20480000

$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -EnableBGP $False -IpsecPolicies $ipsecpolicy5 -UsePolicyBasedTrafficSelectors $True
```

(Optional) Use BGP on S2S VPN connection

When you create the S2S VPN connection, you can optionally use [BGP for the VPN gateway](#). This approach has two differences:

- The on-premises address prefixes can be a single host address. The on-premises BGP peer IP address is specified as follows:

```
New-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1 -Location $Location1 -GatewayIpAddress $LNGIP5 -AddressPrefix $LNGPrefix50 -Asn $LNGASNS -BgpPeeringAddress $BGPPeerIP5
```

- When you create the connection, you must set the **-EnableBGP** option to **\$True**:

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw -Location $Location1 -ConnectionType  
IPsec -SharedKey 'AzureA1b2C3' -EnableBGP $True
```

Next steps

For step-by-step instructions to set up active-active VPN gateways, see [Configuring active-active VPN gateways for cross-premises and VNet-to-VNet connections](#).

Sample configuration: Cisco ASA device (IKEv2/no BGP)

1/18/2018 • 7 minutes to read • [Edit Online](#)

This article provides sample configurations for connecting Cisco Adaptive Security Appliance (ASA) devices to Azure VPN gateways. The example applies to Cisco ASA devices that are running IKEv2 without the Border Gateway Protocol (BGP).

Device at a glance

Device vendor	Cisco
Device model	ASA
Target version	8.4 and later
Tested model	ASA 5505
Tested version	9.2
IKE version	IKEv2
BGP	No
Azure VPN gateway type	Route-based VPN gateway

NOTE

The sample configuration connects a Cisco ASA device to an Azure **route-based** VPN gateway. The connection uses a custom IPsec/IKE policy with the **UsePolicyBasedTrafficSelectors** option, as described in [this article](#).

The sample requires that ASA devices use the **IKEv2** policy with access-list-based configurations, not VTI-based. Consult your VPN device vendor specifications to verify that the IKEv2 policy is supported on your on-premises VPN devices.

VPN device requirements

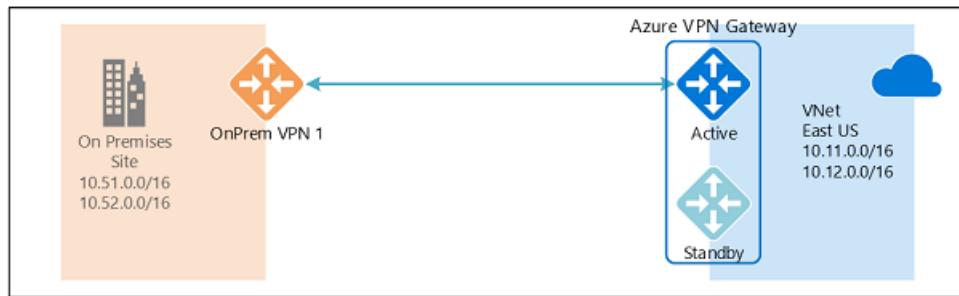
Azure VPN gateways use the standard IPsec/IKE protocol suites to establish Site-to-Site (S2S) VPN tunnels. For the detailed IPsec/IKE protocol parameters and default cryptographic algorithms for Azure VPN gateways, see [About VPN devices](#).

NOTE

You can optionally specify an exact combination of cryptographic algorithms and key strengths for a specific connection, as described in [About cryptographic requirements](#). If you specify an exact combination of algorithms and key strengths, be sure to use the corresponding specifications on your VPN devices.

Single VPN tunnel

This configuration consists of a single S2S VPN tunnel between an Azure VPN gateway and an on-premises VPN device. You can optionally configure the [BGP across the VPN tunnel](#).



For step-by-step instructions to build the Azure configurations, see [Single VPN tunnel setup](#).

Virtual network and VPN gateway information

This section lists the parameters for the sample.

PARAMETER	VALUE
Virtual network address prefixes	10.11.0.0/16 10.12.0.0/16
Azure VPN gateway IP	Azure_Gateway_Public_IP
On-premises address prefixes	10.51.0.0/16 10.52.0.0/16
On-premises VPN device IP	OnPrem_Device_Public_IP
* Virtual network BGP ASN	65010
* Azure BGP peer IP	10.12.255.30
* On-premises BGP ASN	65050
* On-premises BGP peer IP	10.52.255.254

* Optional parameter for BGP only.

IPsec/IKE policy and parameters

The following table lists the IPsec/IKE algorithms and parameters that are used in the sample. Consult your VPN device specifications to verify the algorithms that are supported for your VPN device models and firmware versions.

IPSEC/IKEV2	VALUE
IKEv2 Encryption	AES256
IKEv2 Integrity	SHA384
DH Group	DHGroup24

IPSEC/IKEV2	VALUE
* IPsec Encryption	AES256
* IPsec Integrity	SHA1
PFS Group	PFS24
QM SA Lifetime	7,200 seconds
Traffic Selector	UsePolicyBasedTrafficSelectors \$True
Pre-Shared Key	PreSharedKey

* On some devices, IPsec Integrity must be a null value when the IPsec Encryption algorithm is AES-GCM.

ASA device support

- Support for IKEv2 requires ASA version 8.4 and later.
- Support for DH Group and PFS Group beyond Group 5 requires ASA version 9.x.
- Support for IPsec Encryption with AES-GCM and IPsec Integrity with SHA-256, SHA-384, or SHA-512, requires ASA version 9.x. This support requirement applies to newer ASA devices. At the time of publication, ASA models 5505, 5510, 5520, 5540, 5550, and 5580 do not support these algorithms. Consult your VPN device specifications to verify the algorithms that are supported for your VPN device models and firmware versions.

Sample device configuration

The script provides a sample that is based on the configuration and parameters that are described in the previous sections. The S2S VPN tunnel configuration consists of the following parts:

- Interfaces and routes
- Access lists
- IKE policy and parameters (phase 1 or main mode)
- IPsec policy and parameters (phase 2 or quick mode)
- Other parameters, such as TCP MSS clamping

IMPORTANT

Complete the following steps before you use the sample script. Replace the placeholder values in the script with the device settings for your configuration.

- Specify the interface configuration for both inside and outside interfaces.
- Identify the routes for your inside/private and outside/public networks.
- Ensure all names and policy numbers are unique on your device.
- Ensure that the cryptographic algorithms are supported on your device.
- Replace the following **placeholder values** with actual values for your configuration:
 - Outside interface name: **outside**
 - Azure_Gateway_Public_IP**
 - OnPrem_Device_Public_IP**
 - IKE: **Pre_Shared_Key**

- o Virtual network and local network gateway names: **VNetName** and **LNGName**
- o Virtual network and on-premises network address **prefixes**
- o Proper **netmasks**

Sample script

```

! Sample ASA configuration for connecting to Azure VPN gateway
!
! Tested hardware: ASA 5505
! Tested version: ASA version 9.2(4)
!
! Replace the following place holders with your actual values:
! - Interface names - default are "outside" and "inside"
! - <Azure_Gateway_Public_IP>
! - <OnPrem_Device_Public_IP>
! - <Pre_Shared_Key>
! - <VNetName>*
! - <LNGName>* ==> LocalNetworkGateway - the Azure resource that represents the
!   on-premises network, specifies network prefixes, device public IP, BGP info, etc.
! - <PrivateIPAddress> ==> Replace it with a private IP address if applicable
! - <Netmask> ==> Replace it with appropriate netmasks
! - <Nexthop> ==> Replace it with the actual nexthop IP address
!
! (*) Must be unique names in the device configuration
!
! ==> Interface & route configurations
!
!     > <OnPrem_Device_Public_IP> address on the outside interface or vlan
!     > <PrivateIPAddress> on the inside interface or vlan; e.g., 10.51.0.1/24
!     > Route to connect to <Azure_Gateway_Public_IP> address
!
!     > Example:
!
!         interface Ethernet0/0
!             switchport access vlan 2
!             exit
!
!         interface vlan 1
!             nameif inside
!             security-level 100
!             ip address <PrivateIPAddress> <Netmask>
!             exit
!
!         interface vlan 2
!             nameif outside
!             security-level 0
!             ip address <OnPrem_Device_Public_IP> <Netmask>
!             exit
!
!         route outside 0.0.0.0 0.0.0.0 <NextHop IP> 1
!
! ==> Access lists
!
!     > Most firewall devices deny all traffic by default. Create access lists to
!       (1) Allow S2S VPN tunnels between the ASA and the Azure gateway public IP address
!       (2) Construct traffic selectors as part of IPsec policy or proposal
!
access-list outside_access_in extended permit ip host <Azure_Gateway_Public_IP> host <OnPrem_Device_Public_IP>
!
!     > Object group that consists of all VNet prefixes (e.g., 10.11.0.0/16 &
!       10.12.0.0/16)
!
object-group network Azure-<VNetName>
description Azure virtual network <VNetName> prefixes
network-object 10.11.0.0 255.255.0.0
network-object 10.12.0.0 255.255.0.0
exit

```

```

!
!     > Object group that corresponding to the <LNGName> prefixes.
!     E.g., 10.51.0.0/16 and 10.52.0.0/16. Note that LNG = "local network gateway".
!     In Azure network resource, a local network gateway defines the on-premises
!     network properties (address prefixes, VPN device IP, BGP ASN, etc.)
!
object-group network <LNGName>
description On-Premises network <LNGName> prefixes
network-object 10.51.0.0 255.255.0.0
network-object 10.52.0.0 255.255.0.0
exit
!
!     > Specify the access-list between the Azure VNet and your on-premises network.
!     This access list defines the IPsec SA traffic selectors.
!
access-list Azure-<VNetName>-acl extended permit ip object-group <LNGName> object-group Azure-<VNetName>
!
!     > No NAT required between the on-premises network and Azure VNet
!
nat (inside,outside) source static <LNGName> <LNGName> destination static Azure-<VNetName> Azure-<VNetName>
!
! ==> IKEv2 configuration
!
!     > General IKEv2 configuration - enable IKEv2 for VPN
!
group-policy DfltGrpPolicy attributes
    vpn-tunnel-protocol ikev1 ikev2
exit
!
crypto isakmp identity address
crypto ikev2 enable outside
!
!     > Define IKEv2 Phase 1/Main Mode policy
!     - Make sure the policy number is not used
!     - integrity and prf must be the same
!     - DH group 14 and above require ASA version 9.x.
!
crypto ikev2 policy 1
    encryption      aes-256
    integrity      sha384
    prf            sha384
    group          24
    lifetime       seconds 86400
exit
!
!     > Set connection type and pre-shared key
!
tunnel-group <Azure_Gateway_Public_IP> type ipsec-l2l
tunnel-group <Azure_Gateway_Public_IP> ipsec-attributes
    ikev2 remote-authentication pre-shared-key <Pre_Shared_Key>
    ikev2 local-authentication  pre-shared-key <Pre_Shared_Key>
exit
!
! ==> IPsec configuration
!
!     > IKEv2 Phase 2/Quick Mode proposal
!     - AES-GCM and SHA-2 requires ASA version 9.x on newer ASA models. ASA
!       5505, 5510, 5520, 5540, 5550, 5580 are not supported.
!     - ESP integrity must be null if AES-GCM is configured as ESP encryption
!
crypto ipsec ikev2 ipsec-proposal AES-256
    protocol esp encryption aes-256
    protocol esp integrity sha-1
exit
!
!     > Set access list & traffic selectors, PFS, IPsec protposal, SA lifetime
!     - This sample uses "Azure-<VNetName>-map" as the crypto map name
!     - ASA supports only one crypto map per interface, if you already have
!       an existing crypto map assigned to your outside interface, you must use

```

```
!           the same crypto map name, but with a different sequence number for
!           this policy
! - "match address" policy uses the access-list "Azure-<VNetName>-acl" defined
!           previously
! - "ipsec-proposal" uses the proposal "AES-256" defined previously
! - PFS groups 14 and beyond requires ASA version 9.x.
!
crypto map Azure-<VNetName>-map 1 match address Azure-<VNetName>-acl
crypto map Azure-<VNetName>-map 1 set pfs group24
crypto map Azure-<VNetName>-map 1 set peer <Azure_Gateway_Public_IP>
crypto map Azure-<VNetName>-map 1 set ikev2 ipsec-proposal AES-256
crypto map Azure-<VNetName>-map 1 set security-association lifetime seconds 7200
crypto map Azure-<VNetName>-map interface outside
!
! ==> Set TCP MSS to 1350
!
sysopt connection tcpmss 1350
!
```

Simple debugging commands

Use the following ASA commands for debugging purposes:

- Show the IPsec or IKE security association (SA):

```
show crypto ipsec sa
show crypto ikev2 sa
```

- Enter debug mode:

```
debug crypto ikev2 platform <level>
debug crypto ikev2 protocol <level>
```

The `debug` commands can generate significant output on the console.

- Show the current configurations on the device:

```
show run
```

Use `show` subcommands to list specific parts of the device configuration, for example:

```
show run crypto
show run access-list
show run tunnel-group
```

Next steps

To configure active-active cross-premises and VNet-to-VNet connections, see [Configure active-active VPN gateways](#).

Troubleshoot VPN Gateway

1/31/2018 • 2 minutes to read • [Edit Online](#)

VPN Gateway connections can fail for a variety of reasons. This article contains links to get you started with troubleshooting. For a full list, see the articles contained in the table of contents under **Troubleshoot**, to the left of this page.

Troubleshooting scenarios and solutions

- [Validate VPN throughput to a VNet](#)

A VPN gateway connection enables you to establish secure, cross-premises connectivity between your Virtual Network within Azure and your on-premises IT infrastructure. This article shows how to validate network throughput from the on-premises resources to an Azure virtual machine (VM). It also provides troubleshooting guidance.

- [VPN and Firewall device settings](#)

This article provides several suggested solutions for third-party VPN or firewall devices that are used with VPN Gateway. Technical support for third-party VPN or firewall devices is provided by the device vendor.

- [Point-to-Site connections](#)

This article lists common point-to-site connection problems that you might experience. It also discusses possible causes and solutions for these problems.

- [Site-to-Site connections](#)

After you configure a site-to-site VPN connection between an on-premises network and an Azure virtual network, the VPN connection suddenly stops working and cannot be reconnected. This article provides troubleshooting steps to help you resolve this problem.

Next steps

You can also use these steps to [Validate VNet and VPN connections](#).

Community-suggested third-party VPN or firewall device settings for Azure VPN gateway

6/15/2018 • 2 minutes to read • [Edit Online](#)

This article provides several suggested solutions for third-party VPN or firewall devices that are used with Azure VPN gateway.

NOTE

Technical support for third-party VPN or firewall devices is provided by the device vendor.

More information

The following table lists several common devices and related help:

PRODUCT	REFERENCE
Cisco ASA	Community suggested solutions for Cisco ASA on Azure VPN
Cisco ISR	Community suggested solutions for Cisco ISR on Azure VPN
Cisco ASR	Community suggested solutions for Cisco ASR on Azure VPN
Sonicwall	Search for Azure VPN on Sonicwall site
Checkpoint	Search for Azure VPN on Checkpoint site
Juniper	Search for Azure VPN on Juniper site
Barracuda	Community suggested solutions for Barracuda on Azure VPN
F5	Community suggested solutions for F5 on Azure VPN
Palo	Community suggested solutions for Palo on Azure VPN
Watchguard	Community suggested solutions for Watchguard on Azure VPN

Next step

[Azure Gateways settings](#)

[Known compatible devices](#)

How to validate VPN throughput to a virtual network

7/13/2018 • 4 minutes to read • [Edit Online](#)

A VPN gateway connection enables you to establish secure, cross-premises connectivity between your Virtual Network within Azure and your on-premises IT infrastructure.

This article shows how to validate network throughput from the on-premises resources to an Azure virtual machine (VM). It also provides troubleshooting guidance.

NOTE

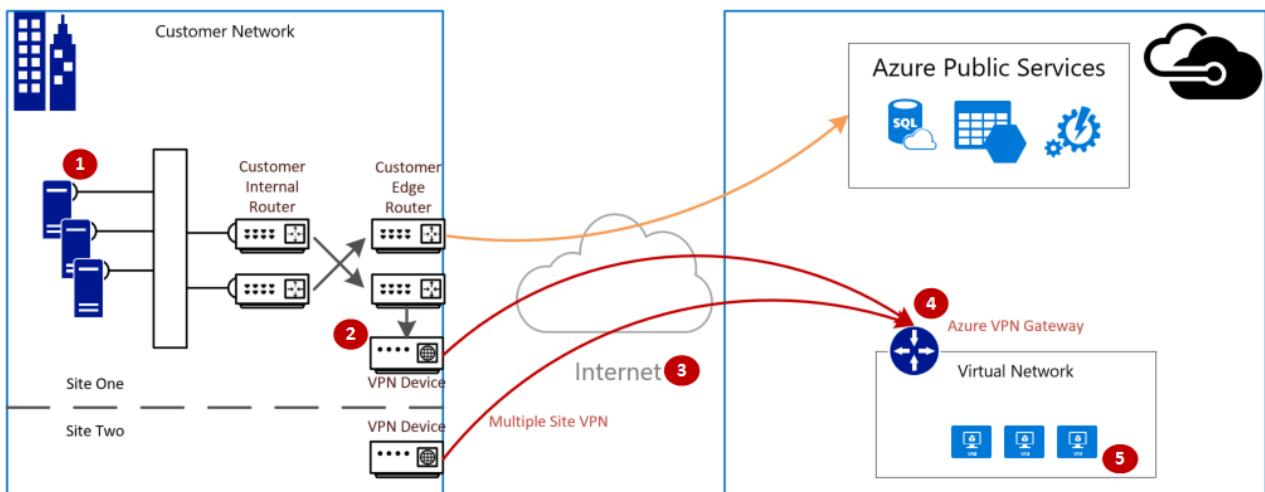
This article is intended to help diagnose and fix common issues. If you're unable to solve the issue by using the following information, [contact support](#).

Overview

The VPN gateway connection involves the following components:

- On-premises VPN device (view a list of [validated VPN devices](#)).
- Public Internet
- Azure VPN gateway
- Azure VM

The following diagram shows the logical connectivity of an on-premises network to an Azure virtual network through VPN.



Calculate the maximum expected ingress/egress

1. Determine your application's baseline throughput requirements.
2. Determine your Azure VPN gateway throughput limits. For help, see the "Aggregate throughput by SKU and VPN type" section of [Planning and design for VPN Gateway](#).
3. Determine the [Azure VM throughput guidance](#) for your VM size.
4. Determine your Internet Service Provider (ISP) bandwidth.
5. Calculate your expected throughput - Least bandwidth of (VM, Gateway, ISP) * 0.8.

If your calculated throughput does not meet your application's baseline throughput requirements, you need to

increase the bandwidth of the resource that you identified as the bottleneck. To resize an Azure VPN Gateway, see [Changing a gateway SKU](#). To resize a virtual machine, see [Resize a VM](#). If you are not experiencing expected Internet bandwidth, you may also want to contact your ISP.

Validate network throughput by using performance tools

This validation should be performed during non-peak hours, as VPN tunnel throughput saturation during testing does not give accurate results.

The tool we use for this test is iPerf, which works on both Windows and Linux and has both client and server modes. It is limited to 3 Gbps for Windows VMs.

This tool does not perform any read/write operations to disk. It solely produces self-generated TCP traffic from one end to the other. It generates statistics based on experimentation that measures the bandwidth available between client and server nodes. When testing between two nodes, one acts as the server and the other as a client. Once this test is completed, we recommend that you reverse the roles to test both upload and download throughput on both nodes.

Download iPerf

Download [iPerf](#). For details, see [iPerf documentation](#).

NOTE

The third-party products that this article discusses are manufactured by companies that are independent of Microsoft. Microsoft makes no warranty, implied or otherwise, about the performance or reliability of these products.

Run iPerf (iperf3.exe)

1. Enable an NSG/ACL rule allowing the traffic (for public IP address testing on Azure VM).
2. On both nodes, enable a firewall exception for port 5001.

Windows: Run the following command as an administrator:

```
netsh advfirewall firewall add rule name="Open Port 5001" dir=in action=allow protocol=TCP  
localport=5001
```

To remove the rule when testing is complete, run this command:

```
netsh advfirewall firewall delete rule name="Open Port 5001" protocol=TCP localport=5001
```

Azure Linux: Azure Linux images have permissive firewalls. If there is an application listening on a port, the traffic is allowed through. Custom images that are secured may need ports opened explicitly. Common Linux OS-layer firewalls include `iptables`, `ufw`, or `firewalld`.

3. On the server node, change to the directory where iperf3.exe is extracted. Then run iPerf in server mode and set it to listen on port 5001 as the following commands:

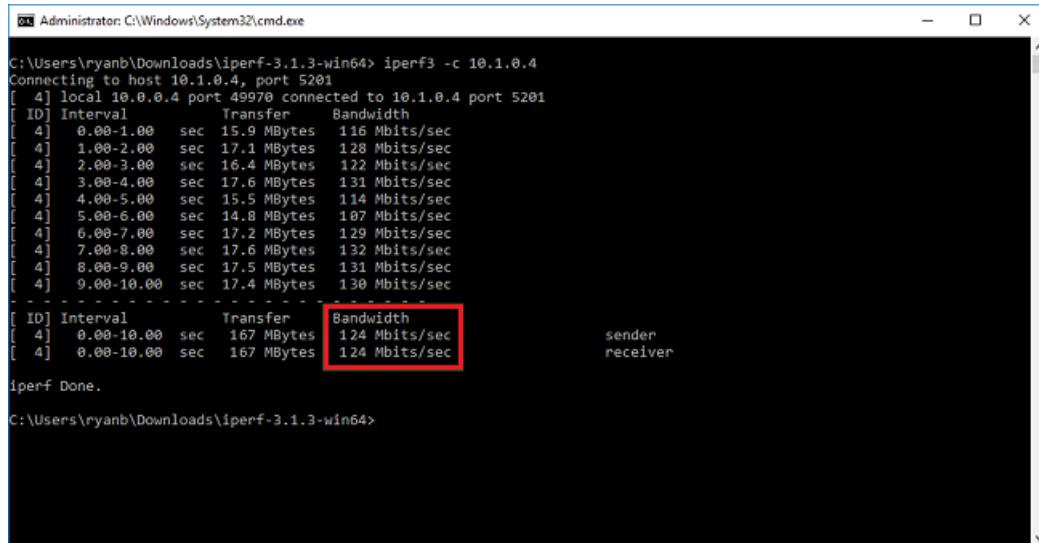
```
cd c:\iperf-3.1.2-win65  
iperf3.exe -s -p 5001
```

4. On the client node, change to the directory where iperf tool is extracted and then run the following command:

```
iperf3.exe -c <IP of the iperf Server> -t 30 -p 5001 -P 32
```

The client is inducing traffic on port 5001 to the server for 30 seconds. The flag '-P' that indicates we are using 32 simultaneous connections to the server node.

The following screen shows the output from this example:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Users\ryanb\Downloads\iperf-3.1.3-win64> iperf3 -c 10.1.0.4
Connecting to host 10.1.0.4, port 5201
[ 4] local 10.0.0.4 port 49970 connected to 10.1.0.4 port 5201
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-1.00 sec 15.9 MBytes 116 Mbits/sec
[ 4] 1.00-2.00 sec 17.1 MBytes 128 Mbits/sec
[ 4] 2.00-3.00 sec 16.4 MBytes 122 Mbits/sec
[ 4] 3.00-4.00 sec 17.6 MBytes 131 Mbits/sec
[ 4] 4.00-5.00 sec 15.5 MBytes 114 Mbits/sec
[ 4] 5.00-6.00 sec 14.8 MBytes 107 Mbits/sec
[ 4] 6.00-7.00 sec 17.2 MBytes 129 Mbits/sec
[ 4] 7.00-8.00 sec 17.6 MBytes 132 Mbits/sec
[ 4] 8.00-9.00 sec 17.5 MBytes 131 Mbits/sec
[ 4] 9.00-10.00 sec 17.4 MBytes 130 Mbits/sec
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-10.00 sec 167 MBytes 124 Mbits/sec
[ 4] 0.00-10.00 sec 167 MBytes 124 Mbits/sec
sender receiver
iperf Done.
C:\Users\ryanb\Downloads\iperf-3.1.3-win64>
```

5. (OPTIONAL) To preserve the testing results, run this command:

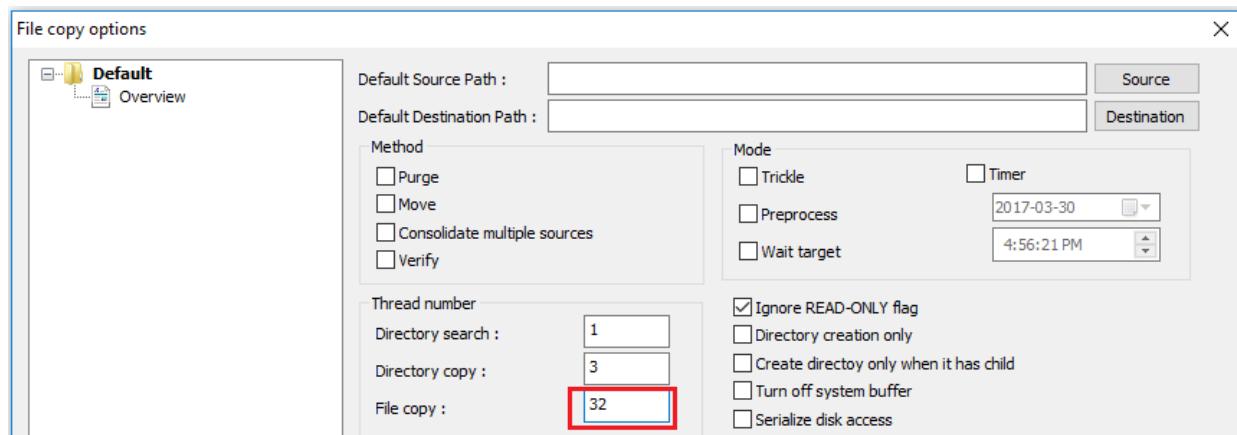
```
iperf3.exe -c IPofTheServerToReach -t 30 -p 5001 -P 32 >> output.txt
```

6. After completing the previous steps, execute the same steps with the roles reversed, so that the server node will now be the client and vice-versa.

Address slow file copy issues

You may experience slow file coping when using Windows Explorer or dragging and dropping through an RDP session. This problem is normally due to one or both of the following factors:

- File copy applications, such as Windows Explorer and RDP, do not use multiple threads when copying files. For better performance, use a multi-threaded file copy application such as [Richcopy](#) to copy files by using 16 or 32 threads. To change the thread number for file copy in Richcopy, click **Action** > **Copy options** > **File copy**.



- Insufficient VM disk read/write speed. For more information, see [Azure Storage Troubleshooting](#).

On-premises device external facing interface

If the on-premises VPN device Internet-facing IP address is included in the [local network](#) definition in Azure, you may experience inability to bring up the VPN, sporadic disconnects, or performance issues.

Checking latency

Use tracert to trace to Microsoft Azure Edge device to determine if there are any delays exceeding 100 ms between hops.

From the on-premises network, run *tracert* to the VIP of the Azure Gateway or VM. Once you see only * returned, you know you have reached the Azure edge. When you see DNS names that include "MSN" returned, you know you have reached the Microsoft backbone.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\chad>tracert 13.72.98.5

Tracing route to 13.72.98.56 over a maximum of 30 hops

 1   3 ms    27 ms   13 ms  sdg-wks2044.europe.corp.microsoft.com [192.168.1.1]
 2   12 ms   2 ms    5 ms   freezeray.fareast.corp.microsoft.com [192.168.0.1]
 3   11 ms   11 ms   13 ms  10.111.192.1
 4   10 ms   9 ms    10 ms  192.168.37.169
 5   27 ms   30 ms   24 ms  10.224.252.26
 6   25 ms   25 ms   27 ms  ae8.er2.ord7.us.zip.zayo.com [128.177.105.165]
 7   27 ms   25 ms   29 ms  ae11.er1.ord7.us.zip.zayo.com [64.125.21.217]
 8   24 ms   24 ms   24 ms  chi-8075.msn.net [206.223.119.27]
 9   27 ms   25 ms   25 ms  ae4-0.ch1-96c-2b.ntwk.msn.net [104.44.224.90]
10   47 ms   48 ms   47 ms  be-64-0.ibr01.ch1.ntwk.msn.net [104.44.8.30]
11   48 ms   49 ms   48 ms  be-4-0.ibr02.was02.ntwk.msn.net [104.44.4.36]
12   47 ms   47 ms   48 ms  ae74-0.bl4-96cbe-1b.ntwk.msn.net [104.44.9.29]
13   *       *       *       Request timed out.
14   *       *       *       Request timed out.
15   *       *       *       Request timed out.
16   *       *       *       Request timed out.
17   *       *       *       Request timed out.
18   *       *       *       Request timed out.
19   *       *       *       Request timed out.
20   *       *       *       Request timed out.
21   *       *       *       Request timed out.
22   *       *       *       Request timed out.
23   *       *       *       Request timed out.
24   *       *       *       Request timed out.
25   *       *       *       Request timed out.
26   *       *       *       Request timed out.
27   *       *       *       Request timed out.
28   *       *       *       Request timed out.
29   *       *       *       Request timed out.
30   *       *       *       Request timed out.

Trace complete.
```

Next steps

For more information or help, check out the following links:

- [Optimize network throughput for Azure virtual machines](#)
- [Microsoft Support](#)

Troubleshooting: Azure point-to-site connection problems

7/5/2018 • 10 minutes to read • [Edit Online](#)

This article lists common point-to-site connection problems that you might experience. It also discusses possible causes and solutions for these problems.

VPN client error: A certificate could not be found

Symptom

When you try to connect to an Azure virtual network by using the VPN client, you receive the following error message:

A certificate could not be found that can be used with this Extensible Authentication Protocol. (Error 798)

Cause

This problem occurs if the client certificate is missing from **Certificates - Current User\Personal\Certificates**.

Solution

To resolve this problem, follow these steps:

1. Open Certificate Manager: Click **Start**, type **manage computer certificates**, and then click **manage computer certificates** in the search result.
2. Make sure that the following certificates are in the correct location:

CERTIFICATE	LOCATION
AzureClient.pfx	Current User\Personal\Certificates
Azuregateway-GUID.cloudapp.net	Current User\Trusted Root Certification Authorities
AzureGateway-GUID.cloudapp.net, AzureRoot.cer	Local Computer\Trusted Root Certification Authorities

3. Go to `Users<UserName>\AppData\Roaming\Microsoft\Network\Connections\{Cm<GUID>`, manually install the certificate (*.cer file) on the user and computer's store.

For more information about how to install the client certificate, see [Generate and export certificates for point-to-site connections](#).

NOTE

When you import the client certificate, do not select the **Enable strong private key protection** option.

VPN client error: The message received was unexpected or badly formatted

Symptom

When you try to connect to an Azure virtual network by using the VPN client, you receive the following error message:

The message received was unexpected or badly formatted. (Error 0x80090326)

Cause

This problem occurs if one of the following conditions is true:

- The user-defined routes (UDR) with default route on the Gateway Subnet is set incorrectly.
- The root certificate public key is not uploaded into the Azure VPN gateway.
- The key is corrupted or expired.

Solution

To resolve this problem, follow these steps:

1. Remove UDR on the Gateway Subnet. Make sure UDR forwards all traffic properly.
2. Check the status of the root certificate in the Azure portal to see whether it was revoked. If it is not revoked, try to delete the root certificate and reupload. For more information, see [Create certificates](#).

VPN client error: A certificate chain processed but terminated

Symptom

When you try to connect to an Azure virtual network by using the VPN client, you receive the following error message:

A certificate chain processed but terminated in a root certificate which is not trusted by the trust provider.

Solution

1. Make sure that the following certificates are in the correct location:

CERTIFICATE	LOCATION
AzureClient.pfx	Current User\Personal\Certificates
Azuregateway-GUID.cloudapp.net	Current User\Trusted Root Certification Authorities
AzureGateway-GUID.cloudapp.net, AzureRoot.cer	Local Computer\Trusted Root Certification Authorities

2. If the certificates are already in the location, try to delete the certificates and reinstall them. The **azuregateway-GUID.cloudapp.net** certificate is in the VPN client configuration package that you downloaded from the Azure portal. You can use file archivers to extract the files from the package.

File download error: Target URI is not specified

Symptom

You receive the following error message:

File download error. Target URI is not specified.

Cause

This problem occurs because of an incorrect gateway type.

Solution

The VPN gateway type must be **VPN**, and the VPN type must be **RouteBased**.

VPN client error: Azure VPN custom script failed

Symptom

When you try to connect to an Azure virtual network by using the VPN client, you receive the following error message:

Custom script (to update your routing table) failed. (Error 8007026f)

Cause

This problem might occur if you are trying to open the site-to-point VPN connection by using a shortcut.

Solution

Open the VPN package directly instead of opening it from the shortcut.

Cannot install the VPN client

Cause

An additional certificate is required to trust the VPN gateway for your virtual network. The certificate is included in the VPN client configuration package that is generated from the Azure portal.

Solution

Extract the VPN client configuration package, and find the .cer file. To install the certificate, follow these steps:

1. Open mmc.exe.
2. Add the **Certificates** snap-in.
3. Select the **Computer** account for the local computer.
4. Right-click the **Trusted Root Certification Authorities** node. Click **All-Tasks > Import**, and browse to the .cer file you extracted from the VPN client configuration package.
5. Restart the computer.
6. Try to install the VPN client.

Azure portal error: Failed to save the VPN gateway, and the data is invalid

Symptom

When you try to save the changes for the VPN gateway in the Azure portal, you receive the following error message:

Failed to save virtual network gateway <gateway name>. Data for certificate <certificate ID> is invalid.

Cause

This problem might occur if the root certificate public key that you uploaded contains an invalid character, such as a space.

Solution

Make sure that the data in the certificate does not contain invalid characters, such as line breaks (carriage returns). The entire value should be one long line. The following text is a sample of the certificate:

```
-----BEGIN CERTIFICATE-----  
MIIC5zCCAc+gAwIBAgIQFSwsLuUrCIdHwI3hzJbdBjANBgkqhkiG9w0BAQsFADAW  
MRQwEgYDVQQDDATQMlNSb290Q2VydDAeFw0xNzA2MTUwMjU4NDZaFw0xODA2MTUw  
MzE4NDZaMBYxFDASBgNVBAMMC1AyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQE  
AAOCAQ8AMIIBCgKCAQEAz8QUCWxxxxTrxF5yc5uUpL/bzwC5zz804ltB1NpPa/PI  
sa5uwLw/YFb8XG/JCWxUJpUzS/kHUKf1uqkY80U+fAmRmTEMq5wcaMhp3wRfeq+1  
G9OPBNTyqpnHe+154QAnj1DjsHXNL4AL1N8/TSzYTm7dkiq+EAItRRMrZ1Ywje  
407ChxIp0stB84MtMShyoSm2hg1+3zfWuaGxoJQwWiXh715kMHVTSj9zFechYd7  
50L1toRRDyyxsf0qweTFKIgFj13Hn/bq/UJG3AcyQNv1Cv1HwQnX0+hckVBB29wE  
sF8QSYk2MMGimPDYYt4ZM5tmYLxxxvGmrGhc+HWxZMeQIDAQABozEwLzAOBgNVHQ8B  
Af8EBAMCAgQWhQYDVR0QBByEFBE9zZwhQftVLBQNATC/LHLvMb0OMA0GCSqGSIb3  
DQEBCwUA4IBAQB7k0ySFUQu72sfj3BdNxrXSyOT4L2rADLhxxxiK0U6gHUF6ewZ  
/0h6y4mNkg3NgLT3j/Wc1qzHXZruhWAXSF+VbAGkwKA99xGW0cUJ+vKVYL/kDja  
gaZrxH1hTYVmwn4F7DWhteFqhzZ89/W9Mv6p180AimF96qDU8Ez8t860HQaFKU6  
2Nw9ZMsGkvLepZZi78yVBDCWMogBMhrRVXG/xQkBajgvL5syLwFB02kWGdC+wyWY  
U/Z+EK9UuHnn3Hkq/vXEzRVsYuaxchta0X2UNRzRq+o706l+iyLTpe6fnvW6il0i  
e8Jcej7mzunzyjz4chN0/WVF94MtxbUkLkqp  
-----END CERTIFICATE-----
```

Azure portal error: Failed to save the VPN gateway, and the resource name is invalid

Symptom

When you try to save the changes for the VPN gateway in the Azure portal, you receive the following error message:

Failed to save virtual network gateway <gateway name>. Resource name <certificate name you try to upload> is invalid.

Cause

This problem occurs because the name of the certificate contains an invalid character, such as a space.

Azure portal error: VPN package file download error 503

Symptom

When you try to download the VPN client configuration package, you receive the following error message:

Failed to download the file. Error details: error 503. The server is busy.

Solution

This error can be caused by a temporary network problem. Try to download the VPN package again after a few minutes.

Azure VPN Gateway upgrade: All Point to Site clients are unable to connect

Cause

If the certificate is more than 50 percent through its lifetime, the certificate is rolled over.

Solution

To resolve this problem, redeploy the Point to Site package on all clients.

Too many VPN clients connected at once

For each VPN gateway, the maximum number of allowable connections is 128. You can see the total number of connected clients in the Azure portal.

Point-to-site VPN incorrectly adds a route for 10.0.0.0/8 to the route table

Symptom

When you dial the VPN connection on the point-to-site client, the VPN client should add a route toward the Azure virtual network. The IP helper service should add a route for the subnet of the VPN clients.

The VPN client range belongs to a smaller subnet of 10.0.0.0/8, such as 10.0.12.0/24. Instead of a route for 10.0.12.0/24, a route for 10.0.0.0/8 is added that has higher priority.

This incorrect route breaks connectivity with other on-premises networks that might belong to another subnet within the 10.0.0.0/8 range, such as 10.50.0.0/24, that don't have a specific route defined.

Cause

This behavior is by design for Windows clients. When the client uses the PPP IPCP protocol, it obtains the IP address for the tunnel interface from the server (the VPN gateway in this case). However, because of a limitation in the protocol, the client does not have the subnet mask. Because there is no other way to get it, the client tries to guess the subnet mask based on the class of the tunnel interface IP address.

Therefore, a route is added based on the following static mapping:

If address belongs to class A --> apply /8

If address belongs to class B --> apply /16

If address belongs to class C --> apply /24

Solution

Have routes for other networks be injected in the routing table with longest prefix match or lower metric (hence higher priority) than the Point to Site.

VPN client cannot access network file shares

Symptom

The VPN client has connected to the Azure virtual network. However, the client cannot access network shares.

Cause

The SMB protocol is used for file share access. When the connection is initiated, the VPN client adds the session credentials and the failure occurs. After the connection is established, the client is forced to use the cache credentials for Kerberos authentication. This process initiates queries to the Key Distribution Center (a domain controller) to get a token. Because the client connects from the Internet, it might not be able to reach the domain controller. Therefore, the client cannot fail over from Kerberos to NTLM.

The only time that the client is prompted for a credential is when it has a valid certificate (with SAN=UPN) issued by the domain to which it is joined. The client also must be physically connected to the domain network. In this case, the client tries to use the certificate and reaches out to the domain controller. Then the Key Distribution Center returns a "KDC_ERR_C_PRINCIPAL_UNKNOWN" error. The client is forced to fail over to NTLM.

Solution

To work around the problem, disable the caching of domain credentials from the following registry subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\DisableDomainCreds - Set the value to 1
```

Cannot find the point-to-site VPN connection in Windows after reinstalling the VPN client

Symptom

You remove the point-to-site VPN connection and then reinstall the VPN client. In this situation, the VPN connection is not configured successfully. You do not see the VPN connection in the **Network connections** settings in Windows.

Solution

To resolve the problem, delete the old VPN client configuration files from **C:\users\username\AppData\Microsoft\Network\Connections<VirtualNetworkId>**, and then run the VPN client installer again.

Point-to-site VPN client cannot resolve the FQDN of the resources in the local domain

Symptom

When the client connects to Azure by using point-to-site VPN connection, it cannot resolve the FQDN of the resources in your local domain.

Cause

Point-to-site VPN client uses Azure DNS servers that are configured in the Azure virtual network. The Azure DNS servers take precedence over the local DNS servers that are configured in the client, so all DNS queries are sent to the Azure DNS servers. If the Azure DNS servers do not have the records for the local resources, the query fails.

Solution

To resolve the problem, make sure that the Azure DNS servers that used on the Azure virtual network can resolve the DNS records for local resources. To do this, you can use DNS Forwarders or Conditional forwarders. For more information, see [Name resolution using your own DNS server](#)

The point-to-site VPN connection is established, but you still cannot connect to Azure resources

Cause

This problem may occur if VPN client does not get the routes from Azure VPN gateway.

Solution

To resolve this problem, [reset Azure VPN gateway](#).

Error: "The revocation function was unable to check revocation because the revocation server was offline.(Error 0x80092013)"

Causes

This error message occurs if the client cannot access <http://crl3.digicert.com/ssca-sha2-g1.crl> and <http://crl4.digicert.com/ssca-sha2-g1.crl>. The revocation check requires access to these two sites. This problem typically happens on the client that has proxy server configured. In some environments, if the requests are not going through the proxy server, it will be denied at the Edge Firewall.

Solution

Check the proxy server settings, make sure that the client can access <http://crl3.digicert.com/ssca-sha2-g1.crl> and

<http://crl4.digicert.com/ssca-sha2-g1.crl>.

VPN Client Error: The connection was prevented because of a policy configured on your RAS/VPN server. (Error 812)

Cause

This error occurs if the RADIUS server that you used for authenticating VPN client has incorrect settings, or Azure Gateway can't reach the Radius server.

Solution

Make sure that RADIUS server is configured correctly. For More information, see [Integrate RADIUS authentication with Azure Multi-Factor Authentication Server](#).

"Error 405" when you download root certificate from VPN Gateway

Cause

Root certificate had not been installed. The root certificate is installed in the client's **Trusted certificates** store.

VPN Client Error: The remote connection was not made because the attempted VPN tunnels failed. (Error 800)

Cause

The NIC driver is outdated.

Solution

Update the NIC driver:

1. Click **Start**, type **Device Manager**, and select it from the list of results. If you're prompted for an administrator password or confirmation, type the password or provide confirmation.
2. In the **Network adapters** categories, find the NIC that you want to update.
3. Double-click the device name, select **Update driver**, select **Search automatically for updated driver software**.
4. If Windows doesn't find a new driver, you can try looking for one on the device manufacturer's website and follow their instructions.
5. Restart the computer and try the connection again.

Error: 'File download error Target URI is not specified'

Cause

This is caused by an incorrect gateway type is configured.

Solution

The Azure VPN gateway type must be VPN and the VPN type must be **RouteBased**.

VPN package installer doesn't complete

Cause

This problem can be caused by the previous VPN client installations.

Solution

Delete the old VPN client configuration files from
C:\users\username\AppData\Microsoft\Network\Connections<VirtualNetworkId> and run the VPN

client installer again.

The VPN client hibernates or sleep after some time

Solution

Check the sleep and hibernate settings in the computer that the VPN client is running on.

Troubleshoot Point-to-Site VPN connections from Mac OS X VPN clients

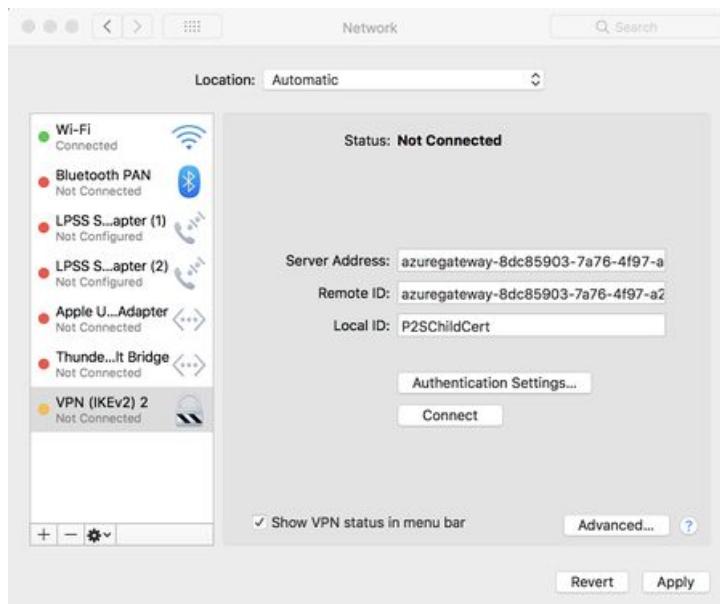
4/9/2018 • 2 minutes to read • [Edit Online](#)

This article helps you troubleshoot Point-to-Site connectivity issues from Mac OS X using the native VPN client and IKEv2. The VPN client in Mac for IKEv2 is very basic and does not allow for much customization. There are only four settings that need to be checked:

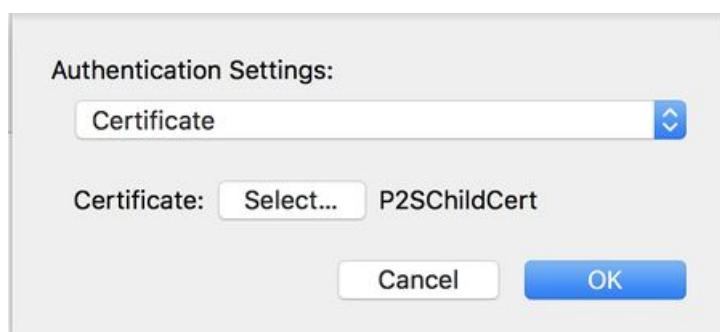
- Server Address
- Remote ID
- Local ID
- Authentication Settings
- OS Version (10.11 or higher)

Troubleshoot certificate-based authentication

1. Check the VPN client settings. Go to the **Network Setting** by pressing Command + Shift, and then type "VPN" to check the VPN client settings. From the list, click the VPN entry that needs to be investigated.



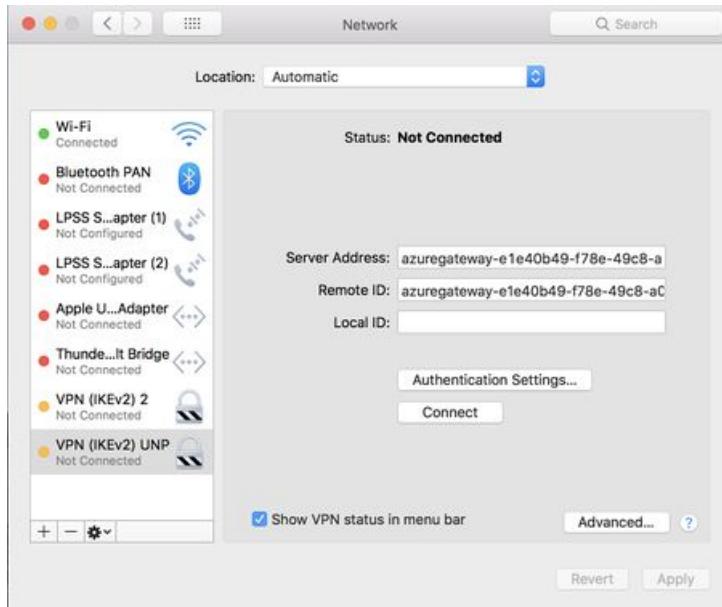
2. Verify that the **Server Address** is the complete FQDN and includes the clouddapp.net.
3. The **Remote ID** should be the same as the Server Address (Gateway FQDN).
4. The **Local ID** should be the same as the **Subject** of the client certificate.
5. Click on **Authentication Settings** to open the Authentication Settings page.



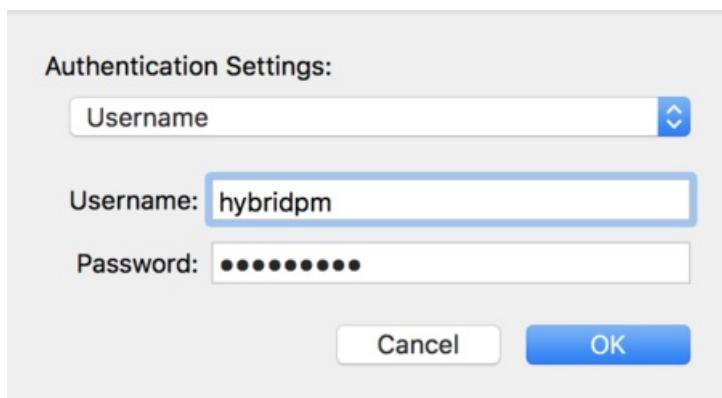
6. Verify that **Certificate** is selected from the dropdown.
7. Click the **Select** button and verify that the correct certificate is selected. Click **OK** to save any changes.

Troubleshoot username and password authentication

1. Check the VPN client settings. Go to the **Network Setting** by pressing Command + Shift, and then type "VPN" to check the VPN client settings. From the list, click the VPN entry that needs to be investigated.



2. Verify that the **Server Address** is the complete FQDN and includes the cloudapp.net.
3. The **Remote ID** should be the same as the Server Address (Gateway FQDN).
4. The **Local ID** can be blank.
5. Click the **Authentication Setting** button and verify that "Username" is selected from the dropdown.



6. Verify that the correct credentials are entered.

Additional steps

If you try the previous steps and everything is configured properly, download [Wireshark](#) and perform a packet capture.

1. Filter on *isakmp* and look at the **IKE_SA** packets. You should be able to look at the SA proposal details under the **Payload: Security Association**.
2. Verify that the client and the server have a common set.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.048622	10.69.0.216	104.42.0.178	ISAKMP	646	IKE_SA_INIT MID=00 Initiator Request
4	0.071347	104.42.0.178	10.69.0.216	ISAKMP	78	IKE_SA_INIT MID=00 Responder Response
5	0.093314	10.69.0.216	104.42.0.178	ISAKMP	646	IKE_SA_INIT MID=00 Initiator Request
6	0.116224	104.42.0.178	10.69.0.216	ISAKMP	78	IKE_SA_INIT MID=00 Responder Response


```

Responder SPI: 0000000000000000
Next payload: Security Association (33)
> Version: 2.0
> Flags: 0x08 (Initiator, No higher version, Request)
> Flags: 0x08 (Initiator, No higher version, Request)
Message ID: 0x00000000
Length: 646
Payload: Security Association (33)
  payload: Key Exchange (34)
    .0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 228
  > Payload: Proposal (2) # 1
    Next payload: Proposal (2)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 44
    Proposal number: 1
    Protocol ID: IKE (1)
    SPI Size: 0
    Proposal transforms: 4
    > Payload: Transform (3)
      Next payload: Transform (3)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0xB8
      Payload length: 12
      Transform Type: Encryption Algorithm (ENCR) (1)
      Reserved: 00
      Transform ID (ENCR): ENCR_AES_CBC (12)
      > Transform Attribute (t=14,l=2): Key Length: 256
    > Payload: Transform (3)
      Next payload: Transform (3)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0x00
  
```

Next steps

For additional help, see [Microsoft Support](#).

Troubleshooting: An Azure site-to-site VPN connection cannot connect and stops working

5/11/2018 • 3 minutes to read • [Edit Online](#)

After you configure a site-to-site VPN connection between an on-premises network and an Azure virtual network, the VPN connection suddenly stops working and cannot be reconnected. 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 steps

To resolve the problem, first try to [reset the Azure VPN gateway](#) and reset the tunnel from the on-premises VPN device. If the problem persists, follow these steps to identify the cause of the problem.

Prerequisite step

Check the type of the Azure VPN gateway.

1. Go to the [Azure portal](#).
2. Check the **Overview** page of the VPN gateway for the type information.

Resource group (change)	SKU
Thomas	VpnGw1
Location	Gateway type
East Asia	VPN
Subscription (change)	VPN type
<Subscription name>	Route-based
Subscription ID	Virtual network
<ID>	Thomas
	Public IP address
	<IP>

Step 1. Check whether the on-premises VPN device is validated

1. Check whether you are using a [validated VPN device and operating system version](#). If the device is not a validated VPN device, you might have to contact the device manufacturer to see if there is a compatibility issue.
2. Make sure that the VPN device is correctly configured. For more information, see [Edit device configuration samples](#).

Step 2. Verify the shared key

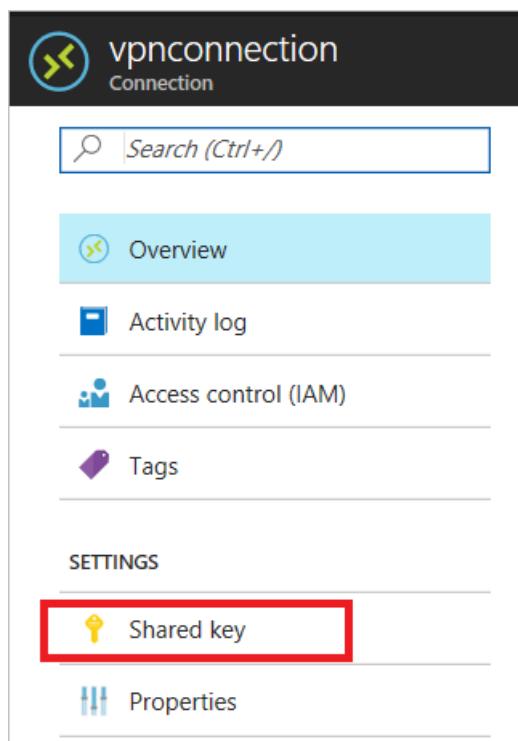
Compare the shared key for the on-premises VPN device to the Azure Virtual Network VPN to make sure that the

keys match.

To view the shared key for the Azure VPN connection, use one of the following methods:

Azure portal

1. Go to the VPN gateway site-to-site connection that you created.
2. In the **Settings** section, click **Shared key**.



Azure PowerShell

For the Azure Resource Manager deployment model:

```
Get-AzureRmVirtualNetworkGatewayConnectionSharedKey -Name <Connection name> -ResourceGroupName <Resource group name>
```

For the classic deployment model:

```
Get-AzureVNetGatewayKey -VNetName -LocalNetworkSiteName
```

Step 3. Verify the VPN peer IPs

- The IP definition in the **Local Network Gateway** object in Azure should match the on-premises device IP.
- The Azure gateway IP definition that is set on the on-premises device should match the Azure gateway IP.

Step 4. Check UDR and NSGs on the gateway subnet

Check for and remove user-defined routing (UDR) or Network Security Groups (NSGs) on the gateway subnet, and then test the result. If the problem is resolved, validate the settings that UDR or NSG applied.

Step 5. Check the on-premises VPN device external interface address

- If the Internet-facing IP address of the VPN device is included in the **Local network** definition in Azure, you might experience sporadic disconnections.
- The device's external interface must be directly on the Internet. There should be no network address translation or firewall between the Internet and the device.
- To configure firewall clustering to have a virtual IP, you must break the cluster and expose the VPN appliance

directly to a public interface that the gateway can interface with.

Step 6. Verify that the subnets match exactly (Azure policy-based gateways)

- Verify that the virtual network address space(s) match exactly between the Azure virtual network and on-premises definitions.
- Verify that the subnets match exactly between the **Local Network Gateway** and on-premises definitions for the on-premises network.

Step 7. Verify the Azure gateway health probe

1. Open health probe by browsing to the following URL:

```
https://<YourVirtualNetworkGatewayIP>:8081/healthprobe
```

2. Click through the certificate warning.
3. If you receive a response, the VPN gateway is considered healthy. If you don't receive a response, the gateway might not be healthy or an NSG on the gateway subnet is causing the problem. The following text is a sample response:

```
<?xml version="1.0"?> Primary Instance: GatewayTenantWorker_IN_1 GatewayTenantVersion:  
14.7.24.6</string>
```

Step 8. Check whether the on-premises VPN device has the perfect forward secrecy feature enabled

The perfect forward secrecy feature can cause disconnection problems. If the VPN device has perfect forward secrecy enabled, disable the feature. Then update the VPN gateway IPsec policy.

Next steps

- [Configure a site-to-site connection to a virtual network](#)
- [Configure an IPsec/IKE policy for site-to-site VPN connections](#)

Troubleshooting: Azure Site-to-Site VPN disconnects intermittently

5/21/2018 • 2 minutes to read • [Edit Online](#)

You might experience the problem that a new or existing Microsoft Azure Site-to-Site VPN connection is not stable or disconnects regularly. This article provides troubleshoot steps to help you identify and resolve the cause of the 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 steps

Prerequisite step

Check the type of Azure virtual network gateway:

1. Go to [Azure portal](#).
2. Check the **Overview** page of the virtual network gateway for the type information.

Resource group (change) Thomas	SKU VpnGw1
Location East Asia	Gateway type VPN
Subscription (change) <Subscription name>	VPN type Route-based
Subscription ID <ID>	Virtual network Thomas
	Public IP address <IP>

Step 1 Check whether the on-premises VPN device is validated

1. Check whether you are using a [validated VPN device and operating system version](#). If the VPN device is not validated, you may have to contact the device manufacturer to see if there is any compatibility issue.
2. Make sure that the VPN device is correctly configured. For more information, see [Editing device configuration samples](#).

Step 2 Check the Security Association settings(for policy-based Azure virtual network gateways)

1. Make sure that the virtual network, subnets and, ranges in the **Local network gateway** definition in Microsoft Azure are same as the configuration on the on-premises VPN device.
2. Verify that the Security Association settings match.

Step 3 Check for User-Defined Routes or Network Security Groups on Gateway Subnet

A user-defined route on the gateway subnet may be restricting some traffic and allowing other traffic. This makes it appear that the VPN connection is unreliable for some traffic and good for others.

Step 4 Check the "one VPN Tunnel per Subnet Pair" setting (for policy-based virtual network gateways)

Make sure that the on-premises VPN device is set to have **one VPN tunnel per subnet pair** for policy-based virtual network gateways.

Step 5 Check for Security Association Limitation (for policy-based virtual network gateways)

The Policy-based virtual network gateway has limit of 200 subnet Security Association pairs. If the number of Azure virtual network subnets multiplied times by the number of local subnets is greater than 200, you see sporadic subnets disconnecting.

Step 6 Check on-premises VPN device external interface address

- If the Internet facing IP address of the VPN device is included in the **Local network gateway** definition in Azure, you may experience sporadic disconnections.
- The device's external interface must be directly on the Internet. There should be no Network Address Translation (NAT) or firewall between the Internet and the device.
- If you configure Firewall Clustering to have a virtual IP, you must break the cluster and expose the VPN appliance directly to a public interface that the gateway can interface with.

Step 7 Check whether the on-premises VPN device has Perfect Forward Secrecy enabled

The **Perfect Forward Secrecy** feature can cause the disconnection problems. If the VPN device has **Perfect forward Secrecy** enabled, disable the feature. Then [update the virtual network gateway IPsec policy](#).

Next steps

- [Configure a Site-to-Site connection to a virtual network](#)
- [Configure IPsec/IKE policy for Site-to-Site VPN connections](#)

Create a Site-to-Site connection using the Azure portal (classic)

2/16/2018 • 13 minutes to read • [Edit Online](#)

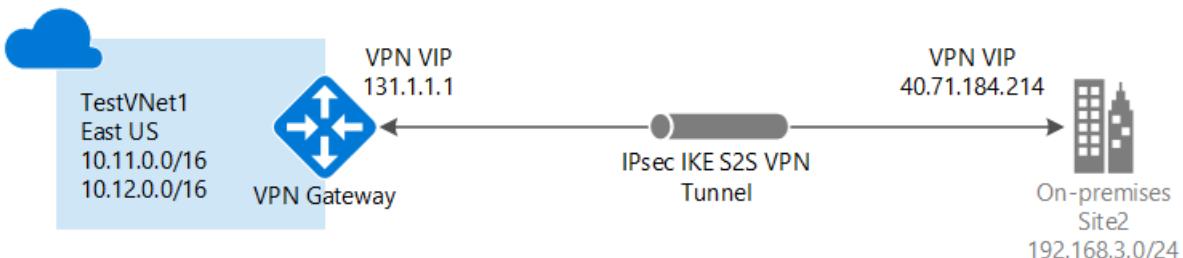
NOTE

This article is written for the classic deployment model. If you're new to Azure, we recommend that you use the Resource Manager deployment model instead. The Resource Manager deployment model is the most current deployment model and offers more options and feature compatibility than the classic deployment model. For more information about the deployment models, see [Understanding deployment models](#).

For the Resource Manager version of this article, select it from the drop-down list below, or from the table of contents on the left.

This article 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. The steps in this article apply to the classic deployment model. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it. For more information about VPN gateways, see [About VPN gateway](#).



Before you begin

Verify that you have met the following criteria before beginning configuration:

- Verify that you want to work in the classic deployment model. If you want to work in the Resource Manager deployment model, see [Create a Site-to-Site connection \(Resource Manager\)](#). When possible, we recommend that you use the Resource Manager deployment model.
- 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. This IP address cannot be located behind a NAT.
- 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.
- Currently, PowerShell is required to specify the shared key and create the VPN gateway connection. Install the

latest version of the Azure Service Management (SM) PowerShell cmdlets. For more information, see [How to install and configure Azure PowerShell](#). When working with PowerShell for this configuration, make sure that you are running as administrator.

Sample configuration values for this exercise

The examples in this article use the following values. You can use these values to create a test environment, or refer to them to better understand the examples in this article.

- **VNet Name:** TestVNet1
- **Address Space:**
 - 10.11.0.0/16
 - 10.12.0.0/16 (optional for this exercise)
- **Subnets:**
 - FrontEnd: 10.11.0.0/24
 - BackEnd: 10.12.0.0/24 (optional for this exercise)
- **GatewaySubnet:** 10.11.255.0/27
- **Resource Group:** TestRG1
- **Location:** East US
- **DNS Server:** 10.11.0.3 (optional for this exercise)
- **Local site name:** Site2
- **Client address space:** The address space that is located on your on-premises site.

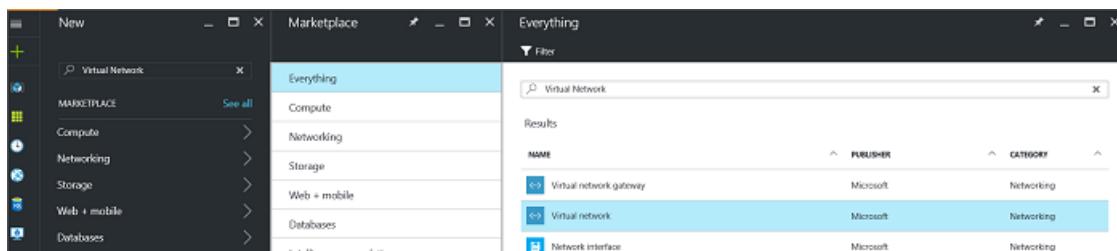
1. Create a virtual network

When you create a virtual network to use for a S2S connection, you need to make sure that the address spaces that you specify do not overlap with any of the client address spaces for the local sites that you want to connect to. If you have overlapping subnets, your connection won't work properly.

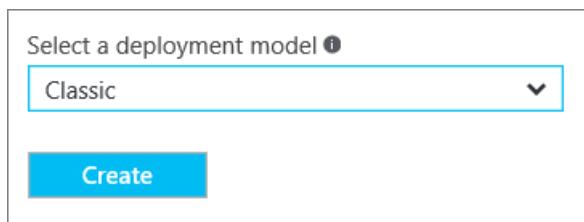
- If you already have a VNet, verify that the settings are compatible with your VPN gateway design. Pay particular attention to any subnets that may overlap with other networks.
- If you don't already have a virtual network, create one. Screenshots are provided as examples. Be sure to replace the values with your own.

To create a virtual network

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **+**. In the **Search the marketplace** field, type 'Virtual Network'. Locate **Virtual Network** from the returned list and click to open the **Virtual Network** page.



3. Near the bottom of the Virtual Network page, from the **Select a deployment model** dropdown list, select **Classic**, and then click **Create**.



4. On the **Create virtual network(classic)** page, configure the VNet settings. On this page, you add your first address space and a single subnet address range. After you finish creating the VNet, you can go back and add additional subnets and address spaces.

Create virtual network (clas... □ X)

* Name
TestVNet1 ✓

* Address space ⓘ
10.11.0.0/16 ✓
10.11.0.0 - 10.11.255.255 (65536 addresses)

* Subnet name
FrontEnd ✓

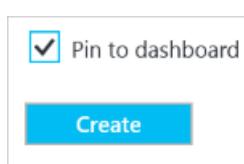
* Subnet address range ⓘ
10.11.0.0/24 ✓
10.11.0.0 - 10.11.0.255 (256 addresses)

* Subscription
Windows Azure Internal Consumption ▾

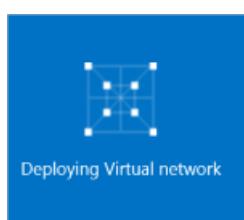
* Resource Group ⓘ
 Create new Use existing
TestRG1 ✓

* Location
East US ▾

5. Verify that the **Subscription** is the correct one. You can change subscriptions by using the drop-down.
6. Click **Resource group** and either select an existing resource group, or create a new one by typing a name. For more information about resource groups, visit [Azure Resource Manager Overview](#).
7. Next, select the **Location** settings for your VNet. The location determines where the resources that you deploy to this VNet will reside.
8. If you want to be able to find your VNet easily on the dashboard, select **Pin to dashboard**. Click **Create** to create your VNet.



9. After clicking 'Create', a tile appears on the dashboard that reflects the progress of your VNet. The tile changes as the VNet is being created.



2. Add additional address space

After you create your virtual network, you can add additional address space. Adding additional address space is not a required part of a S2S configuration, but if you require multiple address spaces, use the following steps:

1. Locate the virtual networks in the portal.
2. On the page for your virtual network, under the **Settings** section, click **Address space**.
3. On the Address space page, click **+Add** and enter additional address space.

3. Specify a DNS server

DNS settings are not a required part of a S2S configuration, but DNS is necessary if you want name resolution. Specifying a value does not create a new DNS server. The DNS server IP address that you specify should be a DNS server that can resolve the names for the resources you are connecting to. For the example settings, we used a private IP address. The IP address we use is probably not the IP address of your DNS server. Be sure to use your own values.

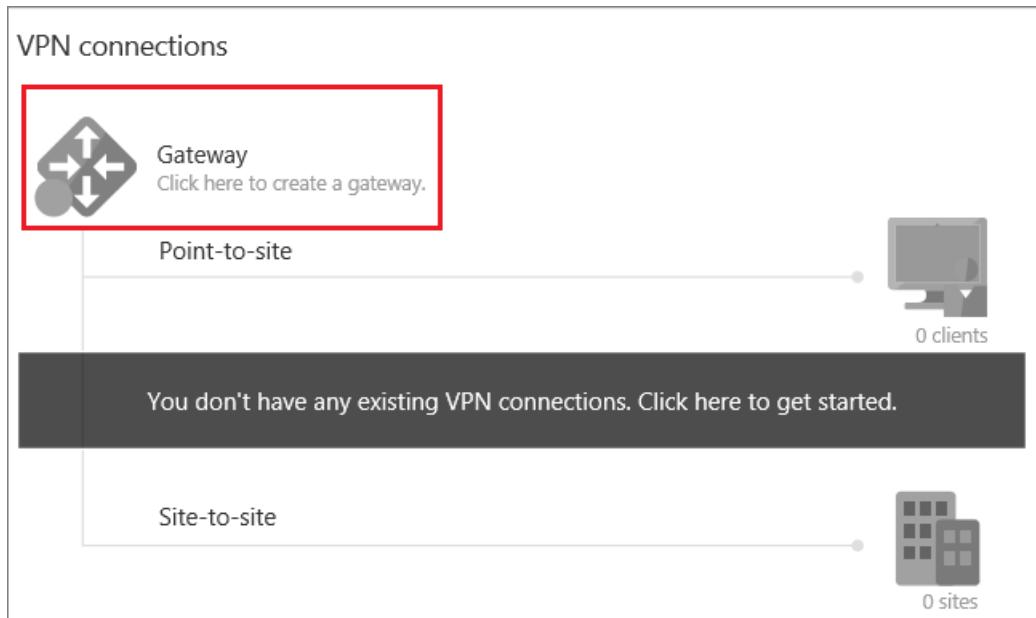
After you create your virtual network, you can add the IP address of a DNS server to handle name resolution. Open the settings for your virtual network, click DNS servers, and add the IP address of the DNS server that you want to use for name resolution.

1. Locate the virtual networks in the portal.
2. On the page for your virtual network, under the **Settings** section, click **DNS servers**.
3. Add a DNS server.
4. To save your settings, click **Save** at the top of the page.

4. Configure the local site

The local site typically refers to your on-premises location. It contains the IP address of the VPN device to which you will create a connection, and the IP address ranges that will be routed through the VPN gateway to the VPN device.

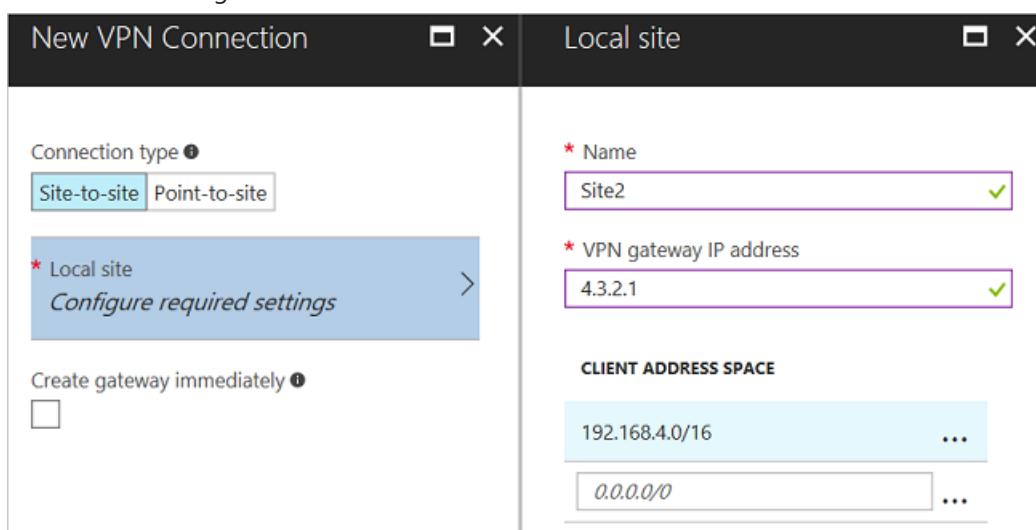
1. In the portal, navigate to the virtual network for which you want to create a gateway.
2. On the page for your virtual network, on the **Overview** page, in the VPN connections section, click **Gateway** to open the **New VPN Connection** page.



3. On the **New VPN Connection** page, select **Site-to-site**.

4. Click **Local site - Configure required settings** to open the **Local site** page. Configure the settings, and then click **OK** to save the settings.

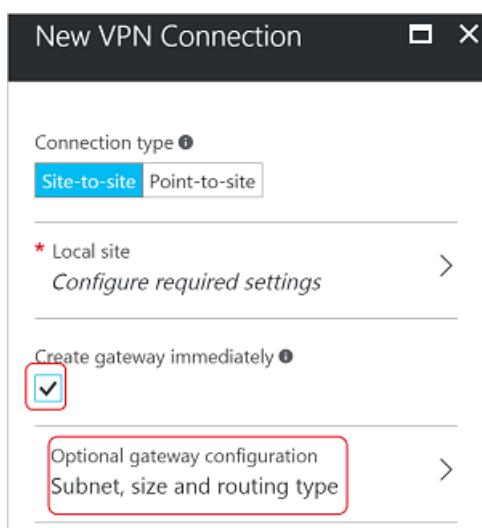
- **Name:** Create a name for your local site to make it easy for you to identify.
- **VPN gateway IP address:** This is the public IP address of the VPN device for your on-premises network. The VPN device requires an IPv4 public IP address. Specify a valid public IP address for the VPN device to which you want to connect. It cannot be behind NAT and has to be reachable by Azure. If you don't know the IP address of your VPN device, you can always put in a placeholder value (as long as it is in the format of a valid public IP address) and then change it later.
- **Client Address space:** List the IP address ranges that you want routed to the local on-premises network through this gateway. You can add multiple address space ranges. Make sure that the ranges you specify here do not overlap with ranges of other networks your virtual network connects to, or with the address ranges of the virtual network itself.



5. Configure the gateway subnet

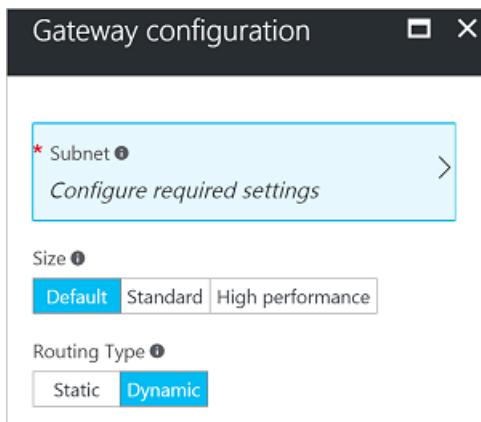
You must create a gateway subnet for your VPN gateway. The gateway subnet contains the IP addresses that the VPN gateway services use.

1. On the **New VPN Connection** page, select the checkbox **Create gateway immediately**. The 'Optional gateway configuration' page appears. If you don't select the checkbox, you won't see the page to configure the gateway subnet.

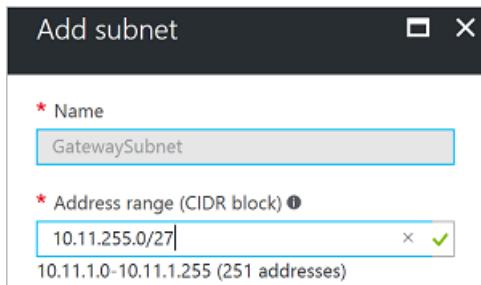


2. To open the **Gateway configuration** page, click **Optional gateway configuration - Subnet, size, and routing type**.
3. On the **Gateway Configuration** page, click **Subnet - Configure required settings** to open the **Add**

subnet page.



- On the **Add subnet** page, add the gateway subnet. The size of the gateway subnet that you specify depends on the VPN gateway configuration that you want to create. While it is possible to create a gateway subnet as small as /29, we recommend that you use /27 or /28. This creates a larger subnet that includes more addresses. Using a larger gateway subnet allows for enough IP addresses to accommodate possible future configurations.



6. Specify the SKU and VPN type

- Select the gateway **Size**. This is the gateway SKU that you use to create your virtual network gateway. In the portal, the 'Default SKU' = **Basic**. Classic VPN gateways use the old (legacy) gateway SKUs. For more information about the legacy gateway SKUs, see [Working with virtual network gateway SKUs \(old SKUs\)](#).

The screenshot shows two overlapping windows: 'New VPN Connection' and 'Gateway configuration'.

New VPN Connection (Left):

- Connection type:** Site-to-site selected (radio button highlighted).
- Local site:** Site2
- Create gateway immediately:** Checked (checkbox highlighted).
- Optional gateway configuration:** Subnet, size and routing type

Gateway configuration (Right):

- Subnet:** 10.11.255.0/27
- Size:** Standard selected (radio button highlighted).
- Routing Type:** Dynamic selected (radio button highlighted).

- Select the **Routing Type** for your gateway. This is also known as the VPN type. It's important to select the correct gateway type because you cannot convert the gateway from one type to another. Your VPN device must be compatible with the routing type you select. For more information about VPN type, see [About VPN Gateway Settings](#). You may see articles referring to 'RouteBased' and 'PolicyBased' VPN types. 'Dynamic' corresponds to 'RouteBased', and 'Static' corresponds to 'PolicyBased'.

3. Click **OK** to save the settings.
4. On the **New VPN Connection** page, click **OK** at the bottom of the page to begin creating your virtual network gateway. Depending on the SKU you select, it can take up to 45 minutes to create a virtual network gateway.

7. 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:

- 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 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](#).

8. Create the connection

In this step, you set the shared key and create the connection. The key you set is must be the same key that was used in your VPN device configuration.

NOTE

Currently, this step is not available in the Azure portal. You must use the Service Management (SM) version of the Azure PowerShell cmdlets.

Step 1. Connect to your Azure account

1. Open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

2. Check the subscriptions for the account.

```
Get-AzureSubscription
```

3. If you have more than one subscription, select the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionId "Replace_with_your_subscription_ID"
```

Step 2. Set the shared key and create the connection

When working with PowerShell and the classic deployment model, sometimes the names of resources in the portal are not the names the Azure expects to see when using PowerShell. The following steps help you export the network configuration file to obtain the exact values for the names.

1. 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
```

2. Open the network configuration file with an xml editor and check the values for 'LocalNetworkSite name' and 'VirtualNetworkSite name'. Modify the example to reflect the values that you need. When specifying a name that contains spaces, use single quotation marks around the value.
3. Set the shared key and create the connection. The '-SharedKey' is a value that you generate and specify. In the example, we used 'abc123', but you can generate (and should) use something more complex. The important thing is that the value you specify here must be the same value that you specified when configuring your VPN device.

```
Set-AzureVNetGatewayKey -VNetName 'Group TestRG1 TestVNet1' `  
-LocalNetworkSiteName 'D1BFC9CB_Site2' -SharedKey abc123
```

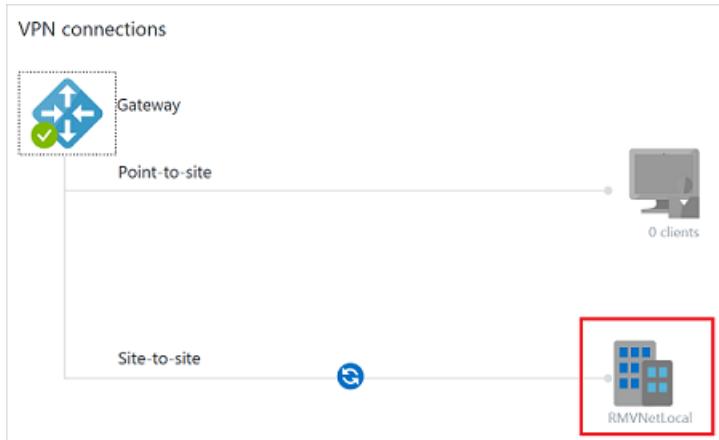
When the connection is created, the result is: **Status: Successful**.

9. Verify your connection

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.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.

3. On the VPN connections graphic, click the site.



4. On the **Site-to-site VPN connections** blade, view the information about your site.

NAME	STATUS	LAST EVENT	DATA IN / OUT
RMVNetLocal	Connected	1/7/2017 12:40:54 AM	177.97 KB / 177.72 KB

5. To view more information about the connection, click the name of the connection to open the **Site-to-site VPN Connection** blade.

RMVNetLocal
Site-to-site VPN connections

Connect Disconnect Delete

Local site
RMVNetLocal >

STATUS
Connected

LAST EVENT TIME STAMP
1/7/2017 12:40:54 AM

LAST EVENT ID
24401

LAST EVENT MESSAGE
The connectivity state for the local network site 'RMVNetLocal' changed from Connecting to Connected.

DATA IN / OUT
283.5 KB / 282.34 KB

CONNECTION ESTABLISHED
1/7/2017 12:40:54 AM

If you are having trouble connecting, see the **Troubleshoot** section of the table of contents in the left pane.

How to reset a VPN 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. For steps, see [Reset a VPN gateway](#).

How to change a gateway SKU

For the steps to change a gateway SKU, see [Resize a gateway SKU](#).

Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).
- For information about Forced Tunneling, see [About Forced Tunneling](#).

Configure a Point-to-Site connection to a VNet using certificate authentication (classic): Azure portal

2/21/2018 • 21 minutes to read • [Edit Online](#)

NOTE

This article is written for the classic deployment model. If you're new to Azure, we recommend that you use the Resource Manager deployment model instead. The Resource Manager deployment model is the most current deployment model and offers more options and feature compatibility than the classic deployment model. For more information about the deployment models, see [Understanding deployment models](#).

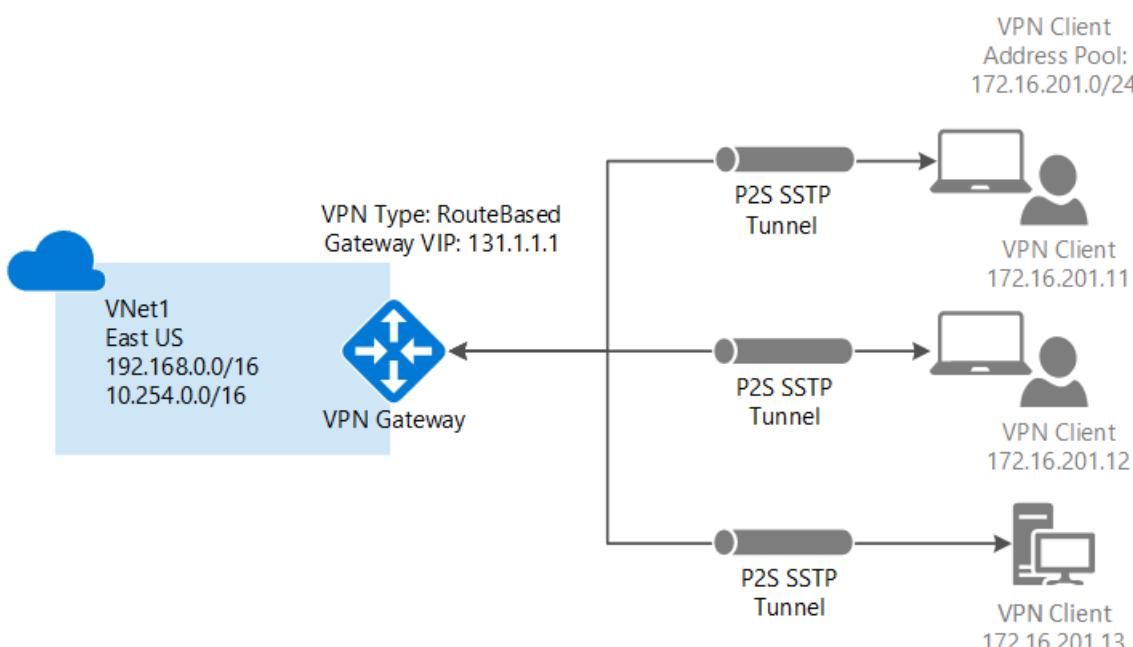
For the Resource Manager version of this article, select it from the drop-down list below, or from the table of contents on the left.

This article shows you how to create a VNet with a Point-to-Site connection in the classic deployment model using the Azure portal. This configuration uses certificates to authenticate the connecting client, either self-signed or CA issued. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:

A Point-to-Site (P2S) VPN gateway lets you create a secure connection to your virtual network from an individual client computer. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location, such as when you are telecommuting from home or a conference. A P2S VPN is also a useful solution to use instead of a Site-to-Site VPN when you have only a few clients that need to connect to a VNet. A P2S VPN connection is established by starting it from the client computer.

IMPORTANT

The classic deployment model supports Windows VPN clients only and uses the Secure Socket Tunneling Protocol (SSTP), an SSL-based VPN protocol. In order to support non-Windows VPN clients, your VNet must be created using the Resource Manager deployment model. The Resource Manager deployment model supports IKEv2 VPN in addition to SSTP. For more information, see [About P2S connections](#).



Point-to-Site certificate authentication connections require the following:

- A Dynamic VPN gateway.
- The public key (.cer file) for a root certificate, which is uploaded to Azure. This is considered a trusted certificate and is used for authentication.
- A client certificate generated from the root certificate, and installed on each client computer that will connect. This certificate is used for client authentication.
- A VPN client configuration package must be generated and installed on every client computer that connects. The client configuration package configures the native VPN client that is already on the operating system with the necessary information to connect to the VNet.

Point-to-Site connections do not require a VPN device or an on-premises public-facing IP address. The VPN connection is created over SSTP (Secure Socket Tunneling Protocol). On the server side, we support SSTP versions 1.0, 1.1, and 1.2. The client decides which version to use. For Windows 8.1 and above, SSTP uses 1.2 by default.

For more information about Point-to-Site connections, see the [Point-to-Site FAQ](#) at the end of this article.

Example settings

You can use the following values to create a test environment, or refer to these values to better understand the examples in this article:

- **Name: VNet1**
- **Address space: 192.168.0.0/16**
For this example, we use only one address space. You can have more than one address space for your VNet, as shown in the diagram.
- **Subnet name: FrontEnd**
- **Subnet address range: 192.168.1.0/24**
- **Subscription:** If you have more than one subscription, verify that you are using the correct one.
- **Resource Group: TestRG**
- **Location: East US**
- **Connection type: Point-to-site**
- **Client Address Space: 172.16.201.0/24.** VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the specified pool.
- **GatewaySubnet: 192.168.200.0/24.** The Gateway subnet must use the name 'GatewaySubnet'.
- **Size:** Select the gateway SKU that you want to use.
- **Routing Type: Dynamic**

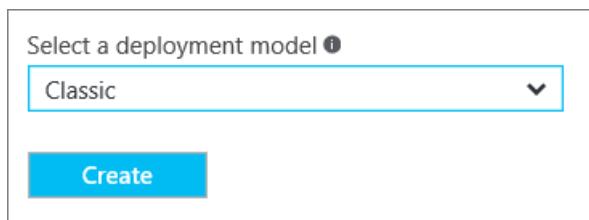
1. Create a virtual network and a VPN gateway

Before beginning, verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).

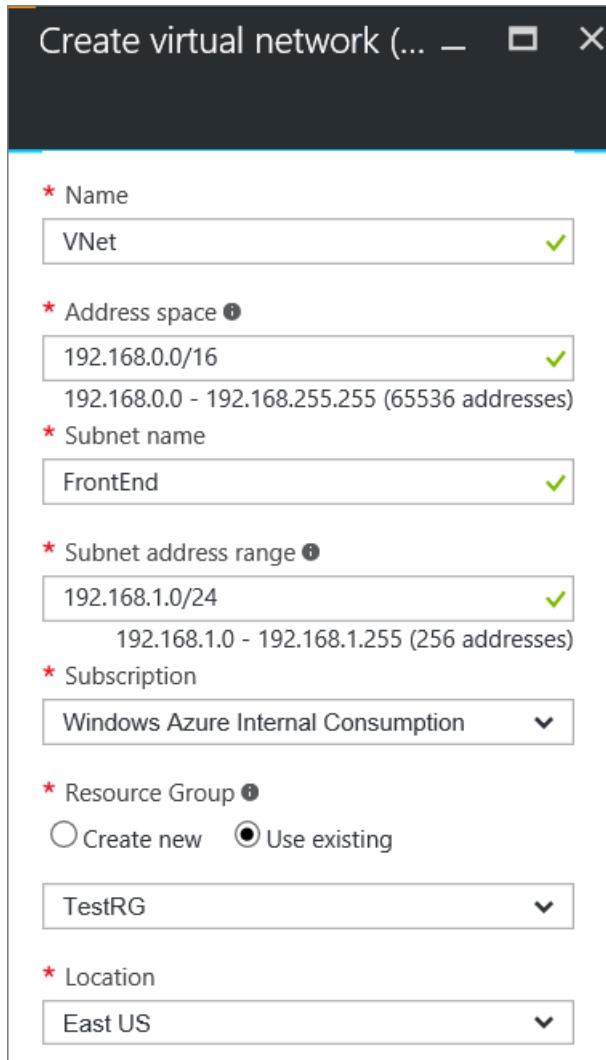
Part 1: Create a virtual network

If you don't already have a virtual network, create one. Screenshots are provided as examples. Be sure to replace the values with your own. To create a VNet by using the Azure portal, use the following steps:

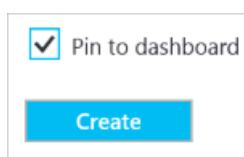
1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **Create a resource > Networking > Virtual Network**.
3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, select **Classic**, and then click **Create**.



4. On the **Create virtual network** page, configure the VNet settings. On this page, you add your first address space and a single subnet address range. After you finish creating the VNet, you can go back and add additional subnets and address spaces.



5. Verify that the **Subscription** is the correct one. You can change subscriptions by using the drop-down.
6. Click **Resource group** and either select an existing resource group, or create a new one by typing a name for your new resource group. If you are creating a new resource group, name the resource group according to your planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](#).
7. Next, select the **Location** settings for your VNet. The location determines where the resources that you deploy to this VNet will reside.
8. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



9. After clicking Create, a tile appears on your dashboard that will reflect the progress of your VNet. The tile changes as the VNet is being created.

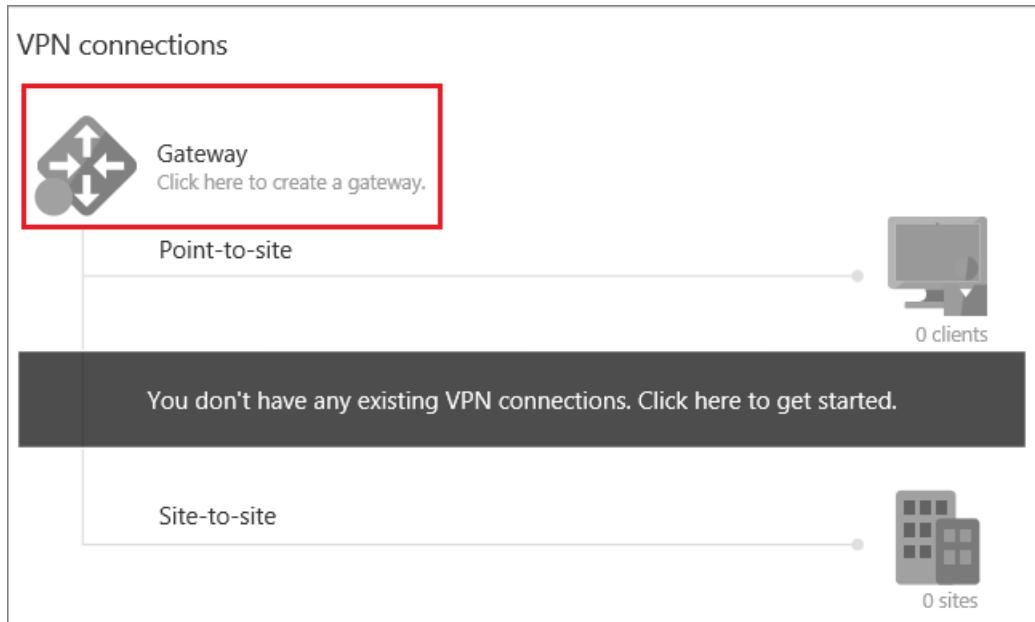


10. Once your virtual network has been created, you see **Created**.
11. Add a DNS server (optional). After you create your virtual network, you can add the IP address of a DNS server for name resolution. The DNS server IP address that you specify should be the address of a DNS server that can resolve the names for the resources in your VNet.
To add a DNS server, open the settings for your virtual network, click DNS servers, and add the IP address of the DNS server that you want to use.

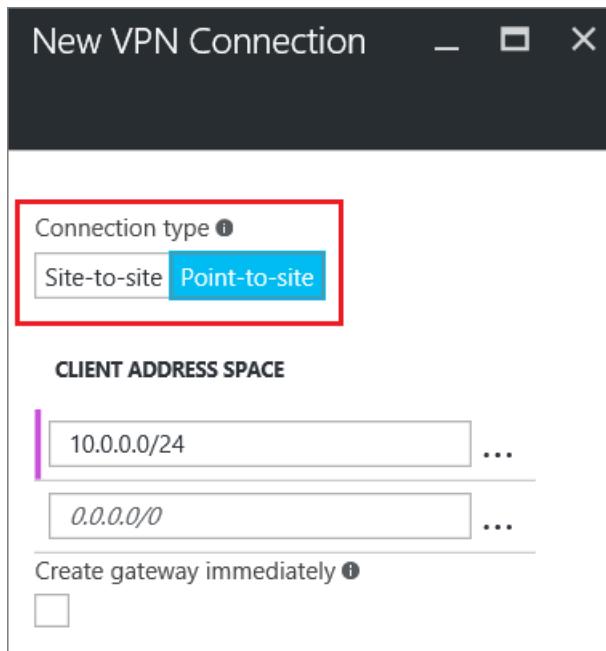
Part 2: Create gateway subnet and a dynamic routing gateway

In this step, you create a gateway subnet and a Dynamic routing gateway. In the Azure portal for the classic deployment model, creating the gateway subnet and the gateway can be done through the same configuration pages. The gateway subnet is used for the gateway services only. Never deploy anything directly to the gateway subnet (such as VMs or other services).

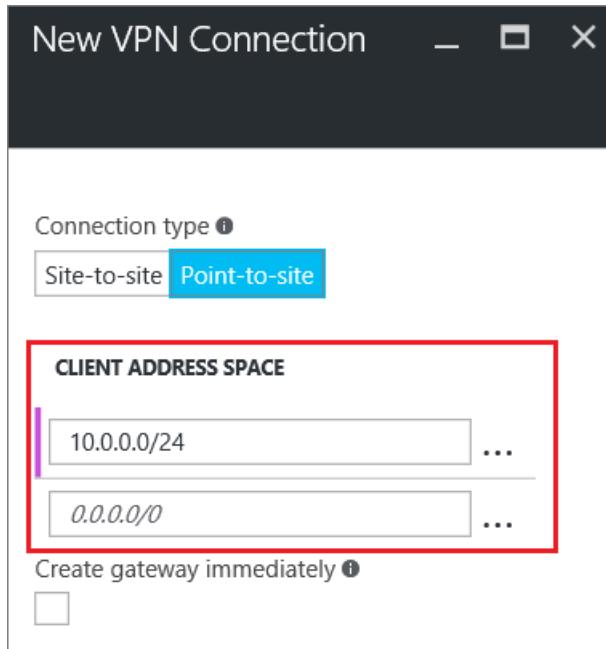
1. In the portal, navigate to the virtual network for which you want to create a gateway.
2. On the page for your virtual network, on the **Overview** page, in the VPN connections section, click **Gateway**.



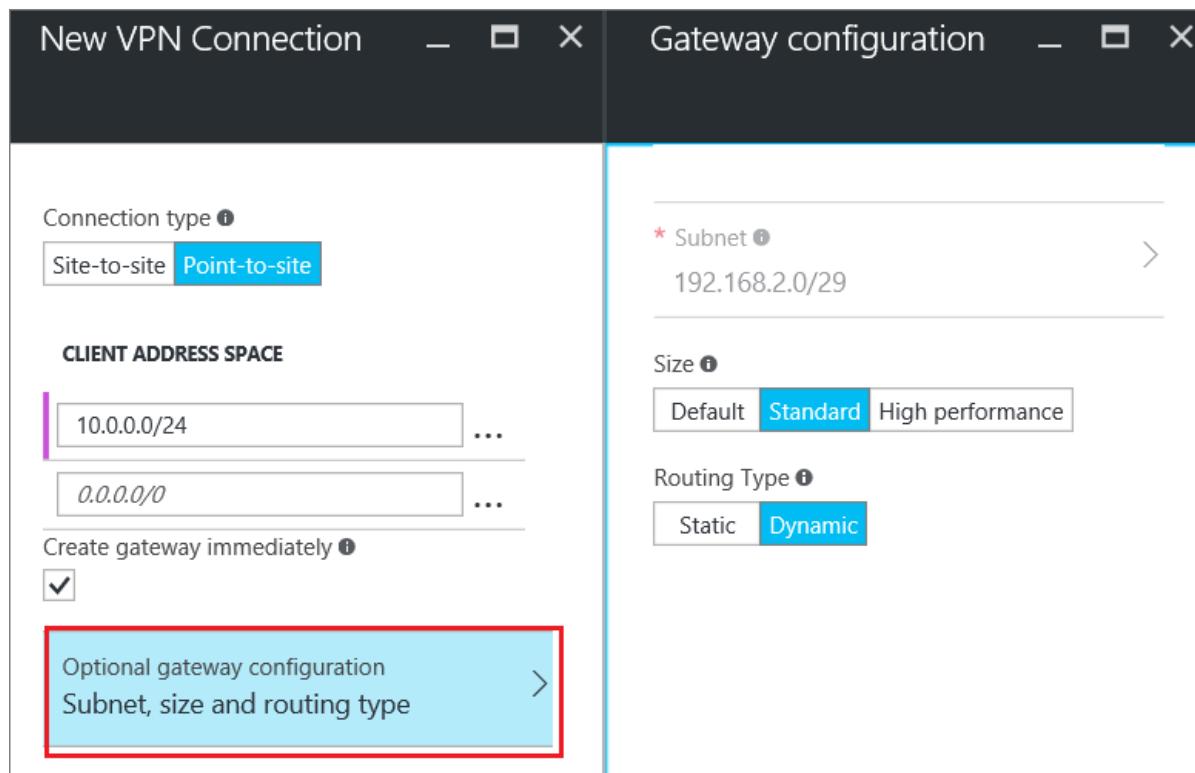
3. On the **New VPN Connection** page, select **Point-to-site**.



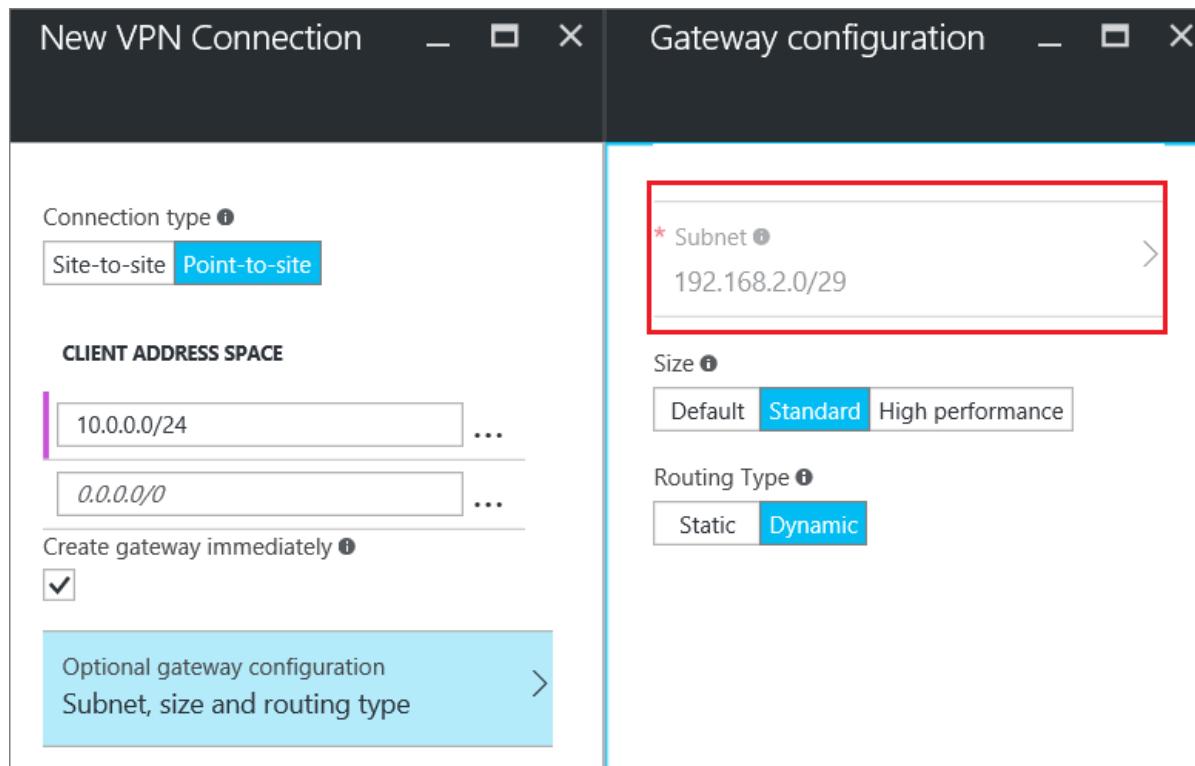
4. For **Client Address Space**, add the IP address range. This is the range from which the VPN clients receive an IP address when connecting. Use a private IP address range that does not overlap with the on-premises location that you will connect from, or with the VNet that you want to connect to. You can delete the auto-filled range, then add the private IP address range that you want to use. This example shows the auto-filled ranged. Delete it to add the value that you want.



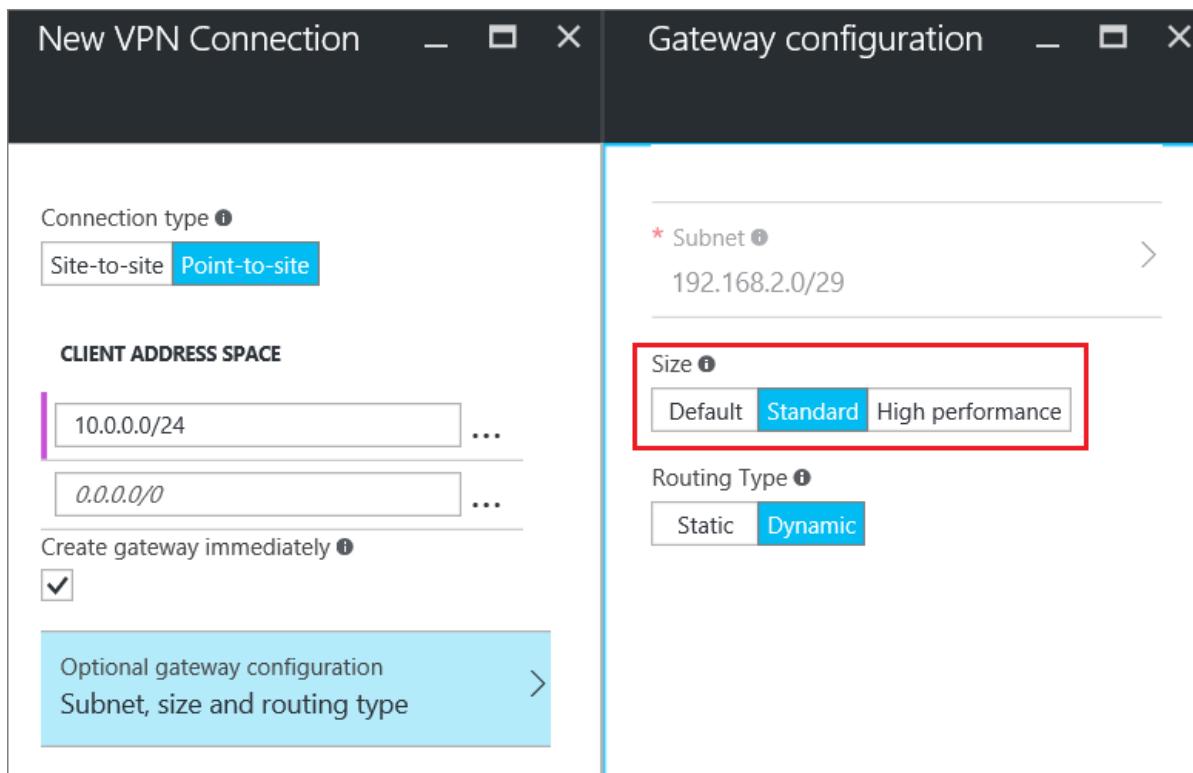
5. Select the **Create gateway immediately** checkbox. Click **Optional gateway configuration** to open the **Gateway configuration** page.



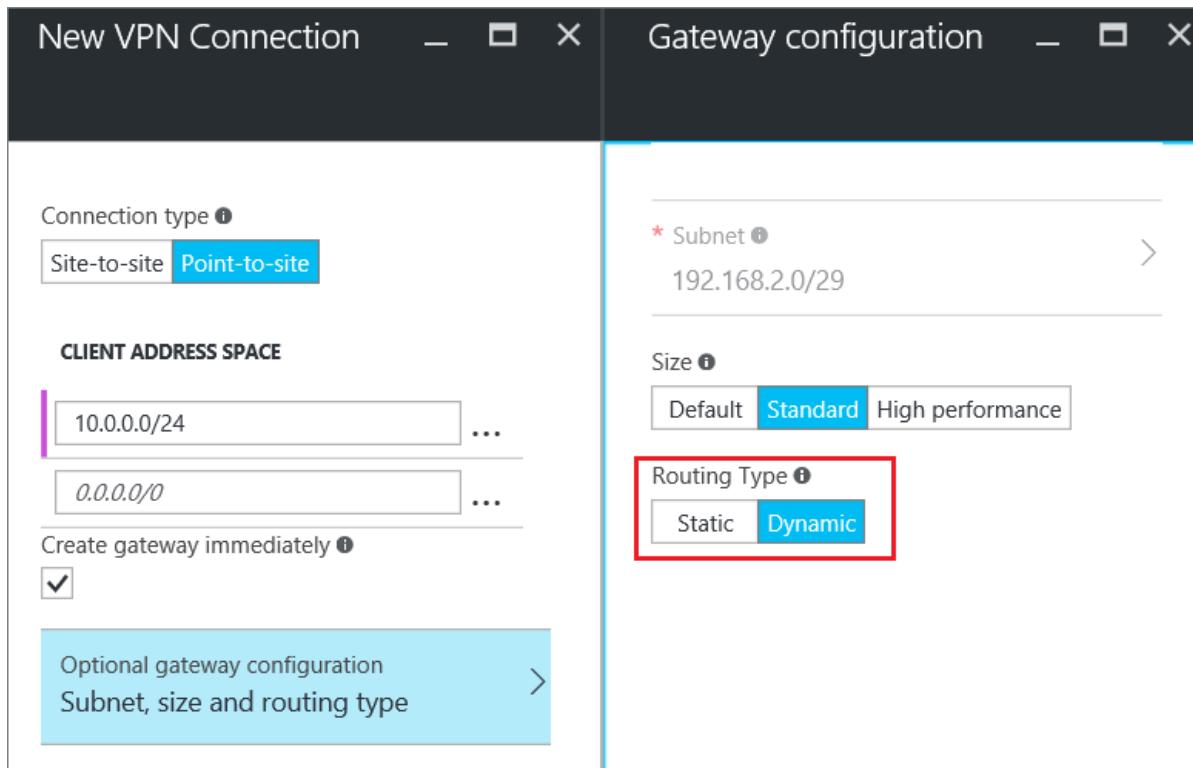
6. Click **Subnet Configure required settings** to add the **gateway subnet**. While it is possible to create a gateway subnet as small as /29, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. This will allow for enough addresses to accommodate possible additional configurations that you may want in the future. 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 VPN gateway to stop functioning as expected.



7. Select the gateway **Size**. The size is the gateway SKU for your virtual network gateway. In the portal, the Default SKU is **Basic**. For more information about gateway SKUs, see [About VPN Gateway Settings](#).



8. Select the **Routing Type** for your gateway. P2S configurations require a **Dynamic** routing type. Click **OK** when you have finished configuring this page.



9. On the **New VPN Connection** page, click **OK** at the bottom of the page to begin creating your virtual network gateway. A VPN gateway can take up to 45 minutes to complete, depending on the gateway sku that you select.

2. Create certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You upload the public key information of the root certificate to Azure. The public key is then considered 'trusted'. Client certificates must be generated from the trusted root certificate, and then installed on each client computer in the Certificates-Current User/Personal certificate store. The certificate is used to authenticate the client when it initiates a connection to the

VNet.

If you use self-signed certificates, they must be created using specific parameters. You can create a self-signed certificate using the instructions for [PowerShell and Windows 10](#), or [MakeCert](#). It's important that you follow the steps in these instructions when working with self-signed root certificates and generating client certificates from the self-signed root certificate. Otherwise, the certificates you create will not be compatible with P2S connections and you will receive a connection error.

Part 1: Obtain the public key (.cer) for the root certificate

You can use either a root certificate that was generated using an enterprise solution (recommended), or you can generate a self-signed certificate. After creating the root certificate, export the public certificate data (not the private key) as a Base-64 encoded X.509 .cer file and upload the public certificate data to Azure.

- **Enterprise certificate:** If you are using an enterprise solution, you can use your existing certificate chain. Obtain the .cer file for the root certificate that you want to use.
- **Self-signed root certificate:** If you aren't using an enterprise certificate solution, you need to create a self-signed root certificate. It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the certificates you create won't be compatible with P2S connections and clients receive a connection error when trying to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the provided articles generate a compatible certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
 - [Linux instructions](#)

Part 2: Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. The client certificate is generated from the root certificate and installed on each client computer. If a valid client certificate is not installed and the client tries to connect to the VNet, authentication fails.

You can either generate a unique certificate for each client, or you can use the same certificate for multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate. Otherwise, if multiple clients are using the same client certificate and you need to revoke it, you have to generate and install new certificates for all the clients that use that certificate to authenticate.

You can generate client certificates using the following methods:

- **Enterprise certificate:**
 - If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
 - Make sure the client certificate is based on the 'User' certificate template that has 'Client Authentication' as the first item in the use list, rather than Smart Card Logon, etc. You can check the certificate by double-clicking the client certificate and viewing **Details > Enhanced Key Usage**.
- **Self-signed root certificate:** It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the client certificates you create won't be compatible with P2S connections and clients receive an error when trying to connect. The steps in either of the following articles generate a compatible client certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to

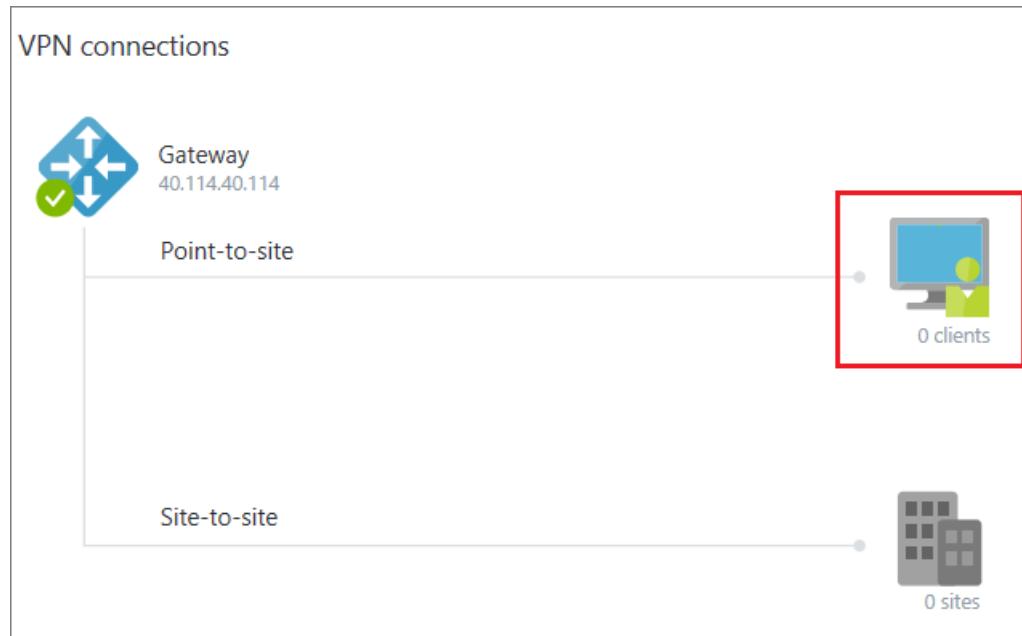
- generate certificates. The certificates that are generated can be installed on any supported P2S client.
- [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. The certificates that are generated can be installed on any supported P2S client.
 - [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate using the preceding instructions, it's automatically installed on the computer that you used to generate it. If you want to install a client certificate on another client computer, you need to export it as a .pfx, along with the entire certificate chain. This creates a .pfx file that contains the root certificate information that is required for the client to successfully authenticate. For steps to export a certificate, see [Certificates - export a client certificate](#).

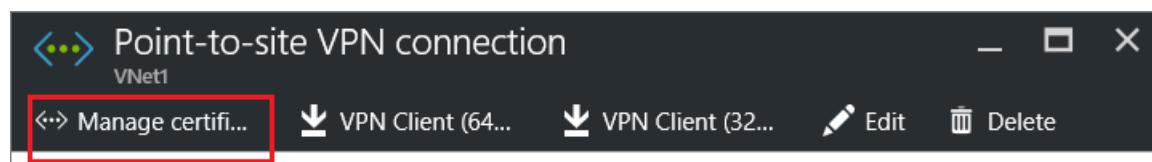
3. Upload the root certificate .cer file

After the gateway has been created, you can upload the .cer file (which contains the public key information) for a trusted root certificate to Azure. You do not upload the private key for the root certificate to Azure. Once a.cer file is uploaded, Azure can use it to authenticate clients that have installed a client certificate generated from the trusted root certificate. You can upload additional trusted root certificate files - up to a total of 20 - later, if needed.

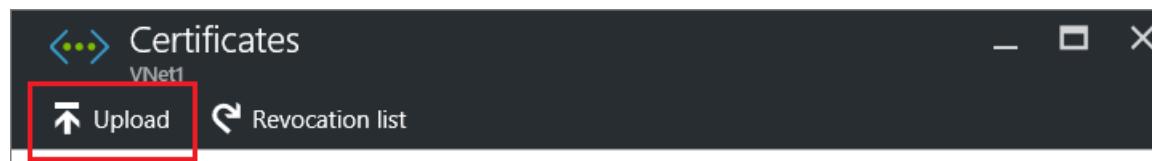
1. On the **VPN connections** section of the page for your VNet, click the **clients** graphic to open the **Point-to-site VPN connection** page.



2. On the **Point-to-site connection** page, click **Manage certificates** to open the **Certificates** page.

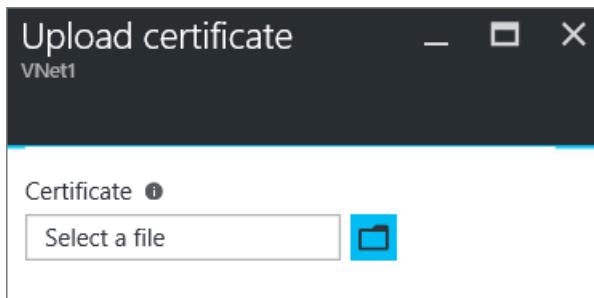


3. On the **Certificates** page, click **Upload** to open the **Upload certificate** page.



4. Click the folder graphic to browse for the .cer file. Select the file, then click **OK**. Refresh the page to see the

uploaded certificate on the **Certificates** page.



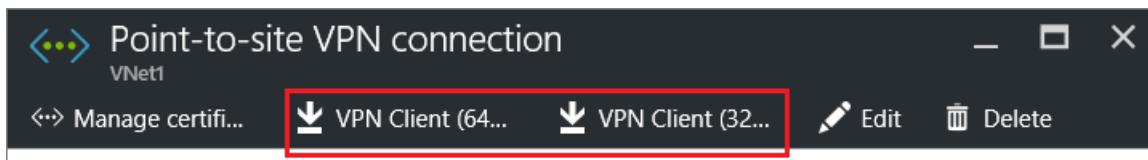
4. Configure the client

To connect to a VNet using a Point-to-Site VPN, each client must install a package to configure the native Windows VPN client. The configuration package configures the native Windows VPN client with the settings necessary to connect to the virtual network.

You can use the same VPN client configuration package on each client computer, as long as the version matches the architecture for the client. For the list of client operating systems that are supported, see the [Point-to-Site connections FAQ](#) at the end of this article.

Part 1: Generate and install the VPN client configuration package

1. In the Azure portal, in the **Overview** page for your VNet, in **VPN connections**, click the client graphic to open the **Point-to-site VPN connection** page.
2. At the top of the **Point-to-site VPN connection** page, click the download package that corresponds to the client operating system on which it will be installed:
 - For 64-bit clients, select **VPN Client (64-bit)**.
 - For 32-bit clients, select **VPN Client (32-bit)**.



3. Once the package generates, download and install it on your client computer. If you see a SmartScreen popup, click **More info**, then **Run anyway**. You can also save the package to install on other client computers.

Part 2: Install the client certificate

If you want to create a P2S connection from a client computer other than the one you used to generate the client certificates, you need to install a client certificate. When installing a client certificate, you need the password that was created when the client certificate was exported. Typically, this is just a matter of double-clicking the certificate and installing it. For more information, see [Install an exported client certificate](#).

5. Connect to Azure

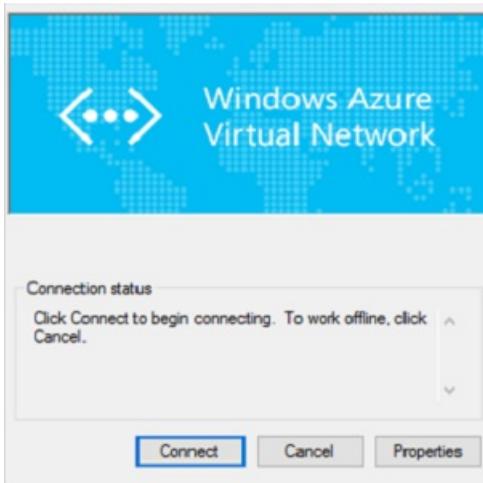
Connect to your VNet

NOTE

You must have Administrator rights on the client computer from which you are connecting.

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.

2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection is established.



Troubleshooting P2S connections

If you are having trouble connecting, check the following items:

- If you exported a client certificate, make sure that you exported it as a .pfx file using the default value 'Include all certificates in the certification path if possible'. When you export it using this value, the root certificate information is also exported. When the certificate is installed on the client computer, the root certificate which is contained in the .pfx file is then also installed on the client computer. The client computer must have the root certificate information installed. To check, go to **Manage user certificates** and navigate to **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed. The root certificate must be present in order for authentication to work.
- If you are using a certificate that was issued using an Enterprise CA solution and are having trouble authenticating, check the authentication order on the client certificate. You can check the authentication list order by double-clicking the client certificate, and going to **Details > Enhanced Key Usage**. Make sure the list shows 'Client Authentication' as the first item. If not, you need to issue a client certificate based on the User template that has Client Authentication as the first item in the list.
- For additional P2S troubleshooting information, see [Troubleshoot P2S connections](#).

Verify the VPN connection

1. To verify that your VPN connection is active, from your client computer, open an elevated command prompt, and run *ipconfig/all*.
2. View the results. Notice that the IP address you received is one of the addresses within the Point-to-Site connectivity address range that you specified when you created your VNet. The results should be similar to this example:

```
PPP adapter VNet1:  
  Connection-specific DNS Suffix .:  
  Description....................: VNet1  
  Physical Address.....:  
  DHCP Enabled.....: No  
  Autoconfiguration Enabled.....: Yes  
  IPv4 Address.....: 192.168.130.2(Preferred)  
  Subnet Mask.....: 255.255.255.255  
  Default Gateway.....:  
  NetBIOS over Tcpip.....: Enabled
```

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 for your VM. 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.
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.

To troubleshoot an RDP connection to a VM

If you are having trouble connecting to a virtual machine over your VPN connection, there are a few things you can check. For more troubleshooting information, see [Troubleshoot Remote Desktop connections to a VM](#).

- Verify that your VPN connection is successful.
- Verify that you are connecting to the private IP address for the VM.
- Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VPNClientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
- 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](#).
- Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.

Add or remove trusted root certificates

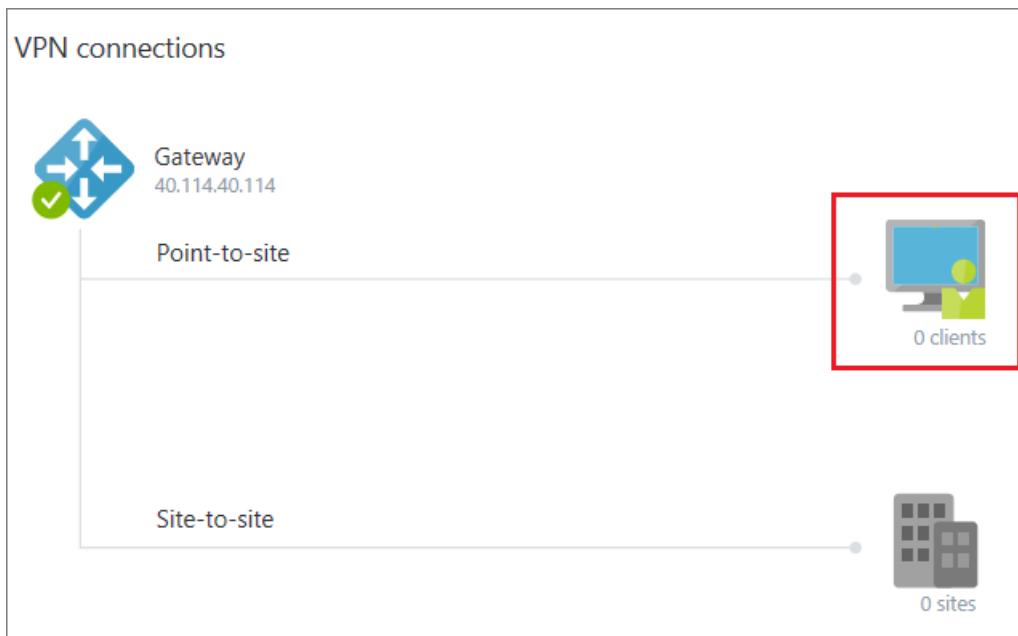
You can add and remove trusted root certificates from Azure. When you remove a root certificate, clients that have a certificate generated from that root won't be able to authenticate, and thus will not be able to connect. If you want a client to authenticate and connect, you need to install a new client certificate generated from a root certificate that is trusted (uploaded) to Azure.

To add a trusted root certificate

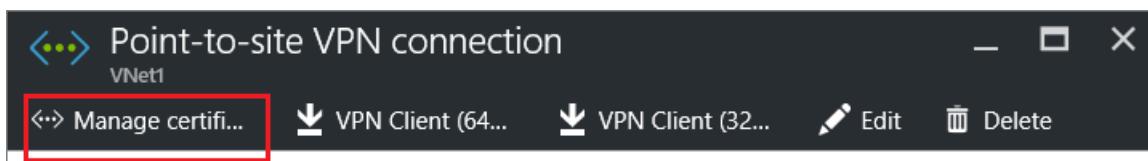
You can add up to 20 trusted root certificate .cer files to Azure. For instructions, see [Section 3 - Upload the root certificate .cer file](#).

To remove a trusted root certificate

1. On the **VPN connections** section of the page for your VNet, click the **clients** graphic to open the **Point-to-site VPN connection** page.



2. On the **Point-to-site connection** page, click **Manage certificates** to open the **Certificates** page.



3. On the **Certificates** page, click the ellipsis next to the certificate that you want to remove, then click **Delete**.

NAME	STATUS	EXPIRATION	THUMBPRINT	
CN=ARMP2SRootCert	Ok	12/31/2039 11:59...	68013DCE85AF8424ECD1B583...	...

Revoke a client certificate

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. This differs from removing a trusted root certificate. If you remove a trusted root certificate .cer from Azure, it revokes the access for all client certificates generated-signed by the revoked root certificate. Revoking a client certificate, rather than the root certificate, allows the other certificates that were generated from the root certificate to continue to be used for authentication for the Point-to-Site connection.

The common practice is to use the root certificate to manage access at team or organization levels, while using revoked client certificates for fine-grained access control on individual users.

To revoke a client certificate

You can revoke a client certificate by adding the thumbprint to the revocation list.

1. Retrieve the client certificate thumbprint. For more information, see [How to: Retrieve the Thumbprint of a](#)

[Certificate](#).

2. Copy the information to a text editor and remove all spaces so that it is a continuous string.
3. Navigate to the '**classic virtual network name**' > **Point-to-site VPN connection** > **Certificates** page and then click **Revocation list** to open the Revocation list page.
4. On the **Revocation list** page, click **+Add certificate** to open the **Add certificate to revocation list** page.
5. On the **Add certificate to revocation list** page, paste the certificate thumbprint as one continuous line of text, with no spaces. Click **OK** at the bottom of the page.
6. After updating has completed, the certificate can no longer be used to connect. Clients that try to connect using this certificate receive a message saying that the certificate is no longer valid.

Point-to-Site FAQ

This FAQ applies to P2S connections using the classic deployment model.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8 (32-bit and 64-bit)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows 10

Can I use any software VPN client for Point-to-Site that supports SSTP?

No. Support is limited only to the Windows operating system versions listed above.

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

Can I traverse proxies and firewalls using Point-to-Site capability?

Yes. We use SSTP (Secure Socket Tunneling Protocol) to tunnel through firewalls. This tunnel will appear as an HTTPS connection.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. Both these solutions will work if you have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or gateways using the `-VpnType PolicyBased` cmdlet.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

Yes, it is possible. But the virtual networks cannot have overlapping IP prefixes and the Point-to-Site address spaces must not overlap between the virtual networks.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet.

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#). To understand more about networking and virtual machines, see [Azure and Linux VM network overview](#).

For P2S troubleshooting information, [Troubleshoot Azure point-to-site connections](#).

Configure a VNet-to-VNet connection (classic)

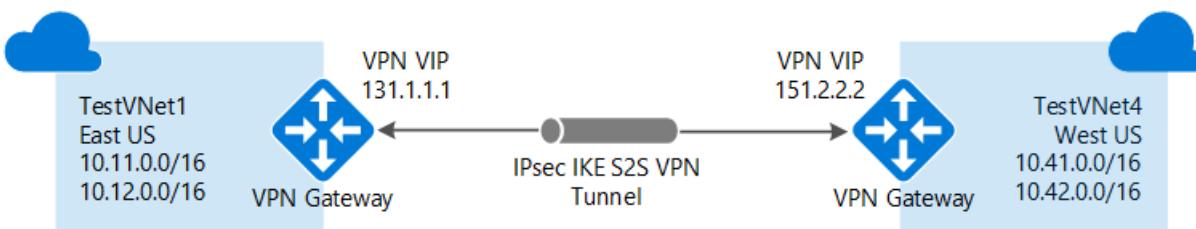
4/18/2018 • 12 minutes to read • [Edit Online](#)

NOTE

This article is written for the classic deployment model. If you're new to Azure, we recommend that you use the Resource Manager deployment model instead. The Resource Manager deployment model is the most current deployment model and offers more options and feature compatibility than the classic deployment model. For more information about the deployment models, see [Understanding deployment models](#).

For the Resource Manager version of this article, select it from the drop-down list below, or from the table of contents on the left.

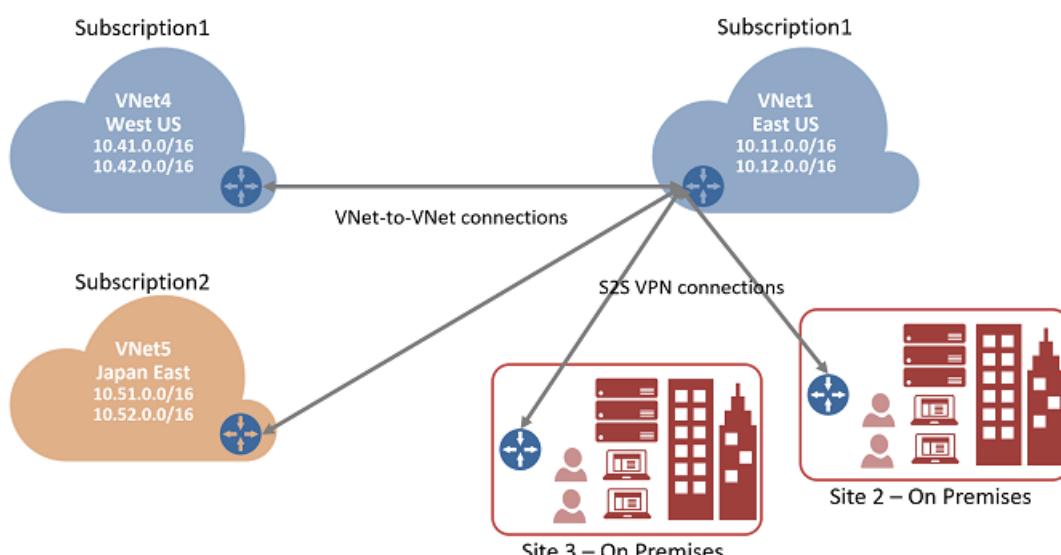
This article helps you create a VPN gateway connection between virtual networks. The virtual networks can be in the same or different regions, and from the same or different subscriptions. The steps in this article apply to the classic deployment model and the Azure portal. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:



About VNet-to-VNet connections

Connecting a virtual network to another virtual network (VNet-to-VNet) in the classic deployment model using a VPN gateway 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 using IPsec/IKE.

The VNets you connect can be in different subscriptions and different regions. You can combine VNet to VNet communication with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.



Why connect virtual networks?

You may want to connect virtual networks 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 Load Balancer and Microsoft or third-party clustering technology, 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 strong isolation boundary**

- Within the same region, you can set up multi-tier applications with multiple VNets connected together with strong isolation and secure inter-tier communication.

- **Cross subscription, inter-organization communication in Azure**

- If you have multiple Azure subscriptions, you can connect workloads from different subscriptions together securely between virtual networks.
- For enterprises or service providers, you can enable cross-organization communication with secure VPN technology within Azure.

For more information about VNet-to-VNet connections, see [VNet-to-VNet considerations](#) at the end of this article.

Before you begin

Before beginning this exercise, download and install the latest version of the Azure Service Management (SM) PowerShell cmdlets. For more information, see [How to install and configure Azure PowerShell](#). We use the portal for most of the steps, but you must use PowerShell to create the connections between the VNets. You can't create the connections using the Azure portal.

Step 1 - Plan your IP address ranges

It's important to decide the ranges that you'll use to configure your virtual networks. For this configuration, you must 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 to define your VNets. Use the ranges as a guideline only. Write down the ranges for your virtual networks. You need this information for later steps.

Example

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE
TestVNet1	TestVNet1 (10.11.0.0/16) (10.12.0.0/16)	East US	VNet4Local (10.41.0.0/16) (10.42.0.0/16)
TestVNet4	TestVNet4 (10.41.0.0/16) (10.42.0.0/16)	West US	VNet1Local (10.11.0.0/16) (10.12.0.0/16)

Step 2 - Create the virtual networks

Create two virtual networks in the [Azure portal](#). For the steps to create classic virtual networks, see [Create a classic virtual network](#).

When using the portal to create a classic virtual network, you must navigate to the virtual network page by using the following steps, otherwise the option to create a classic virtual network does not appear:

1. Click the '+' to open the 'New' page.
2. 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.
3. Locate 'Virtual Network' from the returned list and click it to open the Virtual Network page.
4. On the virtual network page, select 'Classic' to create a classic VNet.

If you are using this article as an exercise, you can use the following example values:

Values for TestVNet1

Name: TestVNet1
Address space: 10.11.0.0/16, 10.12.0.0/16 (optional)
Subnet name: default
Subnet address range: 10.11.0.1/24
Resource group: ClassicRG
Location: East US
GatewaySubnet: 10.11.1.0/27

Values for TestVNet4

Name: TestVNet4
Address space: 10.41.0.0/16, 10.42.0.0/16 (optional)
Subnet name: default
Subnet address range: 10.41.0.1/24
Resource group: ClassicRG
Location: West US
GatewaySubnet: 10.41.1.0/27

When creating your VNets, keep in mind the following settings:

- **Virtual Network Address Spaces** – On the Virtual Network Address Spaces page, specify the address range that you want to use for your virtual network. These are the dynamic IP addresses that will be assigned to the VMs and other role instances that you deploy to this virtual network.
The address spaces you select cannot overlap with the address spaces for any of the other VNets or on-premises locations that this VNet will connect to.
- **Location** – When you create a virtual network, you associate it with an Azure location (region). For example, if you want your VMs that are deployed to your virtual network to be physically located in West US, select that location. You can't change the location associated with your virtual network after you create it.

After creating your VNets, you can add the following settings:

- **Address space** – Additional address space is not required for this configuration, but you can add additional address space after creating the VNet.
- **Subnets** – Additional subnets are not required for this configuration, but you might want to have your VMs in a subnet that is separate from your other role instances.
- **DNS servers** – Enter the DNS server name and IP address. This setting does not create a DNS server. It allows you to specify the DNS servers that you want to use for name resolution for this virtual network.

In this section, you configure the connection type, the local site, and create the gateway.

Step 3 - Configure the local site

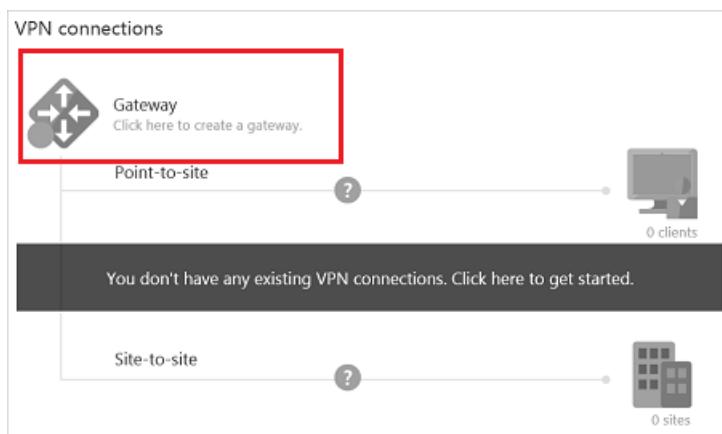
Azure uses the settings specified in each local network site to determine how to route traffic between the VNets. Each VNet must point to the respective local network that you want to route traffic to. You determine the name you want to use to refer to each local network site. It's best to use something descriptive.

For example, TestVNet1 connects to a local network site that you create named 'VNet4Local'. The settings for VNet4Local contain the address prefixes for TestVNet4.

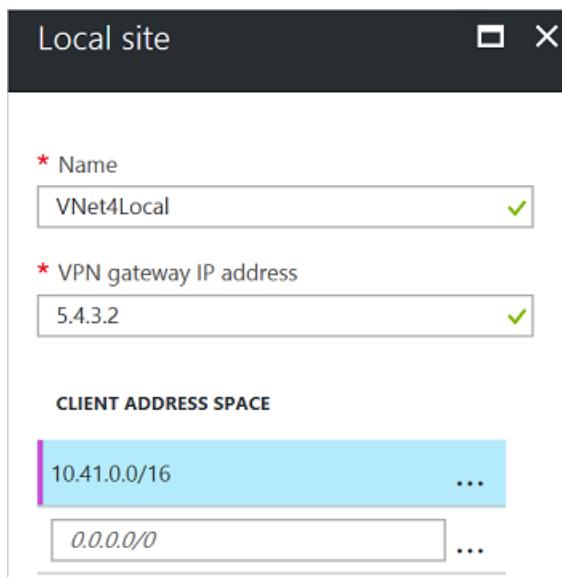
The local site for each VNet is the other VNet. The following example values are used for our configuration:

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE
TestVNet1	TestVNet1 (10.11.0.0/16) (10.12.0.0/16)	East US	VNet4Local (10.41.0.0/16) (10.42.0.0/16)
TestVNet4	TestVNet4 (10.41.0.0/16) (10.42.0.0/16)	West US	VNet1Local (10.11.0.0/16) (10.12.0.0/16)

1. Locate TestVNet1 in the Azure portal. In the **VPN connections** section of the page, click **Gateway**.



2. On the **New VPN Connection** page, select **Site-to-Site**.
3. Click **Local site** to open the Local site page and configure the settings.
4. On the **Local site** page, name your local site. In our example, we name the local site 'VNet4Local'.
5. For **VPN gateway IP address**, you can use any IP address that you want, as long as it's in a valid format. Typically, you'd use the actual external IP address for a VPN device. But, for a classic VNet-to-VNet configuration, you use the public IP address that is assigned to the gateway for your VNet. Given that you've not yet created the virtual network gateway, you specify any valid public IP address as a placeholder. Don't leave this blank - it's not optional for this configuration. In a later step, you go back into these settings and configure them with the corresponding virtual network gateway IP addresses once Azure generates it.
6. For **Client Address Space**, use the address space of the other VNet. Refer to your planning example. Click **OK** to save your settings and return back to the **New VPN Connection** page.



Step 4 - Create the virtual network gateway

Each virtual network must have a virtual network gateway. The virtual network gateway routes and encrypts traffic.

1. On the **New VPN Connection** page, select the checkbox **Create gateway immediately**.
2. Click **Subnet, size and routing type**. On the **Gateway configuration** page, click **Subnet**.
3. The gateway subnet name is filled in automatically with the required name 'GatewaySubnet'. The **Address range** contains the IP addresses that are allocated to the VPN gateway services. Some configurations allow a gateway subnet of /29, but it's best to use a /28 or /27 to accommodate future configurations that may require more IP addresses for the gateway services. In our example settings, we use 10.11.1.0/27. Adjust the address space, then click **OK**.
4. Configure the **Gateway Size**. This setting refers to the **Gateway SKU**.
5. Configure the **Routing Type**. The routing type for this configuration must be **Dynamic**. You can't change the routing type later unless you tear down the gateway and create a new one.
6. Click **OK**.
7. On the **New VPN Connection** page, click **OK** to begin creating the virtual network gateway. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

Step 5 - Configure TestVNet4 settings

Repeat the steps to [Create a local site](#) and [Create the virtual network gateway](#) to configure TestVNet4, substituting the values when necessary. If you are doing this as an exercise, use the [Example values](#).

Step 6 - Update the local sites

After your virtual network gateways have been created for both VNets, you must adjust the local sites **VPN gateway IP address** values.

VNET NAME	CONNECTED SITE	GATEWAY IP ADDRESS
TestVNet1	VNet4Local	VPN gateway IP address for TestVNet4
TestVNet4	VNet1Local	VPN gateway IP address for TestVNet1

Part 1 - Get the virtual network gateway public IP address

1. Locate your virtual network in the Azure portal.

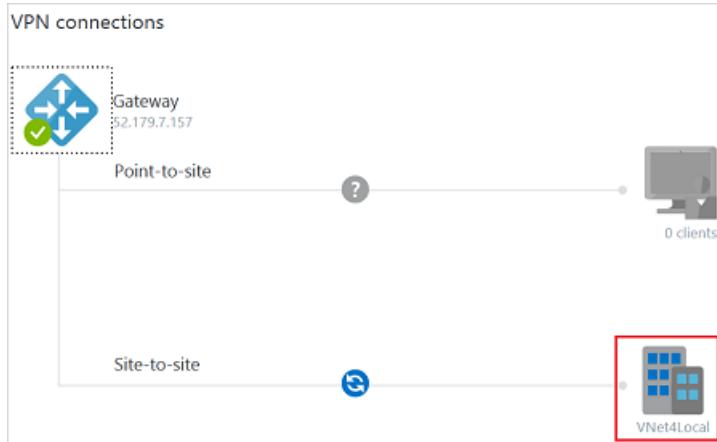
2. Click to open the VNet **Overview** page. On the page, in **VPN connections**, you can view the IP address for your virtual network gateway.



3. Copy the IP address. You will use it in the next section.
4. Repeat these steps for TestVNet4

Part 2 - Modify the local sites

1. Locate your virtual network in the Azure portal.
2. On the VNet **Overview** page, click the local site.

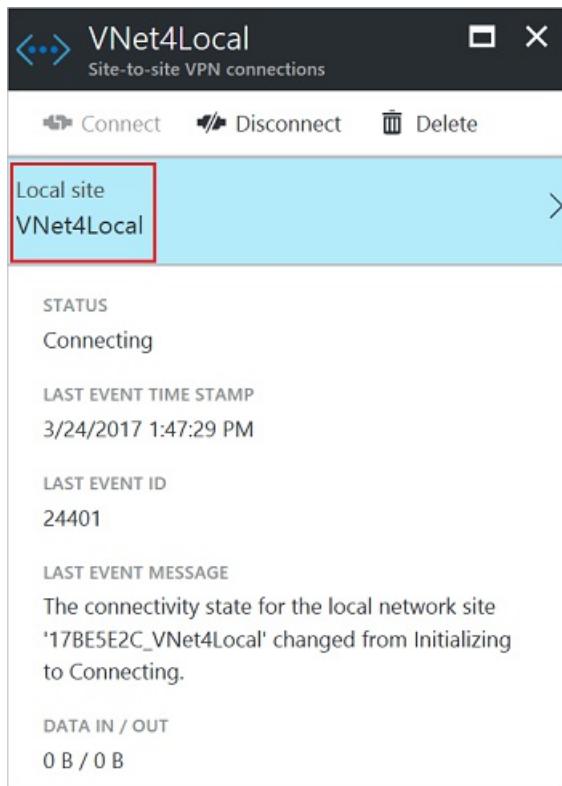


3. On the **Site-to-Site VPN Connections** page, click the name of the local site that you want to modify.

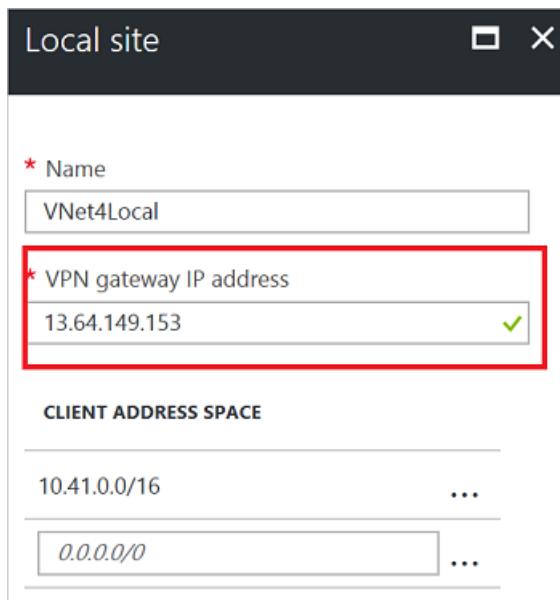
A screenshot of the 'Site-to-site VPN connections' page for the 'TestVNet4' network. The page displays a table of local sites. The first row, 'VNet1Local', has its 'NAME' column highlighted with a red box. The table includes columns for NAME, STATUS, LAST EVENT, and DATA IN / OUT. The 'VNet1Local' entry shows a status of 'Connecting' and a last event timestamp of 3/24/2017 1:59:08 PM.

NAME	STATUS	LAST EVENT	DATA IN / OUT
VNet1Local	Connecting	3/24/2017 1:59:08 PM	0 B / 0 B

4. Click the **Local site** that you want to modify.



5. Update the **VPN gateway IP address** and click **OK** to save the settings.



6. Close the other pages.
7. Repeat these steps for TestVNet4.

Step 7 - Retrieve values from the network configuration file

When you create classic VNets in the Azure portal, the name that you view is not the full name that you use for PowerShell. For example, a VNet that appears to be named **TestVNet1** in the portal, may have a much longer name in the network configuration file. The name might look something like: **Group ClassicRG TestVNet1**. When you create your connections, it's important to use the values that you see in the network configuration file.

In the following steps, you will connect to your Azure account and download and view the network configuration file to obtain the values that are required for your connections.

1. Download and install the latest version of the Azure Service Management (SM) PowerShell cmdlets. For more information, see [How to install and configure Azure PowerShell](#).

2. Open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect:

```
Connect-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

If you have more than one subscription, select the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Next, use the following cmdlet to add your Azure subscription to PowerShell for the classic deployment model.

```
Add-AzureAccount
```

3. Export and view the network configuration file. 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
```

4. Open the file with a text editor and view the names for your VNets and sites. These will be the name you use when you create your connections.

VNet names are listed as **VirtualNetworkSite name =**

Site names are listed as **LocalNetworkSiteRef name =**

Step 8 - Create the VPN gateway connections

When all the previous steps have been completed, you can set the IPsec/IKE pre-shared keys and create the connection. This set of steps uses PowerShell. VNet-to-VNet connections for the classic deployment model cannot be configured in the Azure portal.

In the examples, notice that the shared key is exactly the same. The shared key must always match. Be sure to replace the values in these examples with the exact names for your VNets and Local Network Sites.

1. Create the TestVNet1 to TestVNet4 connection.

```
Set-AzureVNetGatewayKey -VNetName 'Group ClassicRG TestVNet1' `  
-LocalNetworkSiteName '17BE5E2C_VNet4Local' -SharedKey A1b2C3D4
```

2. Create the TestVNet4 to TestVNet1 connection.

```
Set-AzureVNetGatewayKey -VNetName 'Group ClassicRG TestVNet4' `  
-LocalNetworkSiteName 'F7F7BFC7_VNet1Local' -SharedKey A1b2C3D4
```

3. Wait for the connections to initialize. Once the gateway has initialized, the Status is 'Successful'.

```
Error      : 
HttpStatusCode : OK
Id          : 
Status      : Successful
RequestId   : 
StatusCode  : OK
```

VNet-to-VNet considerations for classic VNets

- The virtual networks can be in the same or different subscriptions.
- The virtual networks can be in the same or different Azure regions (locations).
- A cloud service or a load balancing endpoint can't span across virtual networks, even if they are connected together.
- Connecting multiple virtual networks together doesn't require any VPN devices.
- VNet-to-VNet supports connecting Azure Virtual Networks. It does not support connecting virtual machines or cloud services that are not deployed to a virtual network.
- VNet-to-VNet requires dynamic routing gateways. Azure static routing gateways are not supported.
- Virtual network connectivity can be used simultaneously with multi-site VPNs. There is a maximum of 10 VPN tunnels for a virtual network VPN gateway connecting to either other virtual networks, or on-premises sites.
- The address spaces of the virtual networks and on-premises local network sites must not overlap. Overlapping address spaces will cause the creation of virtual networks or uploading netcfg configuration files to fail.
- Redundant tunnels between a pair of virtual networks are not supported.
- All VPN tunnels for the VNet, including P2S VPNs, share the available bandwidth for the VPN gateway, and the same VPN gateway uptime SLA in Azure.
- VNet-to-VNet traffic travels across the Azure backbone.

Next steps

Verify your connections. See [Verify a VPN Gateway connection](#).

Configure forced tunneling using the classic deployment model

8/2/2017 • 5 minutes to read • [Edit Online](#)

Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies. Without forced tunneling, Internet-bound traffic from your VMs in Azure will always traverse from Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic. Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches.

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

This article walks you through configuring forced tunneling for virtual networks created using the classic deployment model. Forced tunneling can be configured by using PowerShell, not through the portal. If you want to configure forced tunneling for the Resource Manager deployment model, select classic article from the following dropdown list:

Requirements and considerations

Forced tunneling in Azure is configured via virtual network user-defined routes (UDR). Redirecting traffic to an on-premises site is expressed as a Default Route to the Azure VPN gateway. The following section lists the current limitation of the routing table and routes for an Azure Virtual Network:

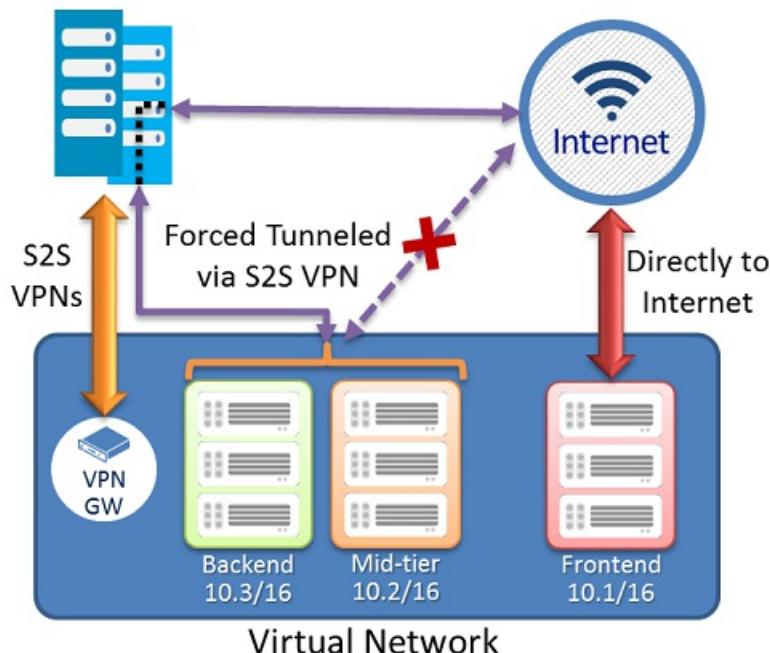
- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network.
 - **On-premises routes:** To the Azure VPN gateway.
 - **Default route:** Directly to the Internet. Packets destined to the private IP addresses not covered by the previous two routes will be dropped.
- With the release of user-defined routes, you can create a routing table to add a default route, and then associate the routing table to your VNet subnet(s) to enable forced tunneling on those subnets.
- You need to set a "default site" among the cross-premises local sites connected to the virtual network.
- Forced tunneling must be associated with a VNet that has a dynamic routing VPN gateway (not a static gateway).
- ExpressRoute forced tunneling is not configured via this mechanism, but instead, is enabled by advertising a default route via the ExpressRoute BGP peering sessions. Please see the [ExpressRoute Documentation](#) for more information.

Configuration overview

In the following example, the Frontend subnet is not forced tunneled. The workloads in the Frontend subnet can continue to accept and respond to customer requests from the Internet directly. The Mid-tier and Backend subnets are forced tunneled. Any outbound connections from these two subnets to the Internet will be forced or redirected back to an on-premises site via one of the S2S VPN tunnels.

This allows you to restrict and inspect Internet access from your virtual machines or cloud services in Azure, while continuing to enable your multi-tier service architecture required. You also can apply forced tunneling to the entire virtual networks if there are no Internet-facing workloads in your virtual networks.

On Premises



Before you begin

Verify that you have the following items before beginning configuration.

- An Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- A configured virtual network.
- The latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Configure forced tunneling

The following procedure will help you specify forced tunneling for a virtual network. The configuration steps correspond to the VNet network configuration file.

```

<VirtualNetworkSite name="MultiTier-VNet" Location="North Europe">
  <AddressSpace>
    <AddressPrefix>10.1.0.0/16</AddressPrefix>
  </AddressSpace>
  <Subnets>
    <Subnet name="Frontend">
      <AddressPrefix>10.1.0.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Midtier">
      <AddressPrefix>10.1.1.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Backend">
      <AddressPrefix>10.1.2.0/23</AddressPrefix>
    </Subnet>
    <Subnet name="GatewaySubnet">
      <AddressPrefix>10.1.200.0/28</AddressPrefix>
    </Subnet>
  </Subnets>
  <Gateway>
    <ConnectionsToLocalNetwork>
      <LocalNetworkSiteRef name="DefaultSiteHQ">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch1">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch2">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch3">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
    </ConnectionsToLocalNetwork>
  </Gateway>
  </VirtualNetworkSite>
</VirtualNetworkSite>

```

In this example, the virtual network 'MultiTier-VNet' has three subnets: 'Frontend', 'Midtier', and 'Backend' subnets, with four cross premises connections: 'DefaultSiteHQ', and three Branches.

The steps will set the 'DefaultSiteHQ' as the default site connection for forced tunneling, and configure the Midtier and Backend subnets to use forced tunneling.

1. Create a routing table. Use the following cmdlet to create your route table.

```

New-AzureRouteTable -Name "MyRouteTable" -Label "Routing Table for Forced Tunneling" -Location "North Europe"

```

2. Add a default route to the routing table.

The following example adds a default route to the routing table created in Step 1. Note that the only route supported is the destination prefix of "0.0.0.0/0" to the "VPNGateway" NextHop.

```

Get-AzureRouteTable -Name "MyRouteTable" | Set-AzureRoute -RouteTable "MyRouteTable" -RouteName "DefaultRoute" -AddressPrefix "0.0.0.0/0" -NextHopType VPNGateway

```

3. Associate the routing table to the subnets.

After a routing table is created and a route added, use the following example to add or associate the route table to a VNet subnet. The example adds the route table "MyRouteTable" to the Midtier and Backend subnets of VNet MultiTier-VNet.

```
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Midtier" -RouteTableName "MyRouteTable"
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Backend" -RouteTableName "MyRouteTable"
```

4. Assign a default site for forced tunneling.

In the preceding step, the sample cmdlet scripts created the routing table and associated the route table to two of the VNet subnets. The remaining step is to select a local site among the multi-site connections of the virtual network as the default site or tunnel.

```
$DefaultSite = @("DefaultSiteHQ")
Set-AzureVNetGatewayDefaultSite -VNetName "MultiTier-VNet" -DefaultSite "DefaultSiteHQ"
```

Additional PowerShell cmdlets

To delete a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To list a route table

```
Get-AzureRouteTable [-Name <routeTableName> [-DetailLevel <detailLevel>]]
```

To delete a route from a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To remove a route from a subnet

```
Remove-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To list the route table associated with a subnet

```
Get-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To remove a default site from a VNet VPN gateway

```
Remove-AzureVNetGatewayDefaultSite -VNetName <virtualNetworkName>
```

Delete a virtual network gateway using PowerShell (classic)

7/13/2018 • 3 minutes to read • [Edit Online](#)

This article helps you delete a VPN gateway in the classic deployment model by using PowerShell. After the virtual network gateway has been deleted, modify the network configuration file to remove elements that you are no longer using.

Step 1: Connect to Azure

1. Install the latest PowerShell cmdlets.

Download and install the latest version of the Azure Service Management (SM) PowerShell cmdlets. For more information, see [How to install and configure Azure PowerShell](#).

2. Connect to your Azure account.

Open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

Step 2: Export and view the network configuration file

Create a directory on your computer and then export the network configuration file to the directory. You use this file to both view the current configuration information, and also to modify the network configuration.

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. When you create a VNet in the Azure portal, the full name that Azure uses is not visible in the portal. For example, a VNet that appears to be named 'ClassicVNet1' in the Azure portal, may have a much longer name in the network configuration file. The name might look something like: 'Group ClassicRG1 ClassicVNet1'. Virtual network names are listed as '**VirtualNetworkSite name =**'. Use the names in the network configuration file when running your PowerShell cmdlets.

Step 3: Delete the virtual network gateway

When you delete a virtual network gateway, all connections to the VNet through the gateway are disconnected. If you have P2S clients connected to the VNet, they will be disconnected without warning.

This example deletes the virtual network gateway. Make sure to use the full name of the virtual network from the network configuration file.

```
Remove-AzureVNetGateway -VNetName "Group ClassicRG1 ClassicVNet1"
```

If successful, the return shows:

Status : Successful

Step 4: Modify the network configuration file

When you delete a virtual network gateway, the cmdlet does not modify the network configuration file. You need to modify the file to remove the elements that are no longer being used. The following sections help you modify the network configuration file that you downloaded.

Local Network Site References

To remove site reference information, make configuration changes to

ConnectionsToLocalNetwork/LocalNetworkSiteRef. Removing a local site reference triggers Azure to delete a tunnel. Depending on the configuration that you created, you may not have a **LocalNetworkSiteRef** listed.

```
<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="D1BFC9CB_Site2">
      <Connection type="IPsec" />
    </LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>
```

Example:

```
<Gateway>
  <ConnectionsToLocalNetwork>
  </ConnectionsToLocalNetwork>
</Gateway>
```

Local Network Sites

Remove any local sites that you are no longer using. Depending on the configuration you created, it is possible that you don't have a **LocalNetworkSite** listed.

```
<LocalNetworkSites>
  <LocalNetworkSite name="Site1">
    <AddressSpace>
      <AddressPrefix>192.168.0.0/16</AddressPrefix>
    </AddressSpace>
    <VPNGatewayAddress>5.4.3.2</VPNGatewayAddress>
  </LocalNetworkSite>
  <LocalNetworkSite name="Site3">
    <AddressSpace>
      <AddressPrefix>192.168.0.0/16</AddressPrefix>
    </AddressSpace>
    <VPNGatewayAddress>57.179.18.164</VPNGatewayAddress>
  </LocalNetworkSite>
</LocalNetworkSites>
```

In this example, we removed only Site3.

```
<LocalNetworkSites>
  <LocalNetworkSite name="Site1">
    <AddressSpace>
      <AddressPrefix>192.168.0.0/16</AddressPrefix>
    </AddressSpace>
    <VPNGatewayAddress>5.4.3.2</VPNGatewayAddress>
  </LocalNetworkSite>
</LocalNetworkSites>
```

Client AddressPool

If you had a P2S connection to your VNet, you will have a **VPNClientAddressPool**. Remove the client address pools that correspond to the virtual network gateway that you deleted.

```
<Gateway>
  <VPNClientAddressPool>
    <AddressPrefix>10.1.0.0/24</AddressPrefix>
  </VPNClientAddressPool>
  <ConnectionsToLocalNetwork />
</Gateway>
```

Example:

```
<Gateway>
  <ConnectionsToLocalNetwork />
</Gateway>
```

GatewaySubnet

Delete the **GatewaySubnet** that corresponds to the VNet.

```
<Subnets>
  <Subnet name="FrontEnd">
    <AddressPrefix>10.11.0.0/24</AddressPrefix>
  </Subnet>
  <Subnet name="GatewaySubnet">
    <AddressPrefix>10.11.1.0/29</AddressPrefix>
  </Subnet>
</Subnets>
```

Example:

```
<Subnets>
  <Subnet name="FrontEnd">
    <AddressPrefix>10.11.0.0/24</AddressPrefix>
  </Subnet>
</Subnets>
```

Step 5: Upload the network configuration file

Save your changes and upload the network configuration file to Azure. Make sure you change the file path as necessary for your environment.

```
Set-AzureVNetConfig -ConfigurationPath C:\AzureNet\NetworkConfig.xml
```

If successful, the return shows something similar to this example:

OperationDescription	OperationId	OperationStatus
Set-AzureVNetConfig	e0ee6e66-9167-cfa7-a746-7casb9	Succeeded

Add a Site-to-Site connection to a VNet with an existing VPN gateway connection (classic)

2/16/2018 • 6 minutes to read • [Edit Online](#)

NOTE

This article is written for the classic deployment model. If you're new to Azure, we recommend that you use the Resource Manager deployment model instead. The Resource Manager deployment model is the most current deployment model and offers more options and feature compatibility than the classic deployment model. For more information about the deployment models, see [Understanding deployment models](#).

For the Resource Manager version of this article, select it from the drop-down list below, or from the table of contents on the left.

This article walks you through using PowerShell to add Site-to-Site (S2S) connections to a VPN gateway that has an existing connection. This type of connection is often referred to as a "multi-site" configuration. The steps in this article apply to virtual networks created using the classic deployment model (also known as Service Management). These steps do not apply to ExpressRoute/Site-to-Site coexisting connection configurations.

Deployment models and methods

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

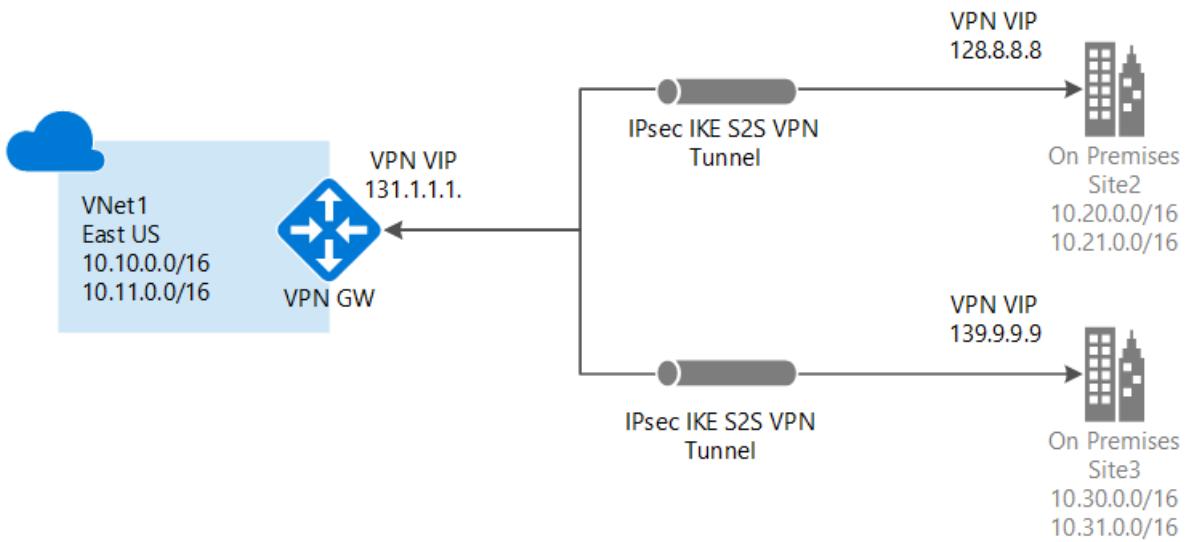
We update this table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from this table.

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Article	Supported
Classic	Not Supported	Article

About connecting

You can connect multiple on-premises sites to a single virtual network. This is especially attractive for building hybrid cloud solutions. Creating a multi-site connection to your Azure virtual network gateway is similar to creating other Site-to-Site connections. In fact, you can use an existing Azure VPN gateway, as long as the gateway is dynamic (route-based).

If you already have a static gateway connected to your virtual network, you can change the gateway type to dynamic without needing to rebuild the virtual network in order to accommodate multi-site. Before changing the routing type, make sure that your on-premises VPN gateway supports route-based VPN configurations.



Points to consider

You won't be able to use the portal to make changes to this virtual network. You need to make changes to the network configuration file instead of using the portal. If you make changes in the portal, they'll overwrite your multi-site reference settings for this virtual network.

You should feel comfortable using the network configuration file by the time you've completed the multi-site procedure. However, if you have multiple people working on your network configuration, you'll need to make sure that everyone knows about this limitation. This doesn't mean that you can't use the portal at all. You can use it for everything else, except making configuration changes to this particular virtual network.

Before you begin

Before you begin configuration, verify that you have the following:

- Compatible VPN hardware for each on-premises location. Check [About VPN Devices for Virtual Network Connectivity](#) to verify if the device that you want to use is something that is known to be compatible.
- An externally facing public IPv4 IP address for each VPN device. The IP address cannot be located behind a NAT. This is requirement.
- You'll need to install the latest version of the Azure PowerShell cmdlets. Make sure you install the Service Management (SM) version in addition to the Resource Manager version. See [How to install and configure Azure PowerShell](#) for more information.
- Someone who is proficient at configuring your VPN hardware. You'll have to have a strong understanding of how to configure your VPN device, or work with someone who does.
- The IP address ranges that you want to use for your virtual network (if you haven't already created one).
- The IP address ranges for each of the local network sites that you'll be connecting to. You'll need to make sure that the IP address ranges for each of the local network sites that you want to connect to do not overlap. Otherwise, the portal or the REST API will reject the configuration being uploaded.

For example, if you have two local network sites that both contain the IP address range 10.2.3.0/24 and you have a package with a destination address 10.2.3.3, Azure wouldn't know which site you want to send the package to because the address ranges are overlapping. To prevent routing issues, Azure doesn't allow you to upload a configuration file that has overlapping ranges.

1. Create a Site-to-Site VPN

If you already have a Site-to-Site VPN with a dynamic routing gateway, great! You can proceed to [Export the virtual network configuration settings](#). If not, do the following:

If you already have a Site-to-Site virtual network, but it has a static (policy-based) routing gateway:

1. Change your gateway type to dynamic routing. A multi-site VPN requires a dynamic (also known as route-based) routing gateway. To change your gateway type, you'll need to first delete the existing gateway, then create a new one.
2. Configure your new gateway and create your VPN tunnel. For instructions, see [Specify the SKU and VPN type](#). Make sure you specify the Routing Type as 'Dynamic'.

If you don't have a Site-to-Site virtual network:

1. Create your Site-to-Site virtual network using these instructions: [Create a Virtual Network with a Site-to-Site VPN Connection](#).
2. Configure a dynamic routing gateway using these instructions: [Configure a VPN Gateway](#). Be sure to select **dynamic routing** for your gateway type.

2. Export the network configuration file

Export your Azure network configuration file by running the following command. You can change the location of the file to export to a different location if necessary.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

3. Open the network configuration file

Open the network configuration file that you downloaded in the last step. Use any xml editor that you like. The file should look similar to the following:

```

<NetworkConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration">
    <VirtualNetworkConfiguration>
        <LocalNetworkSites>
            <LocalNetworkSite name="Site1">
                <AddressSpace>
                    <AddressPrefix>10.0.0.0/16</AddressPrefix>
                    <AddressPrefix>10.1.0.0/16</AddressPrefix>
                </AddressSpace>
                <VPNGatewayAddress>131.2.3.4</VPNGatewayAddress>
            </LocalNetworkSite>
            <LocalNetworkSite name="Site2">
                <AddressSpace>
                    <AddressPrefix>10.2.0.0/16</AddressPrefix>
                    <AddressPrefix>10.3.0.0/16</AddressPrefix>
                </AddressSpace>
                <VPNGatewayAddress>131.4.5.6</VPNGatewayAddress>
            </LocalNetworkSite>
        </LocalNetworkSites>
        <VirtualNetworkSites>
            <VirtualNetworkSite name="VNet1" AffinityGroup="USWest">
                <AddressSpace>
                    <AddressPrefix>10.20.0.0/16</AddressPrefix>
                    <AddressPrefix>10.21.0.0/16</AddressPrefix>
                </AddressSpace>
                <Subnets>
                    <Subnet name="FE">
                        <AddressPrefix>10.20.0.0/24</AddressPrefix>
                    </Subnet>
                    <Subnet name="BE">
                        <AddressPrefix>10.20.1.0/24</AddressPrefix>
                    </Subnet>
                    <Subnet name="GatewaySubnet">
                        <AddressPrefix>10.20.2.0/29</AddressPrefix>
                    </Subnet>
                </Subnets>
                <Gateway>
                    <ConnectionsToLocalNetwork>
                        <LocalNetworkSiteRef name="Site1">
                            <Connection type="IPsec" />
                        </LocalNetworkSiteRef>
                    </ConnectionsToLocalNetwork>
                </Gateway>
            </VirtualNetworkSite>
        </VirtualNetworkSites>
    </VirtualNetworkConfiguration>
</NetworkConfiguration>

```

4. Add multiple site references

When you add or remove site reference information, you'll make configuration changes to the `ConnectionsToLocalNetwork/LocalNetworkSiteRef`. Adding a new local site reference triggers Azure to create a new tunnel. In the example below, the network configuration is for a single-site connection. Save the file once you have finished making your changes.

```

<Gateway>
    <ConnectionsToLocalNetwork>
        <LocalNetworkSiteRef name="Site1"><Connection type="IPsec" /></LocalNetworkSiteRef>
    </ConnectionsToLocalNetwork>
</Gateway>

```

To add additional site references (create a multi-site configuration), simply add additional "LocalNetworkSiteRef"

lines, as shown in the example below:

```
<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="Site1"><Connection type="IPsec" /></LocalNetworkSiteRef>
    <LocalNetworkSiteRef name="Site2"><Connection type="IPsec" /></LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>
```

5. Import the network configuration file

Import the network configuration file. When you import this file with the changes, the new tunnels will be added. The tunnels will use the dynamic gateway that you created earlier. You can use PowerShell to import the file.

6. Download keys

Once your new tunnels have been added, use the PowerShell cmdlet 'Get-AzureVNetGatewayKey' to get the IPsec/IKE pre-shared keys for each tunnel.

For example:

```
Get-AzureVNetGatewayKey -VNetName "VNet1" -LocalNetworkSiteName "Site1"
Get-AzureVNetGatewayKey -VNetName "VNet1" -LocalNetworkSiteName "Site2"
```

If you prefer, you can also use the *Get Virtual Network Gateway Shared Key* REST API to get the pre-shared keys.

7. Verify your connections

Check the multi-site tunnel status. After downloading the keys for each tunnel, you'll want to verify connections. Use 'Get-AzureVnetConnection' to get a list of virtual network tunnels, as shown in the example below. VNet1 is the name of the VNet.

```
Get-AzureVnetConnection -VNetName VNET1
```

Example return:

```
ConnectivityState      : Connected
EgressBytesTransferred : 661530
IngressBytesTransferred : 519207
LastConnectionEstablished : 5/2/2014 2:51:40 PM
LastEventID           : 23401
LastEventMessage       : The connectivity state for the local network site 'Site1' changed from Not
Connected to Connected.
LastEventTimeStamp     : 5/2/2014 2:51:40 PM
LocalNetworkSiteName   : Site1
OperationDescription   : Get-AzureVNetConnection
OperationId            : 7f68a8e6-51e9-9db4-88c2-16b8067fed7f
OperationStatus         : Succeeded

ConnectivityState      : Connected
EgressBytesTransferred : 789398
IngressBytesTransferred : 143908
LastConnectionEstablished : 5/2/2014 3:20:40 PM
LastEventID           : 23401
LastEventMessage       : The connectivity state for the local network site 'Site2' changed from Not
Connected to Connected.
LastEventTimeStamp     : 5/2/2014 2:51:40 PM
LocalNetworkSiteName   : Site2
OperationDescription   : Get-AzureVNetConnection
OperationId            : 7893b329-51e9-9db4-88c2-16b8067fed7f
OperationStatus         : Succeeded
```

Next steps

To learn more about VPN Gateways, see [About VPN Gateways](#).

2 minutes to read

VPN Gateway classic to Resource Manager migration

6/27/2017 • 4 minutes to read • [Edit Online](#)

VPN Gateways can now be migrated from classic to Resource Manager deployment model. You can read more about Azure Resource Manager [features and benefits](#). In this article, we detail how to migrate from classic deployments to newer Resource Manager based model.

VPN Gateways are migrated as part of VNet migration from classic to Resource Manager. This migration is done one VNet at a time. There is no additional requirement in terms of tools or prerequisites to migration. Migration steps are identical to existing VNet migration and are documented at [IaaS resources migration page](#). There is no data path downtime during migration and thus existing workloads would continue to function without loss of on-premises connectivity during migration. The public IP address associated with the VPN gateway does not change during the migration process. This implies that you will not need to reconfigure your on-premises router once the migration is completed.

The model in Resource Manager is different from classic model and is composed of virtual network gateways, local network gateways and connection resources. These represent the VPN gateway itself, the local-site representing on-premises address space and connectivity between the two respectively. Once migration is completed your gateways would not be available in classic model and all management operations on virtual network gateways, local network gateways, and connection objects must be performed using Resource Manager model.

Supported scenarios

Most common VPN connectivity scenarios are covered by classic to Resource Manager migration. The supported scenarios include -

- Point to site connectivity
- Site to site connectivity with VPN Gateway connected to on-premises location
- VNet to VNet connectivity between two VNets using VPN gateways
- Multiple VNets connected to same on-premises location
- Multi-site connectivity
- Forced tunneling enabled VNets

Scenarios which are not supported include -

- VNet with both ExpressRoute Gateway and VPN Gateway is not currently supported.
- Transit scenarios where VM extensions are connected to on-premises servers. Transit VPN connectivity limitations are detailed below.

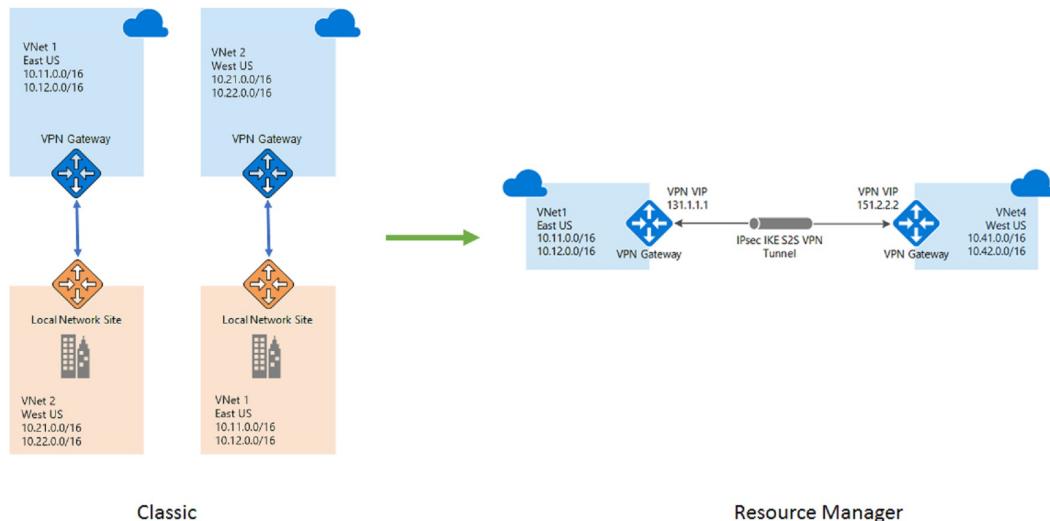
NOTE

CIDR validation in Resource Manager model is more strict than the one in classic model. Before migrating ensure that classic address ranges given conform to valid CIDR format before beginning the migration. CIDR can be validated using any common CIDR validators. VNet or local sites with invalid CIDR ranges when migrated would result in failed state.

VNet to VNet connectivity migration

VNet to VNet connectivity in classic was achieved by creating a local site representation of the connected VNet. Customers were required to create two local sites which represented the two VNets which needed to be connected together. These were then connected to the corresponding VNets using IPsec tunnel to establish connectivity.

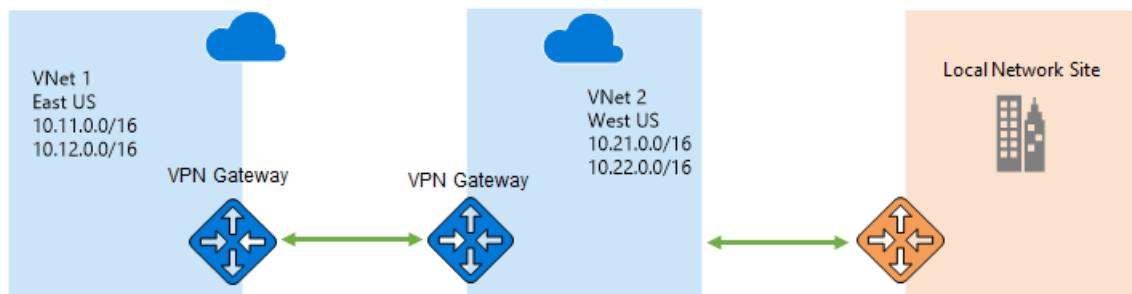
between the two VNets. This model has manageability challenges since any address range changes in one VNet must also be maintained in the corresponding local site representation. In Resource Manager model this workaround is no longer needed. The connection between the two VNets can be directly achieved using 'Vnet2Vnet' connection type in Connection resource.



During VNet migration we detect that the connected entity to current VNet's VPN gateway is another VNet and ensure that once migration of both VNets is completed, you would no longer see two local sites representing the other VNet. The classic model of two VPN gateways, two local sites and two connections between them is transformed to Resource Manager model with two VPN gateways and two connections of type Vnet2Vnet.

Transit VPN connectivity

You can configure VPN gateways in a topology such that on-premises connectivity for a VNet is achieved by connecting to another VNet that is directly connected to on-premises. This is transit VPN connectivity where instances in first VNet are connected to on-premises resources via transit to the VPN gateway in connected VNet that is directly connected to on-premises. To achieve this configuration in classic deployment model, you would need to create a local site which has aggregated prefixes representing both the connected VNet and on-premises address space. This representational local site is then connected to the VNet to achieve transit connectivity. This classic model also has similar manageability challenges since any change in on-premises address range must also be maintained on the local site representing the aggregate of VNet and on-premises. Introduction of BGP support in Resource Manager supported gateways simplifies manageability since the connected gateways can learn routes from on premises without manual modification to prefixes.



Since we transform VNet to VNet connectivity without requiring local sites, the transit scenario loses on-premises

connectivity for VNet that is indirectly connected to on-premises. The loss of connectivity can be mitigated in the following two ways, after migration is completed -

- Enable BGP on VPN gateways that are connected together and to on-premises. Enabling BGP restores connectivity without any other configuration change since routes are learned and advertised between VNet gateways. Note that BGP option is only available on Standard and higher SKUs.
- Establish an explicit connection from affected VNet to the local network gateway representing on-premises location. This would also require changing configuration on the on-premises router to create and configure the IPsec tunnel.

Next steps

After learning about VPN gateway migration support, go to [platform-supported migration of IaaS resources from classic to Resource Manager](#) to get started.

VPN Gateway FAQ

7/30/2018 • 36 minutes to read • [Edit Online](#)

Connecting to virtual networks

Can I connect virtual networks in different Azure regions?

Yes. In fact, there is no region constraint. One virtual network can connect to another virtual network in the same region, or in a different Azure region.

Can I connect virtual networks in different subscriptions?

Yes.

Can I connect to multiple sites from a single virtual network?

You can connect to multiple sites by using Windows PowerShell and the Azure REST APIs. See the [Multi-Site and VNet-to-VNet Connectivity](#) FAQ section.

What are my cross-premises connection options?

The following cross-premises connections are supported:

- Site-to-Site – VPN connection over IPsec (IKE v1 and IKE v2). This type of connection requires a VPN device or RRAS. For more information, see [Site-to-Site](#).
- Point-to-Site – VPN connection over SSTP (Secure Socket Tunneling Protocol) or IKE v2. This connection does not require a VPN device. For more information, see [Point-to-Site](#).
- VNet-to-VNet – This type of connection is the same as a Site-to-Site configuration. VNet to VNet is a VPN connection over IPsec (IKE v1 and IKE v2). It does not require a VPN device. For more information, see [VNet-to-VNet](#).
- Multi-Site – This is a variation of a Site-to-Site configuration that allows you to connect multiple on-premises sites to a virtual network. For more information, see [Multi-Site](#).
- ExpressRoute – ExpressRoute is a direct connection to Azure from your WAN, not a VPN connection over the public Internet. For more information, see the [ExpressRoute Technical Overview](#) and the [ExpressRoute FAQ](#).

For more information about VPN gateway connections, see [About VPN Gateway](#).

What is the difference between a Site-to-Site connection and Point-to-Site?

Site-to-Site (IPsec/IKE VPN tunnel) configurations are between your on-premises location and Azure. This means that you can connect from any of your computers located on your premises to any virtual machine or role instance within your virtual network, depending on how you choose to configure routing and permissions. It's a great option for an always-available cross-premises connection and is well-suited for hybrid configurations. This type of connection relies on an IPsec VPN appliance (hardware device or soft appliance), which must be deployed at the edge of your network. To create this type of connection, you must have an externally facing IPv4 address that is not behind a NAT.

Point-to-Site (VPN over SSTP) configurations let you connect from a single computer from anywhere to anything located in your virtual network. It uses the Windows in-box VPN client. As part of the Point-to-Site configuration, you install a certificate and a VPN client configuration package, which contains the settings that allow your computer to connect to any virtual machine or role instance within the virtual network. It's great when you want to connect to a virtual network, but aren't located on-premises. It's also a good option when you don't have access to VPN hardware or an externally facing IPv4 address, both of which are required for a Site-to-Site connection.

You can configure your virtual network to use both Site-to-Site and Point-to-Site concurrently, as long as you

create your Site-to-Site connection using a route-based VPN type for your gateway. Route-based VPN types are called dynamic gateways in the classic deployment model.

Virtual network gateways

Is a VPN gateway a virtual network gateway?

A VPN gateway is a type of virtual network gateway. A VPN gateway sends encrypted traffic between your virtual network and your on-premises location across a public connection. You can also use a VPN gateway to send traffic between virtual networks. When you create a VPN gateway, you use the -GatewayType value 'Vpn'. For more information, see [About VPN Gateway configuration settings](#).

What is a policy-based (static-routing) gateway?

Policy-based gateways implement policy-based VPNs. Policy-based VPNs encrypt and direct packets through IPsec tunnels based on the combinations of address prefixes between your on-premises network and the Azure VNet. The policy (or Traffic Selector) is usually defined as an access list in the VPN configuration.

What is a route-based (dynamic-routing) gateway?

Route-based gateways implement the route-based VPNs. Route-based VPNs use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels. The policy or traffic selector for route-based VPNs are configured as any-to-any (or wild cards).

Can I update my Policy-based VPN gateway to Route-based?

No. An Azure Vnet gateway type cannot be changed from policy-based to route-based or the other way. The gateway must be deleted and recreated, a process taking around 60 minutes. The IP address of the gateway will not be preserved nor will the Pre-Shared Key (PSK).

1. Delete any connections associated with the gateway to be deleted.
2. Delete the gateway:
3. [Azure portal](#)
4. [Azure PowerShell](#)
5. [Azure Powershell - classic](#)
6. [Create a new gateway of desired type and complete the VPN setup](#)

Do I need a 'GatewaySubnet'?

Yes. The gateway subnet contains the IP addresses that the virtual network gateway services use. You need to create a gateway subnet for your VNet in order to configure a virtual network gateway. All gateway subnets must be named 'GatewaySubnet' to work properly. Don't name your gateway subnet something else. And don't deploy VMs or anything else to the gateway subnet.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The IP addresses in the gateway subnet are allocated to the gateway service. Some configurations require more IP addresses to be allocated to the gateway services than do others. You want to make sure your gateway subnet contains enough IP addresses to accommodate future growth and possible additional new connection configurations. So, while you can create a gateway subnet as small as /29, we recommend that you create a gateway subnet of /27 or larger (/27, /26, /25 etc.). Look at the requirements for the configuration that you want to create and verify that the gateway subnet you have will meet those requirements.

Can I deploy Virtual Machines or role instances to my gateway subnet?

No.

Can I get my VPN gateway IP address before I create it?

No. You have to create your gateway first to get the IP address. The IP address changes if you delete and recreate your VPN gateway.

Can I request a Static Public IP address for my VPN gateway?

No. Only Dynamic IP address assignment is supported. However, this does not mean that the IP address changes after it has been assigned to your VPN gateway. The only time the VPN gateway IP address changes is when the gateway is deleted and re-created. The VPN gateway public IP address doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

How does my VPN tunnel get authenticated?

Azure VPN uses PSK (Pre-Shared Key) authentication. We generate a pre-shared key (PSK) when we create the VPN tunnel. You can change the auto-generated PSK to your own with the Set Pre-Shared Key PowerShell cmdlet or REST API.

Can I use the Set Pre-Shared Key API to configure my policy-based (static routing) gateway VPN?

Yes, the Set Pre-Shared Key API and PowerShell cmdlet can be used to configure both Azure policy-based (static) VPNs and route-based (dynamic) routing VPNs.

Can I use other authentication options?

We are limited to using pre-shared keys (PSK) for authentication.

How do I specify which traffic goes through the VPN gateway?

Resource Manager deployment model

- PowerShell: use "AddressPrefix" to specify traffic for the local network gateway.
- Azure portal: navigate to the Local network gateway > Configuration > Address space.

Classic deployment model

- Azure portal: navigate to the classic virtual network > VPN connections > Site-to-site VPN connections > Local site name > Local site > Client address space.

Can I configure Force Tunneling?

Yes. See [Configure force tunneling](#).

Can I set up my own VPN server in Azure and use it to connect to my on-premises network?

Yes, you can deploy your own VPN gateways or servers in Azure either from the Azure Marketplace or creating your own VPN routers. You need to configure user-defined routes in your virtual network to ensure traffic is routed properly between your on-premises networks and your virtual network subnets.

Why are certain ports opened on my VPN gateway?

They are required for Azure infrastructure communication. They are protected (locked down) by Azure certificates. Without proper certificates, external entities, including the customers of those gateways, will not be able to cause any effect on those endpoints.

A VPN gateway is fundamentally a multi-homed device with one NIC tapping into the customer private network, and one NIC facing the public network. Azure infrastructure entities cannot tap into customer private networks for compliance reasons, so they need to utilize public endpoints for infrastructure communication. The public endpoints are periodically scanned by Azure security audit.

More information about gateway types, requirements, and throughput

For more information, see [About VPN Gateway configuration settings](#).

Site-to-Site connections and VPN devices

What should I consider when selecting a VPN device?

We have validated a set of standard Site-to-Site VPN devices in partnership with device vendors. A list of known compatible VPN devices, their corresponding configuration instructions or samples, and device specs can be found in the [About VPN devices](#) article. All devices in the device families listed as known compatible should work with Virtual Network. To help configure your VPN device, refer to the device configuration sample or link that

corresponds to appropriate device family.

Where can I find VPN device configuration settings?

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](#).

How do I edit VPN device configuration samples?

For information about editing device configuration samples, see [Editing samples](#).

Where do I find IPsec and IKE parameters?

For IPsec/IKE parameters, see [Parameters](#).

Why does my policy-based VPN tunnel go down when traffic is idle?

This is expected behavior for policy-based (also known as static routing) VPN gateways. When the traffic over the tunnel is idle for more than 5 minutes, the tunnel will be torn down. When traffic starts flowing in either direction, the tunnel will be reestablished immediately.

Can I use software VPNs to connect to Azure?

We support Windows Server 2012 Routing and Remote Access (RRAS) servers for Site-to-Site cross-premises configuration.

Other software VPN solutions should work with our gateway as long as they conform to industry standard IPsec implementations. Contact the vendor of the software for configuration and support instructions.

Point-to-Site using native Azure certificate authentication

This section applies to the Resource Manager deployment model.

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2.

Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\ IKEv2\DisableCertReqPayload" REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

What tools can I use to create certificates?

You can use your Enterprise PKI solution (your internal PKI), Azure PowerShell, MakeCert, and OpenSSL.

Are there instructions for certificate settings and parameters?

- **Internal PKI/Enterprise PKI solution:** See the steps to [Generate certificates](#).
- **Azure PowerShell:** See the [Azure PowerShell](#) article for steps.
- **MakeCert:** See the [MakeCert](#) article for steps.
- **OpenSSL:**
 - When exporting certificates, be sure to convert the root certificate to Base64.
 - For the client certificate:
 - When creating the private key, specify the length as 4096.
 - When creating the certificate, for the `-extensions` parameter, specify `usr_cert`.

Point-to-Site using RADIUS authentication

This section applies to the Resource Manager deployment model.

How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)
- Windows Server 2016 (64-bit only)
- Windows 10
- Mac OS X version 10.11 (El Capitan)
- Mac OS X version 10.12 (Sierra)
- Linux (StrongSwan)
- iOS

NOTE

Starting July 1, 2018, support is being removed for TLS 1.0 and 1.1 from Azure VPN Gateway. VPN Gateway will support only TLS 1.2. To maintain support, see the [updates to enable support for TLS1.2](#).

Additionally, the following legacy algorithms will also be deprecated for TLS on July 1, 2018:

- RC4 (Rivest Cipher 4)
- DES (Data Encryption Algorithm)
- 3DES (Triple Data Encryption Algorithm)
- MD5 (Message Digest 5)

How do I enable support for TLS 1.2 in Windows 7 and Windows 8.1?

1. Open a command prompt with elevated privileges by right-clicking on **Command Prompt** and selecting **Run as administrator**.
2. Run the following commands in the command prompt:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13 /v TlsVersion /t REG_DWORD /d 0xfc0
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
if %PROCESSOR_ARCHITECTURE% EQU AMD64 reg add
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" /v
DefaultSecureProtocols /t REG_DWORD /d 0xaa0
```

3. Install the following updates:

- [KB3140245](#)
- [KB2977292](#)

4. Reboot the computer.

5. Connect to the VPN.

Can I traverse proxies and firewalls using Point-to-Site capability?

Azure supports two types of Point-to-site VPN options:

- Secure Socket Tunneling Protocol (SSTP). SSTP is a Microsoft proprietary SSL-based solution that can penetrate firewalls since most firewalls open the TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and IP protocol no. 50. Firewalls do not always open these ports, so there is a possibility of IKEv2 VPN not being able to traverse proxies and firewalls.

If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. For the Resource Manager deployment model, you must have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or PolicyBased VPN gateways.

Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

No. A Point-to-Site client can only connect to resources in the VNet in which the virtual network gateway resides.

How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet. For a VPN Gateway with only IKEv2 Point-to-Site VPN connections, the total throughput that you can expect depends on the Gateway SKU. For more information on throughput, see [Gateway SKUs](#).

Can I use any software VPN client for Point-to-Site that supports SSTP and/or IKEv2?

No. You can only use the native VPN client on Windows for SSTP, and the native VPN client on Mac for IKEv2. Refer to the list of supported client operating systems.

Does Azure support IKEv2 VPN with Windows?

IKEv2 is supported on Windows 10 and Server 2016. However, in order to use IKEv2, you must install updates and set a registry key value locally. OS versions prior to Windows 10 are not supported and can only use SSTP.

To prepare Windows 10 or Server 2016 for IKEv2:

1. Install the update.

OS VERSION	DATE	NUMBER/LINK
Windows Server 2016 Windows 10 Version 1607	January 17, 2018	KB4057142
Windows 10 Version 1703	January 17, 2018	KB4057144

2. Set the registry key value. Create or set

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\ IKEv2\DisableCertReqPayload"
REG_DWORD key in the registry to 1.

What happens when I configure both SSTP and IKEv2 for P2S VPN connections?

When you configure both SSTP and IKEv2 in a mixed environment (consisting of Windows and Mac devices), the Windows VPN client will always try IKEv2 tunnel first, but will fall back to SSTP if the IKEv2 connection is not successful. MacOSX will only connect via IKEv2.

Other than Windows and Mac, which other platforms does Azure support for P2S VPN?

Azure supports Windows, Mac and Linux for P2S VPN.

I already have an Azure VPN Gateway deployed. Can I enable RADIUS and/or IKEv2 VPN on it?

Yes, you can enable these new features on already deployed gateways using Powershell or the Azure portal, provided that the gateway SKU that you are using supports RADIUS and/or IKEv2. For example, the VPN gateway Basic SKU does not support RADIUS or IKEv2.

Is RADIUS authentication supported on all Azure VPN Gateway SKUs?

RADIUS authentication is supported for VpnGw1, VpnGw2, and VpnGw3 SKUs. If you are using legacy SKUs, RADIUS authentication is supported on Standard and High Performance SKUs. It is not supported on the Basic Gateway SKU.

Is RADIUS authentication supported for the classic deployment model?

No. RADIUS authentication is not supported for the classic deployment model.

Are 3rd-party RADIUS servers supported?

Yes, 3rd-party RADIUS servers are supported.

What are the connectivity requirements to ensure that the Azure gateway is able to reach an on-premises RADIUS server?

A VPN Site-to-Site connection to the on-premises site, with the proper routes configured, is required.

Can traffic to an on-premises RADIUS server (from the Azure VPN gateway) be routed over an ExpressRoute connection?

No. It can only be routed over a Site-to-Site connection.

Is there a change in the number of SSTP connections supported with RADIUS authentication? What is the maximum number of SSTP and IKEv2 connections supported?

There is no change in the maximum number of SSTP connections supported on a gateway with RADIUS authentication. It remains 128. The maximum number of connections supported is 128, irrespective of whether the gateway is configured for SSTP, IKEv2, or both.

What is the difference between doing certificate authentication using a RADIUS server vs. using Azure native certificate authentication (by uploading a trusted certificate to Azure)?

In RADIUS certificate authentication, the authentication request is forwarded to a RADIUS server that handles the actual certificate validation. This option is useful if you want to integrate with a certificate authentication infrastructure that you already have through RADIUS.

When using Azure for certificate authentication, the Azure VPN gateway performs the validation of the certificate. You need to upload your certificate public key to the gateway. You can also specify list of revoked certificates that shouldn't be allowed to connect.

Does RADIUS authentication work with both IKEv2, and SSTP VPN?

Yes, RADIUS authentication is supported for both IKEv2, and SSTP VPN.

VNet-to-VNet and Multi-Site connections

The VNet-to-VNet FAQ applies to VPN Gateway connections. If you are looking for 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 using 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. Refer to the [VPN Gateway pricing page](#) for details. If you are connecting your VNets using VNet Peering, rather than VPN Gateway, see the [Virtual Network pricing page](#).

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 AAD Tenants?

Yes, VNet-to-VNet connections using Azure VPN gateways work across AAD Tenants.

Is VNet-to-VNet traffic secure?

Yes, it is 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 are not in the same subscription, do the subscriptions need to be associated with the same AD 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 public Azure and the Chinese / German / US Gov Azure instances. For these scenarios, consider using a Site-to-Site VPN connection.

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 [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 does not support connecting virtual machines or cloud services that are not 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 are connected together.

Can I used 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 be using 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.

Can I use Azure VPN gateway to transit traffic between my on-premises sites or to another virtual network?

Resource Manager deployment model

Yes. See the [BGP](#) section for more information.

Classic deployment model

Transit traffic via Azure VPN gateway is possible using the classic deployment model, but relies on statically defined address spaces in the network configuration file. BGP is not yet supported with Azure Virtual Networks and VPN gateways using the classic deployment model. Without BGP, manually defining transit address spaces is very error prone, and not recommended.

Does Azure generate the same IPsec/IKE pre-shared key for all my VPN connections for the same virtual network?

No, Azure by default generates different pre-shared keys for different VPN connections. However, you can use the Set VPN Gateway Key REST API or PowerShell cmdlet to set the key value you prefer. The key MUST be alphanumerical string of length between 1 to 128 characters.

Do I get more bandwidth with more Site-to-Site VPNs than for a single virtual network?

No, all VPN tunnels, including Point-to-Site VPNs, share the same Azure VPN gateway and the available bandwidth.

Can I configure multiple tunnels between my virtual network and my on-premises site using multi-site VPN?

Yes, but you must configure BGP on both tunnels to the same location.

Can I use Point-to-Site VPNs with my virtual network with multiple VPN tunnels?

Yes, Point-to-Site (P2S) VPNs can be used with the VPN gateways connecting to multiple on-premises sites and other virtual networks.

Can I connect a virtual network with IPsec VPNs to my ExpressRoute circuit?

Yes, this is supported. For more information, see [Configure ExpressRoute and Site-to-Site VPN connections that coexist](#).

IPsec/IKE policy

Is Custom IPsec/IKE policy supported on all Azure VPN Gateway SKUs?

Custom IPsec/IKE policy is supported on Azure **VpnGw1**, **VpnGw2**, **VpnGw3**, **Standard**, and **HighPerformance** VPN gateways. The **Basic** SKU is **not** supported.

How many policies can I specify on a connection?

You can only specify **one** policy combination for a given connection.

Can I specify a partial policy on a connection? (for example, only IKE algorithms, but not IPsec)

No, you must specify all algorithms and parameters for both IKE (Main Mode) and IPsec (Quick Mode). Partial policy specification is not allowed.

What are the algorithms and key strengths supported in the custom policy?

The following table lists the supported cryptographic algorithms and key strengths configurable by the customers. You must select one option for every field.

IPSEC/IKEV2	OPTIONS
IKEv2 Encryption	AES256, AES192, AES128, DES3, DES
IKEv2 Integrity	SHA384, SHA256, SHA1, MD5
DH Group	DHGroup24, ECP384, ECP256, DHGroup14 (DHGroup2048), DHGroup2, DHGroup1, None
IPsec Encryption	GCMAES256, GCMAES192, GCMAES128, AES256, AES192, AES128, DES3, DES, None
IPsec Integrity	GCMAES256, GCMAES192, GCMAES128, SHA256, SHA1, MD5
PFS Group	PFS24, ECP384, ECP256, PFS2048, PFS2, PFS1, None
QM SA Lifetime	Seconds (integer; min. 300 /default 27000 seconds) KBytes (integer; min. 1024 /default 102400000 KBytes)
Traffic Selector	UsePolicyBasedTrafficSelectors (\$True/\$False; default \$False)

IMPORTANT

1. DHGroup2048 & PFS2048 are the same as Diffie-Hellman Group **14** in IKE and IPsec PFS. See [Diffie-Hellman Groups](#) for the complete mappings.
2. For GCMAES algorithms, you must specify the same GCMAES algorithm and key length for both IPsec Encryption and Integrity.
3. IKEv2 Main Mode SA lifetime is fixed at 28,800 seconds on the Azure VPN gateways
4. QM SA Lifetimes are optional parameters. If none was specified, default values of 27,000 seconds (7.5 hrs) and 102400000 KBytes (102GB) are used.
5. UsePolicyBasedTrafficSelector is an option parameter on the connection. See the next FAQ item for "UsePolicyBasedTrafficSelectors"

Does everything need to match between the Azure VPN gateway policy and my on-premises VPN device configurations?

Your on-premises VPN device configuration must match or contain the following algorithms and parameters that you specify on the Azure IPsec/IKE policy:

- IKE encryption algorithm
- IKE integrity algorithm
- DH Group
- IPsec encryption algorithm
- IPsec integrity algorithm
- PFS Group
- Traffic Selector (*)

The SA lifetimes are local specifications only, do not need to match.

If you enable **UsePolicyBasedTrafficSelectors**, you need to ensure your VPN device has the matching traffic selectors defined with all combinations of your on-premises network (local network gateway) prefixes to/from the Azure virtual network prefixes, instead of any-to-any. For example, if your on-premises network prefixes are 10.1.0.0/16 and 10.2.0.0/16, and your virtual network prefixes are 192.168.0.0/16 and 172.16.0.0/16, you need to specify the following traffic selectors:

- 10.1.0.0/16 <=====> 192.168.0.0/16
- 10.1.0.0/16 <=====> 172.16.0.0/16
- 10.2.0.0/16 <=====> 192.168.0.0/16
- 10.2.0.0/16 <=====> 172.16.0.0/16

For more information, see [Connect multiple on-premises policy-based VPN devices](#).

Which Diffie-Hellman Groups are supported?

The table below lists the supported Diffie-Hellman Groups for IKE (DHGroup) and IPsec (PFSGroup):

DIFFIE-HELLMAN GROUP	DHGROUP	PFSGROUP	KEY LENGTH
1	DHGroup1	PFS1	768-bit MODP
2	DHGroup2	PFS2	1024-bit MODP
14	DHGroup14 DHGroup2048	PFS2048	2048-bit MODP
19	ECP256	ECP256	256-bit ECP

Diffie-Hellman group	DHGroup	PFSGroup	Key length
20	ECP384	ECP284	384-bit ECP
24	DHGroup24	PFS24	2048-bit MODP

For more information, see [RFC3526](#) and [RFC5114](#).

Does the custom policy replace the default IPsec/IKE policy sets for Azure VPN gateways?

Yes, once a custom policy is specified on a connection, Azure VPN gateway will only use the policy on the connection, both as IKE initiator and IKE responder.

If I remove a custom IPsec/IKE policy, does the connection become unprotected?

No, the connection will still be protected by IPsec/IKE. Once you remove the custom policy from a connection, the Azure VPN gateway reverts back to the [default list of IPsec/IKE proposals](#) and restart the IKE handshake again with your on-premises VPN device.

Would adding or updating an IPsec/IKE policy disrupt my VPN connection?

Yes, it could cause a small disruption (a few seconds) as the Azure VPN gateway tears down the existing connection and restarts the IKE handshake to re-establish the IPsec tunnel with the new cryptographic algorithms and parameters. Ensure your on-premises VPN device is also configured with the matching algorithms and key strengths to minimize the disruption.

Can I use different policies on different connections?

Yes. Custom policy is applied on a per-connection basis. You can create and apply different IPsec/IKE policies on different connections. You can also choose to apply custom policies on a subset of connections. The remaining ones use the Azure default IPsec/IKE policy sets.

Can I use the custom policy on VNet-to-VNet connection as well?

Yes, you can apply custom policy on both IPsec cross-premises connections or VNet-to-VNet connections.

Do I need to specify the same policy on both VNet-to-VNet connection resources?

Yes. A VNet-to-VNet tunnel consists of two connection resources in Azure, one for each direction. Make sure both connection resources have the same policy, otherwise the VNet-to-VNet connection won't establish.

Does custom IPsec/IKE policy work on ExpressRoute connection?

No. IPsec/IKE policy only works on S2S VPN and VNet-to-VNet connections via the Azure VPN gateways.

BGP

Is BGP supported on all Azure VPN Gateway SKUs?

No, BGP is supported on Azure **VpnGw1**, **VpnGw2**, **VpnGw3**, **Standard** and **HighPerformance** VPN gateways. **Basic** SKU is NOT supported.

Can I use BGP with Azure Policy-Based VPN gateways?

No, BGP is supported on Route-Based VPN gateways only.

Can I use private ASNs (Autonomous System Numbers)?

Yes, you can use your own public ASNs or private ASNs for both your on-premises networks and Azure virtual networks.

Can I use 32-bit ASNs (Autonomous System Numbers)?

No, the Azure VPN Gateways support 16-Bit ASNs today.

Are there ASNs reserved by Azure?

Yes, the following ASNs are reserved by Azure for both internal and external peerings:

- Public ASNs: 8074, 8075, 12076
- Private ASNs: 65515, 65517, 65518, 65519, 65520

You cannot specify these ASNs for your on-premises VPN devices when connecting to Azure VPN gateways.

Are there any other ASNs that I can't use?

Yes, the following ASNs are [reserved by IANA](#) and can't be configured on your Azure VPN Gateway:

23456, 64496-64511, 65535-65551 and 429496729

Can I use the same ASN for both on-premises VPN networks and Azure VNets?

No, you must assign different ASNs between your on-premises networks and your Azure VNets if you are connecting them together with BGP. Azure VPN Gateways have a default ASN of 65515 assigned, whether BGP is enabled or not for your cross-premises connectivity. You can override this default by assigning a different ASN when creating the VPN gateway, or change the ASN after the gateway is created. You will need to assign your on-premises ASNs to the corresponding Azure Local Network Gateways.

What address prefixes will Azure VPN gateways advertise to me?

Azure VPN gateway will advertise the following routes to your on-premises BGP devices:

- Your VNet address prefixes
- Address prefixes for each Local Network Gateways connected to the Azure VPN gateway
- Routes learned from other BGP peering sessions connected to the Azure VPN gateway, **except default route or routes overlapped with any VNet prefix.**

Can I advertise default route (0.0.0.0/0) to Azure VPN gateways?

Yes.

Please note this will force all VNet egress traffic towards your on-premises site, and will prevent the VNet VMs from accepting public communication from the Internet directly, such RDP or SSH from the Internet to the VMs.

Can I advertise the exact prefixes as my Virtual Network prefixes?

No, advertising the same prefixes as any one of your Virtual Network address prefixes will be blocked or filtered by the Azure platform. However you can advertise a prefix that is a superset of what you have inside your Virtual Network.

For example, if your virtual network used the address space 10.0.0.0/16, you could advertise 10.0.0.0/8. But you cannot advertise 10.0.0.0/16 or 10.0.0.0/24.

Can I use BGP with my VNet-to-VNet connections?

Yes, you can use BGP for both cross-premises connections and VNet-to-VNet connections.

Can I mix BGP with non-BGP connections for my Azure VPN gateways?

Yes, you can mix both BGP and non-BGP connections for the same Azure VPN gateway.

Does Azure VPN gateway support BGP transit routing?

Yes, BGP transit routing is supported, with the exception that Azure VPN gateways will **NOT** advertise default routes to other BGP peers. To enable transit routing across multiple Azure VPN gateways, you must enable BGP on all intermediate VNet-to-VNet connections.

Can I have more than one tunnel between Azure VPN gateway and my on-premises network?

Yes, you can establish more than one S2S VPN tunnel between an Azure VPN gateway and your on-premises network. Please note that all these tunnels will be counted against the total number of tunnels for your Azure VPN

gateways and you must enable BGP on both tunnels.

For example, if you have two redundant tunnels between your Azure VPN gateway and one of your on-premises networks, they will consume 2 tunnels out of the total quota for your Azure VPN gateway (10 for Standard and 30 for HighPerformance).

Can I have multiple tunnels between two Azure VNets with BGP?

Yes, but at least one of the virtual network gateways must be in active-active configuration.

Can I use BGP for S2S VPN in an ExpressRoute/S2S VPN co-existence configuration?

Yes.

What address does Azure VPN gateway use for BGP Peer IP?

The Azure VPN gateway will allocate a single IP address from the GatewaySubnet range defined for the virtual network. By default, it is the second last address of the range. For example, if your GatewaySubnet is 10.12.255.0/27, ranging from 10.12.255.0 to 10.12.255.31, the BGP Peer IP address on the Azure VPN gateway will be 10.12.255.30. You can find this information when you list the Azure VPN gateway information.

What are the requirements for the BGP Peer IP addresses on my VPN device?

Your on-premises BGP peer address **MUST NOT** be the same as the public IP address of your VPN device. Use a different IP address on the VPN device for your BGP Peer IP. It can be an address assigned to the loopback interface on the device, but please note that it cannot be an APIPA (169.254.x.x) address. Specify this address in the corresponding Local Network Gateway representing the location.

What should I specify as my address prefixes for the Local Network Gateway when I use BGP?

Azure Local Network Gateway specifies the initial address prefixes for the on-premises network. With BGP, you must allocate the host prefix (/32 prefix) of your BGP Peer IP address as the address space for that on-premises network. If your BGP Peer IP is 10.52.255.254, you should specify "10.52.255.254/32" as the localNetworkAddressSpace of the Local Network Gateway representing this on-premises network. This is to ensure that the Azure VPN gateway establishes the BGP session through the S2S VPN tunnel.

What should I add to my on-premises VPN device for the BGP peering session?

You should add a host route of the Azure BGP Peer IP address on your VPN device pointing to the IPsec S2S VPN tunnel. For example, if the Azure VPN Peer IP is "10.12.255.30", you should add a host route for "10.12.255.30" with a nexthop interface of the matching IPsec tunnel interface on your VPN device.

Cross-premises connectivity and VMs

If my virtual machine is in a virtual network and I have a cross-premises connection, how should I connect to the VM?

You have a few options. If you have RDP enabled for your VM, you can connect to your virtual machine by using the private IP address. In that case, you would specify the private IP address and the port that you want to connect to (typically 3389). You'll need to configure the port on your virtual machine for the traffic.

You can also connect to your virtual machine by private IP address from another virtual machine that's located on the same virtual network. You can't RDP to your virtual machine by using the private IP address if you are connecting from a location outside of your virtual network. For example, if you have a Point-to-Site virtual network configured and you don't establish a connection from your computer, you can't connect to the virtual machine by private IP address.

If my virtual machine is in a virtual network with cross-premises connectivity, does all the traffic from my VM go through that connection?

No. Only the traffic that has a destination IP that is contained in the virtual network Local Network IP address ranges that you specified will go through the virtual network gateway. Traffic has a destination IP located within the virtual network stays within the virtual network. Other traffic is sent through the load balancer to the public

networks, or if forced tunneling is used, sent through the Azure VPN gateway.

How do I troubleshoot an RDP connection to a VM

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](#).

When you connect over Point-to-Site, check the following additional items:

- Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VPNCientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
- Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.

For more information about troubleshooting an RDP connection, see [Troubleshoot Remote Desktop connections to a VM](#).

Virtual Network FAQ

You view additional virtual network information in the [Virtual Network FAQ](#).

Next steps

- For more information about VPN Gateway, see [About VPN Gateway](#).
- For more information about VPN Gateway configuration settings, see [About VPN Gateway configuration settings](#).

Azure subscription and service limits, quotas, and constraints

9/18/2018 • 64 minutes to read • [Edit Online](#)

This document lists some of the most common Microsoft Azure limits, which are also sometimes called quotas. This document doesn't currently cover all Azure services. Over time, the list will be expanded and updated to cover more of the platform.

Please visit [Azure Pricing Overview](#) to learn more about Azure pricing. There, you can estimate your costs using the [Pricing Calculator](#) or by visiting the pricing details page for a service (for example, [Windows VMs](#)). For tips to help manage your costs, see [Prevent unexpected costs with Azure billing and cost management](#).

NOTE

If you want to raise the limit or quota above the **Default Limit**, open an online customer support request at no charge. The limits can't be raised above the **Maximum Limit** value shown in the following tables. If there is no **Maximum Limit** column, then the resource doesn't have adjustable limits.

[Free Trial subscriptions](#) are not eligible for limit or quota increases. If you have a [Free Trial subscription](#), you can upgrade to a [Pay-As-You-Go](#) subscription. For more information, see [Upgrade Azure Free Trial to Pay-As-You-Go](#) and [Free Trial subscription FAQ](#).

Limits and the Azure Resource Manager

It is now possible to combine multiple Azure resources into a single Azure Resource Group. When using Resource Groups, limits that once were global become managed at a regional level with the Azure Resource Manager. For more information about Azure Resource Groups, see [Azure Resource Manager overview](#).

In the limits below, a new table has been added to reflect any differences in limits when using the Azure Resource Manager. For example, there is a **Subscription Limits** table and a **Subscription Limits - Azure Resource Manager** table. When a limit applies to both scenarios, it is only shown in the first table. Unless otherwise indicated, limits are global across all regions.

NOTE

It is important to emphasize that quotas for resources in Azure Resource Groups are per-region accessible by your subscription, and are not per-subscription, as the service management quotas are. Let's use vCPU quotas as an example. If you need to request a quota increase with support for vCPUs, you need to decide how many vCPUs you want to use in which regions, and then make a specific request for Azure Resource Group vCPU quotas for the amounts and regions that you want. Therefore, if you need to use 30 vCPUs in West Europe to run your application there, you should specifically request 30 vCPUs in West Europe. But you will not have a vCPU quota increase in any other region -- only West Europe will have the 30-vCPU quota.

As a result, you may find it useful to consider deciding what your Azure Resource Group quotas need to be for your workload in any one region, and request that amount in each region into which you are considering deployment. See [troubleshooting deployment issues](#) for more help discovering your current quotas for specific regions.

Service-specific limits

- [Active Directory](#)

- [API Management](#)
- [App Service](#)
- [Application Gateway](#)
- [Application Insights](#)
- [Automation](#)
- [Azure Cosmos DB](#)
- [Azure Database for MySQL](#)
- [Azure Database for PostgreSQL](#)
- [Azure Event Grid](#)
- [Azure Maps](#)
- [Azure Monitor](#)
- [Azure Policy](#)
- [Azure Redis Cache](#)
- [Backup](#)
- [Batch](#)
- [Batch AI](#)
- [BizTalk Services](#)
- [CDN](#)
- [Cloud Services](#)
- [Container Instances](#)
- [Container Registry](#)
- [Kubernetes Service](#)
- [Data Factory](#)
- [Data Lake Analytics](#)
- [Data Lake Store](#)
- [Database Migration Service](#)
- [DNS](#)
- [Event Hubs](#)
- [Azure Firewall](#)
- [IoT Hub](#)
- [IoT Hub Device Provisioning Service](#)
- [Key Vault](#)
- [Log Analytics](#)
- [Managed Identity](#)
- [Media Services](#)
- [Mobile Engagement](#)
- [Mobile Services](#)
- [Multi-Factor Authentication](#)
- [Networking](#)
- [Network Watcher](#)
- [Notification Hub Service](#)
- [Resource Group](#)
- [Role-based access control](#)
- [Scheduler](#)
- [Search](#)
- [Service Bus](#)

- [Site Recovery](#)
- [SQL Database](#)
- [SQL Data Warehouse](#)
- [Storage](#)
- [StorSimple System](#)
- [Stream Analytics](#)
- [Subscription](#)
- [Traffic Manager](#)
- [Virtual Machines](#)
- [Virtual Machine Scale Sets](#)

Subscription limits

Subscription limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
vCPUs per subscription ¹	20	10,000
Co-administrators per subscription	200	200
Storage accounts per region per subscription ²	200	250
Cloud services per subscription	20	200
Local networks per subscription	10	500
SQL Database servers per subscription	6	200
DNS servers per subscription	9	100
Reserved IPs per subscription	20	100
Hosted service certificates per subscription	199	199
Affinity groups per subscription	256	256

¹Extra Small instances count as one vCPU towards the vCPU limit despite using a partial CPU core.

²The storage account limit includes both Standard and Premium storage accounts. If you require more than 200 storage accounts in a given region, make a request through [Azure Support](#). The Azure Storage team will review your business case and may approve up to 250 storage accounts for a given region.

Subscription limits - Azure Resource Manager

The following limits apply when using the Azure Resource Manager and Azure Resource Groups. Limits that have not changed with the Azure Resource Manager are not listed below. Please refer to the previous table for those limits.

For information about handling limits on Resource Manager requests, see [Throttling Resource Manager requests](#).

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
VMs per subscription	10,000 ¹ per Region	10,000 per Region

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
VM total cores per subscription	20 ¹ per Region	Contact support
VM per series (Dv2, F, etc.) cores per subscription	20 ¹ per Region	Contact support
Co-administrators per subscription	Unlimited	Unlimited
Storage accounts per region per subscription	200	200 ²
Resource Groups per subscription	980	980
Availability Sets per subscription	2,000 per Region	2,000 per Region
Resource Manager API Reads	15,000 per hour	15,000 per hour
Resource Manager API Writes	1,200 per hour	1,200 per hour
Resource Manager API request size	4,194,304 bytes	4,194,304 bytes
Tags per subscription ³	unlimited	unlimited
Unique tag calculations per subscription ³	10,000	10,000
Cloud services per subscription	Not Applicable ⁴	Not Applicable ⁴
Affinity groups per subscription	Not Applicable ⁴	Not Applicable ⁴
Subscription level deployments per location	800	800

¹Default limits vary by offer Category Type, such as Free Trial, Pay-As-You-Go, and series, such as Dv2, F, G, etc.

²This includes both Standard and Premium storage accounts. If you require more than 200 storage accounts, make a request through [Azure Support](#). The Azure Storage team will review your business case and may approve up to 250 storage accounts.

³You can apply an unlimited number of tags per subscription. The number of tags per resource or resource group is limited to 15. Resource Manager only returns a [list of unique tag name and values](#) in the subscription when the number of tags is 10,000 or less. However, you can still find a resource by tag when the number exceeds 10,000.

⁴These features are no longer required with Azure Resource Groups and the Azure Resource Manager.

NOTE

It is important to emphasize that virtual machine cores have a regional total limit as well as a regional per size series (Dv2, F, etc.) limit that are separately enforced. For example, consider a subscription with a US East total VM core limit of 30, an A series core limit of 30, and a D series core limit of 30. This subscription would be allowed to deploy 30 A1 VMs, or 30 D1 VMs, or a combination of the two not to exceed a total of 30 cores (for example, 10 A1 VMs and 20 D1 VMs).

Resource Group limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resources per resource group (per resource type)	800	Varies per resource type
Deployments per resource group in the deployment history	800	800
Resources per deployment	800	800
Management Locks (per unique scope)	20	20
Number of Tags (per resource or resource group)	15	15
Tag key length	512	512
Tag value length	256	256

Template limits

VALUE	DEFAULT LIMIT	MAXIMUM LIMIT
Parameters	256	256
Variables	256	256
Resources (including copy count)	800	800
Outputs	64	64
Template expression	24,576 chars	24,576 chars
Resources in exported templates	200	200
Template size	1 MB	1 MB
Parameter file size	64 KB	64 KB

You can exceed some template limits by using a nested template. For more information, see [Using linked templates when deploying Azure resources](#). To reduce the number of parameters, variables, or outputs, you can combine several values into an object. For more information, see [Objects as parameters](#).

If you reach the limit of 800 deployments per resource group, delete deployments from the history that are no longer needed. You can delete entries from the history with `az group deployment delete` for Azure CLI, or `Remove-AzureRmResourceGroupDeployment` in PowerShell. Deleting an entry from the deployment history does not affect the deployed resources.

Virtual Machines limits

Virtual Machine limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual machines per cloud service ¹	50	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Input endpoints per cloud service ²	150	150

¹Virtual machines created in Service Management (instead of Resource Manager) are automatically stored in a cloud service. You can add more virtual machines to that cloud service for load balancing and availability.

²Input endpoints allow communications to a virtual machine from outside the virtual machine's cloud service. Virtual machines in the same cloud service or virtual network can automatically communicate with each other. See [How to Set Up Endpoints to a Virtual Machine](#).

Virtual Machines limits - Azure Resource Manager

The following limits apply when using the Azure Resource Manager and Azure Resource Groups. Limits that have not changed with the Azure Resource Manager are not listed below. Please refer to the previous table for those limits.

RESOURCE	DEFAULT LIMIT
Virtual machines per availability set	200
Certificates per subscription	Unlimited ¹

¹With Azure Resource Manager, certificates are stored in the Azure Key Vault. Although the number of certificates is unlimited for a subscription, there is still a 1 MB limit of certificates per deployment (which consists of either a single VM or an availability set).

Virtual Machine Scale Sets limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of VMs in a scale set	1000	1000
Maximum number of VMs based on a custom VM image in a scale set	600	600
Maximum number of scale sets in a region	2000	2000

Container Instances limits

RESOURCE	DEFAULT LIMIT
Container groups per subscription	100 ¹
Number of containers per container group	60
Number of volumes per container group	20
Ports per IP	5
Container creates per hour	300 ¹
Container creates per 5 minutes	100 ¹

RESOURCE	DEFAULT LIMIT
Container deletes per hour	300 ¹
Container deletes per 5 minutes	100 ¹
Multiple containers per container group	Linux only ²
Azure Files volumes	Linux only ²
GitRepo volumes	Linux only ²
Secret volumes	Linux only ²

¹ Create an [Azure support request](#) to request a limit increase.

² Windows support for this feature is planned.

Container Registry limits

The following table details the features and limits of the Basic, Standard, and Premium [service tiers](#).

RESOURCE	BASIC	STANDARD	PREMIUM
Storage ¹	10 GiB	100 GiB	500 GiB
Max image layer size	20 GiB	20 GiB	50 GiB
ReadOps per minute ^{2, 3}	1,000	3,000	10,000
WriteOps per minute ^{2, 4}	100	500	2,000
Download bandwidth MBps ²	30	60	100
Upload bandwidth MBps ²	10	20	50
Webhooks	2	10	100
Geo-replication	N/A	N/A	Supported
Content trust (preview)	N/A	N/A	Supported

¹ The specified storage limits are the amount of *included* storage for each tier. You're charged an additional daily rate per GiB for image storage above these limits. For rate information, see [Container Registry pricing](#).

² *ReadOps*, *WriteOps*, and *Bandwidth* are minimum estimates. ACR strives to improve performance as usage requires.

³ [docker pull](#) translates to multiple read operations based on the number of layers in the image, plus the manifest retrieval.

⁴ [docker push](#) translates to multiple write operations, based on the number of layers that must be pushed. A [docker push](#) includes *ReadOps* to retrieve a manifest for an existing image.

Kubernetes Service limits

RESOURCE	DEFAULT LIMIT
Max clusters per subscription	100
Max nodes per cluster	100
Max pods per node: Basic networking with Kubenet	110
Max pods per node: Advanced networking with Azure CNI	Azure CLI deployment: 110 ¹ Resource Manager template: 110 ¹ Portal deployment: 30

¹ This value is configurable at cluster deployment when deploying an AKS cluster with the Azure CLI or a Resource Manager template.

Networking limits

ExpressRoute Limits

The following limits apply to ExpressRoute resources per subscription.

RESOURCE	DEFAULT/MAX LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription (Azure Resource Manager)	10
Maximum number of routes for Azure private peering with ExpressRoute standard	4,000
Maximum number of routes for Azure private peering with ExpressRoute premium add-on	10,000
Maximum number of routes for Azure Microsoft peering with ExpressRoute standard	200
Maximum number of routes for Azure Microsoft peering with ExpressRoute premium add-on	200
Maximum number of ExpressRoute circuits linked to the same virtual network in different peering locations	4
Number of virtual network links allowed per ExpressRoute circuit	see table below

Number of Virtual Networks per ExpressRoute circuit

CIRCUIT SIZE	NUMBER OF VNET LINKS FOR STANDARD	NUMBER OF VNET LINKS WITH PREMIUM ADD-ON
50 Mbps	10	20
100 Mbps	10	25
200 Mbps	10	25

CIRCUIT SIZE	NUMBER OF VNET LINKS FOR STANDARD	NUMBER OF VNET LINKS WITH PREMIUM ADD-ON
500 Mbps	10	40
1 Gbps	10	50
2 Gbps	10	60
5 Gbps	10	75
10 Gbps	10	100

Networking limits

The following limits apply only for networking resources managed through the classic deployment model per subscription. Learn how to [view your current resource usage against your subscription limits](#).

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual networks	50	100
Local network sites	20	contact support
DNS Servers per virtual network	20	100
Private IP Addresses per virtual network	4096	4096
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500K	500K
Network Security Groups (NSG)	100	200
NSG rules per NSG	200	1000
User defined route tables	100	200
User defined routes per route table	100	400
Public IP addresses (dynamic)	5	contact support
Reserved public IP addresses	20	contact support
Public VIP per deployment	5	contact support
Private VIP (ILB) per deployment	1	1
Endpoint Access Control Lists (ACLs)	50	50

Networking Limits - Azure Resource Manager

The following limits apply only for networking resources managed through Azure Resource Manager per region per subscription. Learn how to [view your current resource usage against your subscription limits](#).

NOTE

We recently increased all default limits to their maximum limits. If there is no **Maximum Limit** column, then the resource doesn't have adjustable limits. If you have had these limits increased by support in the past and do not see updated limits as below, you can [open an online customer support request at no charge](#)

RESOURCE	DEFAULT LIMIT
Virtual networks	1000
Subnets per virtual network	3000
Virtual network peerings per Virtual Network	100
DNS Servers per virtual network	25
Private IP Addresses per virtual network	65536
Private IP Addresses per network interface	256
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500K
Network Interfaces (NIC)	24000
Network Security Groups (NSG)	5000
NSG rules per NSG	1000
IP addresses and ranges specified for source or destination in a security group	4000
Application security groups	3000
Application security groups per IP configuration, per NIC	20
IP configurations per application security group	4000
Application security groups that can be specified within all security rules of a network security group	100
User defined route tables	200
User defined routes per route table	400
Point-to-Site Root Certificates per VPN Gateway	20

Public IP address limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP addresses - dynamic	(Basic) 200	contact support

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP addresses - static	(Basic) 200	contact support
Public IP addresses - static	(Standard) 200	contact support

Load Balancer limits

The following limits apply only for networking resources managed through Azure Resource Manager per region per subscription. Learn how to [view your current resource usage against your subscription limits](#)

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Load Balancers	100	1000
Rules per resource, Basic	250	250
Rules per resource, Standard	1500	1500
Rules per IP configuration	299	299
Frontend IP configurations, Basic	10	200
Frontend IP configurations, Standard	10	600
Backend pool, Basic	100, single Availability Set	100, single Availability Set
Backend pool, Standard	1000, single VNet	1000, single VNet
Backend resources per Load Balancer, Standard *	150	150
HA Ports, Standard	1 per internal frontend	1 per internal frontend

** Up to 150 resources, any combination of standalone virtual machines, availability sets, and virtual machine scale sets.

[Contact support](#) in case you need to increase limits from default.

Application Gateway limits

RESOURCE	DEFAULT LIMIT	NOTE
Application Gateway	50 per subscription	Maximum 1000
Frontend IP Configurations	2	1 public and 1 private
Frontend Ports	20	
Backend Address Pools	20	
Backend Servers per pool	100	
HTTP Listeners	20	

RESOURCE	DEFAULT LIMIT	NOTE
HTTP load balancing rules	200	# of HTTP Listeners * n, n=10 Default
Backend HTTP settings	20	1 per Backend Address Pool
Instances per gateway	10	For more instances, open support ticket
SSL certificates	20	1 per HTTP Listeners
Authentication certificates	5	Maximum 10
Request time out min	1 second	
Request time out max	24 hrs	
Number of sites	20	1 per HTTP Listeners
URL Maps per listener	1	
Maximum URL length	8000	
Maximum file upload size Standard	2 GB	
Maximum file upload size WAF	100 MB	
WAF body size limit (without files)	128 KB	

Network Watcher limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT	NOTE
Network Watcher	1 per region	1 per region	Network Watcher resource is created to enable access to the service. Only 1 Network Watcher resource is required per subscription per region
Packet Capture sessions	10 per region		# of sessions only, not saved captures

Traffic Manager limits

RESOURCE	DEFAULT LIMIT
Profiles per subscription	200 ¹
Endpoints per profile	200

¹Contact support in case you need to increase these limits.

DNS limits

RESOURCE	DEFAULT LIMIT
Zones per subscription	100 ¹
Record sets per zone	5000 ¹
Records per record set	20

¹ Contact Azure Support in case you need to increase these limits.

Azure Firewall limits

RESOURCE	DEFAULT LIMIT
Data processed	1000 TB/firewall/month ¹
Rules	10k application rules, 10k network rules
VNet peering	For hub and spoke implementations, max of 50 spoke VNETs.
Global peering	Not supported. You should have at least one firewall deployment per region.

¹ Contact Azure Support in case you need to increase these limits.

Storage limits

The following table describes default limits for Azure Storage. The *ingress* limit refers to all data (requests) being sent to a storage account. The *egress* limit refers to all data (responses) being received from a storage account.

RESOURCE	DEFAULT LIMIT
Number of storage accounts per region per subscription, including both standard and premium accounts	200
Max storage account capacity	500 TiB
Max number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate per storage account	20,000 requests per second
Max ingress per storage account (US Regions)	10 Gbps if RA-GRS/GRS enabled, 20 Gbps for LRS/ZRS ¹
Max egress per storage account (US Regions)	50 Gbps
Max ingress per storage account (Non-US regions)	5 Gbps if RA-GRS/GRS enabled, 10 Gbps for LRS/ZRS ¹
Max egress per storage account (Non-US regions)	50 Gbps

¹Azure Storage replication options include:

- **RA-GRS**: Read-access geo-redundant storage. If RA-GRS is enabled, egress targets for the secondary location are identical to those for the primary location.
- **GRS**: Geo-redundant storage.
- **ZRS**: Zone-redundant storage.

- **LRS:** Locally redundant storage.

NOTE

If you require expanded limits for your storage account, or a greater number of storage accounts in a specific region, please contact [Azure Support](#). The Azure Storage team will review your request and may approve higher limits on a case by case basis. Both general-purpose and Blob storage accounts support increased capacity, ingress/egress, and request rate upon request.

For additional details on storage account limits, see [Azure Storage Scalability and Performance Targets](#).

Storage resource provider limits

The following limits apply only when performing management operations using Azure Resource Manager with Azure Storage.

RESOURCE	DEFAULT LIMIT
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	200 per hour
Storage account management operations (list)	100 per 5 minutes

Azure Blob storage limits

RESOURCE	TARGET
Max size of single blob container	Same as max storage account capacity
Max number of blocks in a block blob or append blob	50,000 blocks
Max size of a block in a block blob	100 MiB
Max size of a block blob	50,000 X 100 MiB (approx. 4.75 TiB)
Max size of a block in an append blob	4 MiB
Max size of an append blob	50,000 x 4 MiB (approx. 195 GiB)
Max size of a page blob	8 TiB
Max number of stored access policies per blob container	5
Target throughput for single blob	Up to 60 MiB per second, or up to 500 requests per second

Azure Files limits

For additional details on Azure Files limits, see [Azure Files scalability and performance targets](#).

RESOURCE	TARGET
Max size of a file share	5 TiB
Max size of a file in a file share	1 TiB

RESOURCE	TARGET
Max number of files in a file share	No limit
Max IOPS per share	1000 IOPS
Max number of stored access policies per file share	5
Maximum request rate per storage account	20,000 requests per second for files of any valid size ³
Target throughput for single file share	Up to 60 MiB per second
Maximum open handles per file	2000 open handles
Maximum number of share snapshots	200 share snapshots

Azure File Sync limits

RESOURCE	TARGET	HARD LIMIT
Storage Sync Services per subscription	15 Storage Sync Services	No
Sync groups per Storage Sync Service	100 sync groups	Yes
Registered servers per Storage Sync Service	99 servers	Yes
Cloud endpoints per Sync Group	1 cloud endpoint	Yes
Server endpoints per Sync Group	50 server endpoints	No
Server endpoints per server	33-99 server endpoints	Yes, but varies based on configuration (CPU, memory, volumes, file churn, file count, etc.)
Endpoint size	4 TiB	No
File system objects (directories and files) per sync group	25 million objects	No
Maximum number of file system objects (directories and files) in a directory	200,000 objects	Yes
Maximum object (directories and files) name length	255 characters	Yes
Maximum object (directories and files) security descriptor size	4 KiB	Yes
File size	100 GiB	No
Minimum file size for a file to be tiered	64 KiB	Yes

Azure Queue storage limits

RESOURCE	TARGET
Max size of single queue	500 TiB
Max size of a message in a queue	64 KiB
Max number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second assuming 1 KiB message size
Target throughput for single queue (1 KiB messages)	Up to 2000 messages per second

Azure Table storage limits

RESOURCE	TARGET
Max size of single table	500 TiB
Max size of a table entity	1 MiB
Max number of properties in a table entity	255 (including 3 system properties: PartitionKey, RowKey and Timestamp)
Max number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second (assuming 1 KiB entity size)
Target throughput for single table partition (1 KiB entities)	Up to 2000 entities per second

Virtual machine disk limits

An Azure virtual machine supports attaching a number of data disks. This article describes scalability and performance targets for a VM's data disks. Use these targets to help decide the number and type of disk that you need to meet your performance and capacity requirements.

IMPORTANT

For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks are not highly utilized at the same time, then the virtual machine can support a larger number of disks.

- **For Azure Managed Disks:**

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Standard Managed Disks	10,000	50,000
Standard SSD Managed Disks	10,000	50,000
Premium Managed Disks	10,000	50,000
Standard_LRS Snapshots	10,000	50,000
Standard_ZRS Snapshots	10,000	50,000

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Premium_LRS Snapshots	10,000	50,000
Managed Image	10,000	50,000

- For standard storage accounts:** A standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a standard storage account should not exceed this limit.

You can roughly calculate the number of highly utilized disks supported by a single standard storage account based on the request rate limit. For example, for a Basic Tier VM, the maximum number of highly utilized disks is about 66 ($20,000/300$ IOPS per disk), and for a Standard Tier VM, it is about 40 ($20,000/500$ IOPS per disk).

- For premium storage accounts:** A premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

See [Virtual machine sizes](#) for additional details.

Managed virtual machine disks

Standard managed virtual machine disks

STANDARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50
Disk size	32 GB	64 GB	128 GB	256 GB	512 GB	1024 GB (1 TB)	2048 GB (2TB)	4095 GB (4 TB)
IOPS per disk	500	500	500	500	500	500	500	500
Throughput per disk	60 MB/sec	60 MB/sec	60 MB/sec					

Premium managed virtual machine disks: per disk limits

PREMIUM DISKS TYPE	P4	P6	P10	P15	P20	P30	P40	P50
Disk size	32 GB	64 GB	128 GB	256 GB	512 GB	1024 GB (1 TB)	2048 GB (2 TB)	4095 GB (4 TB)
IOPS per disk	120	240	500	1100	2300	5000	7500	7500
Throughput per disk	25 MB/sec	50 MB/sec	100 MB/sec	125MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec

Premium managed virtual machine disks: per VM limits

RESOURCE	DEFAULT LIMIT
Max IOPS Per VM	80,000 IOPS with GS5 VM
Max throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: per disk limits

VM TIER	BASIC TIER VM	STANDARD TIER VM
Disk size	4095 GB	4095 GB
Max 8 KB IOPS per persistent disk	300	500
Max number of disks performing max IOPS	66	40

Premium unmanaged virtual machine disks: per account limits

RESOURCE	DEFAULT LIMIT
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB
Max bandwidth per account (ingress + egress ¹)	<=50 Gbps

¹Ingress refers to all data (requests) being sent to a storage account. Egress refers to all data (responses) being received from a storage account.

Premium unmanaged virtual machine disks: per disk limits

PREMIUM STORAGE DISK TYPE	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1024 GiB (1 TB)	2048 GiB (2 TB)	4095 GiB (4 TB)
Max IOPS per disk	500	2300	5000	7500	7500
Max throughput per disk	100 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s
Max number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: per VM limits

RESOURCE	DEFAULT LIMIT
Max IOPS Per VM	80,000 IOPS with GS5 VM

RESOURCE	DEFAULT LIMIT
Max throughput per VM	2,000 MB/s with GS5 VM

Cloud Services limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Web/worker roles per deployment ¹	25	25
Instance Input Endpoints per deployment	25	25
Input Endpoints per deployment	25	25
Internal Endpoints per deployment	25	25

¹Each Cloud Service with Web/Worker roles can have two deployments, one for production and one for staging. Also note that this limit refers to the number of distinct roles (configuration) and not the number of instances per role (scaling).

App Service limits

The following App Service limits include limits for Web Apps, Mobile Apps, and API Apps.

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
Web, mobile, or API apps per App Service plan ¹	10	100	Unlimited ²	Unlimited ²	Unlimited ²	Unlimited ²
App Service plan	1 per region	10 per resource group	100 per resource group	100 per resource group	100 per resource group	100 per resource group
Compute instance type	Shared	Shared	Dedicated ³	Dedicated ³	Dedicated ³	Dedicated ³
Scale-Out (max instances)	1 shared	1 shared	3 dedicated ³	10 dedicated ³	20 dedicated ³	100 dedicated ⁴
Storage ⁵	1 GB ⁵	1 GB ⁵	10 GB ⁵	50 GB ⁵	250 GB ⁵	1 TB ⁵
CPU time (5 min) ⁶	3 minutes	3 minutes	Unlimited, pay at standard rates			
CPU time (day) ⁶	60 minutes	240 minutes	Unlimited, pay at standard rates			
Memory (1 hour)	1024 MB per App Service plan	1024 MB per app	N/A	N/A	N/A	N/A

Resource	Free	Shared	Basic	Standard	Premium (V2)	Isolated
Bandwidth	165 MB	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply
Application architecture	32-bit	32-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit
Web Sockets per instance ⁷	5	35	350	Unlimited	Unlimited	Unlimited
Concurrent debugger connections per application	1	1	1	5	5	5
Custom domain SSL support	Not supported. Wildcard certificate for *.azurewebsite.s.net available by default.	Not supported. Wildcard certificate for *.azurewebsite.s.net available by default.	Unlimited SNI SSL connections	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included
Integrated Load Balancer		X	X	X	X	X ⁹
Always On			X	X	X	X
Scheduled Backups				Scheduled backups every 2 hours, a max of 12 backups per day (manual + scheduled)	Scheduled backups every hour, a max of 50 backups per day (manual + scheduled)	Scheduled backups every hour, a max of 50 backups per day (manual + scheduled)
Auto Scale				X	X	X
WebJobs ⁸	X	X	X	X	X	X
Azure Scheduler support		X	X	X	X	X
Endpoint monitoring			X	X	X	X
Staging Slots				5	20	20
Custom domains per app	0 (azurewebsites.net subdomain only)	500	500	500	500	500

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
SLA			99.9%	99.95%	99.95%	99.95%

¹Apps and storage quotas are per App Service plan unless noted otherwise.

²The actual number of apps that you can host on these machines depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.

³Dedicated instances can be of different sizes. See [App Service Pricing](#) for more details.

⁴More allowed upon request. ⁵The storage limit is the total content size across all apps in the same App Service plan. More storage options are available in [App Service Environment](#)

⁶These resources are constrained by physical resources on the dedicated instances (the instance size and the number of instances).

⁷If you scale an app in the Basic tier to two instances, you have 350 concurrent connections for each of the two instances.

⁸Run custom executables and/or scripts on demand, on a schedule, or continuously as a background task within your App Service instance. Always On is required for continuous WebJobs execution. Azure Scheduler Free or Standard is required for scheduled WebJobs. There is no predefined limit on the number of WebJobs that can run in an App Service instance, but there are practical limits that depend on what the application code is trying to do.

⁹App Service Isolated SKUs have the ability to be internally load balanced (ILB) with Azure Load Balancer, which means no public connectivity from the internet. As a result, some features of an ILB Isolated App Service must be used from machines that have direct access to the ILB network endpoint.

Scheduler limits

The following table describes each of the major quotas, limits, defaults, and throttles in Azure Scheduler.

RESOURCE	LIMIT DESCRIPTION
Job size	Maximum job size is 16K. If a PUT or a PATCH results in a job larger than these limits, a 400 Bad Request status code is returned.
Request URL size	Maximum size of the request URL is 2048 chars.
Aggregate header size	Maximum aggregate header size is 4096 chars.
Header count	Maximum header count is 50 headers.
Body size	Maximum body size is 8192 chars.
Recurrence span	Maximum recurrence span is 18 months.
Time to start time	Maximum "time to start time" is 18 months.
Job history	Maximum response body stored in job history is 2048 bytes.
Frequency	The default max frequency quota is 1 hour in a free job collection and 1 minute in a standard job collection. The max frequency is configurable on a job collection to be lower than the maximum. All jobs in the job collection are limited the value set on the job collection. If you attempt to create a job with a higher frequency than the maximum frequency on the job collection then request will fail with a 409 Conflict status code.

RESOURCE	LIMIT DESCRIPTION
Jobs	The default max jobs quota is 5 jobs in a free job collection and 50 jobs in a standard job collection. The maximum number of jobs is configurable on a job collection. All jobs in the job collection are limited the value set on the job collection. If you attempt to create more jobs than the maximum jobs quota, then the request fails with a 409 Conflict status code.
Job collections	Maximum number of job collection per subscription is 200,000.
Job history retention	Job history is retained for up to 2 months or up to the last 1000 executions.
Completed and faulted job retention	Completed and faulted jobs are retained for 60 days.
Timeout	There's a static (not configurable) request timeout of 60 seconds for HTTP actions. For longer running operations, follow HTTP asynchronous protocols; for example, return a 202 immediately but continue working in the background.

Batch limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Batch accounts per region per subscription	1 - 3	50
Dedicated cores per Batch account	10 - 100	N/A ¹
Low-priority cores per Batch account	10 - 100	N/A ²
Active jobs and job schedules ³ per Batch account	100 - 300	2500 ⁴
Pools per Batch account	20 - 100	500

NOTE

Default limits vary depending on the type of subscription you use to create a Batch account. Cores quotas shown are for Batch accounts in Batch service mode. [View the quotas in your Batch account](#).

¹ The number of dedicated cores per Batch account can be increased, but the maximum number is unspecified. Contact Azure support to discuss increase options.

² The number of low-priority cores per Batch account can be increased, but the maximum number is unspecified. Contact Azure support to discuss increase options.

³ Completed jobs and job schedules are not limited.

⁴ Contact Azure support if you want to request an increase beyond this limit.

Batch AI limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Dedicated cores per region	10 - 24	N/A ¹
Low-priority cores per region	10 - 24	N/A ²
Clusters per region	20	200 ³

NOTE

Default limits vary depending on the type of subscription you have.

¹ The number of dedicated cores per region can be increased, but the maximum number is unspecified. Contact Azure support to discuss increase options.

² The number of low-priority cores per region can be increased, but the maximum number is unspecified. Contact Azure support to discuss increase options.

³ Contact Azure support if you want to request an increase beyond this limit.

BizTalk Services limits

The following table shows the limits for Azure Biztalk Services.

RESOURCE	FREE (PREVIEW)	DEVELOPER	BASIC	STANDARD	PREMIUM
Scale out	N/A	N/A	Yes, in increments of 1 Basic Unit	Yes, in increments of 1 Standard Unit	Yes, in increments of 1 Premium Unit
Scale Limit	N/A	N/A	Up to 8 units	Up to 8 units	Up to 8 units
EAI Bridges per Unit	N/A	25	25	125	500
EDI Agreements per Unit	N/A	10	50	250	1000
Hybrid Connections per Unit	5	5	10	50	100
Hybrid Connection Data Transfer (GBs) per Unit	5	5	50	250	500
Number of connections using BizTalk Adapter Service per Unit	N/A	1	2	5	25
Archiving	N/A	Available	N/A	N/A	Available
High Availability	N/A	N/A	Available	Available	Available

Azure Cosmos DB limits

Azure Cosmos DB is a global scale database in which throughput and storage can be scaled to handle whatever your application requires. If you have any questions about the scale Azure Cosmos DB provides, please send email to askcosmosdb@microsoft.com.

Azure Database for MySQL

For Azure Database for MySQL limits, see [Limitations in Azure Database for MySQL](#).

Azure Database for PostgreSQL

For Azure Database for PostgreSQL limits, see [Limitations in Azure Database for PostgreSQL](#).

Mobile Engagement limits

RESOURCE	MAXIMUM LIMIT
App Collection Users	5 per App Collection
Average Data points	200 per Active User/Day
Average App-Info set	50 per Active User/Day
Average Messages pushed	20 per Active User/Day
Segments	100 per app
Criteria per segment	10
Active Push Campaigns	50 per app
Total Push Campaigns (includes Active & Completed)	1000 per app

Search limits

Pricing tiers determine the capacity and limits of your search service. Tiers include:

- *Free* multi-tenant service, shared with other Azure subscribers, intended for evaluation and small development projects.
- *Basic* provides dedicated computing resources for production workloads at a smaller scale, with up to three replicas for highly available query workloads.
- *Standard (S1, S2, S3, S3 High Density)* is for larger production workloads. Multiple levels exist within the standard tier so that you can choose a resource configuration that best matches your workload profile.

Limits per subscription

You can create multiple services within a subscription, each one provisioned at a specific tier, limited only by the number of services allowed at each tier. For example, you could create up to 12 services at the Basic tier and another 12 services at the S1 tier within the same subscription. For more information about tiers, see [Choose a SKU or tier for Azure Search](#).

Maximum service limits can be raised upon request. Contact Azure Support if you need more services within the same subscription.

RESOURCE	FREE ¹	BASIC	S1	S2	S3	S3 HD
Maximum services	1	12	12	6	6	6
Maximum scale in SU ²	N/A	3 SU	36 SU	36 SU	36 SU	36 SU

¹ Free is based on shared, not dedicated, resources. Scale-up is not supported on shared resources.

² Search units (SU) are billing units, allocated as either a *replica* or a *partition*. You need both resources for storage, indexing, and query operations. To learn more about SU computations, see [Scale resource levels for query and index workloads](#).

Limits per search service

Storage is constrained by disk space or by a hard limit on the *maximum number* of indexes, document, or other high-level resources, whichever comes first. The following table documents storage limits. For maximum limits on indexes, documents, and other objects, see [limits by resource](#).

RESOURCE	FREE	BASIC ¹	S1	S2	S3	S3 HD ²
Service Level Agreement (SLA) ³	No	Yes	Yes	Yes	Yes	Yes
Storage per partition	50 MB	2 GB	25 GB	100 GB	200 GB	200 GB
Partitions per service	N/A	1	12	12	12	3
Partition size	N/A	2 GB	25 GB	100 GB	200 GB	200 GB
Replicas	N/A	3	12	12	12	12

¹ Basic has one fixed partition. At this tier, additional SUs are used for allocating more replicas for increased query workloads.

² S3 HD has a hard limit of 3 partitions, which is lower than the partition limit for S3. The lower partition limit is imposed because the index count for S3 HD is substantially higher. Given that service limits exist for both computing resources (storage and processing) and content (indexes and documents), the content limit is reached first.

³ Service level agreements (SLAs) are offered for billable services on dedicated resources. Free services and preview features have no SLA. For billable services, SLAs take effect when you provision sufficient redundancy for your service. Two or more replicas are required for query (read) SLA. Three or more replicas are required for query and indexing (read-write) SLA. The number of partitions is not an SLA consideration.

To learn more about limits on a more granular level, such as document size, queries per second, keys, requests, and responses, see [Service limits in Azure Search](#).

Media Services limits

NOTE

For resources that are not fixed, you may ask for the quotas to be raised, by opening a support ticket. Do **not** create additional Azure Media Services accounts in an attempt to obtain higher limits.

RESOURCE	DEFAULT LIMIT
Azure Media Services (AMS) accounts in a single subscription	25 (fixed)
Media Reserved Units (RUs) per AMS account	25 (S1) 10 (S2, S3) ⁽¹⁾
Jobs per AMS account	50,000 ⁽²⁾
Chained tasks per job	30 (fixed)
Assets per AMS account	1,000,000
Assets per task	50
Assets per job	100
Unique locators associated with an asset at one time	5 ⁽⁴⁾
Live channels per AMS account	5
Programs in stopped state per channel	50
Programs in running state per channel	3
Streaming endpoints in running state per AMS account	2
Streaming units per streaming endpoint	10
Storage accounts	1,000 ⁽⁵⁾ (fixed)
Policies	1,000,000 ⁽⁶⁾
File size	In some scenarios, there is a limit on the maximum file size supported for processing in Media Services. ⁷

¹ If you change the type (for example, from S2 to S1,) the max RU limits are reset.

² This number includes queued, finished, active, and canceled jobs. It does not include deleted jobs. You can delete the old jobs using **IJob.Delete** or the **DELETE** HTTP request.

As of April 1, 2017, any Job record in your account older than 90 days will be automatically deleted, along with its associated Task records, even if the total number of records is below the maximum quota. If you need to archive the job/task information, you can use the code described [here](#).

³ When making a request to list Job entities, a maximum of 1,000 jobs is returned per request. If you need to keep track of all submitted Jobs, you can use top/skip as described in [OData system query options](#).

⁴ Locators are not designed for managing per-user access control. To give different access rights to individual users,

use Digital Rights Management (DRM) solutions. For more information, see [this](#) section.

⁵ The storage accounts must be from the same Azure subscription.

⁶ There is a limit of 1,000,000 policies for different AMS policies (for example, for Locator policy or ContentKeyAuthorizationPolicy).

NOTE

You should use the same policy ID if you are always using the same days / access permissions / etc. For information and an example, see [this](#) section.

⁷If you are uploading content to an Asset in Azure Media Services to process it with one of the media processors in the service (that is, encoders like Media Encoder Standard and Media Encoder Premium Workflow, or analysis engines like Face Detector), then you should be aware of the constraints on the maximum file sizes supported.

The maximum size supported for a single blob is currently up to 5 TB in Azure Blob Storage. However, additional limits apply in Azure Media Services based on the VM sizes that are used by the service. The following table shows the limits on each of the Media Reserved Units (S1, S2, S3.) If your source file is larger than the limits defined in the table, your encoding job will fail. If you are encoding 4K resolution sources of long duration, you are required to use S3 Media Reserved Units to achieve the performance needed. If you have 4K content that is larger than 260 GB limit on the S3 Media Reserved Units, contact us at amshelp@microsoft.com for potential mitigations to support your scenario.

MEDIA RESERVED UNIT TYPE	MAXIMUM INPUT SIZE (GB)
S1	325
S2	640
S3	260

CDN limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
CDN profiles	25	25
CDN endpoints per profile	10	25
Custom domains per endpoint	10	25

A CDN subscription can contain one or more CDN profiles and a CDN profile can contain one or more CDN endpoints. You may wish to use multiple profiles to organize your CDN endpoints by internet domain, web application, or some other criteria.

To request an update to your subscription's default limits, open a support ticket.

Mobile Services limits

TIER:	FREE	BASIC	STANDARD
API Calls	500 K	1.5 M / unit	15 M / unit
Active Devices	500	Unlimited	Unlimited

TIER:	FREE	BASIC	STANDARD
Scale	N/A	Up to 6 units	Unlimited units
Push Notifications	Notification Hubs Free Tier included, up to 1 M pushes	Notification Hubs Basic Tier included, up to 10 M pushes	Notification Hubs Standard Tier included, up to 10 M pushes
Real time messaging/ Web Sockets	Limited	350 / mobile service	Unlimited
Offline synchronizations	Limited	Included	Included
Scheduled jobs	Limited	Included	Included
SQL Database (required) Standard rates apply for additional capacity	20 MB included	20 MB included	20 MB included
CPU capacity	60 minutes / day	Unlimited	Unlimited
Outbound data transfer	165 MB per day (daily Rollover)	Included	Included

For additional details on these limits and for information on pricing, see [Mobile Services Pricing](#).

Monitor limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Autoscale Settings	100 per region per subscription	same as default
Metric Alerts (classic)	100 active alert rules per subscription	call support
Metric Alerts	100 active alert rules per subscription	call support
Action Groups	2000 action groups per subscription	call support

Notification Hub Service limits

TIER:	FREE	BASIC	STANDARD
Included Pushes	1 Million	10 Million	10 Million
Active Devices	500	200,000	10 million
Tag quota per installation/registration	60	60	60

For additional details on these limits and for information on pricing, see [Notification Hubs Pricing](#).

Event Hubs limits

The following table lists quotas and limits specific to [Azure Event Hubs](#). For information about Event Hubs pricing, see [Event Hubs pricing](#).

LIMIT	SCOPE	NOTES	VALUE
Number of event hubs per namespace	Namespace	Subsequent requests for creation of a new event hub will be rejected.	10
Number of partitions per event hub	Entity	-	32
Number of consumer groups per event hub	Entity	-	20
Number of AMQP connections per namespace	Namespace	Subsequent requests for additional connections will be rejected and an exception is received by the calling code.	5,000
Maximum size of Event Hubs event	Entity	-	256 KB
Maximum size of an event hub name	Entity	-	50 characters
Number of non-epoch receivers per consumer group	Entity	-	5
Maximum retention period of event data	Entity	-	1-7 days
Maximum throughput units	Namespace	Exceeding the throughput unit limit causes your data to be throttled and generates a ServerBusyException . You can request a larger number of throughput units for a Standard tier by filing a support request . Additional throughput units are available in blocks of 20 on a committed purchase basis.	20
Number of authorization rules per namespace	Namespace	Subsequent requests for authorization rule creation will be rejected.	12

Service Bus limits

The following table lists quota information specific to Service Bus messaging. For information about pricing and other quotas for Service Bus, see the [Service Bus Pricing](#) overview.

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum number of basic / standard namespaces per Azure subscription	Namespace	Subsequent requests for additional basic / standard namespaces are rejected by the portal.	100

Quota Name	Scope	Notes	Value
Maximum number of premium namespaces per Azure subscription	Namespace	Subsequent requests for additional premium namespaces are rejected by the portal.	10
Queue/topic size	Entity	Defined upon creation of the queue/topic. Subsequent incoming messages are rejected and an exception is received by the calling code.	1, 2, 3, 4 GB or 5 GB. In the Premium SKU, as well as Standard with partitioning enabled, the maximum queue/topic size is 80 GB.
Number of concurrent connections on a namespace	Namespace	Subsequent requests for additional connections are rejected and an exception is received by the calling code. REST operations do not count towards concurrent TCP connections.	NetMessaging: 1,000 AMQP: 5,000
Number of concurrent receive requests on a queue/topic/subscription entity	Entity	Subsequent receive requests are rejected and an exception is received by the calling code. This quota applies to the combined number of concurrent receive operations across all subscriptions on a topic.	5,000
Number of topics/queues per namespace	Namespace	Subsequent requests for creation of a new topic or queue on the namespace are rejected. As a result, if configured through the Azure portal , an error message is generated. If called from the management API, an exception is received by the calling code.	10,000 (Basic/Standard tier). The total number of topics and queues in a namespace must be less than or equal to 10,000. For premium tier, 1000 per messaging unit(MU). Maximum limit is 4000.
Number of partitioned topics/queues per namespace	Namespace	Subsequent requests for creation of a new partitioned topic or queue on the namespace are rejected. As a result, if configured through the Azure portal , an error message is generated. If called from the management API, a QuotaExceededException exception is received by the calling code.	Basic and Standard Tiers - 100 Partitioned entities are not supported in the Premium tier. Each partitioned queue or topic counts towards the quota of 10,000 entities per namespace.
Maximum size of any messaging entity path: queue or topic	Entity	-	260 characters

Quota Name	Scope	Notes	Value
Maximum size of any messaging entity name: namespace, subscription, or subscription rule	Entity	-	50 characters
Maximum size of a message SessionID	Entity	-	128
Message size for a queue/topic/subscription entity	Entity	<p>Incoming messages that exceed these quotas are rejected and an exception is received by the calling code.</p> <p>Due to system overhead, this limit is less than these values.</p> <p>Maximum header size: 64 KB</p> <p>Maximum number of header properties in property bag: byte/int.MaxValue</p> <p>Maximum size of property in property bag: No explicit limit. Limited by maximum header size.</p>	<p>Maximum message size: 256 KB (Standard tier) / 1 MB (Premium tier).</p>
Message property size for a queue/topic/subscription entity	Entity	A SerializationException exception is generated.	<p>Maximum message property size for each property is 32 K. Cumulative size of all properties cannot exceed 64 K. This limit applies to the entire header of the BrokeredMessage, which has both user properties as well as system properties (such as SequenceNumber, Label, MessageId, and so on).</p>
Number of subscriptions per topic	Entity	Subsequent requests for creating additional subscriptions for the topic are rejected. As a result, if configured through the portal, an error message is shown. If called from the management API an exception is received by the calling code.	2,000
Number of SQL filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected and an exception is received by the calling code.	2,000

Quota name	Scope	Notes	Value
Number of correlation filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected and an exception is received by the calling code.	100,000
Size of SQL filters/actions	Namespace	Subsequent requests for creation of additional filters are rejected and an exception is received by the calling code.	Maximum length of filter condition string: 1024 (1 K). Maximum length of rule action string: 1024 (1 K). Maximum number of expressions per rule action: 32.
Number of SharedAccessAuthorizationRule rules per namespace, queue, or topic	Entity, namespace	Subsequent requests for creation of additional rules are rejected and an exception is received by the calling code.	Maximum number of rules: 12. Rules that are configured on a Service Bus namespace apply to all queues and topics in that namespace.
Number of messages per transaction	Transaction	Additional incoming messages are rejected and an exception stating "Cannot send more than 100 messages in a single transaction" is received by the calling code.	100 For both Send() and SendAsync() operations.

IoT Hub limits

The following table lists the limits associated with the different service tiers (S1, S2, S3, F1). For information about the cost of each *unit* in each tier, see [IoT Hub Pricing](#).

Resource	S1 Standard	S2 Standard	S3 Standard	F1 Free
Messages/day	400,000	6,000,000	300,000,000	8,000
Maximum units	200	200	10	1

Note

If you anticipate using more than 200 units with an S1 or S2 or 10 units with an S3 tier hub, contact Microsoft support.

The following table lists the limits that apply to IoT Hub resources:

Resource	Limit
Maximum paid IoT hubs per Azure subscription	50
Maximum free IoT hubs per Azure subscription	1

RESOURCE	LIMIT
Maximum number of characters in a Device Id	128
Maximum number of device identities returned in a single call	1000
IoT Hub message maximum retention for device-to-cloud messages	7 days
Maximum size of device-to-cloud message	256 KB
Maximum size of device-to-cloud batch	256 KB
Maximum messages in device-to-cloud batch	500
Maximum size of cloud-to-device message	64 KB
Maximum TTL for cloud-to-device messages	2 days
Maximum delivery count for cloud-to-device messages	100
Maximum delivery count for feedback messages in response to a cloud-to-device message	100
Maximum TTL for feedback messages in response to a cloud-to-device message	2 days
Maximum size of device twin (tags, reported properties, and desired properties)	8 KB
Maximum size of device twin string value	4 KB
Maximum depth of object in device twin	5
Maximum size of direct method payload	128 KB
Job history maximum retention	30 days
Maximum concurrent jobs	10 (for S3), 5 for (S2), 1 (for S1)
Maximum additional endpoints	10 (for S1, S2, S3)
Maximum message routing rules	100 (for S1, S2, S3)

NOTE

If you need more than 50 paid IoT hubs in an Azure subscription, contact Microsoft support.

NOTE

Currently, the maximum number of devices you can connect to a single IoT hub is 500,000. If you want to increase this limit, contact [Microsoft Support](#).

The IoT Hub service throttles requests when the following quotas are exceeded:

THROTTLE	PER-HUB VALUE
Identity registry operations (create, retrieve, list, update, delete), individual or bulk import/export	83.33/sec/unit (5000/min/unit) (for S3) 1.67/sec/unit (100/min/unit) (for S1 and S2).
Device connections	6000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Device-to-cloud sends	6000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Cloud-to-device sends	83.33/sec/unit (5000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S1 and S2).
Cloud-to-device receives	833.33/sec/unit (50000/min/unit) (for S3), 16.67/sec/unit (1000/min/unit) (for S1 and S2).
File upload operations	83.33 file upload notifications/sec/unit (5000/min/unit) (for S3), 1.67 file upload notifications/sec/unit (100/min/unit) (for S1 and S2). 10000 SAS URIs can be out for an Azure Storage account at one time. 10 SAS URIs/device can be out at one time.
Direct methods	24MB/sec/unit (for S3), 480KB/sec/unit (for S2), 160KB/sec/unit (for S1) Based on 8KB throttling meter size.
Device twin reads	50/sec/unit (for S3), Maximum of 10/sec or 1/sec/unit (for S2), 10/sec (for S1)
Device twin updates	50/sec/unit (for S3), Maximum of 10/sec or 1/sec/unit (for S2), 10/sec (for S1)
Jobs operations (create, update, list, delete)	83.33/sec/unit (5000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S2), 1.67/sec/unit (100/min/unit) (for S1)
Jobs per-device operation throughput	50/sec/unit (for S3), Maximum of 10/sec or 1/sec/unit (for S2), 10/sec (for S1)

IoT Hub Device Provisioning Service limits

The following table lists the limits that apply to IoT Hub Device Provisioning Service resources:

RESOURCE	LIMIT
Maximum Device Provisioning Services per Azure subscription	10
Maximum number of enrollments	500,000
Maximum number of registrations	500,000
Maximum number of enrollment groups	100
Maximum number of CAs	25

NOTE

You can contact [Microsoft Support](#) to increase the number of instances in your subscription.

NOTE

You can contact [Microsoft Support](#) to increase the number of enrollments and registrations on your provisioning service.

The Device Provisioning Service throttles requests when the following quotas are exceeded:

THROTTLE	PER-SERVICE VALUE
Operations	100/min
Device registrations	100/min

Data Factory limits

Data factory is a multi-tenant service that has the following default limits in place to make sure customer subscriptions are protected from each other's workloads. Many of the limits can be easily raised for your subscription up to the maximum limit by contacting support.

Version 2

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Data factories in an Azure subscription	50	Contact support
Total number of entities (Pipeline, Datasets, Triggers, Linked Services, Integration runtimes) within a data factory	5000	Contact support
Total CPU cores for Azure-SSIS Integration Runtime(s) under one subscription	128	Contact support
Concurrent pipeline runs per pipeline	100	Contact support
Concurrent pipeline runs per data factory	10,000	Contact support

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Max activities per pipeline (includes inner activities for containers)	40	40
Max parameters per pipeline	50	50
ForEach items	100,000	100,000
ForEach parallelism	20	50
Characters per expression	8,192	8,192
Minimum Tumbling Window Trigger interval	15 min	15 min
Max Timeout for pipeline activity runs	7 days	7 days
Bytes per object for pipeline objects ¹	200 KB	200 KB
Bytes per object for dataset and linked service objects ¹	100 KB	2000 KB
Data integration units per copy activity run ³	256	Contact support
Write API calls	2500/hr This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support .
Read API calls	12,500/hr This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support
Monitoring queries per minute	1000	Contact support
Entity CRUD operations per minute	50	Contact support

Version 1

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Data factories in an Azure subscription	50	Contact support
Pipelines within a data factory	2500	Contact support
Datasets within a data factory	5000	Contact support
Concurrent slices per dataset	10	10
Bytes per object for pipeline objects ¹	200 KB	200 KB

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Bytes per object for dataset and linked service objects ¹	100 KB	2000 KB
HDInsight on-demand cluster cores within a subscription ²	60	Contact support
Cloud data movement units per copy activity run ³	32	Contact support
Retry count for pipeline activity runs	1000	MaxInt (32 bit)

¹ Pipeline, dataset, and linked service objects represent a logical grouping of your workload. Limits for these objects do not relate to amount of data you can move and process with the Azure Data Factory service. Data factory is designed to scale to handle petabytes of data.

² On-demand HDInsight cores are allocated out of the subscription that contains the data factory. As a result, the above limit is the Data Factory enforced core limit for on-demand HDInsight cores and is different from the core limit associated with your Azure subscription.

³ Data Integration Unit (DIU) for v2 or Cloud Data Movement Unit (DMU) for v1 is being used in a cloud-to-cloud copy operation. It is a measure that represents the power (a combination of CPU, memory, and network resource allocation) of a single unit in Data Factory. You can achieve higher copy throughput by using more DMUs for some scenarios. Refer to [Data integration units \(V2\)](#) and [Cloud data movement units \(V1\)](#) section on details, and [Azure Data Factory pricing page](#) for billing implication.

⁴ The Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory to provide the following data integration capabilities across different network environments: data movement, dispatching activities to compute services, execution of SSIS packages. For more information, see [Integration Runtime overview](#).

RESOURCE	DEFAULT LOWER LIMIT	MINIMUM LIMIT
Scheduling interval	15 minutes	15 minutes
Interval between retry attempts	1 second	1 second
Retry timeout value	1 second	1 second

Web service call limits

Azure Resource Manager has limits for API calls. You can make API calls at a rate within the [Azure Resource Manager API limits](#).

Data Lake Analytics limits

Data Lake Analytics makes the complex task of managing distributed infrastructure and complex code easy. It dynamically provisions resources and lets you do analytics on exabytes of data. When the job completes, it winds down resources automatically, and you pay only for the processing power used. As you increase or decrease the size of data stored or the amount of compute used, you don't have to rewrite code. Many of the default limits can be easily raised for your subscription by contacting support.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of concurrent jobs	20	

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of Analytics Units (AUs) per account	250	Use any combination of up to a maximum of 250 AUs across 20 jobs. Contact Microsoft support to increase this limit.
Maximum script size for job submission	3 MB	
Maximum number of ADLA accounts per region per subscription	5	Contact Microsoft support to increase this limit.

Data Lake Store limits

Azure Data Lake Store is an enterprise-wide hyper-scale repository for big data analytic workloads. Data Lake Store enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. There is no limit to the amount of data you can store in a Data Lake Store account.

RESOURCE	DEFAULT LIMIT	COMMENTS
Max number of Data Lake Store accounts, per subscription, per region	10	Contact Support to request an increase for this limit
Max number of access ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries
Max number of default ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries

Database Migration Service Limits

The Azure Database Migration Service is a fully managed service designed to enable seamless migrations from multiple database sources to Azure Data platforms with minimal downtime.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of services per subscription, per region	2	Contact Support to request an increase for this limit

Stream Analytics limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of Streaming Units per subscription per region	200	A request to increase streaming units for your subscription beyond 200 can be made by contacting Microsoft Support .
Maximum number of inputs per job	60	There is a hard limit of 60 inputs per Stream Analytics job.
Maximum number of outputs per job	60	There is a hard limit of 60 outputs per Stream Analytics job.
Maximum number of functions per job	60	There is a hard limit of 60 functions per Stream Analytics job.

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of Streaming Units per job	120	There is a hard limit of 120 Streaming Units per Stream Analytics job.
Maximum number of jobs per region	1500	Each subscription may have up to 1500 jobs per geographical region.
Reference data blob MB	100	Reference data blobs cannot be larger than 100 MB each.

Active Directory limits

Here are the usage constraints and other service limits for the Azure Active Directory (Azure AD) service.

CATEGORY	LIMITS
Directories	A single user can belong to a maximum of 500 Azure AD directories as a member or a guest. A single user can create a maximum of 20 directories.
Domains	You can add no more than 900 managed domain names. If you're setting up all of your domains for federation with on-premises Active Directory, you can add no more than 450 domain names in each directory.
Objects	<ul style="list-style-type: none"> A maximum of 500,000 objects can be created in a single directory by users of the Free edition of Azure Active Directory. A non-admin user can create no more than 250 objects.
Schema extensions	<ul style="list-style-type: none"> String type extensions can have maximum of 256 characters. Binary type extensions are limited to 256 bytes. 100 extension values (across ALL types and ALL applications) can be written to any single object. Only "User", "Group", "TenantDetail", "Device", "Application" and "ServicePrincipal" entities can be extended with "String" type or "Binary" type single-valued attributes. Schema extensions are available only in Graph API-version 1.21-preview. The application must be granted write access to register an extension.
Applications	A maximum of 100 users can be owners of a single application.
Groups	<ul style="list-style-type: none"> A maximum of 100 users can be owners of a single group. Any number of objects can be members of a single group in Azure Active Directory. The number of members in a group you can synchronize from your on-premises Active Directory to Azure Active Directory using Azure AD Connect is limited to 50 K members.

CATEGORY	LIMITS
Access Panel	<ul style="list-style-type: none"> There is no limit to the number of applications that can be seen in the Access Panel per end user, for users assigned licenses for Azure AD Premium or the Enterprise Mobility Suite. A maximum of 10 app tiles (examples: Box, Salesforce, or Dropbox) can be seen in the Access Panel for each end user for users assigned licenses for Free or Azure AD Basic editions of Azure Active Directory. This limit does not apply to Administrator accounts.
Reports	A maximum of 1,000 rows can be viewed or downloaded in any report. Any additional data is truncated.
Administrative units	An object can be a member of no more than 30 administrative units.

Azure Event Grid limits

RESOURCE	LIMIT
Custom topics per Azure subscription	100
Event subscriptions per topic	500
Publish rate for a custom topic (ingress)	5,000 events per second per topic

Azure Maps limits

Here are the usage constraints for the Azure Maps service. For information about the cost, see [Azure Maps pricing details](#). [Contact us](#) to increase maximum request rate for your subscription.

RESOURCE	LIMIT
Maximum request rate per subscription	50 requests per second

Azure Policy limits

There is a maximum count for each object type for Azure Policy. An entry of *Scope* means either the subscription or the [management group](#).

WHERE	WHAT	MAXIMUM COUNT
Scope	Policy Definitions	250
Scope	Initiative Definitions	100
Tenant	Initiative Definitions	1000
Scope	Policy/Initiative Assignments	100
Policy Definition	Parameters	20
Initiative Definition	Policies	100

WHERE	WHAT	MAXIMUM COUNT
Initiative Definition	Parameters	100
Policy/Initiative Assignments	Exclusions (notScopes)	100
Policy Rule	Nested Conditionals	512

StorSimple System limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of storage account credentials	64	
Maximum number of volume containers	64	
Maximum number of volumes	255	
Maximum number of schedules per bandwidth template	168	A schedule for every hour, every day of the week (24*7).
Maximum size of a tiered volume on physical devices	64 TB for 8100 and 8600	8100 and 8600 are physical devices.
Maximum size of a tiered volume on virtual devices in Azure	30 TB for 8010 64 TB for 8020	8010 and 8020 are virtual devices in Azure that use Standard Storage and Premium Storage respectively.
Maximum size of a locally pinned volume on physical devices	9 TB for 8100 24 TB for 8600	8100 and 8600 are physical devices.
Maximum number of iSCSI connections	512	
Maximum number of iSCSI connections from initiators	512	
Maximum number of access control records per device	64	
Maximum number of volumes per backup policy	24	
Maximum number of backups retained per backup policy	64	
Maximum number of schedules per backup policy	10	
Maximum number of snapshots of any type that can be retained per volume	256	This includes local snapshots and cloud snapshots.
Maximum number of snapshots that can be present in any device	10,000	

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of volumes that can be processed in parallel for backup, restore, or clone	16	<ul style="list-style-type: none"> If there are more than 16 volumes, they will be processed sequentially as processing slots become available. New backups of a cloned or a restored tiered volume cannot occur until the operation is finished. However, for a local volume, backups are allowed after the volume is online.
Restore and clone recover time for tiered volumes	< 2 minutes	<ul style="list-style-type: none"> The volume is made available within 2 minutes of restore or clone operation, regardless of the volume size. The volume performance may initially be slower than normal as most of the data and metadata still resides in the cloud. Performance may increase as data flows from the cloud to the StorSimple device. The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate may be affected by Internet bandwidth to the cloud. The restore or clone operation is complete when all the metadata is on the device. Backup operations cannot be performed until the restore or clone operation is fully complete.

LIMIT IDENTIFIER	LIMIT	COMMENTS
Restore recover time for locally pinned volumes	< 2 minutes	<ul style="list-style-type: none"> The volume is made available within 2 minutes of the restore operation, regardless of the volume size. The volume performance may initially be slower than normal as most of the data and metadata still resides in the cloud. Performance may increase as data flows from the cloud to the StorSimple device. The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate may be affected by Internet bandwidth to the cloud. Unlike tiered volumes, in the case of locally pinned volumes, the volume data is also downloaded locally on the device. The restore operation is complete when all the volume data has been brought to the device. The restore operations may be long and the total time to complete the restore will depend on the size of the provisioned local volume, your Internet bandwidth and the existing data on the device. Backup operations on the locally pinned volume are allowed while the restore operation is in progress.
Thin-restore availability	Last failover	
Maximum client read/write throughput (when served from the SSD tier)*	920/720 MB/s with a single 10GbE network interface	Up to 2x with MPIO and two network interfaces.
Maximum client read/write throughput (when served from the HDD tier)*	120/250 MB/s	
Maximum client read/write throughput (when served from the cloud tier)*	11/41 MB/s	Read throughput depends on clients generating and maintaining sufficient I/O queue depth.

* Maximum throughput per I/O type was measured with 100 percent read and 100 percent write scenarios. Actual throughput may be lower and depends on I/O mix and network conditions.

Log Analytics limits

The following limits apply to Log Analytics resources per subscription:

RESOURCE	DEFAULT LIMIT	COMMENTS
Number of free workspaces per subscription	10	This limit cannot be increased.
Number of paid workspaces per subscription	N/A	You are limited by the number of resources within a resource group and number of resource groups per subscription

NOTE

As of April 2, 2018, new workspaces in a new subscription will automatically use the *Per GB* pricing plan. For existing subscriptions created before April 2, or a subscription that was tied to an existing EA enrollment, you can continue choosing between the three pricing tiers for new workspaces.

The following limits apply to each Log Analytics workspace:

	FREE	STANDARD	PREMIUM	STANDALONE	OMS	PER GB
Data volume collected per day	500 MB ¹	None	None	None	None	None
Data retention period	7 days	1 month	12 months	1 month ²	1 month ²	1 month ²

¹ When customers reach their 500 MB daily data transfer limit, data analysis stops and resumes at the start of the next day. A day is based on UTC.

² The data retention period for the Standalone, OMS, and Per GB pricing plans can be increased to 730 days.

CATEGORY	LIMITS	COMMENTS
Data Collector API	Maximum size for a single post is 30 MB Maximum size for field values is 32 KB	Split larger volumes into multiple posts Fields longer than 32 KB are truncated.
Search API	5000 records returned for non-aggregated data 500000 records for aggregated data	Aggregated data is a search that includes the <code>summarize</code> command

Backup limits

The following limits apply to Azure Backup.

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of servers/machines that can be registered against each vault	50 for Windows Server/Client/SCDPM 1000 for IaaS VMs
Size of a data source for data stored in Azure vault storage	54400 GB max ¹
Number of backup vaults that can be created in each Azure subscription	500 Recovery Services vaults per region

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of times backup can be scheduled per day	3 per day for Windows Server/Client 2 per day for SCDPM Once a day for IaaS VMs
Data disks attached to an Azure virtual machine for backup	16
Size of individual data disk attached to an Azure virtual machine for backup	4095 GB

- ¹The 54400 GB limit does not apply to IaaS VM backup.

Site Recovery limits

The following limits apply to Azure Site Recovery:

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of vaults per subscription	25
Number of servers per Azure vault	250
Number of protection groups per Azure vault	No limit
Number of recovery plans per Azure vault	No limit
Number of servers per protection group	No limit
Number of servers per recovery plan	50

Application Insights limits

There are some limits on the number of metrics and events per application (that is, per instrumentation key). Limits depend on the [pricing plan](#) that you choose.

RESOURCE	DEFAULT LIMIT	NOTE
Total data per day	100 GB	You can reduce data by setting a cap. If you need more data, you can increase the limit in the portal, up to 1,000 GB. For capacities greater than 1,000 GB, send mail to AIDataCap@microsoft.com .
Throttling	32 K events/second	The limit is measured over a minute.
Data retention	90 days	This resource is for Search , Analytics , and Metrics Explorer .
Availability multi-step test detailed results retention	90 days	This resource provides detailed results of each step.
Maximum event size	64 K	
Property and metric name length	150	See type schemas .

RESOURCE	DEFAULT LIMIT	NOTE
Property value string length	8,192	See type schemas .
Trace and exception message length	10 K	See type schemas .
Availability tests count per app	100	
Profiler data retention	5 days	
Profiler data sent per day	10 GB	

For more information, see [About pricing and quotas in Application Insights](#).

API Management limits

RESOURCE	LIMIT
Units of scale	10 per region ¹
Cache	5 GB per unit ¹
Concurrent backend connections ² per HTTP authority	2048 per unit ³
Maximum cached response size	10MB
Maximum policy document size	256KB
Maximum custom gateway domains	20 per service instance ⁴

¹API Management limits are different for each pricing tier. To see the pricing tiers and their scaling limits go to [API Management Pricing](#). ² Connections are pooled and re-used, unless explicitly closed by the backend. ³ Per unit of Basic, Standard and Premium tiers. Developer tier is limited to 1024. ⁴ Available in Premium tier only.

Azure Redis Cache limits

RESOURCE	LIMIT
Cache size	530 GB
Databases	64
Max connected clients	40,000
Redis Cache replicas (for high availability)	1
Shards in a premium cache with clustering	10

Azure Redis Cache limits and sizes are different for each pricing tier. To see the pricing tiers and their associated sizes, see [Azure Redis Cache Pricing](#).

For more information on Azure Redis Cache configuration limits, see [Default Redis server configuration](#).

Because configuration and management of Azure Redis Cache instances is done by Microsoft, not all Redis commands are supported in Azure Redis Cache. For more information, see [Redis commands not supported in](#)

Azure Redis Cache.

Key Vault limits

Key transactions (Max transactions allowed in 10 seconds, per vault per region¹):

KEY TYPE	HSM-KEY CREATE KEY	HSM-KEY ALL OTHER TRANSACTIONS	SOFTWARE-KEY CREATE KEY	SOFTWARE-KEY ALL OTHER TRANSACTIONS
RSA 2048-bit	5	1000	10	2000
RSA 3072-bit	5	250	10	500
RSA 4096-bit	5	125	10	250
ECC P-256	5	1000	10	2000
ECC P-384	5	1000	10	2000
ECC P-521	5	1000	10	2000
ECC SECP256K1	5	1000	10	2000

Secrets, Managed Storage Account Keys, and vault transactions:

TRANSACTIONS TYPE	MAX TRANSACTIONS ALLOWED IN 10 SECONDS, PER VAULT PER REGION ¹
All transactions	2000

See [Azure Key Vault throttling guidance](#) for information on how to handle throttling when these limits are exceeded.

¹ There is a subscription-wide limit for all transaction types, that is 5x per key vault limit. For example, HSM- other transactions per subscription are limited to 5000 transactions in 10 seconds per subscription.

Multi-Factor Authentication

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Max number of Trusted IP addresses/ranges per subscription	0	50
Remember my devices - number of days	14	60
Max number of app passwords?	0	No Limit
Allow X attempts during MFA call	1	99
Two-way Text message Timeout Seconds	60	600

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Default one-time bypass seconds	300	1800
Lock user account after X consecutive MFA denials	Not Set	99
Reset account lockout counter after X minutes	Not Set	9999
Unlock account after X minutes	Not Set	9999

Automation limits

RESOURCE	MAXIMUM LIMIT	NOTES
Max number of new jobs that can be submitted every 30 seconds per Automation Account (non Scheduled jobs)	100	When this limit is hit, the subsequent requests to create a job fail. The client receives an error response.
Max number of concurrent running jobs at the same instance of time per Automation Account (non Scheduled jobs)	200	When this limit is hit, the subsequent requests to create a job fail. The client receives an error response.
Max number of modules that can be imported every 30 seconds per Automation Account	5	
Max size of a Module	100 MB	
Job Run Time - Free tier	500 minutes per subscription per calendar month	
Max amount of disk space allowed per sandbox ¹	1 GB	Applies to Azure sandboxes only
Max amount of memory given to a sandbox ¹	400 MB	Applies to Azure sandboxes only
Max number of network sockets allowed per sandbox ¹	1000	Applies to Azure sandboxes only
Maximum runtime allowed per runbook ¹	3 hours	Applies to Azure sandboxes only
Max number of Automation Accounts in a subscription	No Limit	
Max number of concurrent jobs that can be run on a single Hybrid Runbook Worker	50	
Max Runbook Job parameters size	512 kb	

RESOURCE	MAXIMUM LIMIT	NOTES
Max Runbook parameters	50	You can pass a JSON or XML string to a parameter and parse it with the runbook if you hit the 50 parameter limit
Max webhook payload size	512 kb	

¹ A sandbox is a shared environment that can be used by multiple jobs, jobs using the same sandbox are bound by the resource limitations of the sandbox.

Managed Identity limits

CATEGORY	LIMIT
User assigned managed identities	<ul style="list-style-type: none"> When creating user assigned managed identities, only alphanumeric characters (0-9, a-z, A-Z) and the hyphen (-) are supported. Additionally, the name should be limited to 24 characters in length for the assignment to VM/VMSS to work properly. If using the managed identity virtual machine extension, the supported limit is 32 user assigned managed identities. Without the managed identity virtual machine extension, the supported limit is 512 user assigned identities.

Role-based access control limits

RESOURCE	LIMIT
Role assignments per Azure subscription	2000
Custom roles per tenant	2000

SQL Database limits

For SQL Database limits, see [SQL Database Resource Limits for single databases](#) and [SQL Database Resource Limits for elastic pools and pooled databases](#).

SQL Data Warehouse limits

For SQL Data Warehouse limits, see [SQL Data Warehouse Resource Limits](#).

See also

[Understanding Azure Limits and Increases](#)

[Virtual Machine and Cloud Service Sizes for Azure](#)

[Sizes for Cloud Services](#)