

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

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

About zone-redundant gateways for Availability Zones

Backend Connectivity Interoperability

Preface and Test Setup

Test Setup Configuration

Control Plane Analysis

Data Plane Analysis

Built-in security controls

How-to guides

Create and manage a VPN gateway

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](#)

[Manage legacy Gateway SKUs](#)

[Site-to-Site connections](#)

[Configure S2S connections](#)

[Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Configure multiple Site-to-Site connections](#)

[Connect to multiple policy-based VPN devices](#)

[Site-to-Site and ExpressRoute coexisting connections](#)

[VNet-to-VNet connections](#)

[Configure VNet-to-VNet connections](#)

[Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Configure a VNet-to-VNet connection between deployment models](#)

[Azure portal](#)

[Azure PowerShell](#)

[Point-to-Site connections](#)

[Azure certificate authentication](#)

[Configure a P2S VPN](#)

[Azure portal](#)

[Azure PowerShell](#)

[Configure P2S certificates and clients](#)

[Generate self-signed certificates](#)

[Azure PowerShell](#)

[Makecert](#)

[Linux](#)

- [Install client certificates](#)
- [Create and install VPN client configuration files](#)
- [RADIUS authentication](#)
 - [Configure a P2S VPN](#)
 - [Create and install VPN client configuration files](#)
 - [Integrate P2S VPN RADIUS authentication with NPS server](#)
- [Azure AD authentication](#)
 - [Configure a tenant](#)
 - [Configure a tenant with multiple client apps](#)
 - [Configure Multi-Factor Authentication \(MFA\)](#)
 - [Configure a VPN client](#)
 - [About the client VPN profile download file](#)
- [OpenVPN](#)
 - [Configure OpenVPN for Point-to-Site connections](#)
 - [Configure OpenVPN clients](#)
- [Transition to a public CA gateway certificate for P2S](#)
- [Configure an Always On VPN device tunnel](#)
- [Configure an Always On VPN user tunnel](#)
- [Advertise custom routes to P2S clients](#)
- [Create a zone-redundant gateway](#)
- [Configure IPsec/IKE policies on connections](#)
- [Configure active-active connections](#)
- [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 a local network gateway](#)
 - [Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Configure VPN devices](#)

[Overview & Azure configuration](#)

[Download VPN device configuration scripts](#)

[Sample: Cisco ASA device \(IKEv2/no BGP\)](#)

[Monitoring and alerts](#)

[Set up alerts based on metrics](#)

[Set up alerts based on diagnostic log](#)

[Configure packet captures](#)

[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](#)

[Point-to-Site - Azure AD authentication](#)

[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?

1/14/2020 • 13 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 VMs that are deployed to a specific subnet you create called the *gateway subnet*. Virtual network gateway VMs contain routing tables and run specific gateway services. These VMs are created when you create the virtual network gateway. You can't directly configure the VMs that are part of the virtual network gateway.

One setting that you configure for a virtual network gateway is the gateway type. Gateway type specifies how the virtual network gateway will be used and the actions that the gateway takes. The gateway type 'Vpn' specifies that the type of virtual network gateway created is a 'VPN gateway', rather than an ExpressRoute gateway. A virtual network can have two virtual network gateways; one VPN gateway and one ExpressRoute gateway - as is the case with [coexisting](#) connection configurations. For more information, see [Gateway types](#).

VPN gateways can be deployed 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. see [About zone-redundant virtual network gateways in Azure Availability Zones](#)

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. 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 OpenVPN, 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), OpenVPN 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 the [VPN Gateway Settings - Gateway SKUs](#) article.
- For Legacy SKU information, see [Working with Legacy SKUs](#).

Gateway SKUs by tunnel, connection, and throughput

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
Generation 1	Basic	Max. 10	Max. 128	Not Supported	100 Mbps	Not Supported	No
Generation 1	VpnGw1	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	No
Generation 1	VpnGw2	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	No
Generation 1	VpnGw3	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	No
Generation 1	VpnGw1AZ	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	Yes
Generation 1	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	Yes
Generation 1	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	Yes
Generation 2	VpnGw2	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	No
Generation 2	VpnGw3	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	No
Generation 2	VpnGw4	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	No
Generation 2	VpnGw5	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	No
Generation 2	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	Yes
Generation 2	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	Yes
Generation 2	VpnGw4AZ	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	Yes
Generation 2	VpnGw5AZ	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	Yes

(*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.

- The resizing of VpnGw SKUs is allowed within the same generation, except resizing of the Basic SKU. The Basic SKU is a legacy SKU and has feature limitations. In order to move from Basic to another VpnGw

SKU, you must delete the Basic SKU VPN gateway and create a new gateway with the desired Generation and SKU size combination.

- These connection limits are separate. For example, you can have 128 SSTP connections and also 250 IKEv2 connections on a VpnGw1 SKU.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- On a single tunnel a maximum of 1 Gbps throughput can be achieved. Aggregate Throughput Benchmark in the above table is based on measurements of multiple tunnels aggregated through a single gateway. The Aggregate Throughput Benchmark for a VPN Gateway is S2S + P2S combined. **If you have a lot of P2S connections, it can negatively impact a S2S connection due to throughput limitations.** The Aggregate Throughput Benchmark is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.

To help our customers understand the relative performance of SKUs using different algorithms, we used publicly available iPerf and CTS Traffic tools to measure performances. The table below lists the results of performance tests for Generation 1, VpnGw SKUs. As you can see, the best performance is obtained when we used GCMAES256 algorithm for both IPsec Encryption and Integrity. We got average performance when using AES256 for IPsec Encryption and SHA256 for Integrity. When we used DES3 for IPsec Encryption and SHA256 for Integrity we got lowest performance.

GENERATION	SKU	ALGORITHMS USED	THROUGHPUT OBSERVED	PACKETS PER SECOND OBSERVED
Generation1	VpnGw1	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000
Generation1	VpnGw1AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000

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:

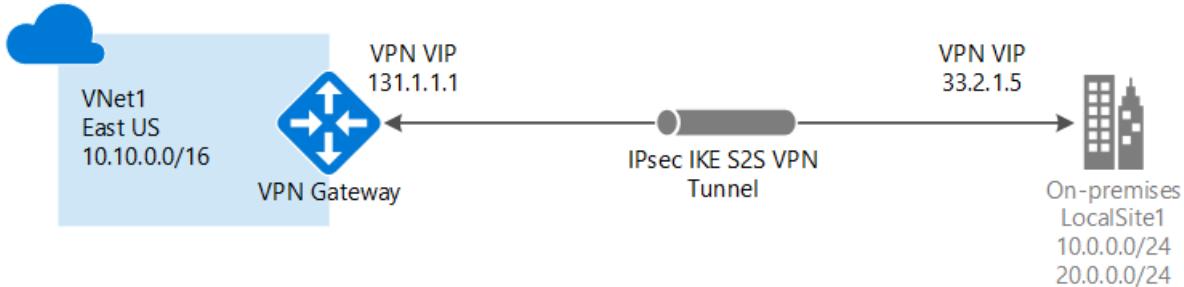
- Available deployment model
- Available configuration tools
- Links that take you directly to an article, if available

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)

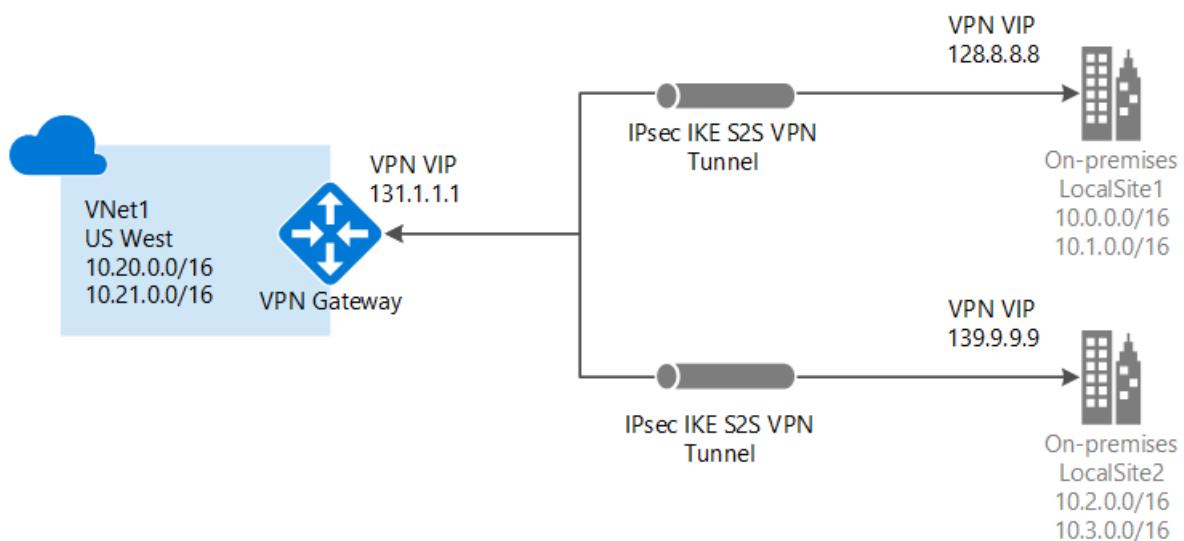
Site-to-Site

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. 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	Tutorial Tutorial+	Tutorial	Tutorial

DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL	AZURE CLI
Classic	Tutorial**	Tutorial+	Not Supported

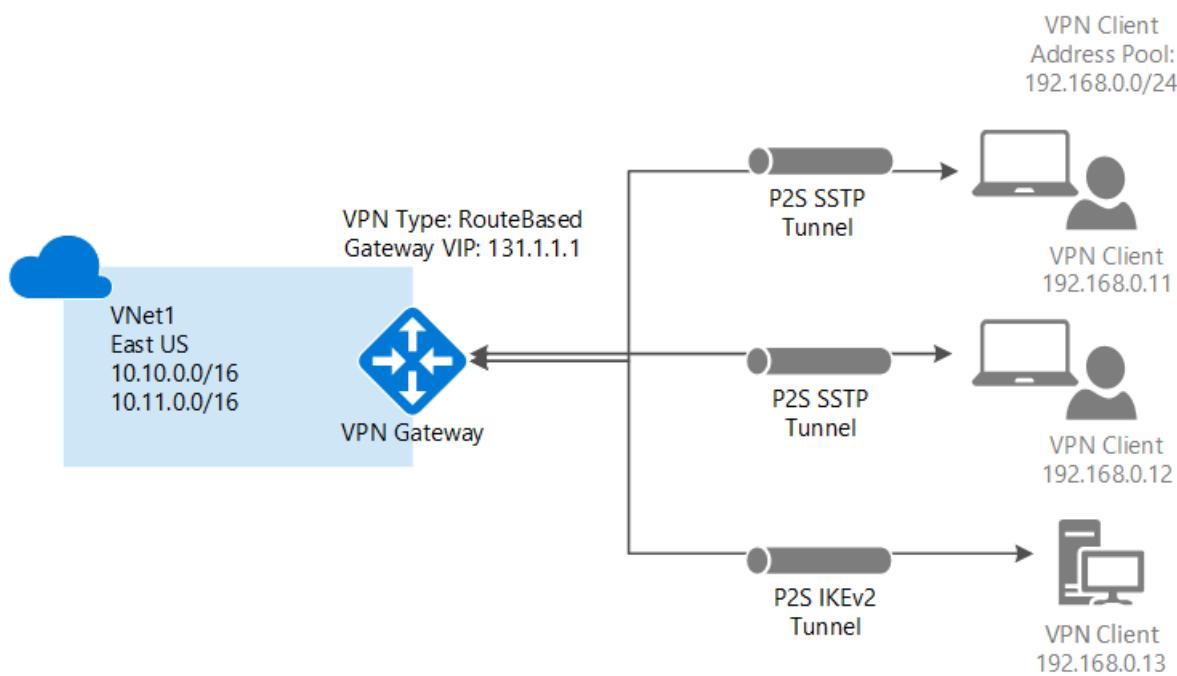
(**) denotes that this method contains steps that require PowerShell.

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

Point-to-Site VPN

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	Tutorial	Tutorial
Classic	Tutorial	Supported

RADIUS authentication

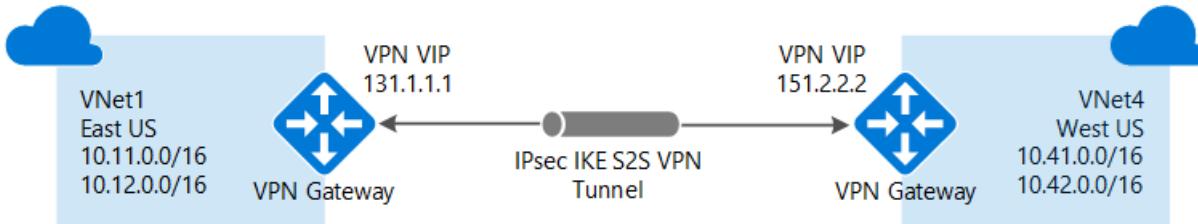
DEPLOYMENT MODEL/METHOD	AZURE PORTAL	POWERSHELL
Resource Manager	Supported	Tutorial
Classic	Not Supported	Not Supported

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	Tutorial*	Supported	Not Supported
Resource Manager	Tutorial+	Tutorial	Tutorial
Connections between different deployment models	Tutorial*	Tutorial	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.

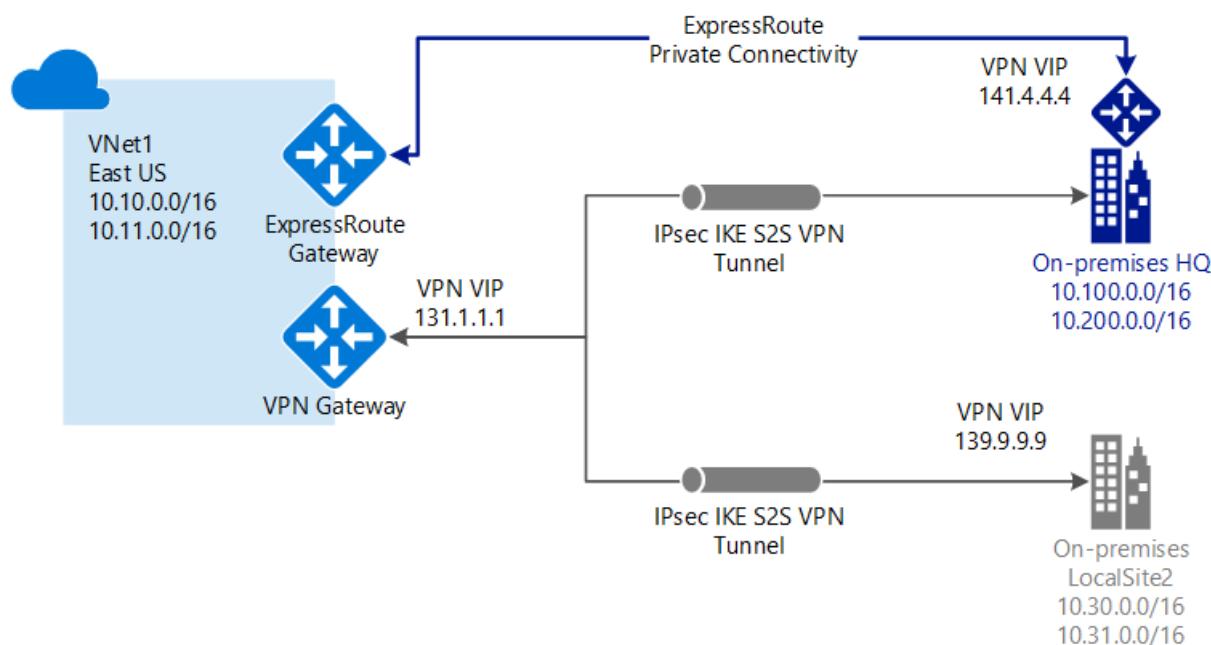
ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

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	Supported	Tutorial
Classic	Not Supported	Tutorial

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. For legacy gateway SKU

pricing, see the [ExpressRoute pricing page](#) and scroll to the **Virtual Network Gateways** section.

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. Cost of an active-active setup is the same as active-passive.

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

- 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.

Tutorial: Create and manage a VPN gateway using PowerShell

11/22/2019 • 6 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
- View the public IP address
- 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

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

Common network parameter values

Below are the parameter values used for this tutorial. In the examples, the variables translate to the following:

```
##$RG1      = The name of the resource group  
##$VNet1    = The name of the virtual network  
##$Location1 = The location region  
##$FESubnet1 = The name of the first subnet  
##$BESubnet1 = The name of the second subnet  
##$VNet1Prefix = The address range for the virtual network  
##$FEPrefix1 = Addresses for the first subnet  
##$BEPrefix1 = Addresses for the second subnet  
##$GwPrefix1 = Addresses for the GatewaySubnet  
##$VNet1ASN   = ASN for the virtual network  
##$DNS1       = The IP address of the DNS server you want to use for name resolution  
##$Gw1        = The name of the virtual network gateway  
##$GwIP1      = The public IP address for the virtual network gateway  
##$GwIPConf1  = The name of the IP configuration
```

Change the values below based on your environment and network setup, then copy and paste to set the variables for this tutorial. If your Cloud Shell session times out, or you need to use a different PowerShell window, copy and paste the variables to your new session and continue the tutorial.

```
$RG1      = "TestRG1"  
$VNet1    = "VNet1"  
$Location1 = "East US"  
$FESubnet1 = "FrontEnd"  
$BESubnet1 = "Backend"  
$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 a resource group

Create a resource group with the [New-AzResourceGroup](#) 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-AzResourceGroup -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. Notice that the example specifies the name of the gateway subnet specifically. You must always specify the name of the gateway subnet as "GatewaySubnet" in order for it to function properly. This example creates a new virtual network with three subnets: Frontend, Backend, and GatewaySubnet using [New-AzVirtualNetworkSubnetConfig](#) and [New-AzVirtualNetwork](#):

```
$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubnet1 -AddressPrefix $FEPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubnet1 -AddressPrefix $BEPrefix1
$gwsb1 = New-AzVirtualNetworkSubnetConfig -Name GatewaySubnet -AddressPrefix $GwPrefix1
$vnet   = New-AzVirtualNetwork ` 
    -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-AzPublicIpAddress](#) and [New-AzVirtualNetworkGatewayIpConfig](#):

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-AzPublicIpAddress -Name $GWIP1 -ResourceGroupName $RG1 ` 
    -Location $Location1 -AllocationMethod Dynamic
$subnet  = Get-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' ` 
    -VirtualNetwork $vnet
$gwipconf = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPConf1 ` 
    -Subnet $subnet -PublicIpAddress $gwpip
```

Create a 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-AzVirtualNetworkGateway](#) cmdlet.

```
New-AzVirtualNetworkGateway -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 another VpnGw SKU if you need higher throughputs or more connections. For more information, see [Gateway SKUs](#).

If you are using the TryIt, your session may time out. That's OK. The gateway will still 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. You can also configure a P2S connection to your VNet from a client computer.

View the gateway public IP address

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

If your session timed out, copy the common network parameters from the beginning of this tutorial into your new session and proceed, then proceed.

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

Resize a 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-AzVirtualNetworkGateway](#) to resize your gateway from VpnGw1 to VpnGw2. For more information, see [Gateway SKUs](#).

```
$gateway = Get-AzVirtualNetworkGateway -Name $Gw1 -ResourceGroup $RG1  
Resize-AzVirtualNetworkGateway -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 a 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-AzVirtualNetworkGateway](#) to reset your gateway.

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

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

Clean up resources

If you're advancing to the [next tutorial](#), you will want to keep these resources because they are the prerequisites.

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

```
Remove-AzResourceGroup -Name $RG1
```

Next steps

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

- Create a VPN gateway
- View the public IP address
- 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](#)

Tutorial: Create and manage S2S VPN connections using PowerShell

1/11/2020 • 7 minutes to read • [Edit Online](#)

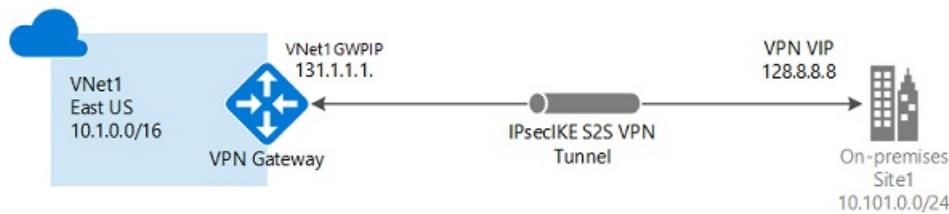
Azure S2S VPN connections provide secure, cross-premises connectivity between customer premises and Azure. This tutorial walks through IPsec S2S VPN connection life cycles such as creating and managing a S2S VPN connection. You learn how to:

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

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

The following diagram shows the topology for this tutorial:



Use Azure Cloud Shell

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

To start Azure Cloud Shell:

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

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

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

Requirements

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

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

The virtual network parameter values are listed below. Note the additional values for the local network gateway which represent your on-premises network. Change the values below based on your environment and network setup, then copy and paste to set the variables for this tutorial. If your Cloud Shell session times out, or you need to use a different PowerShell window, copy and paste the variables to your new session and continue the tutorial.

NOTE

If you are using this to make a connection, be sure to change the values to match your on-premises network. If you are just running these steps as a tutorial, you don't need to make changes, but the connection will not work.

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

# On-premises network - LNGIP1 is the VPN device public IP address
$LNG1      = "VPNsite1"
$LNGprefix1 = "10.101.0.0/24"
$LNGprefix2 = "10.101.1.0/24"
$LNGIP1    = "5.4.3.2"

# 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-AzLocalNetworkGateway](#) command.

```
New-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection](#). Notice that the '-ConnectionType' for Site-to-Site VPN is **IPsec**.

```
$vng1 = Get-AzVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1
$lng1 = Get-AzLocalNetworkGateway -Name $LNG1 -ResourceGroupName $RG1

New-AzVirtualNetworkGatewayConnection -Name $Connection1 -ResourceGroupName $RG1 ` 
    -Location $Location1 -VirtualNetworkGateway1 $vng1 -LocalNetworkGateway2 $lng1 ` 
    -ConnectionType IPsec -SharedKey "Azure@!b2C3" -ConnectionProtocol IKEv2
```

Add the optional "**-EnableBGP \$True**" property to enable BGP for the connection if you are using BGP. It is disabled by default. Parameter '-ConnectionProtocol' is optional with IKEv2 as default. You can create the connection with IKEv1 protocols by specifying **-ConnectionProtocol IKEv1**.

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-AzVirtualNetworkGatewayConnectionSharedKey](#) and [Set-AzVirtualNetworkGatewayConnectionSharedKey](#).

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-AzVirtualNetworkGatewayConnectionSharedKey ` 
    -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-AzVirtualNetworkGatewayConnectionSharedKey ` 
    -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, the following commands add these properties to your VPN gateway and local network gateway: [Set-AzVirtualNetworkGateway](#) and [Set-AzLocalNetworkGateway](#).

Use the following example to configure BGP properties:

```
$vng1 = Get-AzVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $vng1 -Asn $VNet1ASN

$lng1 = Get-AzLocalNetworkGateway -Name $LNG1 -ResourceGroupName $RG1
Set-AzLocalNetworkGateway -LocalNetworkGateway $lng1 `

-Asn $LNGASN1 -BgpPeeringAddress $BGPPeerIP1
```

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

```
$connection = Get-AzVirtualNetworkGatewayConnection `

-Name $Connection1 -ResourceGroupName $RG1

Set-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGatewayConnection -Name $Connection1 `

-ResourceGroupName $RG1

$newpolicy = New-AzIpsecPolicy `

-IkeEncryption AES256 -IkeIntegrity SHA256 -DhGroup DHGroup14 `

-IpsecEncryption AES128 -IpsecIntegrity SHA1 -PfsGroup PFS2048 `

-SALifeTimeSeconds 14400 -SADataSizeKilobytes 102400000

Set-AzVirtualNetworkGatewayConnection -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

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. Use the following examples, making sure to modify the variables to reflect your own network configuration.

```

# On-premises network - LNGIP2 is the VPN device public IP address
$LNG2      = "VPNsite2"
$Location2 = "West US"
$LNGprefix21 = "10.102.0.0/24"
$LNGprefix22 = "10.102.1.0/24"
$LNGIP2     = "4.3.2.1"
$Connection2 = "VNet1ToSite2"

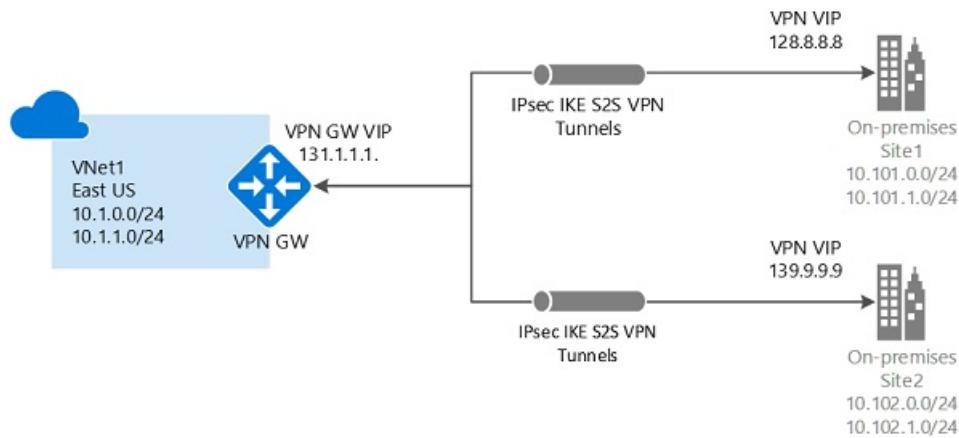
New-AzLocalNetworkGateway -Name $LNG2 -ResourceGroupName $RG1 ` 
    -Location $Location2 -GatewayIpAddress $LNGIP2 -AddressPrefix $LNGprefix21,$LNGprefix22

$vng1 = Get-AzVirtualNetworkGateway -Name $GW1 -ResourceGroupName $RG1
$lng2 = Get-AzLocalNetworkGateway -Name $LNG2 -ResourceGroupName $RG1

New-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGatewayConnection](#).

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

Clean up resources

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

```
Remove-AzResourceGroup -Name $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

1/10/2020 • 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.

About VPN Gateway configuration settings

2/11/2020 • 15 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.

The values in this article apply VPN gateways (virtual network gateways that use the `-GatewayType Vpn`). This article does not cover all gateway types or zone-redundant gateways.

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

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-AzVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg `  
-Location 'West US' -IpConfigurations $gwpipconfig -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. For virtual network gateway SKUs in Azure Availability Zones, see [Azure Availability Zones gateway SKUs](#).

Gateway SKUs by tunnel, connection, and throughput

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
Generation 1	Basic	Max. 10	Max. 128	Not Supported	100 Mbps	Not Supported	No
Generation 1	VpnGw1	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	No
Generation 1	VpnGw2	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	No
Generation 1	VpnGw3	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	No
Generation 1	VpnGw1AZ	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	Yes
Generation 1	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	Yes
Generation 1	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	Yes
Generation 2	VpnGw2	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	No
Generation 2	VpnGw3	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	No
Generation 2	VpnGw4	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	No
Generation 2	VpnGw5	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	No
Generation 2	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	Yes
Generation 2	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	Yes
Generation 2	VpnGw4AZ	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	Yes
Generation 2	VpnGw5AZ	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	Yes

(*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.

- The resizing of VpnGw SKUs is allowed within the same generation, except resizing of the Basic SKU. The Basic SKU is a legacy SKU and has feature limitations. In order to move from Basic to another VpnGw SKU, you must delete the Basic SKU VPN gateway and create a new gateway with the desired Generation and

SKU size combination.

- These connection limits are separate. For example, you can have 128 SSTP connections and also 250 IKEv2 connections on a VpnGw1 SKU.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- On a single tunnel a maximum of 1 Gbps throughput can be achieved. Aggregate Throughput Benchmark in the above table is based on measurements of multiple tunnels aggregated through a single gateway. The Aggregate Throughput Benchmark for a VPN Gateway is S2S + P2S combined. **If you have a lot of P2S connections, it can negatively impact a S2S connection due to throughput limitations.** The Aggregate Throughput Benchmark is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.

To help our customers understand the relative performance of SKUs using different algorithms, we used publicly available iPerf and CTS Traffic tools to measure performances. The table below lists the results of performance tests for Generation 1, VpnGw SKUs. As you can see, the best performance is obtained when we used GCMAES256 algorithm for both IPsec Encryption and Integrity. We got average performance when using AES256 for IPsec Encryption and SHA256 for Integrity. When we used DES3 for IPsec Encryption and SHA256 for Integrity we got lowest performance.

GENERATION	SKU	ALGORITHMS USED	THROUGHPUT OBSERVED	PACKETS PER SECOND OBSERVED
Generation1	VpnGw1	GCMAES256	650 Mbps	58,000
		AES256 & SHA256	500 Mbps	50,000
		DES3 & SHA256	120 Mbps	50,000
Generation1	VpnGw2	GCMAES256	1 Gbps	90,000
		AES256 & SHA256	500 Mbps	80,000
		DES3 & SHA256	120 Mbps	55,000
Generation1	VpnGw3	GCMAES256	1.25 Gbps	105,000
		AES256 & SHA256	550 Mbps	90,000
		DES3 & SHA256	120 Mbps	60,000
Generation1	VpnGw1AZ	GCMAES256	650 Mbps	58,000
		AES256 & SHA256	500 Mbps	50,000
		DES3 & SHA256	120 Mbps	50,000
Generation1	VpnGw2AZ	GCMAES256	1 Gbps	90,000
		AES256 & SHA256	500 Mbps	80,000
		DES3 & SHA256	120 Mbps	55,000
Generation1	VpnGw3AZ	GCMAES256	1.25 Gbps	105,000
		AES256 & SHA256	550 Mbps	90,000
		DES3 & SHA256	120 Mbps	60,000

NOTE

VpnGw SKUs (VpnGw1, VpnGw1AZ, VpnGw2, VpnGw2AZ, VpnGw3, VpnGw3AZ, VpnGw4, VpnGw4AZ, VpnGw5, and VpnGw5AZ) 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 for S2S/connections; no RADIUS authentication for P2S; no IKEv2 for P2S Policy-based VPN: (IKEv1): 1 S2S/connection tunnel; no P2S
All Generation1 and Generation2 SKUs except Basic	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 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	All Generation1 and Generation2 SKUs except Basic
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-AzVirtualNetworkGateway -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. A gateway can take up to 45 minutes to build. In comparison, when you resize a gateway SKU, there is not much 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. With the exception of the Basic SKU, you can resize a VPN gateway SKU to another VPN gateway SKU within the same generation (Generation1 or Generation2). For example, VpnGw1 of Generation1 can be resized to VpnGw2 of Generation1 but not to VpnGw2 of Generation2.
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 VpnGw SKUs. You must instead, [change](#) to the new SKUs.

To resize a gateway

You can use the `Resize-AzVirtualNetworkGateway` PowerShell cmdlet to upgrade or downgrade a Generation1 or Generation2 SKU (all VpnGw SKUs can be resized except Basic SKUs). 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-AzVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg  
Resize-AzVirtualNetworkGateway -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
- VPNClient

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

```
New-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGateway -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
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. 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

User defined routes with a 0.0.0.0/0 destination and NSGs on the GatewaySubnet **are not supported**. Gateways created with this configuration will be blocked from creation. Gateways require access to the management controllers in order to function 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.

When you are planning your gateway subnet size, refer to the documentation for the configuration that you are planning to create. For example, the ExpressRoute/VPN Gateway coexist configuration requires a larger gateway subnet than most other configurations. 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 /27 or larger (/27, /26 etc.) if you have the available address space to do so. This will accommodate most configurations.

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

Local network gateways

A local network gateway is different than a virtual network gateway. When creating a VPN gateway configuration, the local network gateway usually 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-AzLocalNetworkGateway -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

1/10/2020 • 8 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

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 the 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	AR-Series 5.4.7+	Configuration guide	Configuration guide
Barracuda Networks, Inc.	Barracuda CloudGen Firewall	PolicyBased: 5.4.3 RouteBased: 6.2.0	Configuration guide	Configuration guide
Check Point	Security Gateway	R80.10	Configuration guide	Configuration guide
Cisco	ASA	8.3 8.4+ (IKEv2*)	Supported	Configuration guide*

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED CONFIGURATION INSTRUCTIONS	ROUTEBASED CONFIGURATION INSTRUCTIONS
Cisco	ASR	PolicyBased: IOS 15.1 RouteBased: IOS 15.2	Supported	Supported
Cisco	CSR	RouteBased: IOS-XE 16.10	(not tested)	Configuration script
Cisco	ISR	PolicyBased: IOS 15.0 RouteBased*: IOS 15.1	Supported	Supported
Cisco	Meraki	N/A	Not compatible	Not compatible
Cisco	vEdge (Viptela OS)	18.4.0 (Active/Passive Mode) 19.2 (Active/Active Mode)	Not compatible	Manual configuration (Active/Passive) Cloud Onramp configuration (Active/Active)
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	(not tested)	Configuration guide
Hillstone Networks	Next-Gen Firewalls (NGFW)	5.5R7	(not tested)	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	Supported	Configuration script
Juniper	J-Series	PolicyBased: JunOS 10.4r9 RouteBased: JunOS 11.4	Supported	Configuration script
Juniper	ISG	ScreenOS 6.3	Supported	Configuration script
Juniper	SSG	ScreenOS 6.2	Supported	Configuration script
Juniper	MX	JunOS 12.x	Supported	Configuration script
Microsoft	Routing and Remote Access Service	Windows Server 2012	Not compatible	Supported

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED CONFIGURATION INSTRUCTIONS	ROUTEBASED CONFIGURATION INSTRUCTIONS
Open Systems AG	Mission Control Security Gateway	N/A	Configuration guide	Not compatible
Palo Alto Networks	All devices running PAN-OS	PAN-OS PolicyBased: 6.1.5 or later RouteBased: 7.1.4	Supported	Configuration guide
Sentrium (Developer)	VyOS	VyOS 1.2.2	(not tested)	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	(not tested)	Configuration guide Configuration guide - Multiple SAs
Synology	MR2200ac RT2600ac RT1900ac	SRM1.1.5/VpnPlusServer-1.2.0	(not tested)	Configuration guide
Ubiquiti	EdgeRouter	EdgeOS v1.10	(not tested)	BGP over IKEv2/IPsec VTI over IKEv2/IPsec
WatchGuard	All	Fireware XTM PolicyBased: v11.11.x RouteBased: v11.12.x	Configuration guide	Configuration guide
Zyxel	ZyWALL USG series ZyWALL ATP series ZyWALL VPN series	ZLD v4.32+	(not tested)	VTI over IKEv2/IPsec BGP over IKEv2/IPsec

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
Juniper	SRX	JunOS 12.x RouteBased BGP
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased VTI
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased BGP

NOTE

(*) Required: NarrowAzureTrafficSelectors (enable UsePolicyBasedTrafficSelectors option) 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

SAMPLE TEXT	CHANGE TO
<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 .
<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	IKEv1 and IKEv2
Diffie-Hellman Group	Group 2 (1024 bit)	Group 2 (1024 bit)
Authentication Method	Pre-Shared Key	Pre-Shared Key

PROPERTY	POLICYBASED	ROUTEBASED
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	IKEv1 and 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
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

-	ENCRYPTION	AUTHENTICATION	PFS GROUP
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
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.

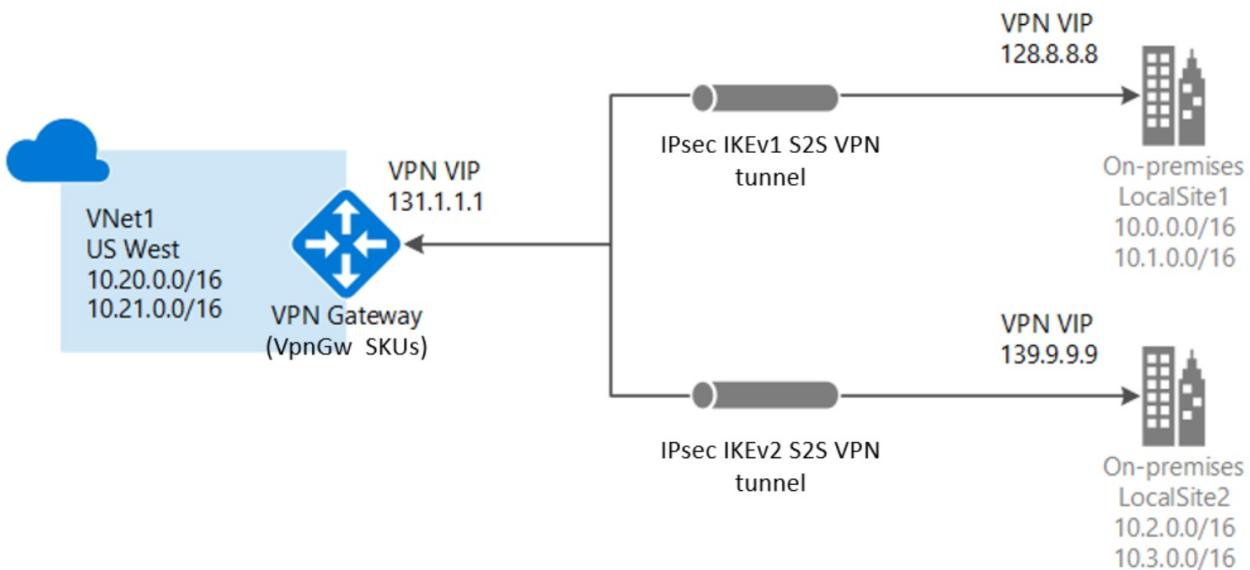
About cryptographic requirements and Azure VPN gateways

1/11/2020 • 7 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 IKEv1 and IKEv2 for Azure VPN connections

Traditionally we allowed IKEv1 connections for Basic SKUs only and allowed IKEv2 connections for all VPN gateway SKUs other than Basic SKUs. The Basic SKUs allow only 1 connection and along with other limitations such as performance, customers using legacy devices that support only IKEv1 protocols were having limited experience. In order to enhance the experience of customers using IKEv1 protocols, we are now allowing IKEv1 connections for all of the VPN gateway SKUs, except Basic SKU. For more information, see [VPN Gateway SKUs](#).



When IKEv1 and IKEv2 connections are applied to the same VPN gateway, the transit between these two connections is auto-enabled.

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 you 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 article: [About VPN devices and IPsec/IKE parameters for Site-to-Site VPN Gateway connections](#).

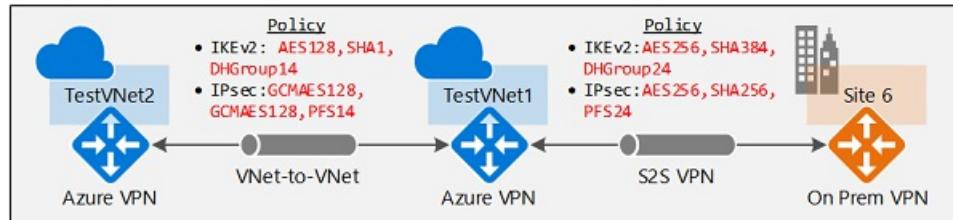
Cryptographic requirements

For communications that require specific cryptographic algorithms or parameters, typically due to compliance or security requirements, you 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 you 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 all Azure SKUs except the Basic SKU.

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/IKEV2	OPTIONS
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	ECP384	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 restarts 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.

How do I create connections with IKEv1 or IKEv2 protocol type?

IKEv1 connections can be created on all RouteBased VPN type SKUs, except the Basic SKU. You can specify a connection protocol type of IKEv1 or IKEv2 while creating connections. If you do not specify a connection protocol

type, IKEv2 is used as default option where applicable. For more information, see the [PowerShell cmdlet](#) documentation. For SKU types and IKEv1/IKEv2 support, see [Connect gateways to policy-based VPN devices](#).

Is transit between between IKEv1 and IKEv2 connections allowed?

Yes. Transit between IKEv1 and IKEv2 connections is supported.

Can I have IKEv1 site-to-site connections on Basic SKUs of RouteBased VPN type?

No. The Basic SKU does not support this.

Can I change the connection protocol type after the connection is created (IKEv1 to IKEv2 and vice versa)?

No. Once the connection is created, IKEv1/IKEv2 protocols cannot be changed. You must delete and recreate a new connection with the desired protocol type.

Where can I find more configuration information for IPsec?

See [Configure IPsec/IKE policy for S2S or VNet-to-VNet connections](#)

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.

About BGP with Azure VPN Gateway

1/9/2020 • 7 minutes to read • [Edit Online](#)

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

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

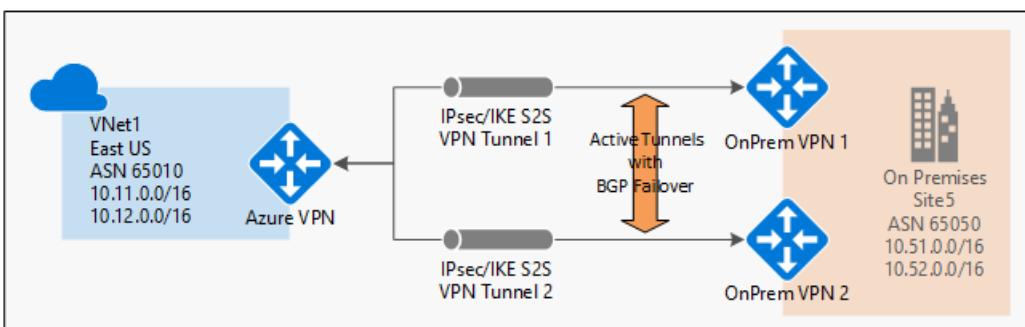
With BGP, you only need to declare a minimum prefix to a specific BGP peer over the IPsec S2S VPN tunnel. It can be as small as a host prefix (/32) of the BGP peer IP address of your on-premises VPN device. You can control which on-premises network prefixes you want to advertise to Azure to allow your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

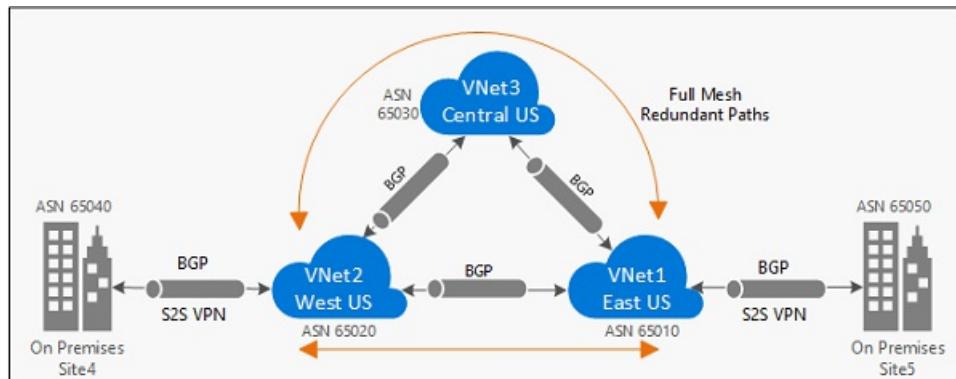


Support transit routing between your on-premises networks and multiple Azure VNets

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

What Private ASNs can I use?

The useable range of Private ASNs that can be used are:

- 64512-65514, 65521-65534

These ASNs are not reserved by IANA or Azure for use and therefore can be used to assign to your Azure VPN Gateway.

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.**

How many prefixes can I advertise to Azure VPN gateway?

We support up to 4000 prefixes. The BGP session is dropped if the number of prefixes exceeds the limit.

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. For more information, see [About BGP](#).

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 for active-standby VPN

gateways, or two IP addresses for active-active VPN gateways. You can get the actual BGP IP address(es) allocated by using PowerShell (Get-AzVirtualNetworkGateway, look for the "bgpPeeringAddress" property), or in the Azure portal (under the "Configure BGP ASN" property on the Gateway Configuration page).

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 or the Vnet address space of the VPN Gateway. 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.

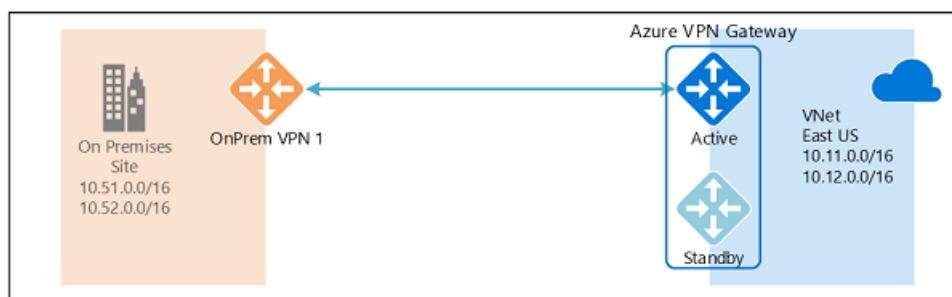
Highly Available Cross-Premises and VNet-to-VNet Connectivity

1/9/2020 • 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.



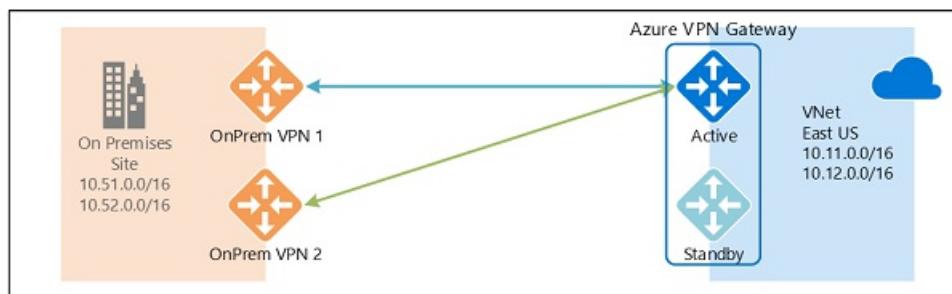
Highly Available Cross-Premises Connectivity

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

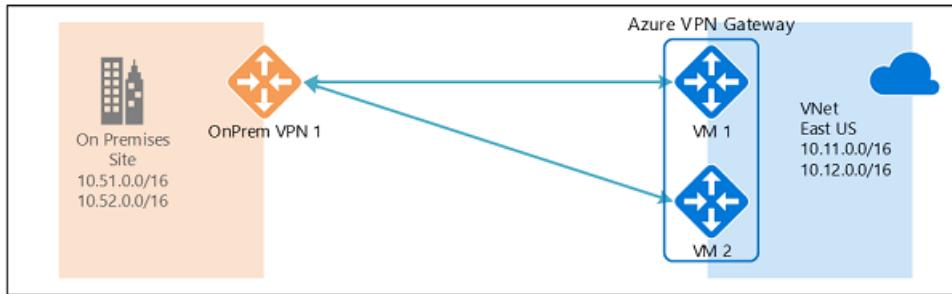
multiple VPN devices from the same on-premises network to Azure, you need to create one local network gateway for each VPN device, and one connection from your Azure VPN gateway to each local network gateway.

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. You must use Equal-cost multi-path routing (ECMP).
7. 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:



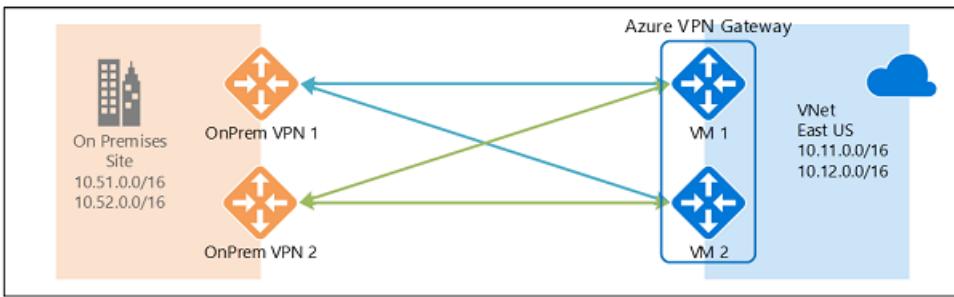
In this configuration, each Azure gateway instance will have a unique public IP address, and each will establish an IPsec/IKE S2S VPN tunnel to your on-premises VPN device specified in your local network gateway and connection. Note that both VPN tunnels are actually part of the same connection. You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



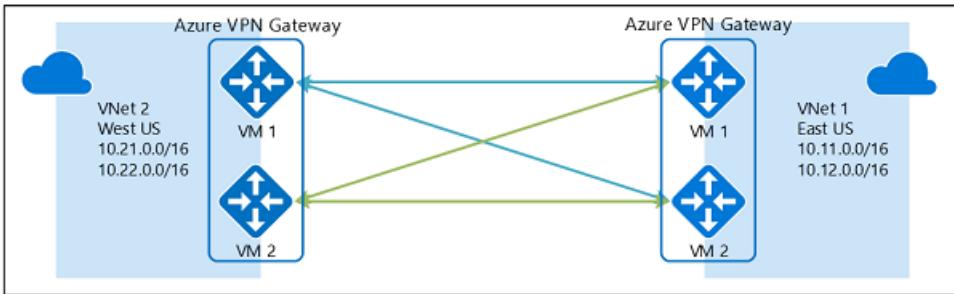
Here you create and setup the Azure VPN gateway in an active-active configuration, and create two local network gateways and two connections for your two on-premises VPN devices as described above. The result is a full mesh connectivity of 4 IPsec tunnels between your Azure virtual network and your on-premises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

2/20/2020 • 21 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:

- **OpenVPN® Protocol**, an SSL/TLS based VPN protocol. An SSL VPN solution can penetrate firewalls, since most firewalls open TCP port 443 outbound, which SSL uses. OpenVPN can be used to connect from Android, iOS (versions 11.0 and above), Windows, Linux and Mac devices (OSX versions 10.13 and above).
- 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 outbound, 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).

NOTE

IKEv2 and OpenVPN for P2S are available for the Resource Manager deployment model only. They are 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 native Azure Active Directory authentication

Azure AD authentication allows users to connect to Azure using their Azure Active Directory credentials. Native Azure AD authentication is only supported for OpenVPN protocol and Windows 10 and requires the use of the [Azure VPN Client](#).

With native Azure AD authentication, you can leverage Azure AD's conditional access as well as Multi-Factor Authentication(MFA) features for VPN.

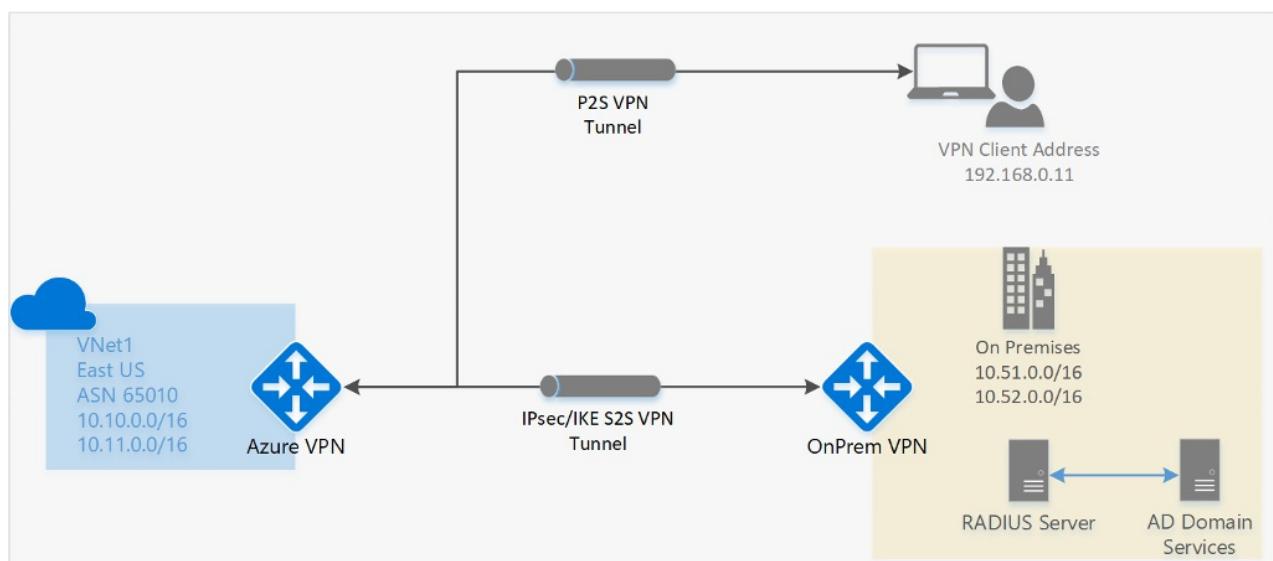
At a high level, you need to perform the following steps to configure Azure AD authentication:

1. [Configure an Azure AD tenant](#)
2. [Enable Azure AD authentication on the gateway](#)
3. [Download and configure Azure VPN Client](#)

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?

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
Generation 1	Basic	Max. 10	Max. 128	Not Supported	100 Mbps	Not Supported	No
Generation 1	VpnGw1	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	No
Generation 1	VpnGw2	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	No
Generation 1	VpnGw3	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	No
Generation 1	VpnGw1AZ	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	Yes
Generation 1	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	Yes
Generation 1	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	Yes
Generation 2	VpnGw2	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	No
Generation 2	VpnGw3	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	No
Generation 2	VpnGw4	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	No
Generation 2	VpnGw5	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	No
Generation 2	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	Yes
Generation 2	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	Yes

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
------------------------	-----	--------------------------	----------------------	--------------------------------	--------------------------------	-----	----------------

Generation 2	VpnGw4AZ	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	Yes
Generation 2	VpnGw5AZ	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	Yes

(*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.

- The resizing of VpnGw SKUs is allowed within the same generation, except resizing of the Basic SKU. The Basic SKU is a legacy SKU and has feature limitations. In order to move from Basic to another VpnGw SKU, you must delete the Basic SKU VPN gateway and create a new gateway with the desired Generation and SKU size combination.
- These connection limits are separate. For example, you can have 128 SSTP connections and also 250 IKEv2 connections on a VpnGw1 SKU.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- On a single tunnel a maximum of 1 Gbps throughput can be achieved. Aggregate Throughput Benchmark in the above table is based on measurements of multiple tunnels aggregated through a single gateway. The Aggregate Throughput Benchmark for a VPN Gateway is S2S + P2S combined. **If you have a lot of P2S connections, it can negatively impact a S2S connection due to throughput limitations.** The Aggregate Throughput Benchmark is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.

To help our customers understand the relative performance of SKUs using different algorithms, we used publicly available iPerf and CTS Traffic tools to measure performances. The table below lists the results of performance tests for Generation 1, VpnGw SKUs. As you can see, the best performance is obtained when we used GCMAES256 algorithm for both IPsec Encryption and Integrity. We got average performance when using AES256 for IPsec Encryption and SHA256 for Integrity. When we used DES3 for IPsec Encryption and SHA256 for Integrity we got lowest performance.

GENERATION	SKU	ALGORITHMS USED	THROUGHPUT OBSERVED	PACKETS PER SECOND OBSERVED
Generation1	VpnGw1	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000

GENERATION	SKU	ALGORITHMS USED	THROUGHPUT OBSERVED	PACKETS PER SECOND OBSERVED
Generation1	VpnGw1AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000

- For Gateway SKU recommendations, see [About VPN Gateway settings](#).

NOTE

The Basic SKU does not support IKEv2 or RADIUS authentication.

What IKE/IPsec policies are configured on VPN gateways for P2S?

IKEv2

CIPHER	INTEGRITY	PRF	DH GROUP
GCM_AES256	GCM_AES256	SHA384	GROUP_24
GCM_AES256	GCM_AES256	SHA384	GROUP_14
GCM_AES256	GCM_AES256	SHA384	GROUP_ECP384
GCM_AES256	GCM_AES256	SHA384	GROUP_ECP256
GCM_AES256	GCM_AES256	SHA256	GROUP_24
GCM_AES256	GCM_AES256	SHA256	GROUP_14
GCM_AES256	GCM_AES256	SHA256	GROUP_ECP384
GCM_AES256	GCM_AES256	SHA256	GROUP_ECP256
AES256	SHA384	SHA384	GROUP_24
AES256	SHA384	SHA384	GROUP_14
AES256	SHA384	SHA384	GROUP_ECP384
AES256	SHA384	SHA384	GROUP_ECP256
AES256	SHA256	SHA256	GROUP_24

CIPHER	INTEGRITY	PRF	DH GROUP
AES256	SHA256	SHA256	GROUP_14
AES256	SHA256	SHA256	GROUP_ECP384
AES256	SHA256	SHA256	GROUP_ECP256
AES256	SHA256	SHA256	GROUP_2

IPsec

CIPHER	INTEGRITY	PFS GROUP
GCM_AES256	GCM_AES256	GROUP_NONE
GCM_AES256	GCM_AES256	GROUP_24
GCM_AES256	GCM_AES256	GROUP_14
GCM_AES256	GCM_AES256	GROUP_ECP384
GCM_AES256	GCM_AES256	GROUP_ECP256
AES256	SHA256	GROUP_NONE
AES256	SHA256	GROUP_24
AES256	SHA256	GROUP_14
AES256	SHA256	GROUP_ECP384
AES256	SHA256	GROUP_ECP256
AES256	SHA1	GROUP_NONE

What TLS policies are configured on VPN gateways for P2S?

TLS

POLICIES
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

POLICIES

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

TLS_RSA_WITH_AES_128_GCM_SHA256

TLS_RSA_WITH_AES_256_GCM_SHA384

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

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](#)
- [Configure OpenVPN](#)

To remove the configuration of a P2S connection

For steps, see the [FAQ](#), below.

FAQ for native Azure certificate authentication

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

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

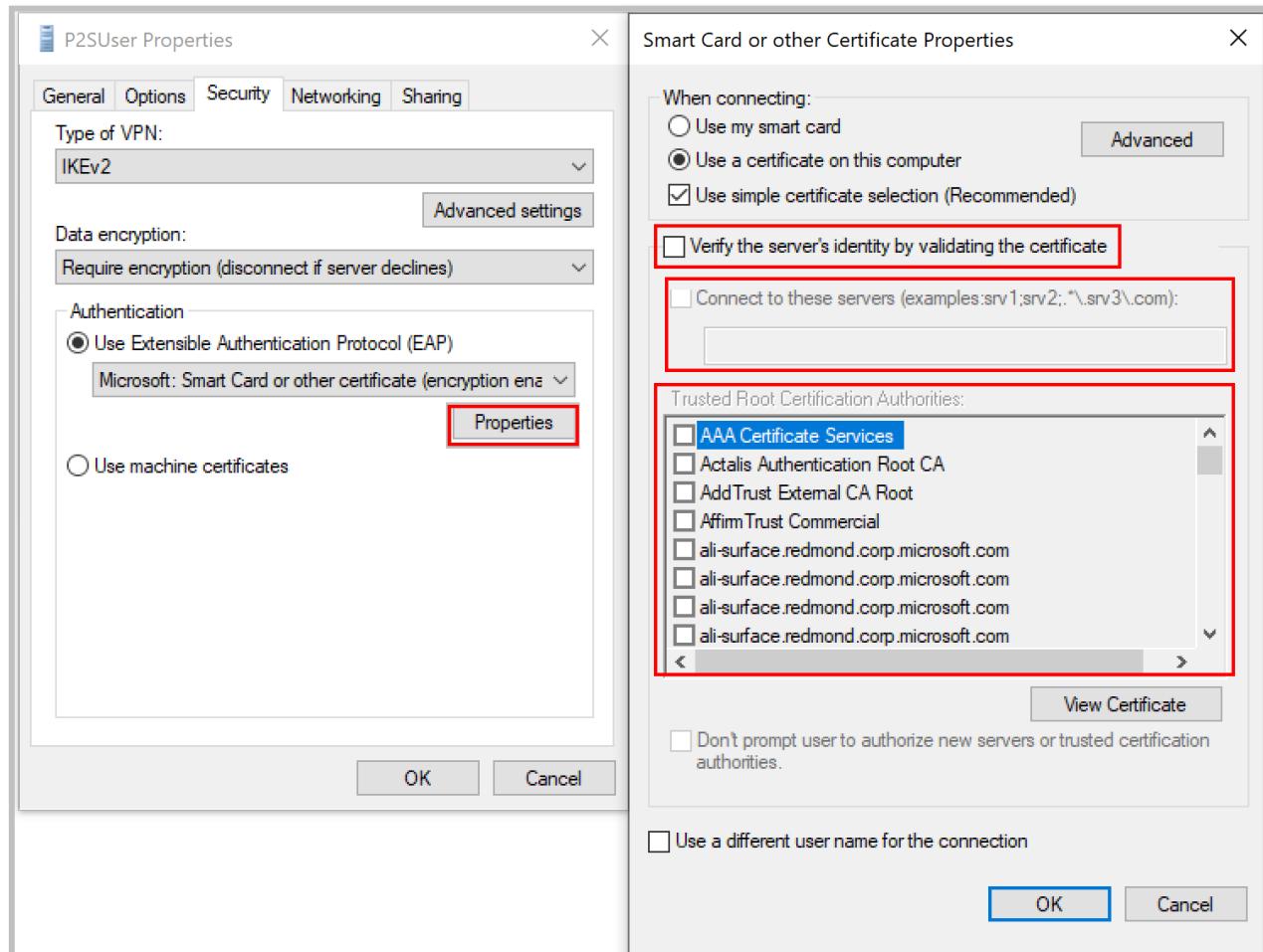
Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

What should I do if I'm getting a certificate mismatch when connecting using certificate authentication?

Uncheck "**Verify the server's identity by validating the certificate**" or **add the server FQDN along with the certificate** when creating a profile manually. You can do this by running **rasphone** from a command prompt and picking the profile from the drop-down list.

Bypassing server identity validation is not recommended in general, but with Azure certificate authentication, the same certificate is being used for server validation in the VPN tunneling protocol (IKEv2/SSTP) and the EAP protocol. Since the server certificate and FQDN is already validated by the VPN tunneling protocol, it is redundant to validate the same again in EAP.



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

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

Can I use certificates from Azure Key Vault?

No.

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?

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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\Wow64Node\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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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.

However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

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 for SSTP, but depends on the gateway SKU for IKEv2. For more information on the number of connections supported, see [Gateway SKUs](#).

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.

Does RADIUS authentication work with the OpenVPN client?

RADIUS authentication is supported for the OpenVPN protocol only through PowerShell.

Next Steps

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

"OpenVPN" is a trademark of OpenVPN Inc.

About Point-to-Site VPN routing

11/17/2019 • 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

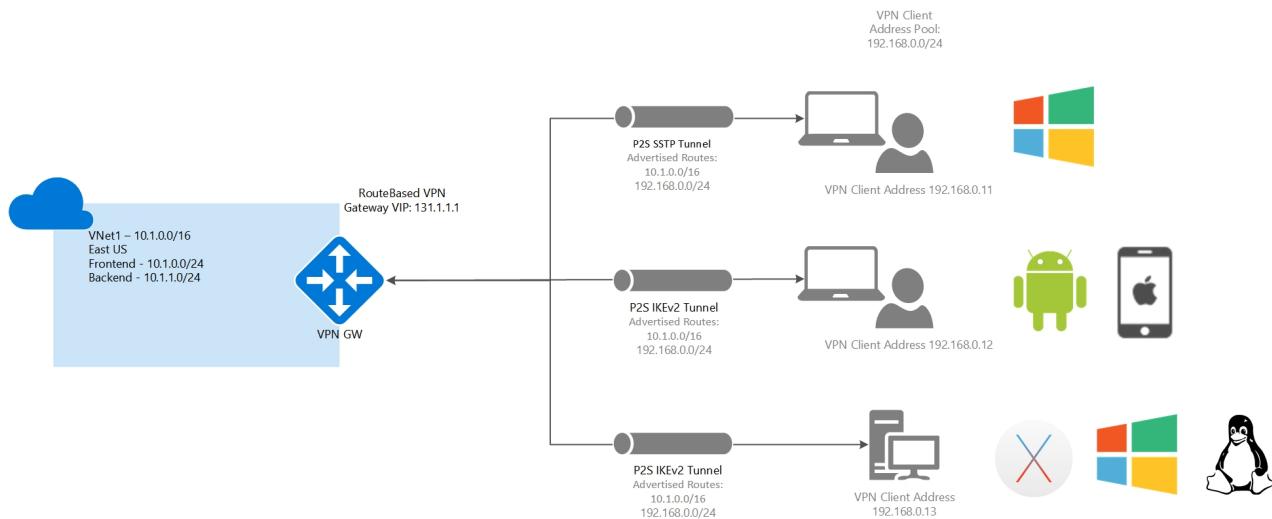
This article applies to IKEv2 only.

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 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

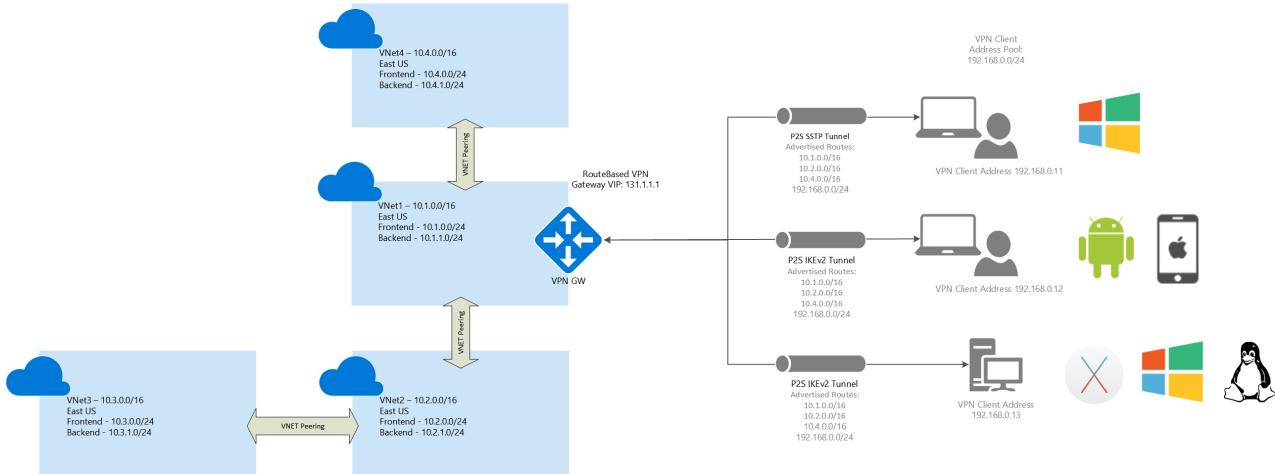
Access

- 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

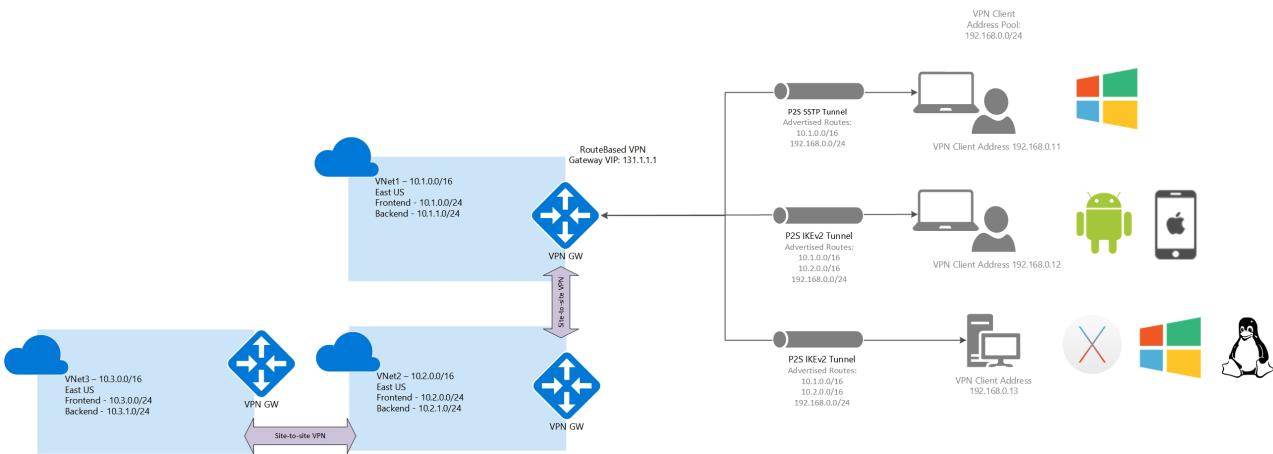
Access

- 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

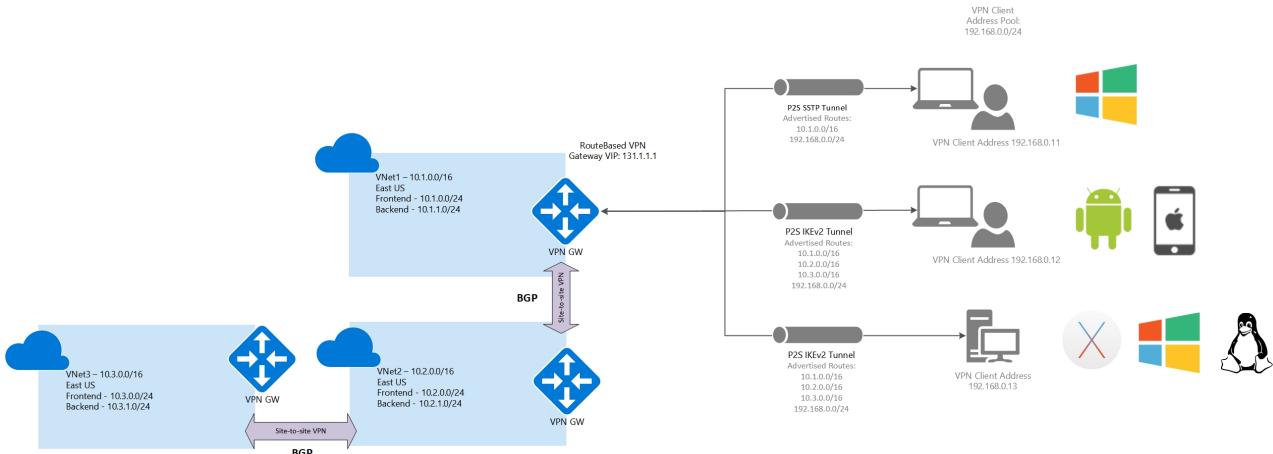
Access

- 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

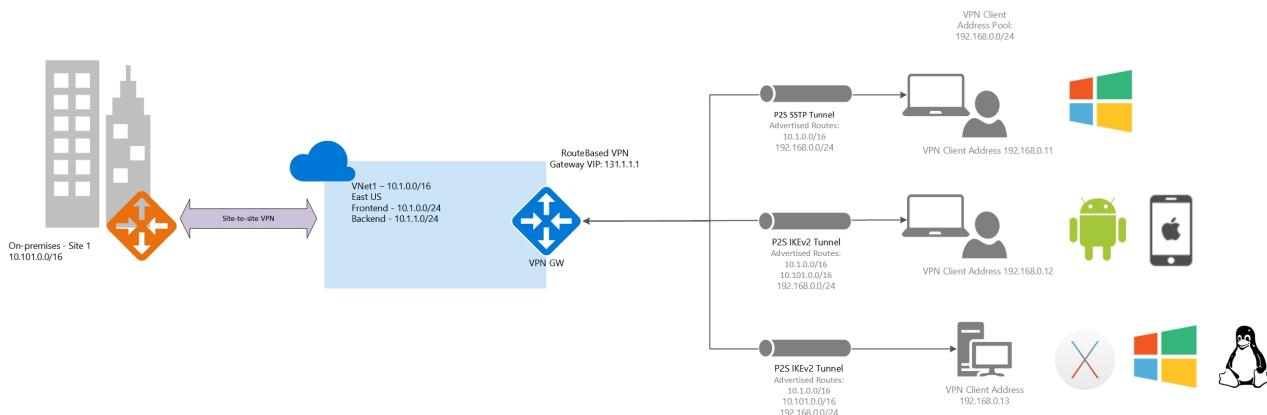
Access

- 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 and non-Windows clients can only access VNet1.



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, 192.168.0.0/24

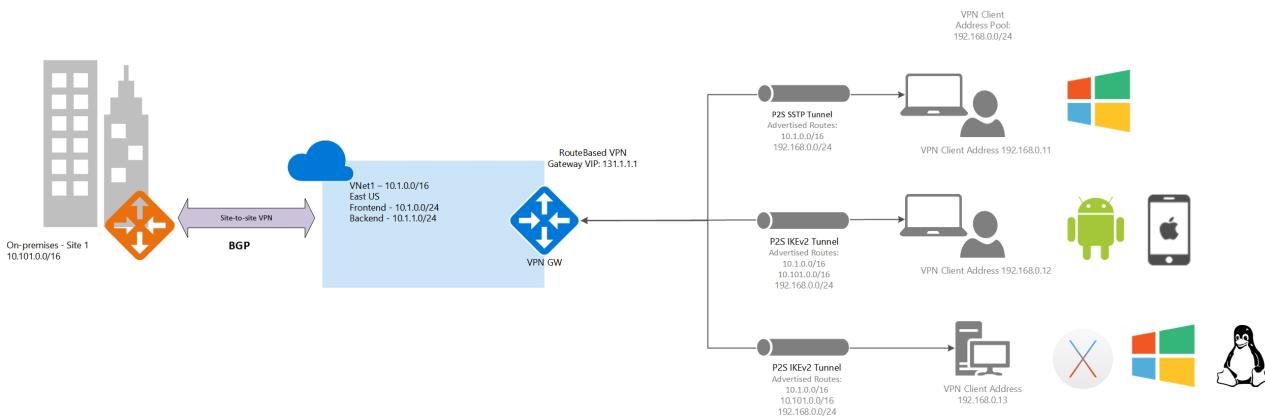
Access

- 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

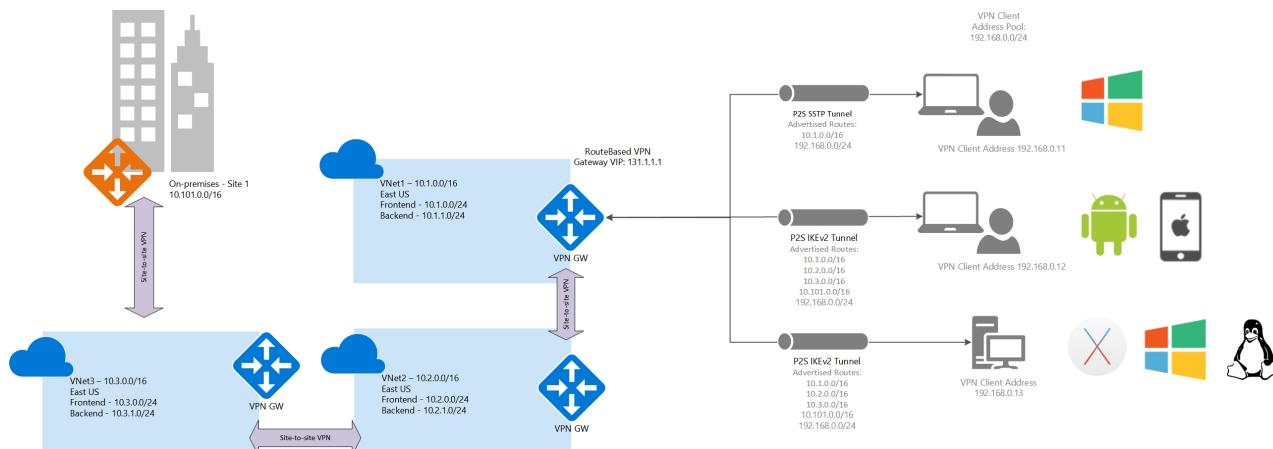
Access

- 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 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, 10.3.0.0/16, 10.101.0.0/16, 192.168.0.0/24

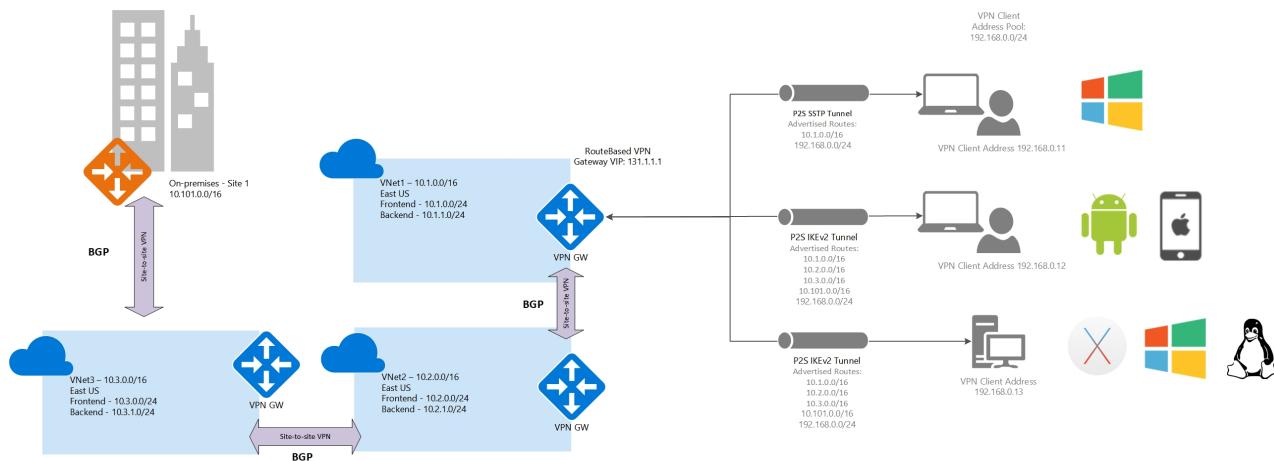
Access

- 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 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, 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.

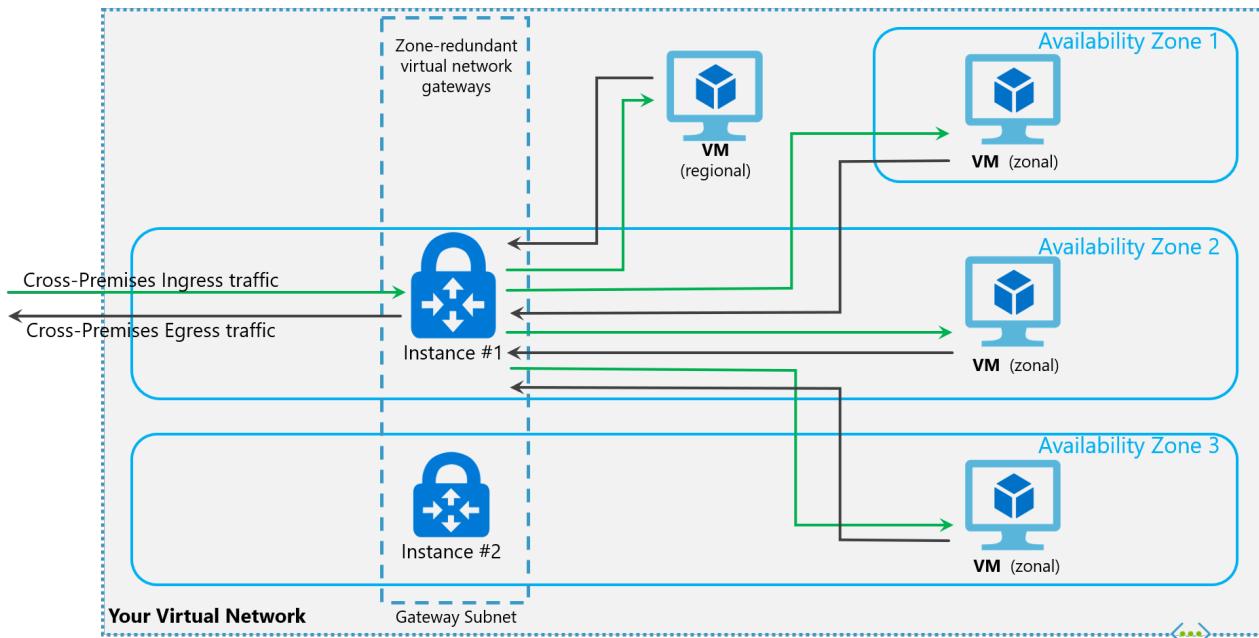
About zone-redundant virtual network gateways in Azure Availability Zones

1/10/2020 • 3 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.

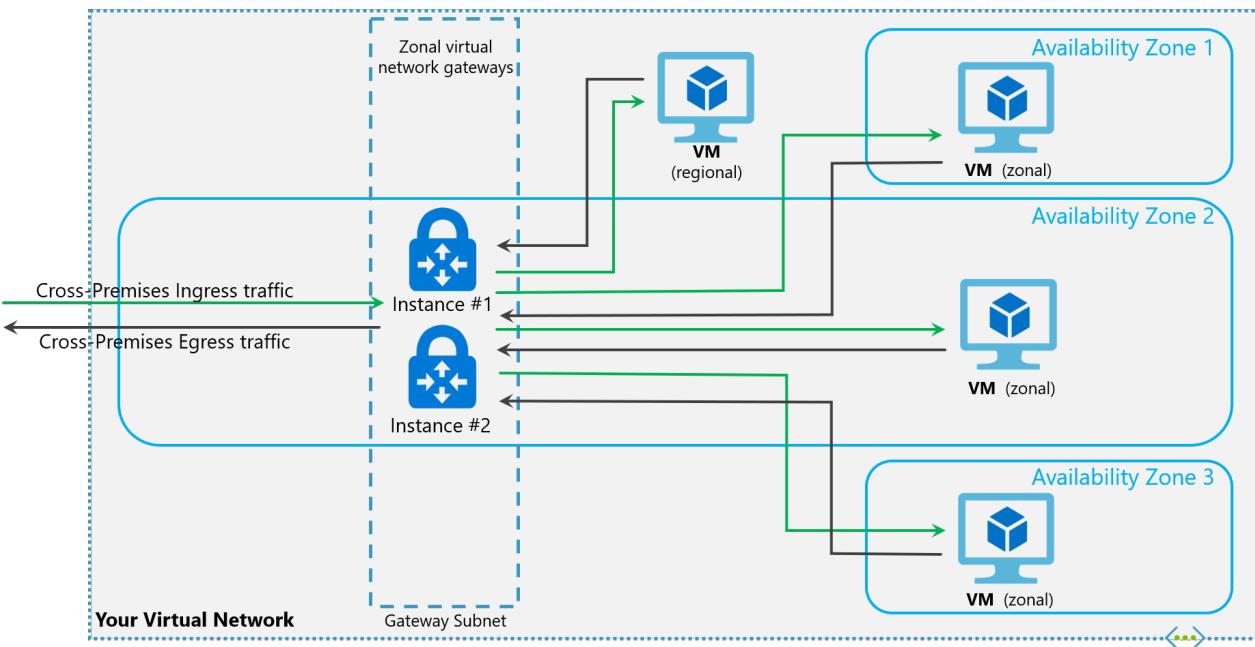
Zone-redundant gateways

To automatically deploy your virtual network gateways across availability zones, you can use zone-redundant virtual network gateways. With zone-redundant gateways, you can benefit from zone-resiliency to access your mission-critical, scalable services on Azure.



Zonal gateways

To deploy gateways in a specific zone, you can use zonal gateways. When you deploy a zonal gateway, all instances of the gateway are deployed in the same Availability Zone.



Gateway SKUs

Zone-redundant and zonal gateways are available as new gateway SKUs. We have added new virtual network gateway SKUs in Azure AZ regions. These SKUs are similar to the corresponding existing SKUs for ExpressRoute and VPN Gateway, except that they are specific to zone-redundant and zonal gateways. You can identify these SKUs by the "AZ" in the SKU name.

For information about gateway SKUs, see [VPN gateway SKUs](#) and [ExpressRoute gateway SKUs](#).

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.

FAQ

What will change when I deploy these new SKUs?

From your perspective, 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 to deploy the new SKUs. However, you will see these new SKUs only in those Azure regions that have Azure Availability Zones.

What regions are available for me to use the new SKUs?

The new SKUs are available in Azure regions that have Azure Availability Zones - Central US, France Central, North Europe, West Europe, and West US 2 regions, East US, East US 2, Southeast Asia, Japan East, UK South. Going forward, we will make zone-redundant gateways available to you in other Azure Public Regions.

Can I change/migrate/upgrade 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. However, you should reserve a /27 IP address range for the gateway subnet.

Next steps

[Create a zone-redundant virtual network gateway](#)

Interoperability in Azure back-end connectivity features: Test setup

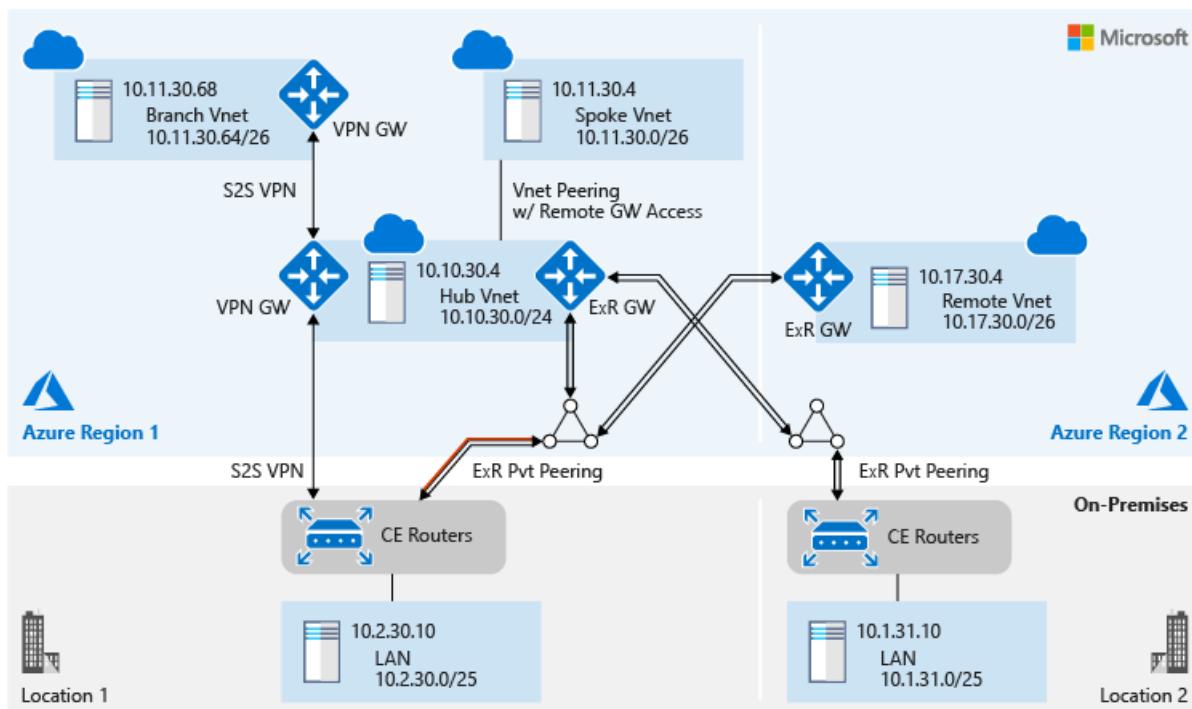
12/5/2019 • 4 minutes to read • [Edit Online](#)

This article describes a test setup you can use to analyze how Azure networking services interoperate at the control plane level and data plane level. Let's look briefly at the Azure networking components:

- **Azure ExpressRoute:** Use private peering in Azure ExpressRoute to directly connect private IP spaces in your on-premises network to your Azure Virtual Network deployments. ExpressRoute can help you achieve higher bandwidth and a private connection. Many ExpressRoute eco partners offer ExpressRoute connectivity with SLAs. To learn more about ExpressRoute and to learn how to configure ExpressRoute, see [Introduction to ExpressRoute](#).
- **Site-to-site VPN:** You can use Azure VPN Gateway as a site-to-site VPN to securely connect an on-premises network to Azure over the internet or by using ExpressRoute. To learn how to configure a site-to-site VPN to connect to Azure, see [Configure VPN Gateway](#).
- **VNet peering:** Use virtual network (VNet) peering to establish connectivity between VNets in Azure Virtual Network. To learn more about VNet peering, see the [tutorial on VNet peering](#).

Test setup

The following figure illustrates the test setup:



The centerpiece of the test setup is the hub VNet in Azure Region 1. The hub VNet is connected to different networks in the following ways:

- The hub VNet is connected to the spoke VNet by using VNet peering. The spoke VNet has remote access to both gateways in the hub VNet.
- The hub VNet is connected to the branch VNet by using site-to-site VPN. The connectivity uses eBGP to exchange routes.
- The hub VNet is connected to the on-premises Location 1 network by using ExpressRoute private peering as

the primary path. It uses site-to-site VPN connectivity as the backup path. In the rest of this article, we refer to this ExpressRoute circuit as ExpressRoute 1. By default, ExpressRoute circuits provide redundant connectivity for high availability. On ExpressRoute 1, the secondary customer edge (CE) router's subinterface that faces the secondary Microsoft Enterprise Edge Router (MSEE) is disabled. A red line over the double-line arrow in the preceding figure represents the disabled CE router subinterface.

- The hub VNet is connected to the on-premises Location 2 network by using another ExpressRoute private peering. In the rest of this article, we refer to this second ExpressRoute circuit as ExpressRoute 2.
- ExpressRoute 1 also connects both the hub VNet and the on-premises Location 1 network to a remote VNet in Azure Region 2.

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [configuration details](#) for the test topology.

Learn about [control plane analysis](#) of the test setup and the views of different VNets or VLANs in the topology.

Learn about the [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure back-end connectivity features: Test configuration details

7/19/2019 • 6 minutes to read • [Edit Online](#)

This article describes the configuration details of the [test setup](#). The test setup helps you analyze how Azure networking services interoperate at the control plane level and data plane level.

Spoke VNet connectivity by using VNet peering

The following figure shows the Azure Virtual Network peering details of a spoke virtual network (VNet). To learn how to set up peering between two VNets, see [Manage VNet peering](#). If you want the spoke VNet to use the gateways that are connected to the hub VNet, select **Use remote gateways**.

The screenshot shows the configuration details for a VNet peering named "Spoke01-VNet01-peering".

General Information:

- Name: Spoke01-VNet01-peering
- Peering status: Connected
- Provisioning state: Succeeded

Peer details:

- Address space: 10.10.30.0/24
- Virtual network: vNet01

Configuration:

- Allow virtual network access: Enabled
- Allow forwarded traffic:
- Allow gateway transit:
- Use remote gateways:

The following figure shows the VNet peering details of the hub VNet. If you want the hub VNet to permit the spoke VNet to use the hub's gateways, select **Allow gateway transit**.

VNet01-Spoke01-Peering

VNet01

Save Discard Delete

Name
VNet01-Spoke01-Peering

Peering status
Connected

Provisioning state
Succeeded

Peer details

Address space
10.11.30.0/26

Virtual network
Spoke01-VNet

Configuration

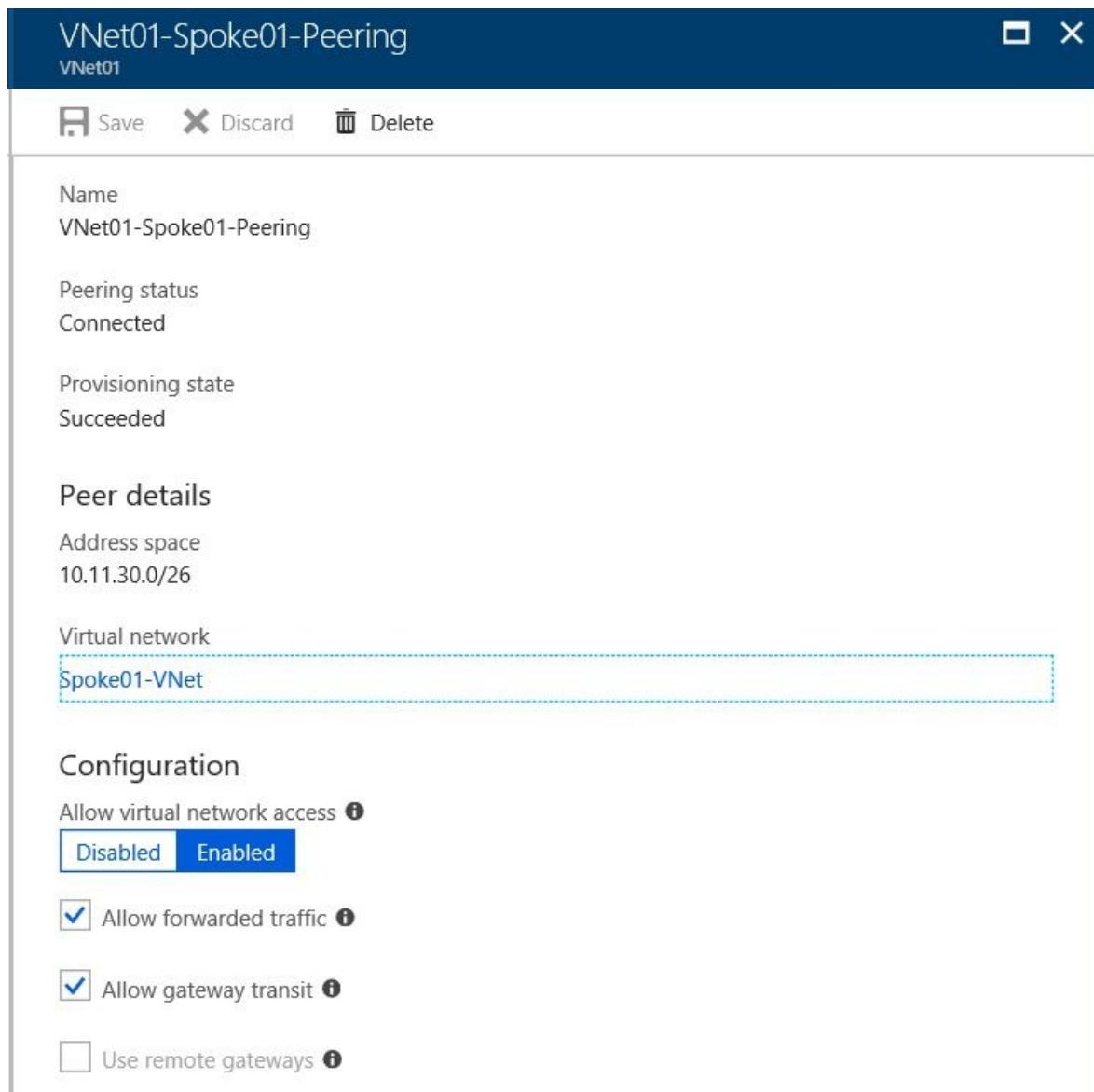
Allow virtual network access ⓘ

Disabled Enabled

Allow forwarded traffic ⓘ

Allow gateway transit ⓘ

Use remote gateways ⓘ



Branch VNet connectivity by using a site-to-site VPN

Set up site-to-site VPN connectivity between the hub and branch VNets by using VPN gateways in Azure VPN Gateway. By default, VPN gateways and Azure ExpressRoute gateways use a private autonomous system number (ASN) value of **65515**. You can change the ASN value in VPN Gateway. In the test setup, the ASN value of the branch VNet VPN gateway is changed to **65516** to support eBGP routing between the hub and branch VNets.

On-premises Location 1 connectivity by using ExpressRoute and a site-to-site VPN

ExpressRoute 1 configuration details

The following figure shows the Azure Region 1 ExpressRoute circuit configuration toward on-premises Location 1 customer edge (CE) routers:

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Provisioned	192.168.30.16/30	192.168.30.20/30	Customer

The following figure shows the connection configuration between the ExpressRoute 1 circuit and the hub VNet:

[Move](#) [Delete](#)

Resource group (change)	Data in
ASH-Cust30	0 B
Status	Data out
Succeeded	0 B
Location	Virtual network
East US	VNet01
Subscription (change)	Virtual network gateway
ExpressRoute-Lab	ASH-Cust30-gw (13.90.87.1)
Subscription ID	Circuit
	ASH-Cust30-ER
Tags (change)	
Click here to add tags	

The following list shows the primary CE router configuration for ExpressRoute private peering connectivity. (Cisco ASR1000 routers are used as CE routers in the test setup.) When site-to-site VPN and ExpressRoute circuits are configured in parallel to connect an on-premises network to Azure, Azure prioritizes the ExpressRoute circuit by default. To avoid asymmetrical routing, the on-premises network also should prioritize ExpressRoute connectivity over site-to-site VPN connectivity. The following configuration establishes prioritization by using the BGP **local-preference** attribute:

```
interface TenGigabitEthernet0/0/0.300
description Customer 30 private peering to Azure
encapsulation dot1Q 30 second-dot1q 300
ip vrf forwarding 30
ip address 192.168.30.17 255.255.255.252
!
interface TenGigabitEthernet1/0/0.30
description Customer 30 to south bound LAN switch
encapsulation dot1Q 30
ip vrf forwarding 30
ip address 192.168.30.0 255.255.255.254
ip ospf network point-to-point
!
router ospf 30 vrf 30
router-id 10.2.30.253
redistribute bgp 65021 subnets route-map BGP20SPF
network 192.168.30.0 0.0.0.1 area 0.0.0.0
default-information originate always
default-metric 10
!
router bgp 65021
!
address-family ipv4 vrf 30
network 10.2.30.0 mask 255.255.255.128
neighbor 192.168.30.18 remote-as 12076
neighbor 192.168.30.18 activate
neighbor 192.168.30.18 next-hop-self
neighbor 192.168.30.18 soft-reconfiguration inbound
neighbor 192.168.30.18 route-map prefer-ER-over-VPN in
neighbor 192.168.30.18 prefix-list Cust30_to_Private out
exit-address-family
!
route-map prefer-ER-over-VPN permit 10
set local-preference 200
!
ip prefix-list Cust30_to_Private seq 10 permit 10.2.30.0/25
!
```

Site-to-site VPN configuration details

The following list shows the primary CE router configuration for site-to-site VPN connectivity:

```

crypto ikev2 proposal Cust30-azure-proposal
  encryption aes-cbc-256 aes-cbc-128 3des
  integrity sha1
  group 2
!
crypto ikev2 policy Cust30-azure-policy
  match address local 66.198.12.106
  proposal Cust30-azure-proposal
!
crypto ikev2 keyring Cust30-azure-keyring
  peer azure
  address 52.168.162.84
  pre-shared-key local IamSecure123
  pre-shared-key remote IamSecure123
!
crypto ikev2 profile Cust30-azure-profile
  match identity remote address 52.168.162.84 255.255.255.255
  identity local address 66.198.12.106
  authentication local pre-share
  authentication remote pre-share
  keyring local Cust30-azure-keyring
!
crypto ipsec transform-set Cust30-azure-ipsec-proposal-set esp-aes 256 esp-sha-hmac
  mode tunnel
!
crypto ipsec profile Cust30-azure-ipsec-profile
  set transform-set Cust30-azure-ipsec-proposal-set
  set ikev2-profile Cust30-azure-profile
!
interface Loopback30
  ip address 66.198.12.106 255.255.255.255
!
interface Tunnel30
  ip vrf forwarding 30
  ip address 10.2.30.125 255.255.255.255
  tunnel source Loopback30
  tunnel mode ipsec ipv4
  tunnel destination 52.168.162.84
  tunnel protection ipsec profile Cust30-azure-ipsec-profile
!
router bgp 65021
!
address-family ipv4 vrf 30
  network 10.2.30.0 mask 255.255.255.128
  neighbor 10.10.30.254 remote-as 65515
  neighbor 10.10.30.254 ebgp-multihop 5
  neighbor 10.10.30.254 update-source Tunnel30
  neighbor 10.10.30.254 activate
  neighbor 10.10.30.254 soft-reconfiguration inbound
exit-address-family
!
ip route vrf 30 10.10.30.254 255.255.255.255 Tunnel30

```

On-premises Location 2 connectivity by using ExpressRoute

A second ExpressRoute circuit, in closer proximity to on-premises Location 2, connects on-premises Location 2 to the hub VNet. The following figure shows the second ExpressRoute configuration:

		Move	Delete	Refresh
Resource group (change)	ASH-Cust30	Provider	Equinix	
Circuit status	Enabled	Provider status	Provisioned	
Location	East US	Peering location	Seattle	
Subscription (change)	ExpressRoute-Lab	Bandwidth	50 Mbps	
Subscription ID		Service key		
Tags (change)				
	Click here to add tags			
<hr/>				
Peerings				
Type	Status	Primary Subnet	Secondary Subnet	Last modified by
Azure private	Provisioned	192.168.31.16/30	192.168.31.20/30	Customer
				...

The following figure shows the connection configuration between the second ExpressRoute circuit and the hub VNet:

Resource group (change)	ASH-Cust30	Data in 0 B
Status	Succeeded	Data out 0 B
Location	East US	Virtual network VNet01
Subscription (change)	ExpressRoute-Lab	Virtual network gateway ASH-Cust30-gw (13.90.87.1)
Subscription ID		Circuit SEA-Cust31-ER
Tags (change)		
	Click here to add tags	

ExpressRoute 1 connects both the hub VNet and on-premises Location 1 to a remote VNet in a different Azure region:

		Move	Delete
Resource group (change)	ASH-Cust30	Data in 0 B	
Status	Succeeded	Data out 0 B	
Location	West US 2	Virtual network USWst2-VNet	
Subscription (change)	ExpressRoute-Lab	Virtual network gateway ASH30-USWst2-ERGW (52.175.245.182)	
Subscription ID		Circuit ASH-Cust30-ER	
Tags (change)			
	Click here to add tags		

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than

ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [control plane analysis](#) of the test setup and the views of different VNets or VLANs in the topology.

Learn about [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure back-end connectivity features: Control plane analysis

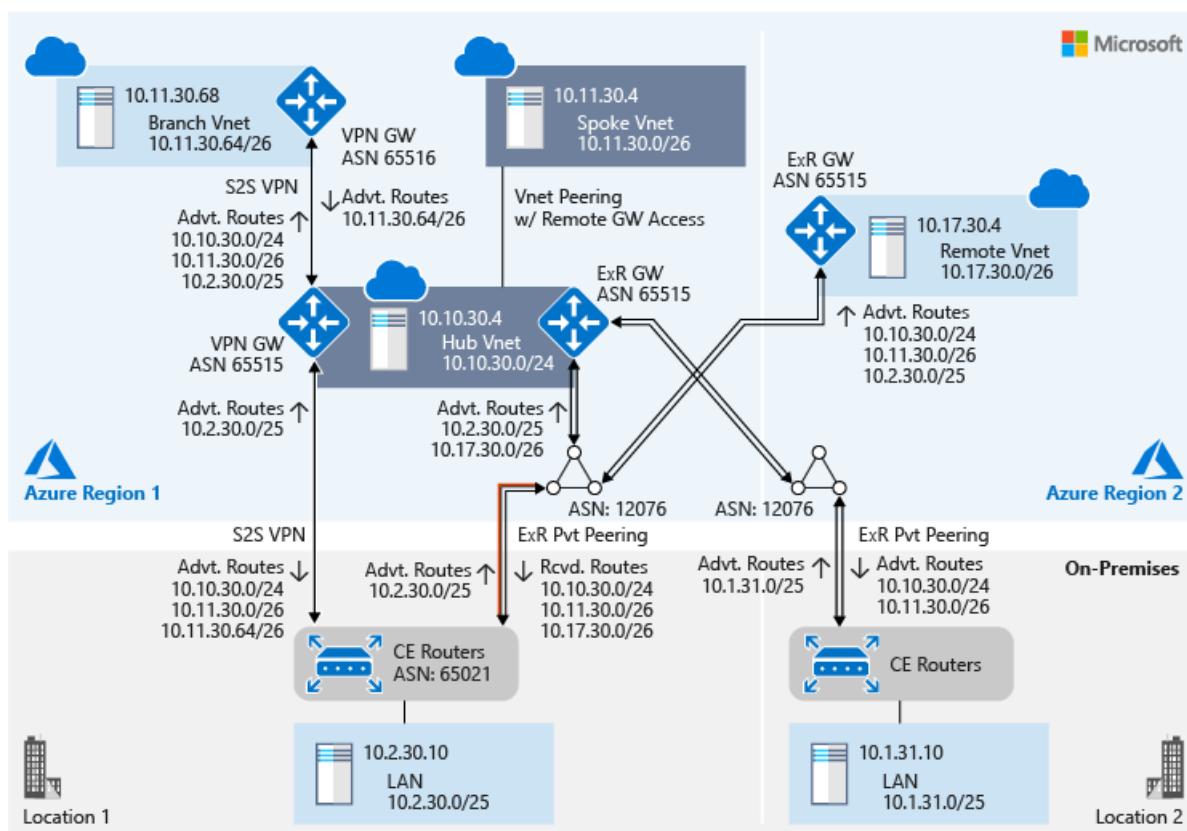
12/5/2019 • 4 minutes to read • [Edit Online](#)

This article describes the control plane analysis of the [test setup](#). You can also review the [test setup configuration](#) and the [data plane analysis](#) of the test setup.

Control plane analysis essentially examines routes that are exchanged between networks within a topology. Control plane analysis can help you understand how different networks view the topology.

Hub and spoke VNet perspective

The following figure illustrates the network from the perspective of a hub virtual network (VNet) and a spoke VNet (highlighted in blue). The figure also shows the autonomous system number (ASN) of different networks and routes that are exchanged between different networks:



The ASN of the VNet's Azure ExpressRoute gateway is different from the ASN of Microsoft Enterprise Edge Routers (MSEEs). An ExpressRoute gateway uses a private ASN (a value of **65515**) and MSEEs use public ASN (a value of **12076**) globally. When you configure ExpressRoute peering, because MSEE is the peer, you use **12076** as the peer ASN. On the Azure side, MSEE establishes eBGP peering with the ExpressRoute gateway. The dual eBGP peering that the MSEE establishes for each ExpressRoute peering is transparent at the control plane level. Therefore, when you view an ExpressRoute route table, you see the VNet's ExpressRoute gateway ASN for the VNet's prefixes.

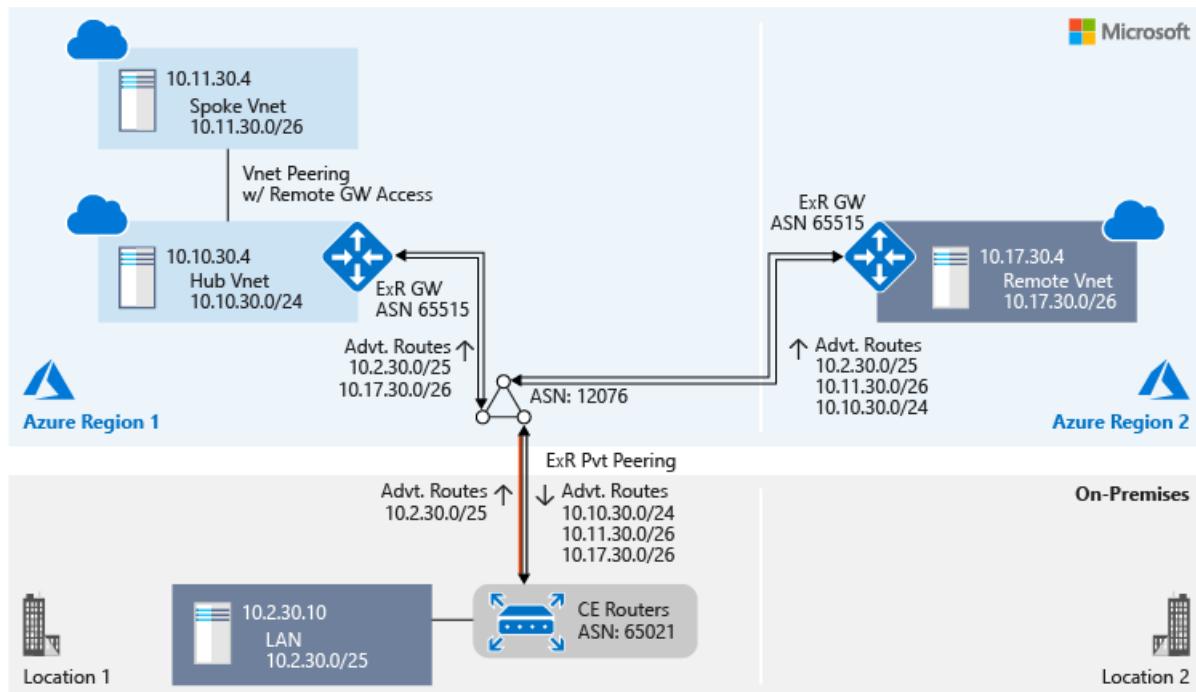
The following figure shows a sample ExpressRoute route table:

Route table (Primary)					
AzurePrivatePeering - ASH-Cust30-ER					
Download	Show secondary				
i Showing only top 200 primary records, click Download above to see all.					
NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH	
10.2.30.0/25	192.168.30.17		0	65021	
10.2.30.125/32	192.168.30.17		0	65021 65515	
10.10.30.0/24	10.10.30.141		0	65515	
	10.10.30.140		0	65515	
10.11.30.0/26	10.10.30.141		0	65515	
	10.10.30.140		0	65515	

Within Azure, the ASN is significant only from a peering perspective. By default, the ASN of both the ExpressRoute gateway and the VPN gateway in Azure VPN Gateway is **65515**.

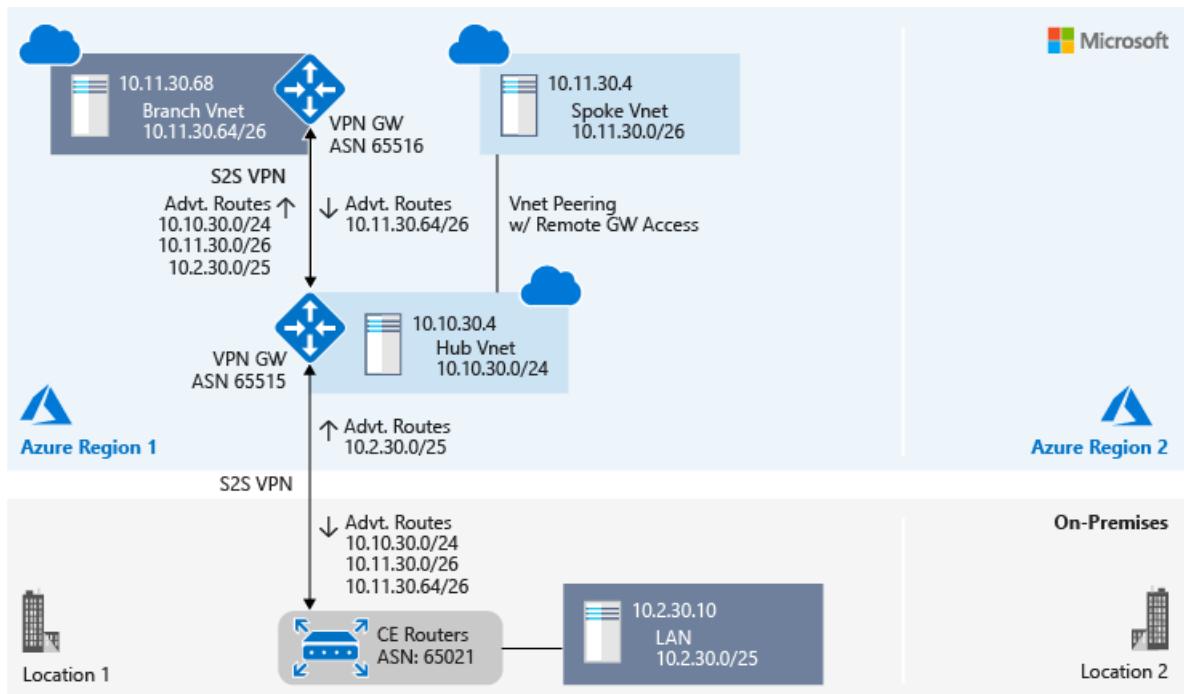
On-premises Location 1 and the remote VNet perspective via ExpressRoute 1

Both on-premises Location 1 and the remote VNet are connected to the hub VNet via ExpressRoute 1. They share the same perspective of the topology, as shown in the following diagram:



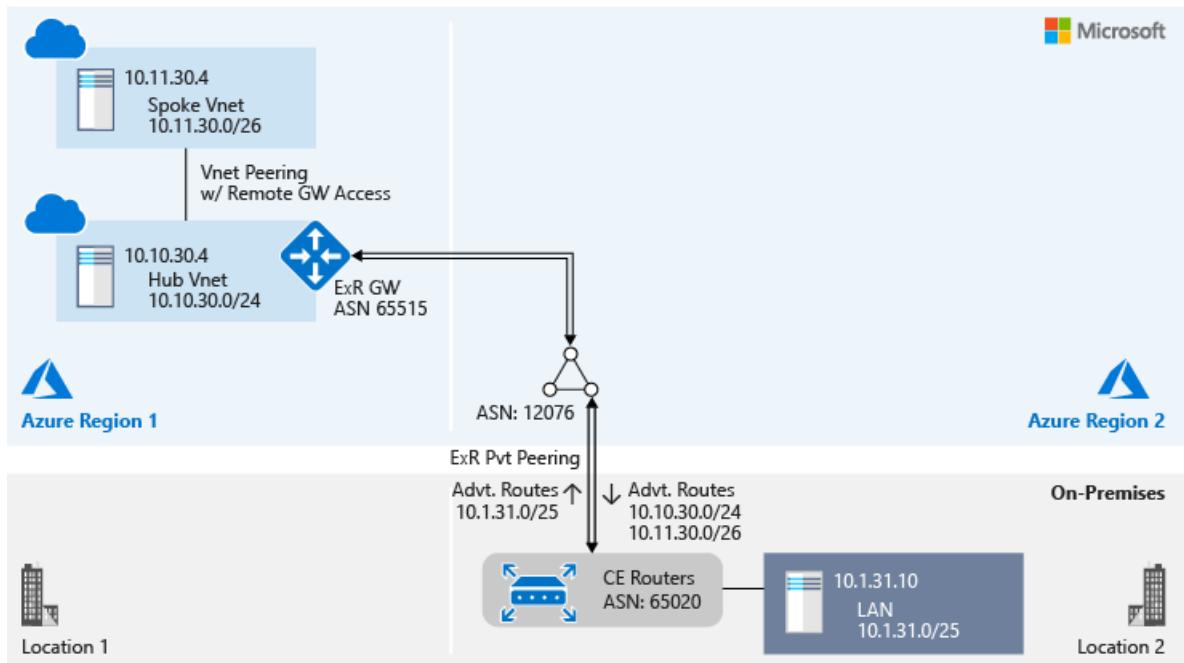
On-premises Location 1 and the branch VNet perspective via a site-to-site VPN

Both on-premises Location 1 and the branch VNet are connected to a hub VNet's VPN gateway via a site-to-site VPN connection. They share the same perspective of the topology, as shown in the following diagram:



On-premises Location 2 perspective

On-premises Location 2 is connected to a hub VNet via private peering of ExpressRoute 2:



ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

Learn about [data plane analysis](#) of the test setup and Azure network monitoring feature views.

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Interoperability in Azure back-end connectivity features: Data plane analysis

2/21/2020 • 16 minutes to read • [Edit Online](#)

This article describes the data plane analysis of the [test setup](#). You can also review the [test setup configuration](#) and the [control plane analysis](#) of the test setup.

Data plane analysis examines the path taken by packets that traverse from one local network (LAN or virtual network) to another within a topology. The data path between two local networks isn't necessarily symmetrical. Therefore, in this article, we analyze a forwarding path from a local network to another network that's separate from the reverse path.

Data path from the hub VNet

Path to the spoke VNet

Virtual network (VNet) peering emulates network bridge functionality between the two VNets that are peered. Traceroute output from a hub VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1      2 ms      1 ms      1 ms  10.11.30.4

Trace complete.
```

The following figure shows the graphical connection view of the hub VNet and the spoke VNet from the perspective of Azure Network Watcher:



Path to the branch VNet

Traceroute output from a hub VNet to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1      1 ms      1 ms      1 ms  10.10.30.142
 2      *          *          *      Request timed out.
 3      2 ms      2 ms      2 ms  10.11.30.68

Trace complete.
```

In this traceroute, the first hop is the VPN gateway in Azure VPN Gateway of the hub VNet. The second hop is the VPN gateway of the branch VNet. The IP address of the VPN gateway of the branch VNet isn't advertised in the hub VNet. The third hop is the VM on the branch VNet.

The following figure shows the graphical connection view of the hub VNet and the branch VNet from the perspective of Network Watcher:



For the same connection, the following figure shows the grid view in Network Watcher:

Hops				
NAME	IP ADDRESS	STATUS	NEXT HOP IP ADDRESS	
ash-cust30-vm1195	10.10.30.4	✓	13.90.87.1	
ASH-Cust30-gw	13.90.87.1	✓	10.11.30.68	
ash-c30-sk2-vm1619	10.11.30.68	✓	-	

Average Latency in milliseconds

3

Minimum Latency in milliseconds

2

Maximum Latency in milliseconds

6

Probes Sent

66

Probes Failed

0

Path to on-premises Location 1

Traceroute output from a hub VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  2 ms    2 ms    2 ms  10.10.30.132
 2  *        *        *      Request timed out.
 3  *        *        *      Request timed out.
 4  2 ms    2 ms    2 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the Azure ExpressRoute gateway tunnel endpoint to a Microsoft Enterprise Edge Router (MSEE). The second and third hops are the customer edge (CE) router and the on-premises Location 1 LAN IPs. These IP addresses aren't advertised in the hub VNet. The fourth hop is the VM in the on-premises Location 1.

Path to on-premises Location 2

Traceroute output from a hub VNet to a VM in on-premises Location 2 is shown here:

```
C:\Users\rb>tracert 10.1.31.10

Tracing route to 10.1.31.10 over a maximum of 30 hops

 1  76 ms   75 ms   75 ms  10.10.30.134
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  75 ms   75 ms   75 ms  10.1.31.10

Trace complete.
```

In this traceroute, the first hop is the ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 2 LAN IPs. These IP addresses aren't advertised in the hub VNet. The fourth hop is the VM on the on-premises Location 2.

Path to the remote VNet

Traceroute output from a hub VNet to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1  2 ms   2 ms   2 ms  10.10.30.132
 2  *         *         *      Request timed out.
 3  69 ms   68 ms   69 ms  10.17.30.4

Trace complete.
```

In this traceroute, the first hop is the ExpressRoute gateway tunnel endpoint to an MSEE. The second hop is the remote VNet's gateway IP. The second hop IP range isn't advertised in the hub VNet. The third hop is the VM on the remote VNet.

Data path from the spoke VNet

The spoke VNet shares the network view of the hub VNet. Through VNet peering, the spoke VNet uses the remote gateway connectivity of the hub VNet as if it's directly connected to the spoke VNet.

Path to the hub VNet

Traceroute output from the spoke VNet to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms   <1 ms   <1 ms  10.10.30.4

Trace complete.
```

Path to the branch VNet

Traceroute output from the spoke VNet to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1  1 ms    <1 ms    <1 ms  10.10.30.142
 2  *         *         *      Request timed out.
 3  3 ms    2 ms    2 ms  10.11.30.68

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the hub VNet. The second hop is the VPN gateway of the branch VNet. The IP address of the VPN gateway of the branch VNet isn't advertised within the hub/spoke VNet. The third hop is the VM on the branch VNet.

Path to on-premises Location 1

Traceroute output from the spoke VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  24 ms    2 ms    3 ms  10.10.30.132
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  3 ms    2 ms    2 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 1 LAN IPs. These IP addresses aren't advertised in the hub/spoke VNet. The fourth hop is the VM in the on-premises Location 1.

Path to on-premises Location 2

Traceroute output from the spoke VNet to a VM in on-premises Location 2 is shown here:

```
C:\Users\rb>tracert 10.1.31.10

Tracing route to 10.1.31.10 over a maximum of 30 hops

 1  76 ms    75 ms    76 ms  10.10.30.134
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  75 ms    75 ms    75 ms  10.1.31.10

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second and third hops are the CE router and the on-premises Location 2 LAN IPs. These IP addresses aren't advertised in the hub/spoke VNets. The fourth hop is the VM in the on-premises Location 2.

Path to the remote VNet

Traceroute output from the spoke VNet to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1  2 ms    1 ms    1 ms  10.10.30.133
 2  *        *        *      Request timed out.
 3  71 ms   70 ms   70 ms  10.17.30.4

Trace complete.
```

In this traceroute, the first hop is the hub VNet's ExpressRoute gateway tunnel endpoint to an MSEE. The second hop is the remote VNet's gateway IP. The second hop IP range isn't advertised in the hub/spoke VNet. The third hop is the VM on the remote VNet.

Data path from the branch VNet

Path to the hub VNet

Traceroute output from the branch VNet to a VM in the hub VNet is shown here:

```
C:\Windows\system32>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.11.30.100
 2  *        *        *      Request timed out.
 3  4 ms    3 ms    3 ms  10.10.30.4

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VM on the hub VNet.

Path to the spoke VNet

Traceroute output from the branch VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  1 ms    <1 ms    1 ms  10.11.30.100
 2  *        *        *      Request timed out.
 3  4 ms    3 ms    2 ms  10.11.30.4

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VM on the spoke VNet.

Path to on-premises Location 1

Traceroute output from the branch VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  1 ms    <1 ms    <1 ms  10.11.30.100
 2  *         *         *      Request timed out.
 3  3 ms    2 ms    2 ms  10.2.30.125
 4  *         *         *      Request timed out.
 5  3 ms    3 ms    3 ms  10.2.30.10

Trace complete.
```

In this traceroute, the first hop is the VPN gateway of the branch VNet. The second hop is the VPN gateway of the hub VNet. The IP address of the VPN gateway of the hub VNet isn't advertised in the remote VNet. The third hop is the VPN tunnel termination point on the primary CE router. The fourth hop is an internal IP address of on-premises Location 1. This LAN IP address isn't advertised outside the CE router. The fifth hop is the destination VM in the on-premises Location 1.

Path to on-premises Location 2 and the remote VNet

As we discussed in the control plane analysis, the branch VNet has no visibility either to on-premises Location 2 or to the remote VNet per the network configuration. The following ping results confirm:

```
C:\Users\rb>ping 10.1.31.10

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.1.31.10:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\rb>ping 10.17.30.4

Pinging 10.17.30.4 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.17.30.4:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Data path from on-premises Location 1

Path to the hub VNet

Traceroute output from on-premises Location 1 to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1 <1 ms    <1 ms    <1 ms  10.2.30.3
 2 <1 ms    <1 ms    <1 ms  192.168.30.0
 3 <1 ms    <1 ms    <1 ms  192.168.30.18
 4 *        *        * Request timed out.
 5 2 ms     2 ms     2 ms  10.10.30.4

Trace complete.
```

In this traceroute, the first two hops are part of the on-premises network. The third hop is the primary MSEE interface that faces the CE router. The fourth hop is the ExpressRoute gateway of the hub VNet. The IP range of the ExpressRoute gateway of the hub VNet isn't advertised to the on-premises network. The fifth hop is the destination VM.

Network Watcher provides only an Azure-centric view. For an on-premises perspective, we use Azure Network Performance Monitor. Network Performance Monitor provides agents that you can install on servers in networks outside Azure for data path analysis.

The following figure shows the topology view of the on-premises Location 1 VM connectivity to the VM on the hub VNet via ExpressRoute:



As discussed earlier, the test setup uses a site-to-site VPN as backup connectivity for ExpressRoute between the on-premises Location 1 and the hub VNet. To test the backup data path, let's induce an ExpressRoute link failure between the on-premises Location 1 primary CE router and the corresponding MSEE. To induce an ExpressRoute link failure, shut down the CE interface that faces the MSEE:

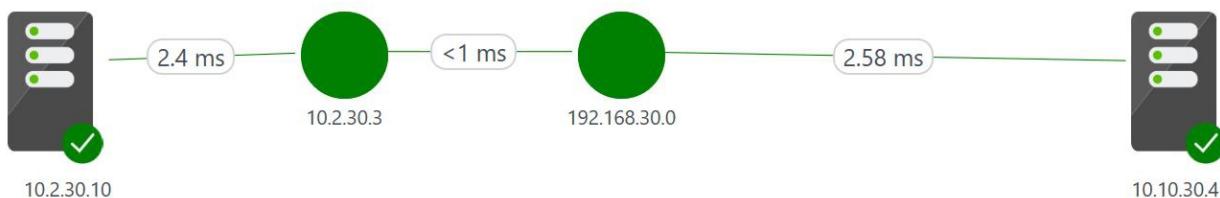
```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1 <1 ms    <1 ms    <1 ms  10.2.30.3
 2 <1 ms    <1 ms    <1 ms  192.168.30.0
 3 3 ms     2 ms     3 ms  10.10.30.4

Trace complete.
```

The following figure shows the topology view of the on-premises Location 1 VM connectivity to the VM on the hub VNet via site-to-site VPN connectivity when ExpressRoute connectivity is down:



Path to the spoke VNet

Traceroute output from on-premises Location 1 to a VM in the spoke VNet is shown here:

Let's bring back the ExpressRoute primary connectivity to do the data path analysis toward the spoke VNet:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  <1 ms    <1 ms    <1 ms  192.168.30.18
 4  *          *          *      Request timed out.
 5  3 ms     2 ms     2 ms  10.11.30.4

Trace complete.
```

Bring up the primary ExpressRoute 1 connectivity for the remainder of the data path analysis.

Path to the branch VNet

Traceroute output from on-premises Location 1 to a VM in the branch VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.68

Tracing route to 10.11.30.68 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  <1 ms    <1 ms    <1 ms  192.168.30.0
 3  3 ms     2 ms     2 ms  10.11.30.68

Trace complete.
```

Path to on-premises Location 2

As we discuss in the [control plane analysis](#), the on-premises Location 1 has no visibility to on-premises Location 2 per the network configuration. The following ping results confirm:

```
C:\Users\rb>ping 10.1.31.10

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

Ping statistics for 10.1.31.10:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Path to the remote VNet

Traceroute output from on-premises Location 1 to a VM in the remote VNet is shown here:

```
C:\Users\rb>tracert 10.17.30.4

Tracing route to 10.17.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.2.30.3
 2  2 ms     5 ms     7 ms   192.168.30.0
 3  <1 ms    <1 ms    <1 ms  192.168.30.18
 4  *         *         *      Request timed out.
 5  69 ms    70 ms    69 ms  10.17.30.4

Trace complete.
```

Data path from on-premises Location 2

Path to the hub VNet

Traceroute output from on-premises Location 2 to a VM in the hub VNet is shown here:

```
C:\Windows\system32>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.1.31.3
 2  <1 ms    <1 ms    <1 ms  192.168.31.4
 3  <1 ms    <1 ms    <1 ms  192.168.31.22
 4  *         *         *      Request timed out.
 5  75 ms    74 ms    74 ms  10.10.30.4

Trace complete.
```

Path to the spoke VNet

Traceroute output from on-premises Location 2 to a VM in the spoke VNet is shown here:

```
C:\Windows\system32>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  <1 ms    <1 ms    1 ms   10.1.31.3
 2  <1 ms    <1 ms    <1 ms  192.168.31.0
 3  <1 ms    <1 ms    <1 ms  192.168.31.18
 4  *         *         *      Request timed out.
 5  75 ms    74 ms    74 ms  10.11.30.4

Trace complete.
```

Path to the branch VNet, on-premises Location 1, and the remote VNet

As we discuss in the [control plane analysis](#), the on-premises Location 1 has no visibility to the branch VNet, to on-premises Location 1, or to the remote VNet per the network configuration.

Data path from the remote VNet

Path to the hub VNet

Traceroute output from the remote VNet to a VM in the hub VNet is shown here:

```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  65 ms    65 ms    65 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  69 ms    68 ms    68 ms  10.10.30.4

Trace complete.
```

Path to the spoke VNet

Traceroute output from the remote VNet to a VM in the spoke VNet is shown here:

```
C:\Users\rb>tracert 10.11.30.4

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1  67 ms    67 ms    67 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  71 ms    69 ms    69 ms  10.11.30.4

Trace complete.
```

Path to the branch VNet and on-premises Location 2

As we discuss in the [control plane analysis](#), the remote VNet has no visibility to the branch VNet or to on-premises Location 2 per the network configuration.

Path to on-premises Location 1

Traceroute output from the remote VNet to a VM in on-premises Location 1 is shown here:

```
C:\Users\rb>tracert 10.2.30.10

Tracing route to 10.2.30.10 over a maximum of 30 hops

 1  67 ms    67 ms    67 ms  10.17.30.36
 2  *         *         *      Request timed out.
 3  *         *         *      Request timed out.
 4  69 ms    69 ms    69 ms  10.2.30.10

Trace complete.
```

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

You can configure a site-to-site VPN by using ExpressRoute Microsoft peering to privately exchange data between your on-premises network and your Azure VNets. With this configuration, you can exchange data with confidentiality, authenticity, and integrity. The data exchange also is anti-replay. For more information about how to configure a site-to-site IPsec VPN in tunnel mode by using ExpressRoute Microsoft peering, see [Site-to-site VPN over ExpressRoute Microsoft peering](#).

The primary limitation of configuring a site-to-site VPN that uses Microsoft peering is throughput. Throughput over the IPsec tunnel is limited by the VPN gateway capacity. The VPN gateway throughput is lower than ExpressRoute throughput. In this scenario, using the IPsec tunnel for highly secure traffic and using private peering for all other traffic helps optimize the ExpressRoute bandwidth utilization.

Site-to-site VPN as a secure failover path for ExpressRoute

ExpressRoute serves as a redundant circuit pair to ensure high availability. You can configure geo-redundant ExpressRoute connectivity in different Azure regions. Also, as demonstrated in our test setup, within an Azure region, you can use a site-to-site VPN to create a failover path for your ExpressRoute connectivity. When the same prefixes are advertised over both ExpressRoute and a site-to-site VPN, Azure prioritizes ExpressRoute. To avoid asymmetrical routing between ExpressRoute and the site-to-site VPN, on-premises network configuration should also reciprocate by using ExpressRoute connectivity before it uses site-to-site VPN connectivity.

For more information about how to configure coexisting connections for ExpressRoute and a site-to-site VPN, see [ExpressRoute and site-to-site coexistence](#).

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

Hub and spoke VNet architecture is widely used. The hub is a VNet in Azure that acts as a central point of connectivity between your spoke VNets and to your on-premises network. The spokes are VNets that peer with the hub, and which you can use to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN connection. For more information about the architecture, see [Implement a hub-spoke network topology in Azure](#).

In VNet peering within a region, spoke VNets can use hub VNet gateways (both VPN and ExpressRoute gateways) to communicate with remote networks.

Branch VNet connectivity by using site-to-site VPN

You might want branch VNets, which are in different regions, and on-premises networks to communicate with each other via a hub VNet. The native Azure solution for this configuration is site-to-site VPN connectivity by using a VPN. An alternative is to use a network virtual appliance (NVA) for routing in the hub.

For more information, see [What is VPN Gateway?](#) and [Deploy a highly available NVA](#).

Next steps

See the [ExpressRoute FAQ](#) to:

- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Security controls for Azure VPN Gateway

1/14/2020 • 2 minutes to read • [Edit Online](#)

This article documents the security controls built into Azure VPN Gateway.

A security control is a quality or feature of an Azure service that contributes to the service's ability to prevent, detect, and respond to security vulnerabilities.

For each control, we use "Yes" or "No" to indicate whether it is currently in place for the service, "N/A" for a control that is not applicable to the service. We might also provide a note or links to more information about an attribute.

Network

SECURITY CONTROL	YES/NO	NOTES
Service endpoint support	N/A	
VNet injection support	N/A	
Network Isolation and Firewalling support	Yes	VPN gateways are dedicated VM instances for each customer Virtual Network
Forced tunneling support	Yes	

Monitoring & logging

SECURITY CONTROL	YES/NO	NOTES
Azure monitoring support (Log analytics, App insights, etc.)	Yes	See Azure Monitor Diagnostics Logs/alert & Azure Monitor Metrics/alert .
Control and management plane logging and audit	Yes	Azure Resource Manager Activity Log.
Data plane logging and audit	Yes	Azure Monitor Diagnostic Logs for VPN connectivity logging and auditing.

Identity

SECURITY CONTROL	YES/NO	NOTES
Authentication	Yes	Azure Active Directory for managing the service and configuring the Azure VPN gateway.
Authorization	Yes	Support Authorization via RBAC .

Data protection

SECURITY CONTROL	YES/NO	NOTES
Server-side encryption at rest: Microsoft-managed keys	N/A	VPN gateway transit customer data, does NOT store customer data
Encryption in transit (such as ExpressRoute encryption, in VNet encryption, and VNet-VNet encryption)	Yes	VPN gateway encrypt customer packets between Azure VPN gateways and customer on-premises VPN devices (S2S) or VPN clients (P2S). VPN gateways also support VNet-to-VNet encryption.
Server-side encryption at rest: customer-managed keys (BYOK)	No	Customer-specified pre-shared keys are encrypted at rest; but not integrated with CMK yet.
Column level encryption (Azure Data Services)	N/A	
API calls encrypted	Yes	Through Azure Resource Manager and HTTPS

Configuration management

SECURITY CONTROL	YES/NO	NOTES
Configuration management support (versioning of configuration, etc.)	Yes	For management operations, the state of an Azure VPN gateway configuration can be exported as an Azure Resource Manager template and versioned over time.

Next steps

- Learn more about the [built-in security controls across Azure services](#).

Create a route-based VPN gateway using the Azure portal

11/17/2019 • 6 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.

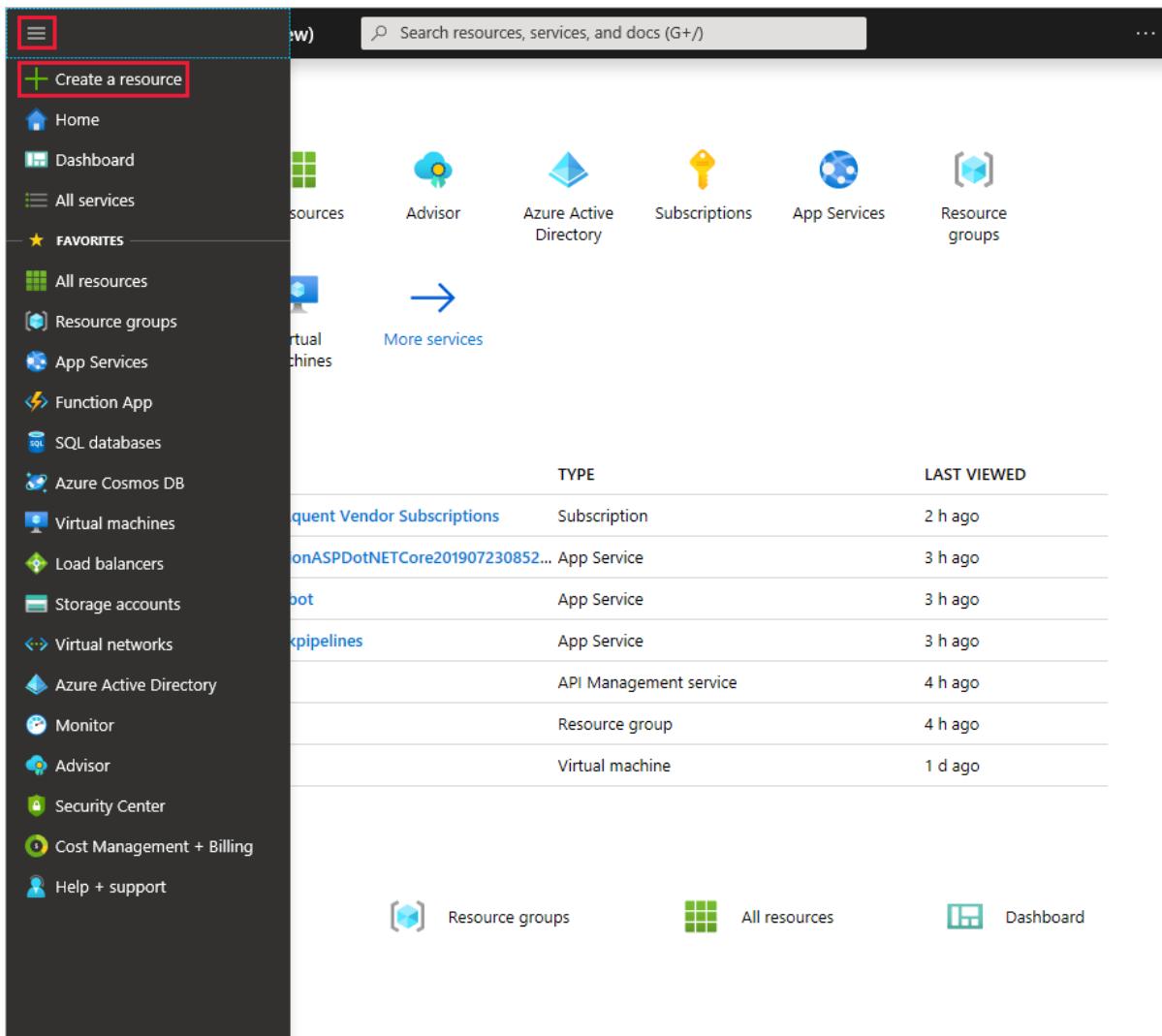
Create a virtual network

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 the [Azure portal](#) menu, select **Create a resource**.



2. 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. 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:** VNet1
 - **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 (click **Create new** to create a new group)
 - **Location:** East US
 - **Subnet:** Frontend
 - **Address range:** 10.1.0.0/24

Create virtual network

* Name
VNet1

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

* Subscription

* Resource group
(New) TestRG1
[Create new](#)

* Location
(US) East US

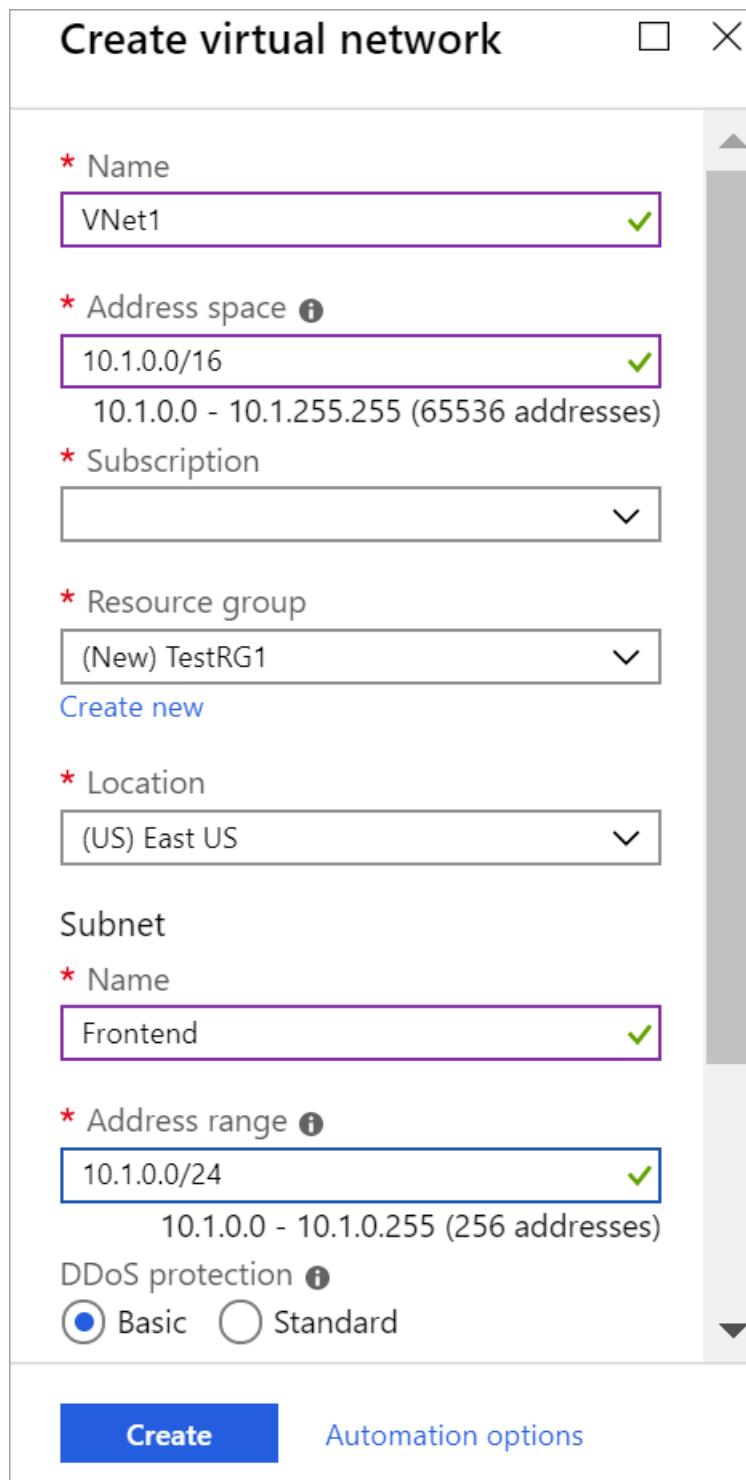
Subnet

* Name
Frontend

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

DDoS protection ⓘ
 Basic Standard

Create [Automation options](#)



5. Leave DDoS as Basic, Service endpoints as Disabled, and Firewall as Disabled.

6. Click **Create** to create the VNet.

Configure and create the gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses

a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

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**. This opens the **Create virtual network gateway** page.

The screenshot shows the 'Create virtual network gateway' wizard in the 'Basics' step. It includes tabs for 'Basics', 'Tags', and 'Review + create'. A note says 'Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)'

Project details

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

* Subscription: A dropdown menu.

Resource group ⓘ: A dropdown menu labeled 'Select a virtual network to get resource group'.

Instance details

* Name: VNet1GW

* Region: (US) East US

* Gateway type ⓘ: VPN ExpressRoute

* VPN type ⓘ: Route-based Policy-based

* SKU ⓘ: VpnGw1

Note: ⓘ Only virtual networks in the currently selected subscription and region are listed.

VIRTUAL NETWORK

* Virtual network 	VNet1 
Gateway subnet address range	10.1.255.0/27 
Public IP address	
* Public IP address 	<input checked="" type="radio"/> Create new <input type="radio"/> Use existing
* Public IP address name	VNet1GWpip 
Public IP address SKU	Basic
* Assignment	<input checked="" type="radio"/> Dynamic <input type="radio"/> Static
* Enable active-active mode 	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled
* Configure BGP ASN 	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled
Azure recommends using a validated VPN device with your virtual network gateway. To view a list of validated devices and instructions for configuration, refer to Azure's documentation regarding validated VPN devices.	

[Review + create](#)  Previous [Next : Tags >](#) [Download a template for automation](#)

2. On the **Create virtual network gateway** page, fill in the values for your virtual network gateway.

Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).

Virtual network: Choose the virtual network to which you want to add this gateway.

Gateway subnet address range: This field only appears if the virtual network you selected does not have a gateway subnet. Fill in the range if you don't already have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.)

Public IP address: This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.

- **Assignment:** VPN gateway supports only Dynamic.

Active-Active mode: Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting unselected.

Leave **Configure BGP ASN** deselected, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

3. Click **Review + Create** to run validation. Once validation passes, click **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

NOTE

The Basic gateway SKU does not support IKEv2 or RADIUS authentication. If you plan on having Mac clients connect to your virtual network, do not use the Basic SKU.

IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

View the VPN gateway

1. After the gateway is created, navigate to VNet1 in the portal. The VPN gateway appears on the Overview page as a connected device.

DEVICE	TYPE	IP ADDRESS	SUBNET
VNet1GW	Virtual network gateway	-	GatewaySubnet

2. In the device list, click **VNet1GW** to view more information.

Resource group (change) TestRG1	Location East US	Subscription (change) Windows Azure Internal Consum...
Subscription ID	SKU VpnGw1	Gateway type VPN
VPN type Route-based	Virtual network VNet1	Public IP address 52.191.12.152 (VNet1GWIP)
Tags (change) 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

2/11/2020 • 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.

Before you begin

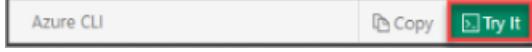
The steps in this article will create a VNet, a subnet, a gateway subnet, and a route-based VPN gateway (virtual network gateway). Once the gateway creation has completed, you can then create connections. These steps require an Azure subscription. If you don't have an Azure subscription, create a [free account](#) before you begin.

Working with Azure PowerShell

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

Create a resource group

Create an Azure resource group with [New-AzResourceGroup](#). A resource group is a logical container into which Azure resources are deployed and managed. Create a resource group. If you are running PowerShell locally, open your PowerShell console with elevated privileges and connect to Azure using the `Connect-AzAccount` command.

```
New-AzResourceGroup -Name TestRG1 -Location EastUS
```

Create a virtual network

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named

VNet1 in the **EastUS** location:

```
$virtualNetwork = New-AzVirtualNetwork `  
    -ResourceGroupName TestRG1 `  
    -Location EastUS `  
    -Name VNet1 `  
    -AddressPrefix 10.1.0.0/16
```

Create a subnet configuration using the [New-AzVirtualNetworkSubnetConfig](#) cmdlet.

```
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `  
    -Name Frontend `  
    -AddressPrefix 10.1.0.0/24 `  
    -VirtualNetwork $virtualNetwork
```

Set the subnet configuration for the virtual network using the [Set-AzVirtualNetwork](#) cmdlet.

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

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-AzVirtualNetwork -ResourceGroupName TestRG1 -Name VNet1
```

Create the gateway subnet using the [Add-AzVirtualNetworkSubnetConfig](#) cmdlet.

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

Set the subnet configuration for the virtual network using the [Set-AzVirtualNetwork](#) cmdlet.

```
$vnet | Set-AzVirtualNetwork
```

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-AzPublicIpAddress -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-AzVirtualNetwork -Name VNet1 -ResourceGroupName TestRG1
$subnet = Get-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gwipconfig = New-AzVirtualNetworkGatewayIpConfig -Name gwipconfig1 -SubnetId $subnet.Id -PublicIpAddressId
$gwip.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-AzVirtualNetworkGateway](#) cmdlet.

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

View the VPN gateway

You can view the VPN gateway using the [Get-AzVirtualNetworkGateway](#) cmdlet.

```
Get-AzVirtualNetworkGateway -Name Vnet1GW -ResourceGroup TestRG1
```

The output will look similar to this example:

```

Name : VNet1GW
ResourceGroupName : TestRG1
Location : eastus
Id : /subscriptions/<subscription ID>/resourceGroups/TestRG1/provide
rs/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/Te
stRG1/providers/Microsoft.Network/virtualNetworks/VNet1/subnets/GatewaySubnet"
        },
        "PublicIpAddress": {
            "Id": "/subscriptions/<subscription ID>/resourceGroups/Te
stRG1/providers/Microsoft.Network/publicIPAddresses/VNet1GWIP"
        },
        "Name": "default",
        "Etag": "W/"0952d-9da8-4d7d-a8ed-28c8ca0413"",
        "Id": "/subscriptions/<subscription ID>/resourceGroups/Test
RG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW/ipConfigurations/de
fault"
    }
]
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-AzPublicIpAddress](#) cmdlet.

```
Get-AzPublicIpAddress -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-AzResourceGroup](#) command to delete the resource group. This will delete the resource group and all of the resources it contains.

```
Remove-AzResourceGroup -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

1/9/2020 • 4 minutes to read • [Edit Online](#)

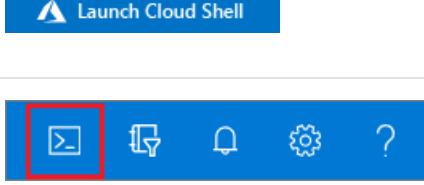
This article helps you quickly create a route-based Azure VPN gateway using the Azure CLI. A VPN gateway is used when creating a VPN connection to your on-premises network. You can also use a VPN gateway to connect VNets.

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.

Use Azure Cloud Shell

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

To start Azure Cloud Shell:

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

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

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

If you choose to install and use 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 the Azure CLI](#).

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

1/9/2020 • 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	Virtual network
East US	
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-AzVirtualNetworkGatewayConnection' cmdlet, with or without '-Debug'.

1. Use the following cmdlet example, configuring the values to match your own. If prompted, select 'A' in order to run 'All'. In the example, '-Name' refers to the name of the connection that you want to test.

```
Get-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. After the cmdlet has finished, view the values. In the example below, the connection status shows as 'Connected' and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

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).

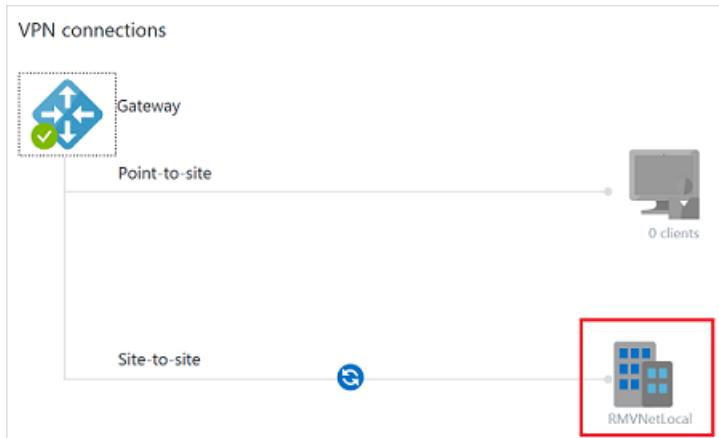
You can verify that your connection succeeded by using the `az network vpn-connection show` command. In the example,'--name' refers to the name of the connection that you want to test. When the connection is in the process of being established, its connection status shows 'Connecting'. Once the connection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

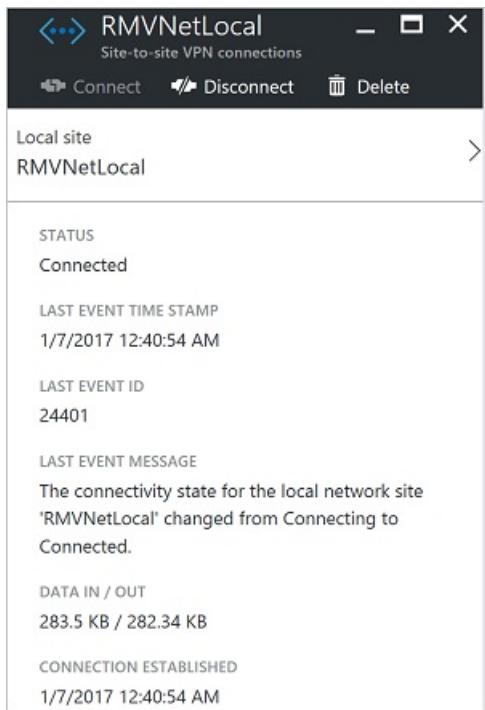


4. On the **Site-to-site VPN connections** blade, view the information about your site.

The screenshot shows the 'Site-to-site VPN connections' blade for the 'ClassicVNet' resource group. It has a header with 'Add' and 'VPN Device Scr...'. Below is a table with columns: NAME, STATUS, LAST EVENT, and DATA IN / OUT. There is one row for 'RMVNetLocal' with a green checkmark in the STATUS column, indicating it is 'Connected'. The LAST EVENT is '1/7/2017 12:40:54 AM' and the DATA IN / OUT is '177.97 KB / 177.72 KB'.

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

1/10/2020 • 4 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 30 to 45 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:

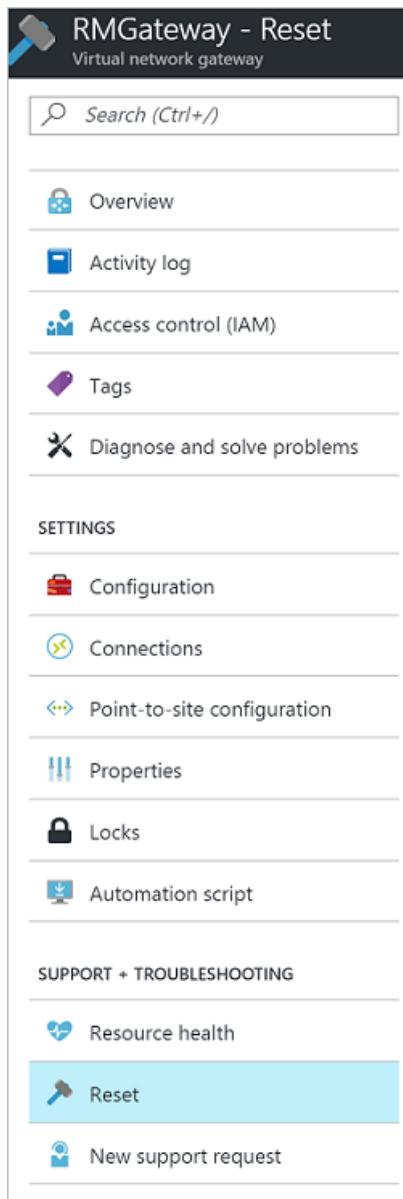
- The Internet IP addresses (VIPs) for both the Azure VPN gateway and the on-premises VPN gateway are configured correctly in both the Azure and the on-premises VPN policies.
- The pre-shared key must be the same on both Azure and on-premises VPN gateways.
- If you apply specific IPsec/IKE configuration, 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

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

The cmdlet for resetting a gateway is **Reset-AzVirtualNetworkGateway**. Before performing a reset, make sure you have the latest version of the [PowerShell Az cmdlets](#). The following example resets a virtual network gateway named VNet1GW in the TestRG1 resource group:

```
$gw = Get-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
Reset-AzVirtualNetworkGateway -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**. The Azure PowerShell cmdlets for Service Management must be installed locally on your desktop. You can't use Azure Cloud Shell. Before performing a reset, make sure you have the latest version of the [Service Management \(SM\) PowerShell cmdlets](#). When using this command, make sure you are using the full name of the virtual network. Classic VNets that were created using the portal have a long name that is required for PowerShell. You can view the long name by using 'Get-AzureVNetConfig -ExportToFile C:\Myfoldername\NetworkConfig.xml'.

The following example resets the gateway for a virtual network named "Group TestRG1 TestVNet1" (which shows as simply "TestVNet1" in the portal):

```
Reset-AzureVNetGateway -VnetName 'Group TestRG1 TestVNet1'
```

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

1/10/2020 • 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

2/12/2020 • 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-AzAccount
```

Check the subscriptions for the account.

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "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-Azvirtualnetworkgateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Check to see if the virtual network gateway has any connections.

```
get-Azvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object {$_.VirtualNetworkGateway1.Id -eq $GW.Id}  
$Conns=get-Azvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object {$_.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-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGateway -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.

```
$LNG=Get-AzLocalNetworkGateway -ResourceGroupName "RG1" | where-object {$_.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-AzLocalNetworkGateway -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.

```
$PubIP=Get-AzPublicIpAddress | where-object {$_.Id -In $GWIpcfgs.PublicIpAddress.Id}
```

Delete the Public IP resources.

```
$PubIP | foreach-object {remove-AzpublicIpAddress -Name $_.Name -ResourceGroupName "RG1"}
```

7. Delete the gateway subnet and set the configuration.

```
$GWSUB = Get-AzVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet"  
Set-AzVirtualNetwork -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-Azvirtualnetworkgateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Check to see if the virtual network gateway has any connections.

```
get-Azvirtualnetworkgatewayconnection -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-Azvirtualnetworkgatewayconnection -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.

```
$ConnsL=get-Azvirtualnetworkgatewayconnection -ResourceGroupName "RG1" | where-object {$_.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.

```
$ConnsR=get-Azvirtualnetworkgatewayconnection -ResourceGroupName "<NameOfResourceGroup2>" | where-object {$_.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-AzVirtualNetworkGatewayConnection -Name $_.name -ResourceGroupName  
$_.ResourceGroupName}  
$ConnsR | ForEach-Object {Remove-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGateway -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.

```
$PubIP=Get-AzPublicIpAddress | where-object {$_.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-AzpublicIpAddress -Name $_.Name -ResourceGroupName "<NameOfResourceGroup1>"}
```

7. Delete the gateway subnet and set the configuration.

```
$GWSUB = Get-AzVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-AzVirtualNetworkSubnetConfig -  
Name "GatewaySubnet"  
Set-AzVirtualNetwork -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-Azvirtualnetworkgateway -Name "GW1" -ResourceGroupName "RG1"
```

2. Delete the virtual network gateway.

You may be prompted to confirm the deletion of the virtual network gateway.

```
Remove-AzVirtualNetworkGateway -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.

```
$PubIP=Get-AzPublicIpAddress | where-object {$_.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-AzpublicIpAddress -Name $_.Name -ResourceGroupName "<NameOfResourceGroup1>"}
```

4. Delete the gateway subnet and set the configuration.

```
$GWSUB = Get-AzVirtualNetwork -ResourceGroupName "RG1" -Name "VNet1" | Remove-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet"  
Set-AzVirtualNetwork -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-AzResourceGroup
```

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-AzResource -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-AzResourceGroup -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-AzResourceGroup -ResourceGroupName RG1
```

The result that is returned shows 'Succeeded'.

```
ResourceGroupName : RG1
Location        : eastus
ProvisioningState : Succeeded
```

Working with virtual network gateway SKUs (legacy SKUs)

1/10/2020 • 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:

- Default (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.

You can view legacy gateway pricing in the **Virtual Network Gateways** section, which is located in on the [ExpressRoute pricing page](#).

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.

Resource Manager

To resize a gateway for the Resource Manager deployment model using PowerShell, use the following command:

```
$gw = Get-AzVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg  
Resize-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -GatewaySku HighPerformance
```

You can also resize a gateway in the Azure portal.

Classic

To resize a gateway for the classic deployment model, you must use the Service Management PowerShell cmdlets. Use the following command:

```
Resize-AzureVirtualNetworkGateway -GatewayId <Gateway ID> -GatewaySKU HighPerformance
```

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](#).

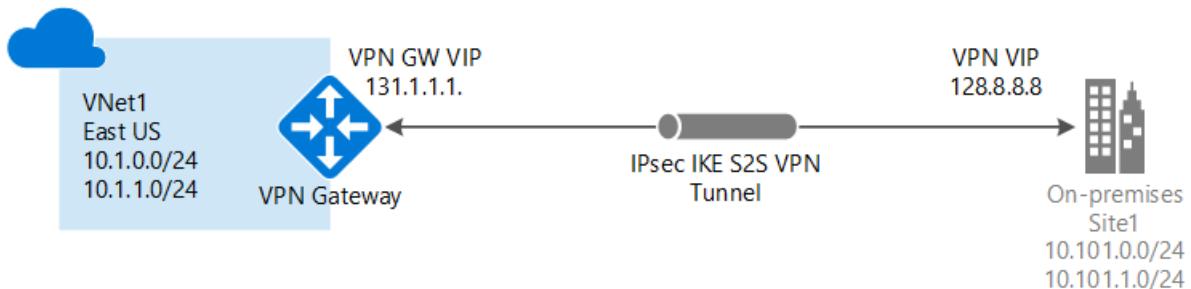
For more information about configuration settings, see [About VPN Gateway configuration settings](#).

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

1/10/2020 • 16 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.
- 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](#).

- **Virtual network name:** VNet1
- **Address Space:** 10.1.0.0/16
- **Subscription:** The subscription you want to use
- **Resource Group:** TestRG1
- **Region:** East US
- **Subnet:** FrontEnd: 10.1.0.0/24, BackEnd: 10.1.1.0/24 (optional for this exercise)
- **Gateway subnet address range:** 10.1.255.0/27
- **Virtual network gateway name:** VNet1GW
- **Public IP address name:** 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.

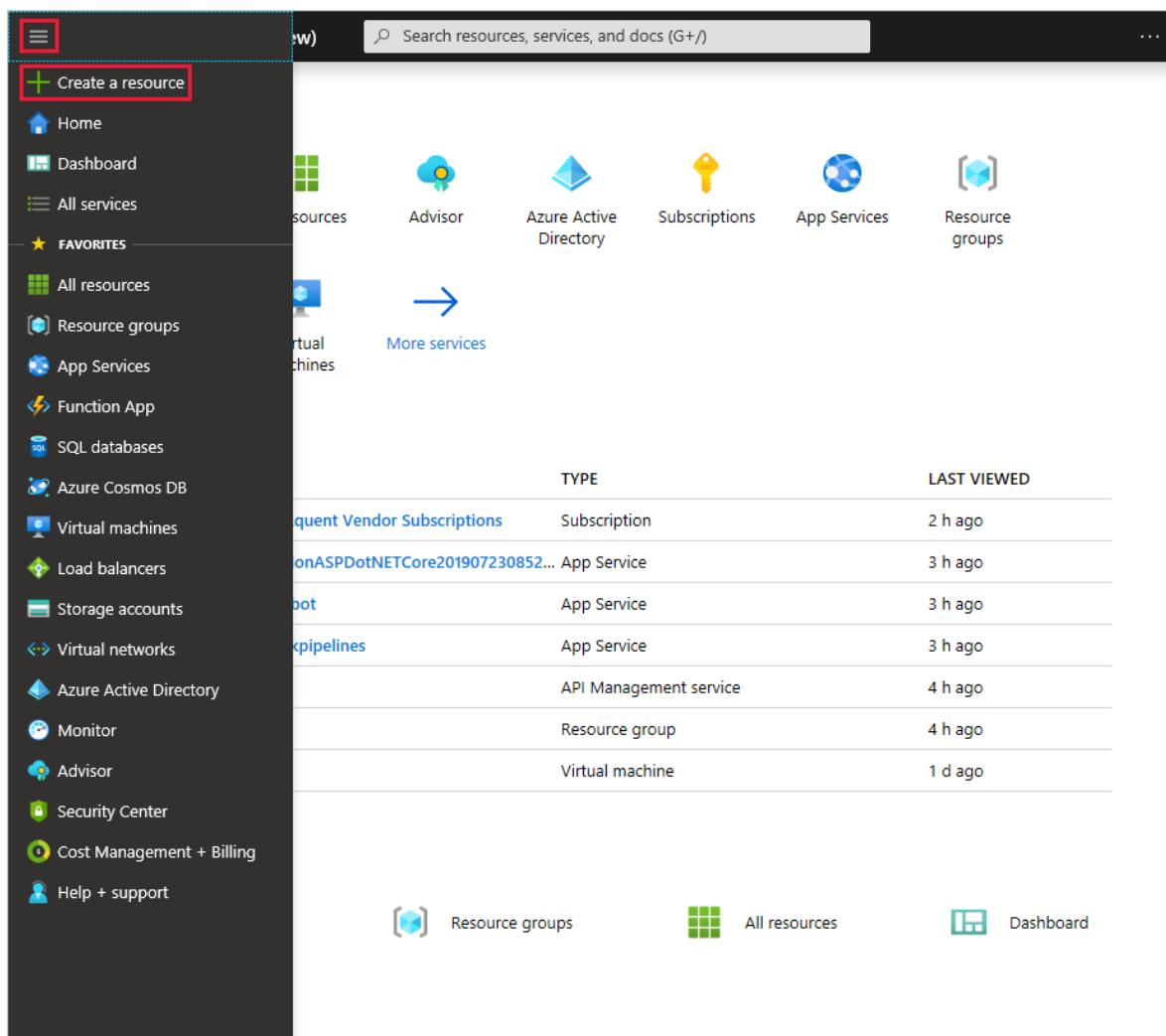
1. Create a virtual network

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 the [Azure portal](#) menu, select **Create a resource**.



2. 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. 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:** VNet1
- **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 (click **Create new** to create a new group)
- **Location:** East US
- **Subnet:** Frontend
- **Address range:** 10.1.0.0/24

Create virtual network

* Name
VNet1 ✓

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

* Subscription
▼

* Resource group
(New) TestRG1 ▼
[Create new](#)

* Location
(US) East US ▼

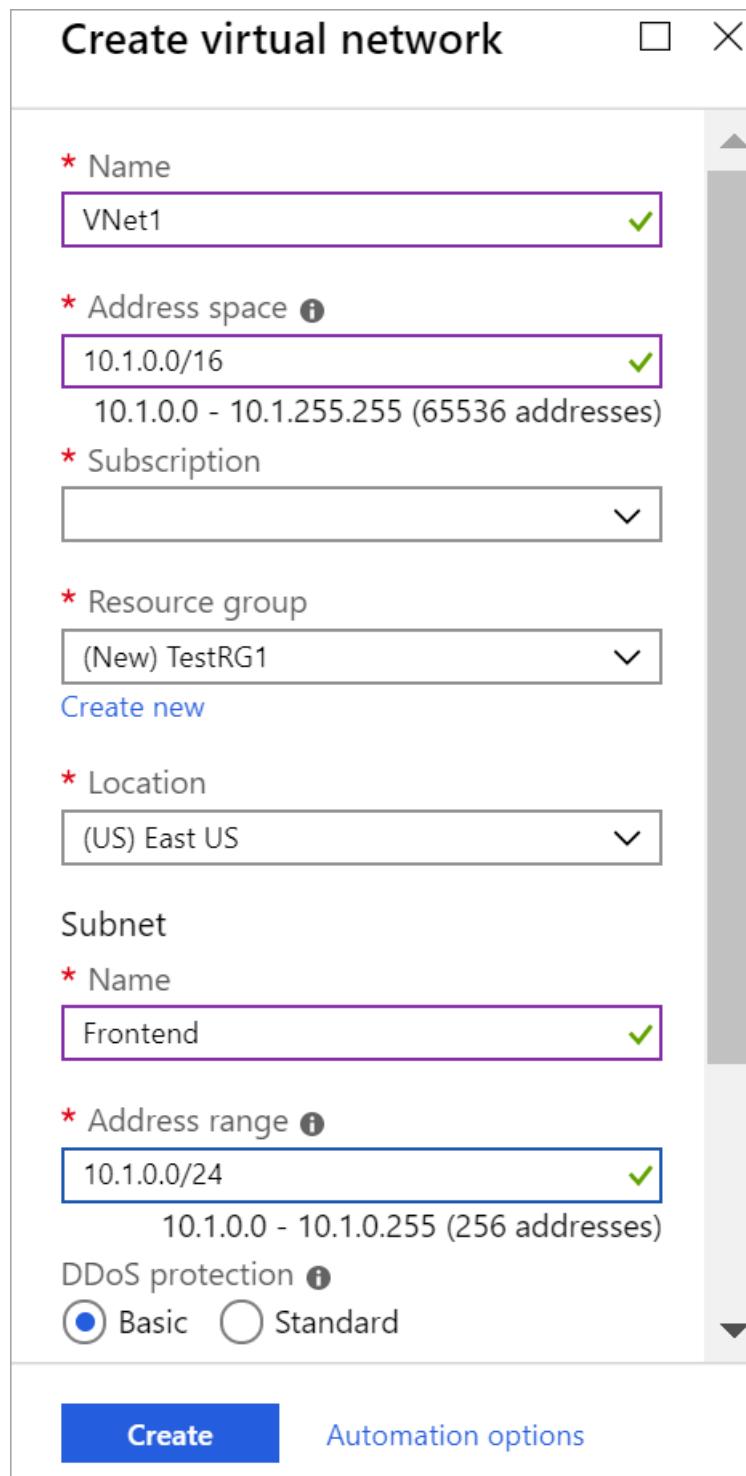
Subnet

* Name
Frontend ✓

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

DDoS protection ⓘ
 Basic Standard

Create Automation options



5. Leave DDoS as Basic, Service endpoints as Disabled, and Firewall as Disabled.

6. Click **Create** to create the VNet.

2. Create the VPN gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that

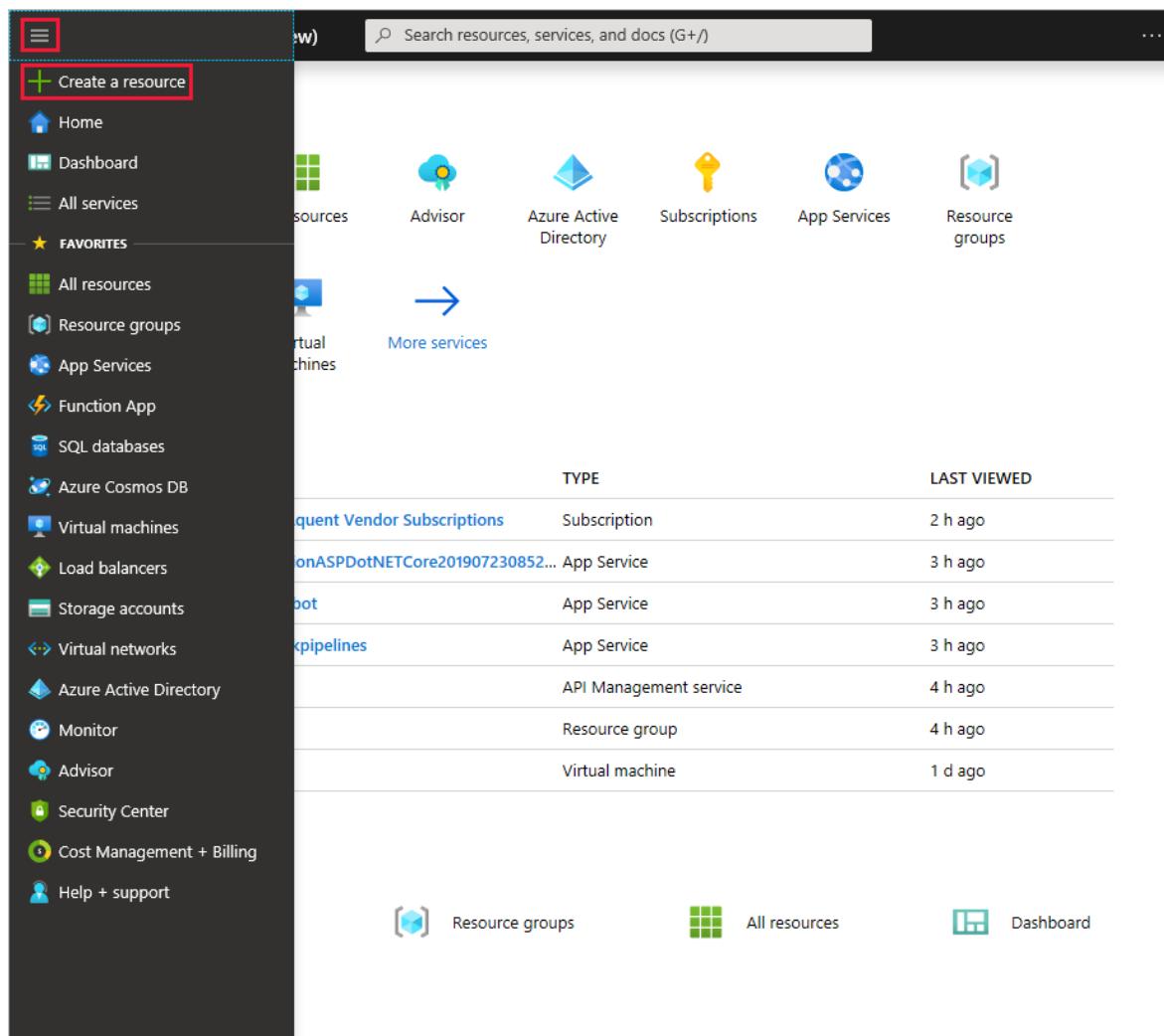
uses a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

Example settings

- **Instance details > Region:** East US
- **Virtual Network > Virtual network:** VNet1
- **Instance details > Name:** VNet1GW
- **Instance details > Gateway type:** VPN
- **Instance details > VPN type:** Route-based
- **Virtual Network > Gateway subnet address range:** 10.1.255.0/27
- **Public IP address > Public IP address name:** VNet1GWIP

1. From the [Azure portal](#) menu, select **Create a resource**.



2. In the **Search the Marketplace** field, type 'Virtual Network Gateway'. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create**. This opens the **Create virtual network gateway** page.

Create virtual network gateway

Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)

Project details

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

Subscription *

Resource group ⓘ

TestRG1 (derived from virtual network's resource group)

Instance details

Name *

 VNet1GW ✓

Region *

 (US) East US ✓

Gateway type * ⓘ

VPN ExpressRoute

VPN type * ⓘ

Route-based Policy-based

SKU * ⓘ

 VpnGw1 ✓

Generation ⓘ

 Generation1 ✓

Virtual network * ⓘ

 VNet1 ✓

[Create virtual network](#)

i Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range * ⓘ

 10.11.255.0/27 ✓

10.11.255.0 - 10.11.255.31 (32 addresses)

Public IP address

Public IP address * ⓘ

Create new Use existing

Public IP address name *

 VNet1GWpip ✓

Public IP address SKU

Basic

Assignment

Dynamic Static

Enable active-active mode * ⓘ

Enabled Disabled

Configure BGP ASN * ⓘ

Enabled Disabled

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

Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.

- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).

Virtual network: Choose the virtual network to which you want to add this gateway.

Gateway subnet address range: This field only appears if your VNet doesn't have a gateway subnet. If possible, make the range /27 or larger (/26,,/25 etc.). We don't recommend creating a range any smaller than /28. If you already have a gateway subnet, you can view GatewaySubnet details by navigating to your virtual network. Click **Subnets** to view the range. If you want to change the range, you can delete and recreate the GatewaySubnet.

Public IP address: This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only Dynamic.

Active-Active mode: Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting unselected.

Leave **Configure BGP ASN** deselected, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

4. Click **Review + create** to run validation. Once validation passes, click **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

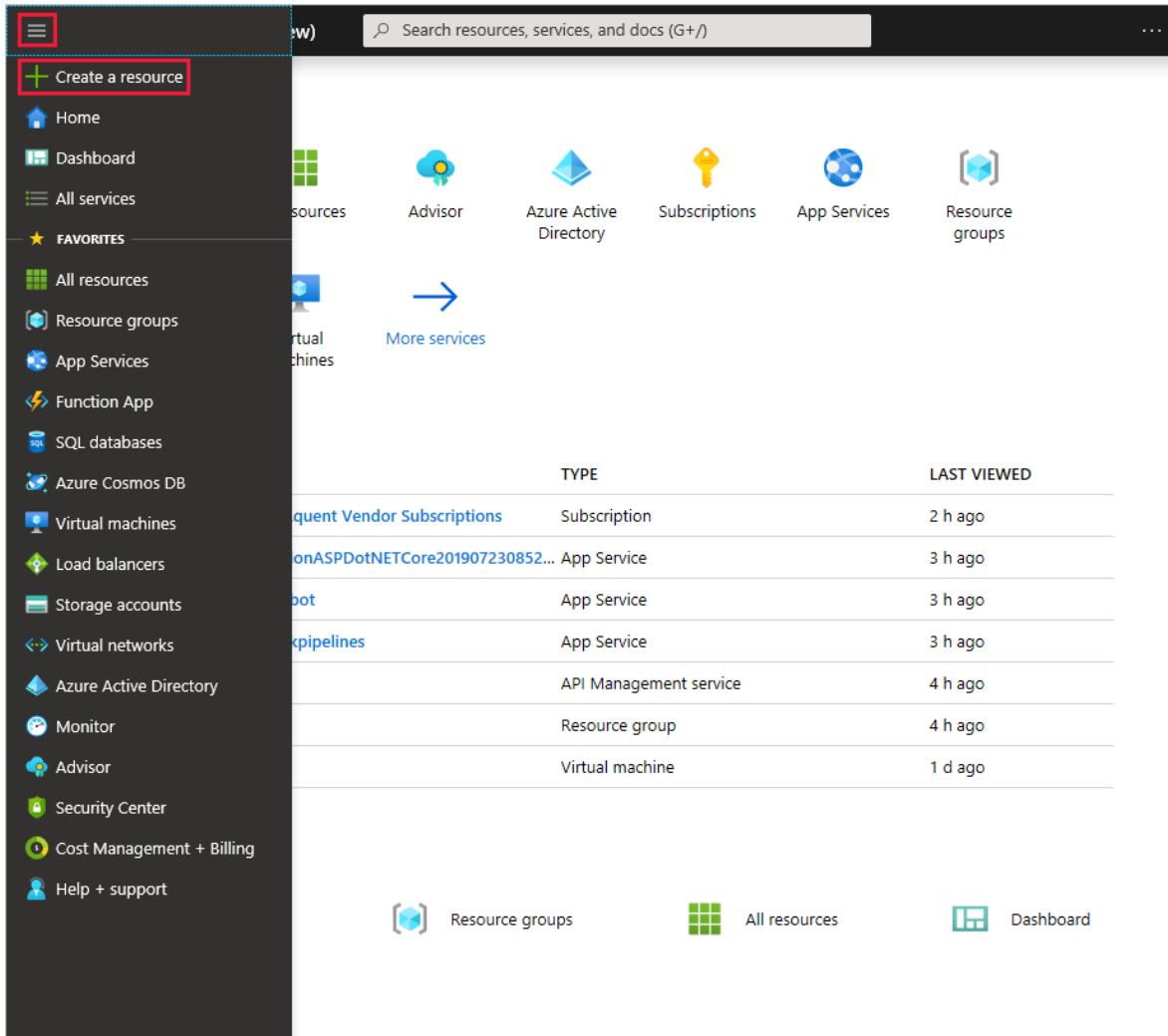
3. Create the local network gateway

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.

Example values

- **Name:** Site1
- **Resource Group:** TestRG1
- **Location:** East US

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



The screenshot shows the Azure portal interface. On the left, a dark sidebar contains a navigation menu with various options like Home, Dashboard, All services, Favorites, and Help + support. A red box highlights the 'Create a resource' button at the top of this menu. At the very top of the page is a search bar labeled 'Search resources, services, and docs (G/?)'. The main content area features several service icons: 'sources' (grid), 'Advisor' (cloud with gear), 'Azure Active Directory' (blue diamond), 'Subscriptions' (key), 'App Services' (globe), and 'Resource groups' (cube). Below these is a large blue arrow pointing right towards a 'More services' link. To the right of the arrow is a table titled 'LAST VIEWED' showing recent activity:

	TYPE	LAST VIEWED
quent Vendor Subscriptions	Subscription	2 h ago
onASPDotNETCore201907230852...	App Service	3 h ago
bot	App Service	3 h ago
ckpipelines	App Service	3 h ago
	API Management service	4 h ago
	Resource group	4 h ago
	Virtual machine	1 d ago

At the bottom of the main content area are three navigation links: 'Resource groups' (cube icon), 'All resources' (grid icon), and 'Dashboard' (dash icon).

2. In the **Search the marketplace** field, 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

This form allows you to create a local network gateway object. A local network gateway is used to connect your on-premises network to Azure.

Name: Site1

IP address: 128.8.8.8

Address space: 10.101.0.0/24

Configure BGP settings:

Subscription:

Resource group: TestRG1

Create new: [Create new](#)

Location: (US) East US

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. If you don't have the IP address right now, you can use the values shown in the example, but you'll need to go back and replace your placeholder IP address with the public IP address of your VPN device. Otherwise, Azure will not be able to connect.
- **Address Space** refers to the address ranges for the network that this local network represents. You can add multiple address space ranges. Make sure that the ranges you specify here do not overlap with ranges of other networks that you want to connect to. Azure will route the address range that you specify to the on-premises VPN device IP address. *Use your own values here if you want to connect to your on-premises site, not the values shown in the example.*
- **Configure BGP settings:** Use only when configuring BGP. Otherwise, don't select this.
- **Subscription:** Verify that the correct subscription is showing.

- **Resource Group:** Select the resource group that you want to use. You can either create a new resource group, or select one that you have already created.
 - **Location:** The location is the same as **Region** in other settings. Select the location that this object will be created in. You may want to select the same location that your VNet resides in, but you are not required to do so.
4. When you have finished specifying the values, click the **Create** button at the bottom of the page to create the local network gateway.

4. 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](#).

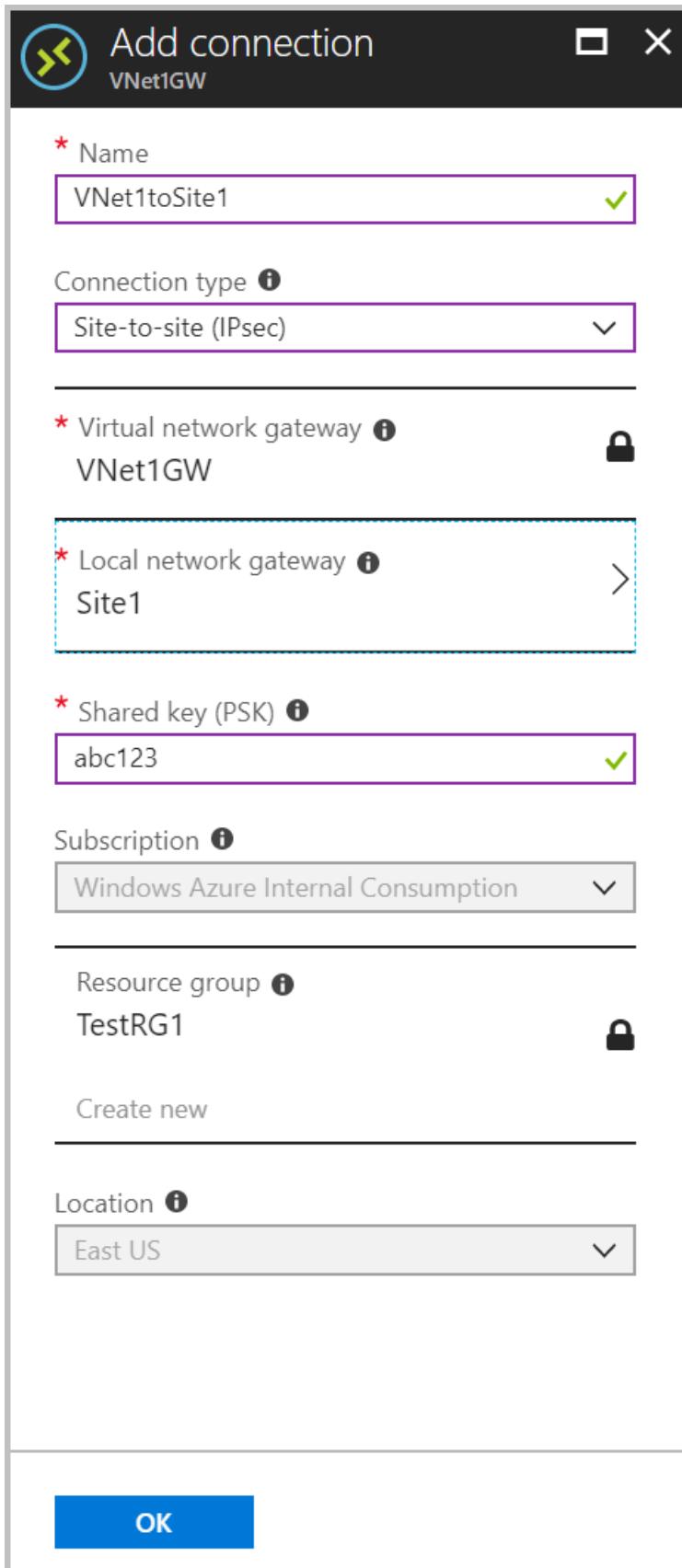
See the following links for additional configuration information:

- 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](#).

5. Create the VPN connection

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

1. Open the page for your virtual network gateway. There are multiple ways to navigate. You can navigate to the gateway by going to **Name of your VNet** -> **Overview** -> **Connected devices** -> **Name of your gateway**.
2. On the page for the gateway, 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*.

6. Verify the VPN connection

In the Azure portal, you can view the connection status of a Resource Manager VPN Gateway by navigating to the connection. The following steps show one way to navigate to your connection and verify.

1. In the [Azure portal](#) menu, select **All resources** or search for and select **All resources** from any page.
2. Select to your virtual network gateway.
3. On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
4. Click the name of the connection that you want to verify to open **Essentials**. In Essentials, you can view more information about your connection. The **Status** is 'Succeeded' and 'Connected' when you have made a successful connection.

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-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}
```

2. Verify that you are connected to your VNet using the 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

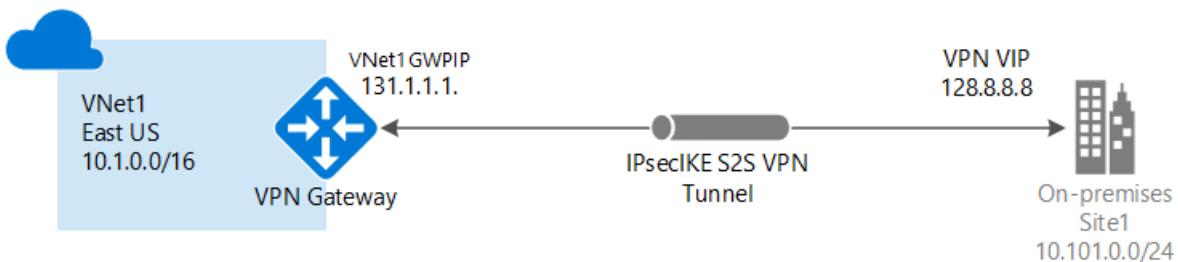
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).
- For information about forced tunneling, see [About forced tunneling](#).
- For information about Highly Available Active-Active connections, see [Highly Available cross-premises and VNet-to-VNet connectivity](#).
- 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

1/16/2020 • 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.
- 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.

Azure PowerShell

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select **Try It** in the upper-right corner of a code block. This **doesn't** automatically copy text to Cloud Shell.

[Azure CLI](#) [Copy](#) [Try It](#)

Open shell.azure.com in your browser.

[Launch Cloud Shell](#)

Select the **Cloud Shell** button on the menu in the upper-right corner of the [Azure portal](#).



You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

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

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-AzResourceGroup -Name TestRG1 -Location 'East US'
```

Create your virtual network.

1. Set the variables.

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

2. Create the VNet.

```
New-AzVirtualNetwork -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-AzVirtualNetwork -ResourceGroupName TestRG1 -Name VNet1
```

2. Create the gateway subnet.

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

3. Set the configuration.

```
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

2. Create the local network gateway

The local network gateway (LNG) typically refers to your on-premises location. It is not the same as a virtual network gateway. 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.
- The *AddressPrefix* is your on-premises address space.

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

```
New-AzLocalNetworkGateway -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-AzLocalNetworkGateway -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 will change 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-AzPublicIpAddress -Name VNet1GWPPIP -ResourceGroupName TestRG1 -Location 'East US' -  
AllocationMethod Dynamic
```

4. Create the gateway IP addressing configuration

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

```
$vnet = Get-AzVirtualNetwork -Name VNet1 -ResourceGroupName TestRG1
$subnet = Get-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gwipconfig = New-AzVirtualNetworkGatewayIpConfig -Name gwipconfig1 -SubnetId $subnet.Id -PublicIpAddressId
$gwpip.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-AzVirtualNetworkGateway -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 items:

- 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. In this example, VNet1GWPIP is the name of the public IP address resource that you created in an earlier step.

```
Get-AzPublicIpAddress -Name VNet1GWPIP -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 [VPN device configuration overview](#).
- 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 **IPsec**.

- Set the variables.

```
$gateway1 = Get-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
$local = Get-AzLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1
```

- Create the connection.

```
New-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGatewayConnection' 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-AzVirtualNetworkGatewayConnection -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.

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-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}
```

2. Verify that you are connected to your VNet using the 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](#).

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. When using these examples, modify the values to match your environment.

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

To add additional address prefixes:

1. Set the variable for the LocalNetworkGateway.

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

2. Modify the prefixes.

```
Set-AzLocalNetworkGateway -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.

1. Set the variable for the LocalNetworkGateway.

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

2. Set the gateway with the updated prefixes.

```
Set-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. Set the local network gateway with the modified address prefixes.

Set the variable for the LocalNetworkGateway.

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

Modify the prefixes.

```
Set-AzLocalNetworkGateway -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-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

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

```
New-AzVirtualNetworkGatewayConnection -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-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection' cmdlet.

```
Remove-AzVirtualNetworkGatewayConnection -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-AzLocalNetworkGateway -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-AzVirtualNetworkGateway' cmdlet.

Set the variables.

```
$local = Get-AzLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1  
$vnetgw = Get-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection.

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

To delete a gateway connection

If you don't know the name of your connection, you can find it by using the 'Get-AzVirtualNetworkGatewayConnection' cmdlet.

```
Remove-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 `  
-ResourceGroupName TestRG1
```

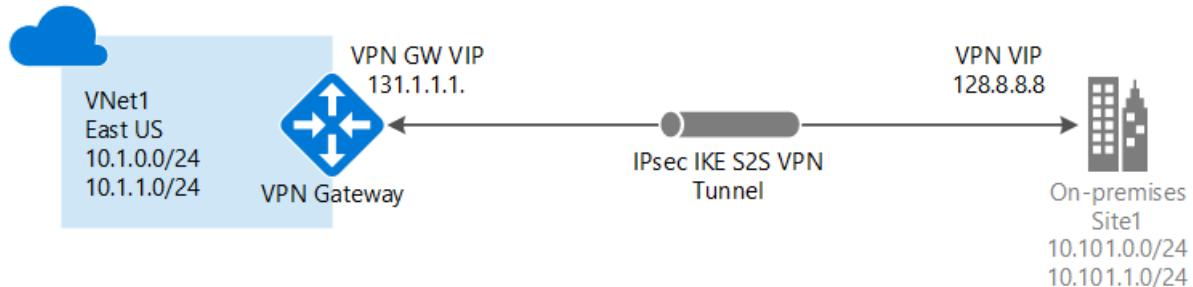
Next steps

- 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

1/9/2020 • 16 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

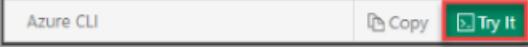
Verify that you have met the following criteria before beginning configuration:

- 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.
- 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.
- You can use Azure Cloud Shell to run your CLI commands (instructions below). However, if you prefer to run your commands locally, verify that you have installed latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install the Azure CLI](#) and [Get Started with Azure CLI](#).

Use Azure Cloud Shell

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

To start Azure Cloud Shell:

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

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

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

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

If you choose to run CLI locally, connect to your subscription. If you are using Azure Cloud Shell in the browser, you don't need to connect to your subscription. You will connect automatically in Azure Cloud Shell. However, you may want to verify that you are using the correct subscription after you connect.

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](#).

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

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.
- 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 [VPN device configuration overview](#).
- 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.

- 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-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}

```

2. Verify that you are connected to your VNet using the 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](#).

Add a Site-to-Site connection to a VNet with an existing VPN gateway connection

1/9/2020 • 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 new ExpressRoute/Site-to-Site coexisting connection configurations. However, if you are merely adding a new VPN connection to an already existing coexist configuration, you can use these steps. 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	Tutorial	Supported
Classic	Not Supported	Tutorial

Before you begin

Verify the following items:

- You are not configuring a new coexisting ExpressRoute and VPN Gateway configuration.
- 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. At the top left is a blue circular icon with a yellow 'X' and a checkmark. To its right is the text 'RMGateway - Connections' and 'Virtual network gateway'. Below this is a search bar with the placeholder 'Search (Ctrl+ /)'. A vertical sidebar on the left contains several navigation links: 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'SETTINGS' (which is expanded), 'Connections' (which is highlighted with a light blue background), 'Point-to-site configuration', 'Properties', 'Locks', and 'Automation script'. Under 'SUPPORT + TROUBLESHOOTING' is a link 'New support request'.

4. On the **Connections** page, click **+Add**.

The screenshot shows the 'Add' page for creating a new connection. At the top left is a large white button with a black plus sign and the word 'Add'. Below it is a search bar with the placeholder 'Search connections'. A table follows, with columns labeled 'NAME', 'STATUS', 'CONNECTION TYPE', and 'PEER'. The table has one row showing 'Site1' in the NAME column, 'Succeeded' in the STATUS column, 'Site-to-site (IPsec)' in the CONNECTION TYPE column, and 'Site1' in the PEER column. There is also a '...' button at the bottom right of the table.

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').

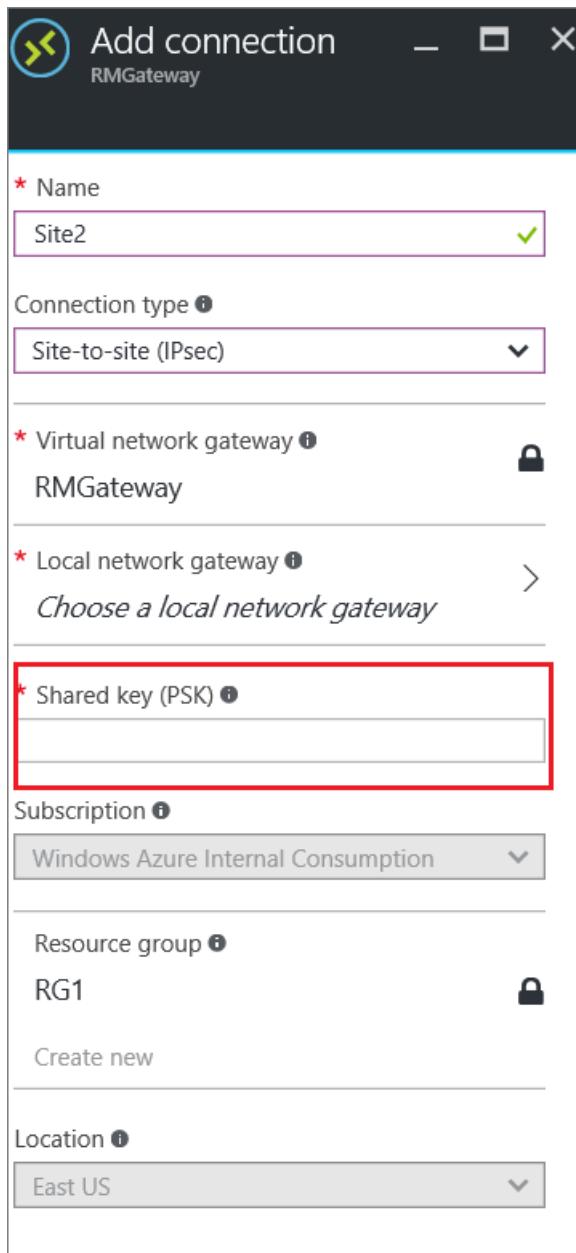
3. On the **Create local network gateway** page, fill out the following fields:

- **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-AzVirtualNetworkGatewayConnection' 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-AzVirtualNetworkGatewayConnection -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

2/27/2020 • 7 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.

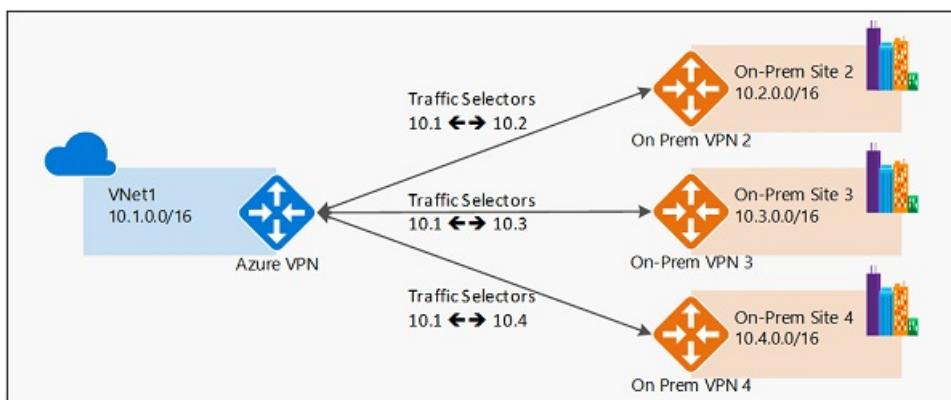
About policy-based and route-based VPN gateways

Policy-based vs. route-based VPN devices differ in how the IPsec traffic selectors are set on a connection:

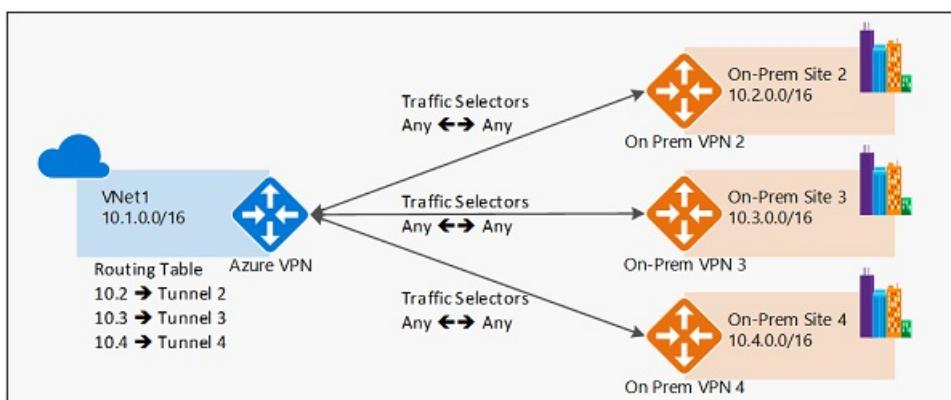
- **Policy-based** VPN devices use the combinations of prefixes from both networks to define how traffic is encrypted/decrypted through IPsec tunnels. It is typically built on firewall devices that perform packet filtering. IPsec tunnel encryption and decryption are added to the packet filtering and 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:

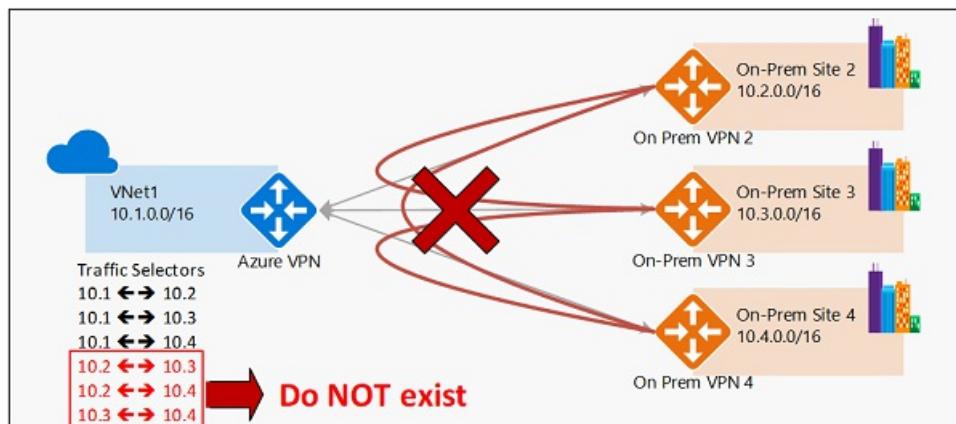
	POLICYBASED VPN GATEWAY	ROUTEBASED VPN GATEWAY	ROUTEBASED VPN GATEWAY
Azure Gateway SKU	Basic	Basic	Standard, HighPerformance, VpnGw1, VpnGw2, VpnGw3
IKE version	IKEv1	IKEv2	IKEv1 and IKEv2
Max. S2S connections	1	10	Standard: 10 Other SKUs: 30

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).

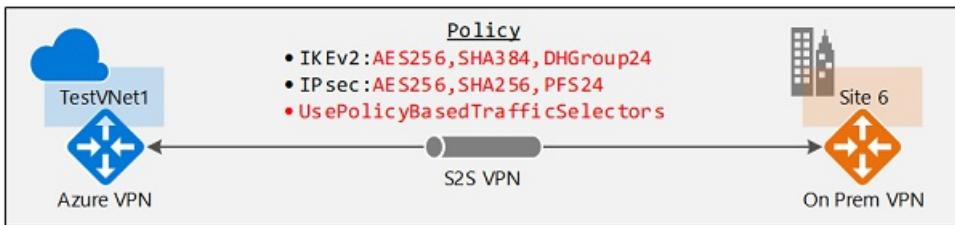
The following diagram shows why transit routing via Azure VPN gateway doesn't work with the policy-based option:



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.

Workflow

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.

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](#).
- This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

Enable policy-based traffic selectors

This section shows you how to enable policy-based traffic selectors on a connection. Make sure you have completed [Part 3 of the Configure IPsec/IKE policy article](#). The steps in this article use the same parameters.

Step 1 - Create the virtual network, VPN gateway, and local network gateway

Connect to your subscription and declare your variables

1. If you are running PowerShell locally on your computer, sign in using the `Connect-AzAccount` cmdlet. Or, instead, use Azure Cloud Shell in your browser.

2. Declare your variables. For this exercise, we use the following variables:

```
$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"
```

Create the virtual network, VPN gateway, and local network gateway

1. Create a resource group.

```
New-AzResourceGroup -Name $RG1 -Location $Location1
```

2. Use the following example to create the virtual network TestVNet1 with three subnets, and the VPN gateway. If you want to substitute 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-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1  
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1  
$gwsu1 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1  
  
New-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix  
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsu1  
  
$gw1pip1 = New-AzPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -  
AllocationMethod Dynamic  
$vnet1   = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1  
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1  
$gw1ipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress  
$gw1pip1  
  
New-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -  
IpConfigurations $gw1ipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance  
  
New-AzLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1 -Location $Location1 -  
GatewayIpAddress $LNGIP6 -AddressPrefix $LNGPrefix61,$LNGPrefix62
```

Step 2 - Create an 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, PFS None, SA Lifetime 14400 seconds & 10240000KB

```
$ipsecpolicy6 = New-AzIpsecPolicy -IkeEncryption AES256 -IkeIntegrity SHA384 -DhGroup DHGroup24 -  
IpsecEncryption AES256 -IpsecIntegrity SHA256 -PfsGroup None -SALifeTimeSeconds 14400 -  
SADataSizeKilobytes 102400000
```

2. Create the S2S VPN connection with policy-based traffic selectors and IPsec/IKE policy 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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$lng6 = Get-AzLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1  
  
New-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng6 -Location $Location1 -ConnectionType  
IPsec -UsePolicyBasedTrafficSelectors $True -IpsecPolicies $ipsecpolicy6 -SharedKey 'AzureA1b2C3'
```

3. 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.

To update policy-based traffic selectors

This section shows you how to update the policy-based traffic selectors option for an existing S2S VPN connection.

1. Get the connection resource.

```
$RG1          = "TestPolicyRG1"  
$Connection16 = "VNet1toSite6"  
$connection6  = Get-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1
```

2. View 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. Once you obtain the connection resource, you can enable or disable the policy-based traffic selectors on a connection.

- To Enable

The following example enables the policy-based traffic selectors option, but leaves the IPsec/IKE

policy unchanged:

```
$RG1          = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6  = Get-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName
$RG1

Set-AzVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -
UsePolicyBasedTrafficSelectors $True
```

- To Disable

The following example disables the policy-based traffic selectors option, but leaves the IPsec/IKE policy unchanged:

```
$RG1          = "TestPolicyRG1"
$Connection16 = "VNet1toSite6"
$connection6  = Get-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName
$RG1

Set-AzVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -
UsePolicyBasedTrafficSelectors $False
```

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 ExpressRoute and Site-to-Site coexisting connections using PowerShell

1/30/2020 • 11 minutes to read • [Edit Online](#)

This article helps you configure ExpressRoute and Site-to-Site VPN connections that coexist. Having the ability to configure Site-to-Site VPN and ExpressRoute has several advantages. You can configure Site-to-Site VPN as a secure failover path for ExpressRoute, or use Site-to-Site VPNs to connect to sites that are not connected through ExpressRoute. We will cover the steps to configure both scenarios in this article. This article applies to the Resource Manager deployment model.

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. You can configure either gateway first. Typically, you will incur no downtime when adding a new gateway or gateway connection.

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](#). You also can use a route-based VPN gateway with a VPN connection configured for 'policy-based traffic selectors' as described in [Connect to multiple policy-based VPN devices](#).
- **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.
- **VPN Gateway defaults to ASN 65515 if not specified.** Azure VPN Gateway supports the BGP routing protocol. You can specify ASN (AS Number) for a virtual network by adding the -Asn switch. If you don't specify this parameter, the default AS number is 65515. You can use any ASN for the configuration, but if you select something other than 65515, you must reset the gateway for the setting to take effect.
- **The gateway subnet must be /27 or a shorter prefix,** (such as /26, /25), or you will receive an error message when you add the ExpressRoute virtual network gateway.

Configuration designs

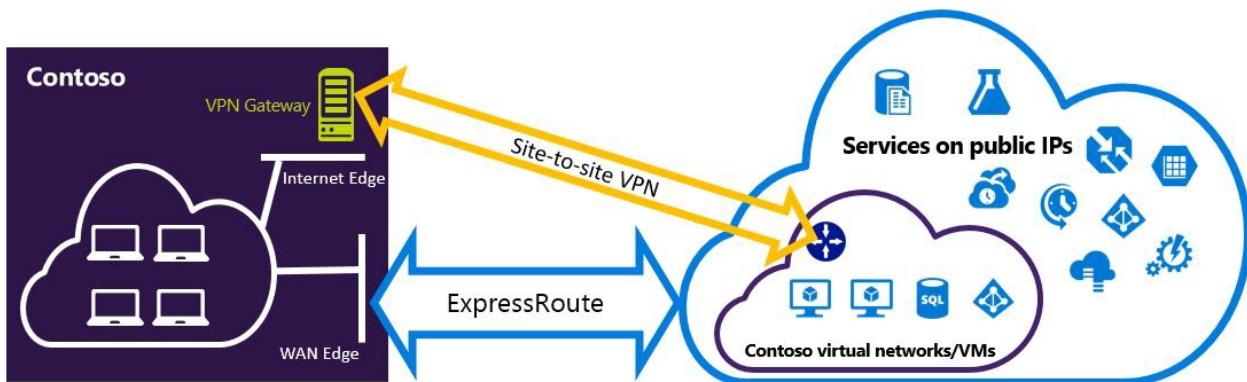
Configure a Site-to-Site VPN as a failover path for ExpressRoute

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 the ExpressRoute.

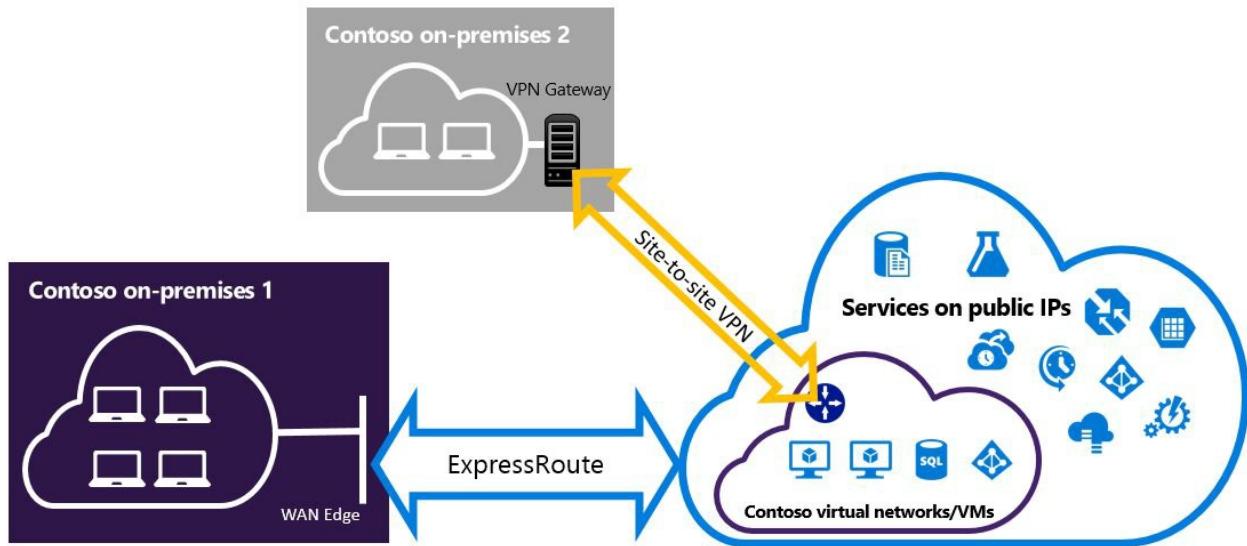
NOTE

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.

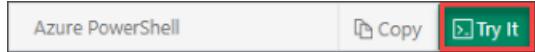
If you delete and recreate your gateway, you will have downtime for your cross-premises connections. However, your VMs and services will still be able to communicate out through the load balancer while you configure your gateway if they are configured to do so.

Before you begin

The steps and examples in this article use Azure PowerShell Az modules. To install the Az modules locally on your computer, see [Install Azure PowerShell](#). To learn more about the new Az module, see [Introducing the new Azure PowerShell Az module](#). PowerShell cmdlets are updated frequently. If you are not running the latest version, the values specified in the instructions may fail. To find the installed versions of PowerShell on your system, use the `Get-Module -ListAvailable Az` cmdlet.

You can use Azure Cloud Shell to run most PowerShell cmdlets and CLI commands, instead of installing Azure PowerShell or CLI locally. Azure Cloud Shell is a free interactive shell that has common Azure tools preinstalled and is configured to use with your account. To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, press **Enter** to run code.

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 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. 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.

1. Sign in and select your subscription.

If you are using the Azure Cloud Shell, you sign in to your Azure account automatically after clicking 'Try it'.

To sign in locally, open your PowerShell console with elevated privileges and run the cmdlet to connect.

```
Connect-AzAccount
```

If you have more than one subscription, get a list of your Azure subscriptions.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Name of subscription"
```

2. Set variables.

```
$location = "Central US"  
$resgrp = New-AzResourceGroup -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-AzVirtualNetwork -Name "CoexVnet" -ResourceGroupName $resgrp.ResourceGroupName -Location  
$location -AddressPrefix "10.200.0.0/16"
```

Add subnets.

```
Add-AzVirtualNetworkSubnetConfig -Name "App" -VirtualNetwork $vnet -AddressPrefix "10.200.1.0/24"  
Add-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix  
"10.200.255.0/24"
```

Save the VNet configuration.

```
$vnet = Set-AzVirtualNetwork -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-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzPublicIpAddress -Name "VPNGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -AllocationMethod Dynamic
$gwConfig = New-AzVirtualNetworkGatewayIpConfig -Name "VPNGatewayIpConfig" -SubnetId $gwSubnet.Id -
PublicIpAddressId $gwIP.Id
New-AzVirtualNetworkGateway -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-AzVirtualNetworkGateway -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-AzLocalNetworkGateway -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-AzLocalNetworkGateway -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-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
New-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzPublicIpAddress -Name "ERGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -Location $location -AllocationMethod Dynamic
$gwConfig = New-AzVirtualNetworkGatewayIpConfig -Name "ERGatewayIpConfig" -SubnetId $gwSubnet.Id - PublicIpAddressId $gwIP.Id
$gw = New-AzVirtualNetworkGateway -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-AzExpressRouteCircuit -Name "YourCircuit" -ResourceGroupName "YourCircuitResourceGroup"
New-AzVirtualNetworkGatewayConnection -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.

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.

1. Delete the existing ExpressRoute or Site-to-Site VPN gateway.

```
Remove-AzVirtualNetworkGateway -Name <yourgatewayname> -ResourceGroupName <yourresourcegroup>
```

2. Delete Gateway Subnet.

```
$vnet = Get-AzVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup> Remove-AzVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet
```

3. 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-AzVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Add-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix
"10.200.255.0/24"
```

Save the VNet configuration.

```
$vnet = Set-AzVirtualNetwork -VirtualNetwork $vnet
```

4. At this point, you have a virtual network with no gateways. To create new gateways and set up the connections, use the following examples:

Set the variables.

```
$gwSubnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzPublicIpAddress -Name "ERGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -Location
$location -AllocationMethod Dynamic
$gwConfig = New-AzVirtualNetworkGatewayIpConfig -Name "ERGatewayIpConfig" -SubnetId $gwSubnet.Id -
PublicIpAddressId $gwIP.Id
```

Create the gateway.

```
$gw = New-AzVirtualNetworkGateway -Name <yourgatewayname> -ResourceGroupName <yourresourcegroup> -
Location <yourlocation> -IpConfigurations $gwConfig -GatewayType "ExpressRoute" -GatewaySku Standard
```

Create the connection.

```
$ckt = Get-AzExpressRouteCircuit -Name "YourCircuit" -ResourceGroupName "YourCircuitResourceGroup"
New-AzVirtualNetworkGatewayConnection -Name "ERConnection" -ResourceGroupName $resgrp.ResourceGroupName
-Location $location -VirtualNetworkGateway1 $gw -PeerId $ckt.Id -ConnectionType ExpressRoute
```

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 coexistence setup. To upload the VPN root certificate, you must either install PowerShell locally to your computer, or use the Azure portal.

1. Add VPN Client address pool.

```
$azureVpn = Get-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
Set-AzVirtualNetworkGatewayVpnClientConfig -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 and that you are running PowerShell locally. You can also upload the certificate using the Azure portal.

```
$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}
$p2sCertData = [System.Convert]::ToString($p2sCertToUpload.RawData)
Add-AzVpnClientRootCertificate -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](#).

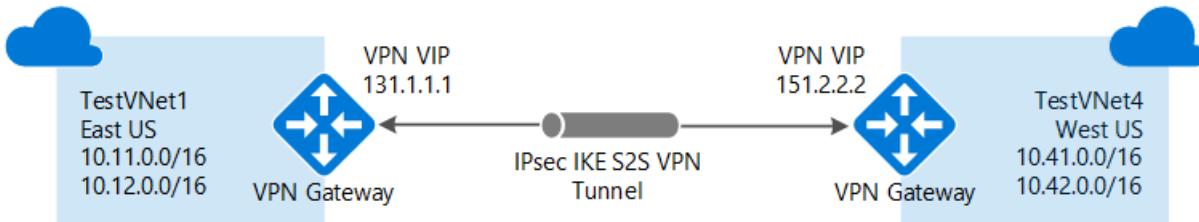
Next steps

For more information about ExpressRoute, see the [ExpressRoute FAQ](#).

Configure a VNet-to-VNet VPN gateway connection by using the Azure portal

2/27/2020 • 16 minutes to read • [Edit Online](#)

This article helps you connect virtual networks (VNets) by using the VNet-to-VNet connection type. Virtual networks can be in different regions and from different subscriptions. When you connect VNets from different subscriptions, the subscriptions don't need to be associated with the same Active Directory tenant.



The steps in this article apply to the Azure Resource Manager deployment model and use the Azure portal. You can create this configuration with a different deployment tool or model by using options that are described in the following articles:

About connecting VNets

The following sections describe the different ways to connect virtual networks.

VNet-to-VNet

Configuring a VNet-to-VNet connection is a simple way to connect VNets. When you connect a virtual network to another virtual network with a VNet-to-VNet connection type (VNet2VNet), it's similar to creating a Site-to-Site IPsec connection to an on-premises location. Both connection types use a VPN gateway to provide a secure tunnel with IPsec/IKE and function the same way when communicating. However, they differ in the way the local network gateway is configured.

When you create a VNet-to-VNet connection, the local network gateway address space is automatically created and populated. If you update the address space for one VNet, the other VNet automatically routes to the updated address space. It's typically faster and easier to create a VNet-to-VNet connection than a Site-to-Site connection.

Site-to-Site (IPsec)

If you're working with a complicated network configuration, you may prefer to connect your VNets by using a [Site-to-Site connection](#) instead. When you follow 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. These steps allow you to specify additional address spaces for the local network gateway to route traffic. If the address space for a VNet changes, you must manually update the corresponding local network gateway.

VNet peering

You can also connect your VNets by using VNet peering. VNet peering doesn't 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 by using a VNet-to-VNet connection for the following reasons:

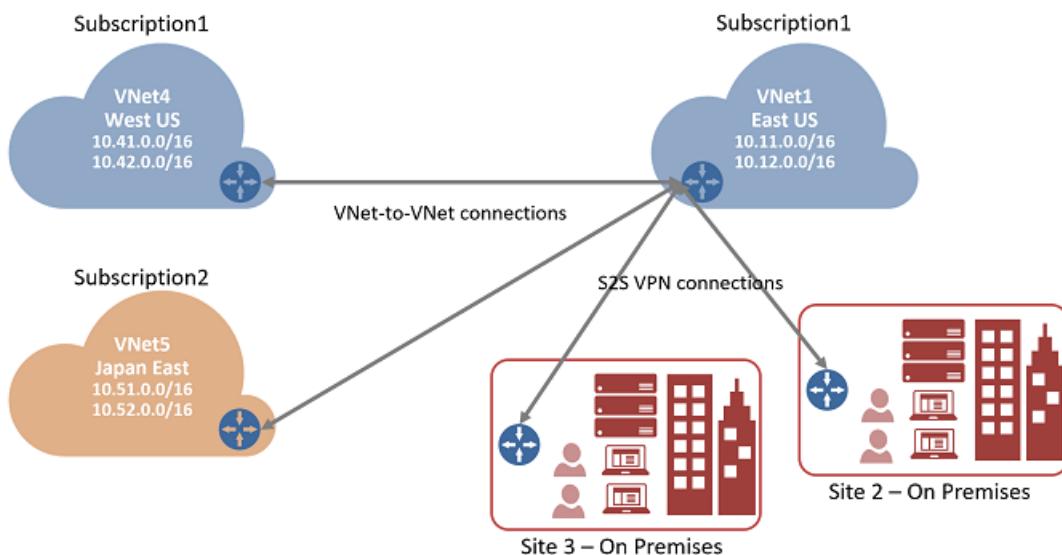
Cross region geo-redundancy and geo-presence

- You can set up your own geo-replication or synchronization with secure connectivity without going over internet-facing endpoints.
- With Azure Traffic Manager and Azure Load Balancer, you can set up highly available workload with geo-redundancy across multiple Azure regions. For example, you can set up SQL Server Always On availability groups across multiple Azure regions.

Regional multi-tier applications with isolation or administrative boundaries

- Within the same region, you can set up multi-tier applications with multiple virtual networks that are connected together because of isolation or administrative requirements.

VNet-to-VNet communication can be combined with multi-site configurations. These configurations lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity, as shown in the following diagram:



This article shows you how to connect VNets by using the VNet-to-VNet connection type. When you follow these steps as an exercise, you can use the following 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. Use [PowerShell](#) or [CLI](#) instead. For more information about VNet-to-VNet connections, see [VNet-to-VNet FAQ](#).

Example settings

Values for VNet1:

- **Virtual network settings**

- **Name:** VNet1
- **Address space:** 10.11.0.0/16
- **Subscription:** Select the subscription you want to use.
- **Resource group:** TestRG1
- **Location:** East US
- **Subnet**
 - **Name:** FrontEnd
 - **Address range:** 10.11.0.0/24
- **Gateway subnet:**
 - **Name:** GatewaySubnet is autofilled
 - **Address range:** 10.11.255.0/27

- **Virtual network gateway settings**

- **Name:** VNet1GW
- **Gateway type:** Select **VPN**.
- **VPN type:** Select **Route-based**.
- **SKU:** Select the gateway SKU you want to use.
- **Public IP address name:** VNet1GWpip
- **Connection**
 - **Name:** VNet1toVNet4
 - **Shared key:** You can create the shared key yourself. When you create the connection between the VNets, the values must match. For this exercise, use abc123.

Values for VNet4:

- **Virtual network settings**

- **Name:** VNet4
- **Address space:** 10.41.0.0/16
- **Subscription:** Select the subscription you want to use.
- **Resource group:** TestRG4
- **Location:** West US
- **Subnet**
 - **Name:** FrontEnd
 - **Address range:** 10.41.0.0/24
- **GatewaySubnet**
 - **Name:** *GatewaySubnet* is autofilled
 - **Address range:** 10.41.255.0/27

- **Virtual network gateway settings**

- **Name:** VNet4GW
- **Gateway type:** Select **VPN**.
- **VPN type:** Select **Route-based**.
- **SKU:** Select the gateway SKU you want to use.
- **Public IP address name:** VNet4GWpip
- **Connection**
 - **Name:** VNet4toVNet1
 - **Shared key:** You can create the shared key yourself. When you create the connection between the VNets, the values must match. For this exercise, use abc123.

Create and configure VNet1

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. Your connection won't work properly if you have overlapping subnets.

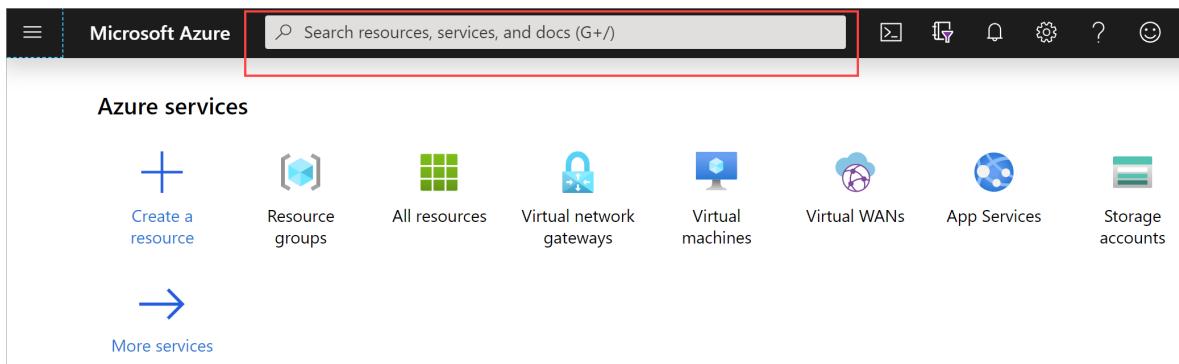
To create a virtual network

You can create a VNet with the Resource Manager deployment model and the Azure portal by following these steps. For more information about virtual networks, see [Virtual Network overview](#).

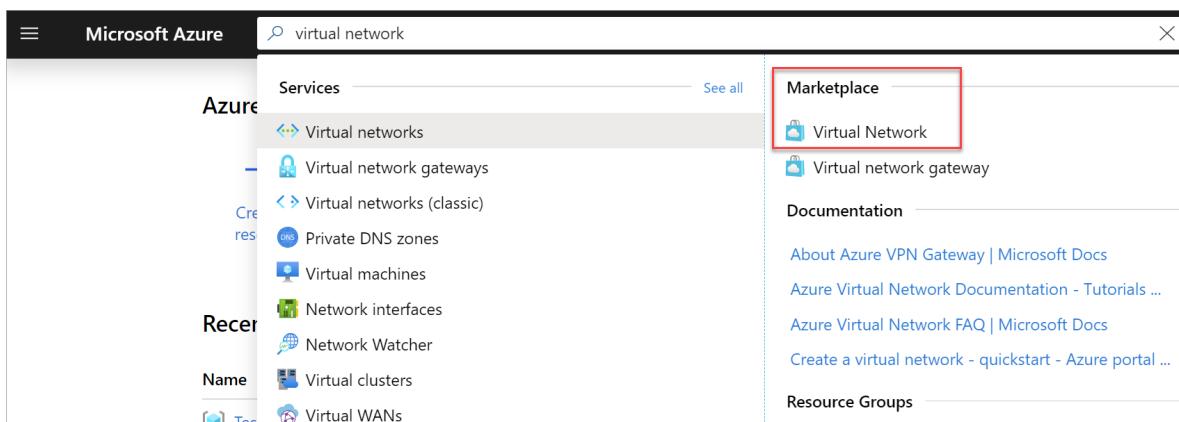
NOTE

For the VNet to connect to an on-premises location, coordinate with your on-premises network administrator to carve out an IP address range that you can use specifically for this virtual network. If a duplicate address range exists on both sides of the VPN connection, traffic will route in an unexpected way. Additionally, if you want to connect this VNet to another VNet, the address space cannot overlap with other VNet. Plan your network configuration accordingly.

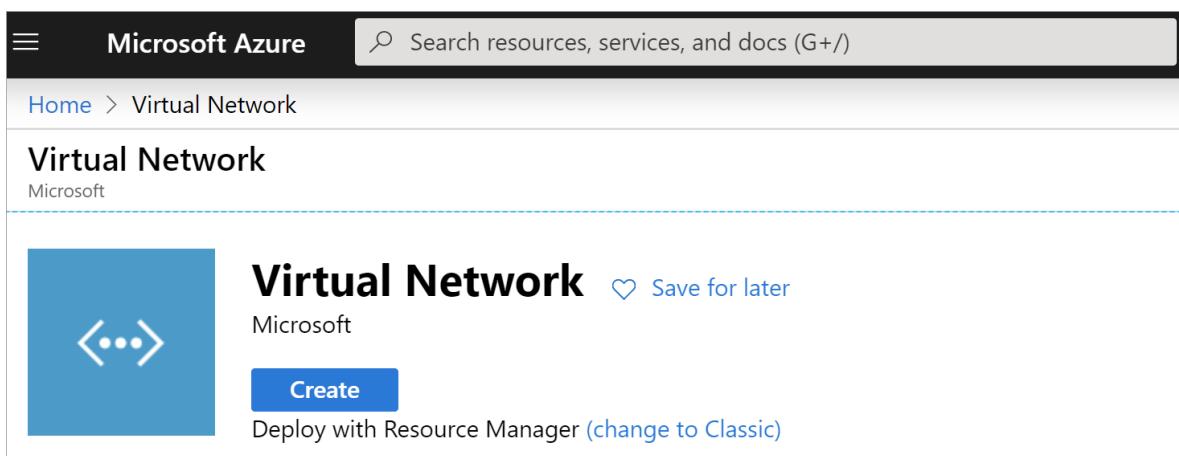
1. Sign in to the [Azure portal](#).
2. In **Search resources, service, and docs (G+/-)**, type *virtual network*.



3. Select **Virtual Network** from the **Marketplace** results.



4. On the **Virtual Network** page, click **Create**.



5. Once you click create, the **Create virtual network** page opens.

Create virtual network X

Name *

Address space * ⓘ

 Add IPv6 address space ⓘ

Subscription *

Resource group *
 ⓘ
[Create new](#)

Location *

Subnet

Name *

Address range * ⓘ
 ⓘ
✓

DDoS protection ⓘ
 Basic Standard

Service endpoints ⓘ
Disabled Enabled

Firewall ⓘ
Disabled Enabled

Create Automation options

6. 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 you enter in the field are validated. Some values are autofilled, which you can replace with your own values:

- **Name:** Enter the name for your virtual network.
- **Address space:** Enter the address space. If you have multiple address spaces to add, enter your first address space here. You can add additional address spaces later, after you create the VNet. If your configuration requires IPv6 address space, check the checkbox to enter that information.
- **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 entering a name for your new resource group. If you're creating a new group, name the resource group according to your planned configuration values. For more information about resource groups, see [Azure Resource Manager overview](#).
- **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will live.
- **Subnet:** Add the subnet **Name** and subnet **Address range**. You can add additional subnets later, after you create the VNet.
- **DDos protection:** Select **Basic**, unless you want to use the Standard service.
- **Service endpoints:** You can leave this setting as **Disabled**, unless your configuration specifies this setting.
- **Firewall:** You can leave this setting as **Disabled**, unless your configuration specifies this setting.

7. Click **Create** to begin the virtual network deployment.

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 ranges to your address space, in the **Settings** section of your virtual network page, select **Address space**. The **Address space** page appears.
2. Add the additional address range, and then select **Save** at the top of the page.

The screenshot shows the Azure portal interface for managing a virtual network. At the top, the navigation bar shows 'Home > Virtual networks > TestVNet1 - Address space'. Below the navigation, there's a breadcrumb trail: '<..> TestVNet1 - Address space' and 'Virtual network'. On the left, a sidebar lists several options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problem..., Settings, and Address space. The 'Address space' option is currently selected. The main content area displays two address ranges: '10.11.0.0/16' and '10.12.0.0/16'. Below these ranges, a button labeled 'Add additional address range' is visible, also highlighted with a red box. At the top right of the main content area, there are 'Save' and 'Discard' buttons, with 'Save' being the one highlighted by a red box.

To create additional subnets

1. To create additional subnets, in the **Settings** section of your virtual network page, select **Subnets**. The **Subnets** page appears.
2. Select **+ Subnet** to open the **Add subnet** page.

 **TestVNet1 - Subnets**
Virtual network

Search (Ctrl+/) << + Subnet + Gateway subnet Refresh

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

Settings

Address space Connected devices Subnets

Search subnets

Name	Address range
FrontEnd	10.11.0.0/24

3. Enter the **Name** of your new subnet and specify the **Address range**. To save your changes, select **OK** at the bottom of the page.

Add subnet

TestVNet1



Name *

Address range (CIDR block) * ⓘ

 ✓

NAT gateway ⓘ

None

Add IPv6 address space

Network security group

None

Route table

None

Service endpoints

Services ⓘ

0 selected

Subnet delegation

Delegate subnet to a service ⓘ

None

OK

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're creating this configuration as an exercise, see the [Example settings](#).

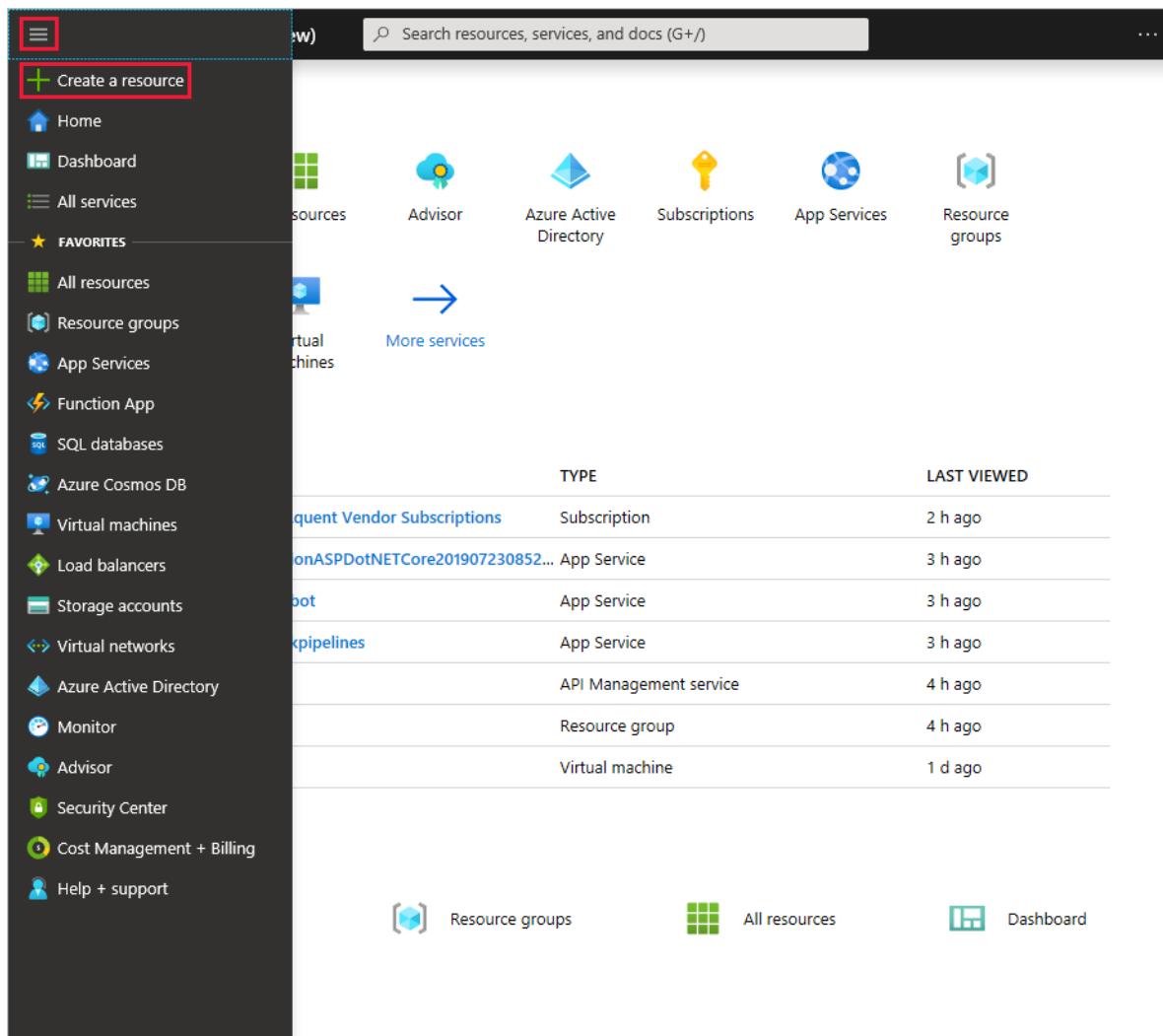
The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

To create a virtual network gateway

1. From the [Azure portal](#) menu, select **Create a resource**.



The screenshot shows the Azure portal interface. On the left, a dark sidebar lists various services with icons: Home, Dashboard, All services, FAVORITES (with All resources, Resource groups, App Services, Function App, SQL databases, Azure Cosmos DB, Virtual machines, Load balancers, Storage accounts, Virtual networks, Azure Active Directory, Monitor, Advisor, Security Center, Cost Management + Billing, Help + support), and More services (indicated by an arrow pointing to a list of services). The 'Create a resource' button is highlighted with a red box. The main content area shows a grid of service icons: Advisor, Azure Active Directory, Subscriptions, App Services, and Resource groups. Below this is a table titled 'LAST VIEWED' showing recent resources: 'quent Vendor Subscriptions' (Subscription, 2 h ago), 'onASPDotNETCore201907230852...' (App Service, 3 h ago), 'bot' (App Service, 3 h ago), 'pipelines' (API Management service, 4 h ago), 'Resource group' (Resource group, 4 h ago), and 'Virtual machine' (Virtual machine, 1 d ago). At the bottom are links for 'Resource groups', 'All resources', and 'Dashboard'.

2. In the **Search the Marketplace** field, type 'Virtual Network Gateway'. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create**. This opens the **Create virtual network gateway** page.

Create virtual network gateway

Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)

Project details

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

Subscription *

Resource group ⓘ

TestRG1 (derived from virtual network's resource group)

Instance details

Name *

 VNet1GW ✓

Region *

 (US) East US ✓

Gateway type * ⓘ

VPN ExpressRoute

VPN type * ⓘ

Route-based Policy-based

SKU * ⓘ

 VpnGw1 ✓

Generation ⓘ

 Generation1 ✓

Virtual network * ⓘ

 VNet1 ✓

[Create virtual network](#)

i Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range * ⓘ

 10.11.255.0/27 ✓

10.11.255.0 - 10.11.255.31 (32 addresses)

Public IP address

Public IP address * ⓘ

Create new Use existing

Public IP address name *

 VNet1GWpip ✓

Public IP address SKU

Basic

Assignment

Dynamic Static

Enable active-active mode * ⓘ

Enabled Disabled

Configure BGP ASN * ⓘ

Enabled Disabled

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

Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.

- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).

Virtual network: Choose the virtual network to which you want to add this gateway.

Gateway subnet address range: This field only appears if your VNet doesn't have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.). We don't recommend creating a range any smaller than /28. If you already have a gateway subnet, you can view GatewaySubnet details by navigating to your virtual network. Click **Subnets** to view the range. If you want to change the range, you can delete and recreate the GatewaySubnet.

Public IP address: This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only Dynamic.

Active-Active mode: Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting unselected.

Leave **Configure BGP ASN** deselected, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

4. Click **Review + create** to run validation. Once validation passes, click **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

Create and configure VNet4

After you've configured VNet1, create VNet4 by repeating the previous steps and replacing the values with VNet4 values. You don't need to wait until the virtual network gateway for VNet1 has finished creating before you configure VNet4. If you're using your own values, make sure the address spaces don't overlap with any of the VNets to which you want to connect.

Configure the VNet1 gateway connection

When the virtual network gateways for both VNet1 and VNet4 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. However, if your VNets are in different resource groups in the same subscription, you can connect them by using the portal.

1. In the Azure portal, select **All resources**, enter *virtual network gateway* in the search box, and then navigate to the virtual network gateway for your VNet. For example, **VNet1GW**. Select the gateway to open the **Virtual network gateway** page. Under **Settings**, select **Connections**.

The screenshot shows the 'VNet1GW - Connections' page in the Azure portal. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, and Connections (which is selected and highlighted with a red box). Below that are Point-to-site configuration and Properties. At the top right, there's a search bar labeled 'Search connections', a '+ Add' button (also highlighted with a red box), and a 'Refresh' button. The main area has a table with columns 'Name', 'Status', and 'Connection type'. A message says 'No results'.

2. Select **+Add** to open the **Add connection** page.

The screenshot shows the 'Add connection' page and the 'Choose virtual network gateway' modal. The 'Add connection' page has fields for Name (VNet1-VNet4), Connection type (VNet-to-VNet), First virtual network gateway (VNet1GW), Second virtual network gateway (VNet4GW), Shared key (PSK) (abc123), IKE Protocol (IKEv2 selected), Subscription (Content Development (cherylmc)), Resource group (TestRG1), and Location (East US). The 'Choose virtual network gateway' modal shows two entries: VNet4GW (TestRG4) and VNet1GW (TestRG1), with VNet4GW selected and highlighted with a red box.

3. On the **Add connection** page, fill in the values for your connection:

- **Name:** Enter a name for your connection. For example, *VNet1toVNet4*.
- **Connection type:** Select **VNet-to-VNet** from the drop-down.
- **First virtual network gateway:** This field value is automatically filled in because you're creating this connection from the specified virtual network gateway.
- **Second virtual network gateway:** This field is the virtual network gateway of the VNet that you want to create a connection to. Select **Choose another virtual network gateway** to open the **Choose virtual network gateway** page.
 - 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 isn't in your subscription, use the [PowerShell](#).
 - Select the virtual network gateway to which you want to connect.
 - **Shared key (PSK):** In this field, enter a shared key for your connection. You can generate or create this key yourself. In a site-to-site connection, the key you use is 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're connecting to another virtual network gateway.

4. Select **OK** to save your changes.

Configure the VNet4 gateway connection

Next, create a connection from VNet4 to VNet1. In the portal, locate the virtual network gateway associated with VNet4. Follow the steps from the previous section, replacing the values to create a connection from VNet4 to VNet1. Make sure that you use the same shared key.

Verify your connections

1. Locate the virtual network gateway in the Azure portal.
2. On the **Virtual network gateway** page, select **Connections** to view the **Connections** page for the virtual network gateway. After the connection is established, you'll see the **Status** values change to **Connected**.

Name	Status	Connection type	Peer
VNet4toVNet1	Not connected	VNet-to-VNet	VNet1GW
VNet1toVNet4	Connected	VNet-to-VNet	VNet1GW

3. Under the **Name** column, select one of the connections to view more information. When data begins flowing, you'll see values for **Data in** and **Data out**.

		Move	Delete	Refresh
Resource group (change)	: TestRG1			
Status	: Connected	Data in	: 0 B	
Location	: East US	Data out	: 0 B	
Subscription (change)	: Content Development	Virtual network	: VNet1, VNet4	
Subscription ID	:	Virtual network gateway 1	: VNet1GW	
Tags (change)	: Click here to add tags	Virtual network gateway 2	: VNet4GW	

Add additional connections

If you want to add additional connections, navigate to the virtual network gateway from which you want to create the connection, then select **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 you create additional connections, verify that the address space for your virtual network doesn't overlap with any of the address spaces 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. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

Next steps

For information about how you can 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](#).

Configure a VNet-to-VNet VPN gateway connection using PowerShell

1/10/2020 • 18 minutes to read • [Edit Online](#)

This article helps you connect virtual networks by using the VNet-to-VNet connection type. The virtual networks can be in the same or different regions, and from the same or different subscriptions. When connecting VNets from different subscriptions, the subscriptions do not need to be associated with the same Active Directory tenant.

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

About connecting VNets

There are multiple ways to connect VNets. The sections below describe different ways to connect virtual networks.

VNet-to-VNet

Configuring a VNet-to-VNet connection is a good way to easily connect VNets. Connecting a virtual network to another virtual network using the VNet-to-VNet connection type (VNet2VNet) is similar to creating a Site-to-Site IPsec connection to an on-premises location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE, and both function the same way when communicating. The difference between the connection types is the way the local network gateway is configured. When you create a VNet-to-VNet connection, you do not see the local network gateway address space. It is automatically created and populated. If you update the address space for one VNet, the other VNet automatically knows to route to the updated address space. Creating a VNet-to-VNet connection is typically faster and easier than creating a Site-to-Site connection between VNets.

Site-to-Site (IPsec)

If you are working with a complicated network configuration, you may prefer to connect your VNets using the [Site-to-Site](#) steps, instead the VNet-to-VNet steps. When you use the Site-to-Site steps, you create and configure the local network gateways manually. The local network gateway for each VNet treats the other VNet as a local site. This lets you specify additional address space for the local network gateway in order to route traffic. If the address space for a VNet changes, you need to update the corresponding local network gateway to reflect the change. It does not automatically update.

VNet peering

You may want to consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway and has different constraints. Additionally, [VNet peering pricing](#) is calculated differently than [VNet-to-VNet VPN Gateway pricing](#). For more information, see [VNet peering](#).

Why create a VNet-to-VNet connection?

You may want to connect virtual networks using a VNet-to-VNet connection for the following reasons:

- **Cross region geo-redundancy and geo-presence**

- You can set up your own geo-replication or synchronization with secure connectivity without going over Internet-facing endpoints.
- With Azure Traffic Manager and Load Balancer, you can set up highly available workload with geo-

redundancy across multiple Azure regions. One important example is to set up SQL Always On with Availability Groups spreading across multiple Azure regions.

- **Regional multi-tier applications with isolation or administrative boundary**

- Within the same region, you can set up multi-tier applications with multiple virtual networks connected together due to isolation or administrative requirements.

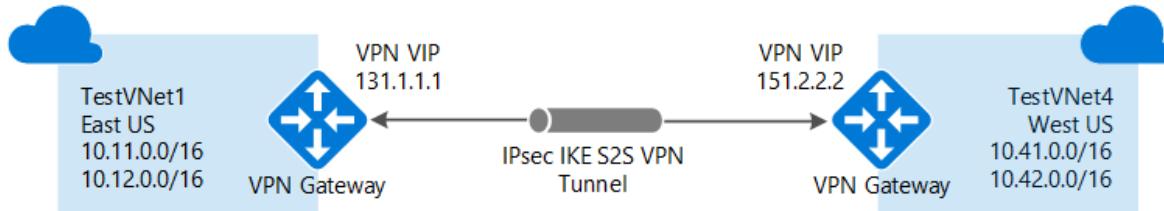
VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

Which VNet-to-VNet steps should I use?

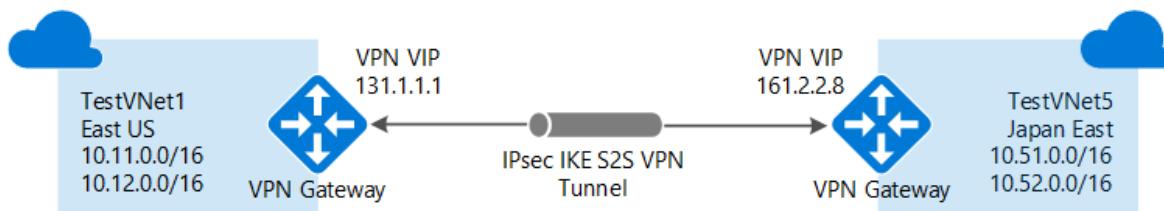
In this article, you see two different sets of steps. One set of steps for [VNets that reside in the same subscription](#) and one for [VNets that reside in different subscriptions](#). The key difference between the sets is that you must use separate PowerShell sessions when configuring the connections for VNets that reside in different subscriptions.

For this exercise, you can combine configurations, or just choose the one that you want to work with. All of the configurations use the VNet-to-VNet connection type. Network traffic flows between the VNets that are directly connected to each other. In this exercise, traffic from TestVNet4 does not route to TestVNet5.

- [VNets that reside in the same subscription](#): The steps for this configuration use TestVNet1 and TestVNet4.



- [VNets that reside in different subscriptions](#): The steps for this configuration use TestVNet1 and TestVNet5.



How to connect VNets that are in the same subscription

Before you begin

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

- Because it takes up to 45 minutes to create a gateway, Azure Cloud Shell will timeout periodically during this exercise. You can restart Cloud Shell by clicking in the upper left of the terminal. Be sure to redeclare any variables when you restart the terminal.
- If you would rather install latest version of the Azure PowerShell module locally, see [How to install and configure Azure PowerShell](#).

Step 1 - Plan your IP address ranges

In the following steps, you create two virtual networks along with their respective gateway subnets and configurations. You then create a VPN connection between the two VNets. It's important to plan the IP address ranges for your network configuration. Keep in mind that you must make sure that none of your VNet ranges or local network ranges overlap in any way. In these examples, we do not include a DNS server. If you want name resolution for your virtual networks, see [Name resolution](#).

We use the following values in the examples:

Values for TestVNet1:

- VNet Name: TestVNet1
- Resource Group: TestRG1
- Location: East US
- TestVNet1: 10.11.0.0/16 & 10.12.0.0/16
- FrontEnd: 10.11.0.0/24
- BackEnd: 10.12.0.0/24
- GatewaySubnet: 10.12.255.0/27
- GatewayName: VNet1GW
- Public IP: VNet1GWIP
- VPNTYPE: RouteBased
- Connection(1to4): VNet1toVNet4
- Connection(1to5): VNet1toVNet5 (For VNets in different subscriptions)
- ConnectionType: VNet2VNet

Values for TestVNet4:

- VNet Name: TestVNet4
- TestVNet2: 10.41.0.0/16 & 10.42.0.0/16
- FrontEnd: 10.41.0.0/24
- BackEnd: 10.42.0.0/24
- GatewaySubnet: 10.42.255.0/27
- Resource Group: TestRG4
- Location: West US
- GatewayName: VNet4GW
- Public IP: VNet4GWIP
- VPNTYPE: RouteBased
- Connection: VNet4toVNet1
- ConnectionType: VNet2VNet

Step 2 - Create and configure TestVNet1

1. Verify your subscription settings.

Connect to your account if you are running PowerShell locally on your computer. If you are using Azure Cloud Shell, you are connected automatically.

```
Connect-AzAccount
```

Check the subscriptions for the account.

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName nameofsubscription
```

2. Declare your variables. This example declares the variables using the values for this exercise. In most cases, you should replace the values with your own. However, you can use these variables if you are running through the steps to become familiar with this type of configuration. Modify the variables if needed, then copy and paste them into your PowerShell console.

```
$RG1 = "TestRG1"
$Location1 = "East US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$VNetPrefix11 = "10.11.0.0/16"
$VNetPrefix12 = "10.12.0.0/16"
$FESubPrefix1 = "10.11.0.0/24"
$BESubPrefix1 = "10.12.0.0/24"
$GWSubPrefix1 = "10.12.255.0/27"
$GWName1 = "VNet1GW"
$GWIPName1 = "VNet1GWIP"
$GWIPconfName1 = "gwipconf1"
$Connection14 = "VNet1toVNet4"
$Connection15 = "VNet1toVNet5"
```

3. Create a resource group.

```
New-AzResourceGroup -Name $RG1 -Location $Location1
```

4. Create the subnet configurations for TestVNet1. This example creates a virtual network named TestVNet1 and three subnets, one called GatewaySubnet, one called FrontEnd, and one called Backend. When substituting values, it's important that you always name your gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation fails. For this reason, it is not assigned via variable below.

The following example uses the variables that you set earlier. In this example, the gateway subnet is using a /27. While it is possible to create a gateway subnet as small as /29, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. This will allow for enough addresses to accommodate possible additional configurations that you may want in the future.

```
$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsu1 = New-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix $GWSubPrefix1
```

5. Create TestVNet1.

```
New-AzVirtualNetwork -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-AzPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 `  
-Location $Location1 -AllocationMethod Dynamic
```

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

```
$vnet1 = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1  
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1  
$gwipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 `  
-Subnet $subnet1 -PublicIpAddress $gwpip1
```

8. Create the gateway for TestVNet1. In this step, you create the virtual network gateway for your TestVNet1. VNet-to-VNet configurations require a RouteBased VpnType. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

```
New-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 `  
-Location $Location1 -IpConfigurations $gwipconf1 -GatewayType Vpn `  
-VpnType RouteBased -GatewaySku VpnGw1
```

After you finish the commands, it will take up to 45 minutes to create this gateway. If you are using Azure Cloud Shell, you can restart your CloudShell session by clicking in the upper left of the Cloud Shell terminal, then configure TestVNet4. You don't need to wait until the TestVNet1 gateway completes.

Step 3 - Create and configure TestVNet4

Once you've configured TestVNet1, create TestVNet4. Follow the steps below, replacing the values with your own when needed.

1. Connect and declare your variables. Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG4 = "TestRG4"  
$Location4 = "West US"  
$VNetName4 = "TestVNet4"  
$FESubName4 = "FrontEnd"  
$BESubName4 = "Backend"  
$VNetPrefix41 = "10.41.0.0/16"  
$VNetPrefix42 = "10.42.0.0/16"  
$FESubPrefix4 = "10.41.0.0/24"  
$BESubPrefix4 = "10.42.0.0/24"  
$GWSubPrefix4 = "10.42.255.0/27"  
$GWName4 = "VNet4GW"  
$GWIPName4 = "VNet4GWIP"  
$GWIPconfName4 = "gwipconf4"  
$Connection41 = "VNet4toVNet1"
```

2. Create a resource group.

```
New-AzResourceGroup -Name $RG4 -Location $Location4
```

3. Create the subnet configurations for TestVNet4.

```
$fesub4 = New-AzVirtualNetworkSubnetConfig -Name $FESubName4 -AddressPrefix $FESubPrefix4  
$besub4 = New-AzVirtualNetworkSubnetConfig -Name $BESubName4 -AddressPrefix $BESubPrefix4  
$gwsub4 = New-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix $GWSubPrefix4
```

4. Create TestVNet4.

```
New-AzVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4 `  
-Location $Location4 -AddressPrefix $VnetPrefix41,$VnetPrefix42 -Subnet $fesub4,$besub4,$gwsb4
```

5. Request a public IP address.

```
$gwpip4 = New-AzPublicIpAddress -Name $GWIPName4 -ResourceGroupName $RG4 `  
-Location $Location4 -AllocationMethod Dynamic
```

6. Create the gateway configuration.

```
$vnet4 = Get-AzVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4  
$subnet4 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet4  
$gwipconf4 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName4 -Subnet $subnet4 -PublicIpAddress  
$gwpip4
```

7. Create the TestVNet4 gateway. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

```
New-AzVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4 `  
-Location $Location4 -IpConfigurations $gwipconf4 -GatewayType Vpn `  
-VpnType RouteBased -GatewaySku VpnGw1
```

Step 4 - Create the connections

Wait until both gateways are completed. Restart your Azure Cloud Shell session and copy and paste the variables from the beginning of Step 2 and Step 3 into the console to redeclare values.

1. Get both virtual network gateways.

```
$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$vnet4gw = Get-AzVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4
```

2. Create the TestVNet1 to TestVNet4 connection. In this step, you create the connection from TestVNet1 to TestVNet4. You'll see a shared key referenced in the examples. You can use your own values for the shared key. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

```
New-AzVirtualNetworkGatewayConnection -Name $Connection14 -ResourceGroupName $RG1 `  
-VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet4gw -Location $Location1 `  
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

3. Create the TestVNet4 to TestVNet1 connection. This step is similar to the one above, except you are creating the connection from TestVNet4 to TestVNet1. Make sure the shared keys match. The connection will be established after a few minutes.

```
New-AzVirtualNetworkGatewayConnection -Name $Connection41 -ResourceGroupName $RG4 `  
-VirtualNetworkGateway1 $vnet4gw -VirtualNetworkGateway2 $vnet1gw -Location $Location4 `  
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

4. Verify your connection. See the section [How to verify your connection](#).

How to connect VNets that are in different subscriptions

In this scenario, you connect TestVNet1 and TestVNet5. TestVNet1 and TestVNet5 reside in different subscriptions. The subscriptions do not need to be associated with the same Active Directory tenant.

The difference between these steps and the previous set is that some of the configuration steps need to be performed in a separate PowerShell session in the context of the second subscription. Especially when the two subscriptions belong to different organizations.

Due to changing subscription context in this exercise, you may find it easier to use PowerShell locally on your computer, rather than using the Azure Cloud Shell, when you get to Step 8.

Step 5 - Create and configure TestVNet1

You must complete [Step 1](#) and [Step 2](#) from the previous section to create and configure TestVNet1 and the VPN Gateway for TestVNet1. For this configuration, you are not required to create TestVNet4 from the previous section, although if you do create it, it will not conflict with these steps. Once you complete Step 1 and Step 2, continue with Step 6 to create TestVNet5.

Step 6 - Verify the IP address ranges

It is important to make sure that the IP address space of the new virtual network, TestVNet5, does not overlap with any of your VNet ranges or local network gateway ranges. In this example, the virtual networks may belong to different organizations. For this exercise, you can use the following values for the TestVNet5:

Values for TestVNet5:

- VNet Name: TestVNet5
- Resource Group: TestRG5
- Location: Japan East
- TestVNet5: 10.51.0.0/16 & 10.52.0.0/16
- FrontEnd: 10.51.0.0/24
- BackEnd: 10.52.0.0/24
- GatewaySubnet: 10.52.255.0.0/27
- GatewayName: VNet5GW
- Public IP: VNet5GWIP
- VPNTYPE: RouteBased
- Connection: VNet5toVNet1
- ConnectionType: VNet2VNet

Step 7 - Create and configure TestVNet5

This step must be done in the context of the new subscription. This part may be performed by the administrator in a different organization that owns the subscription.

1. Declare your variables. Be sure to replace the values with the ones that you want to use for your configuration.

```
$Sub5 = "Replace_With_the_New_Subscription_Name"
$RG5 = "TestRG5"
$Location5 = "Japan East"
$VnetName5 = "TestVNet5"
$FESubName5 = "FrontEnd"
$BESubName5 = "Backend"
$GWSubName5 = "GatewaySubnet"
$VnetPrefix51 = "10.51.0.0/16"
$VnetPrefix52 = "10.52.0.0/16"
$FESubPrefix5 = "10.51.0.0/24"
$BESubPrefix5 = "10.52.0.0/24"
$GWSubPrefix5 = "10.52.255.0/27"
$GWName5 = "VNet5GW"
$GWIPName5 = "VNet5GWIP"
$GWIPconfName5 = "gwipconf5"
$Connection51 = "VNet5stoVNet1"
```

2. Connect to subscription 5. Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Connect-AzAccount
```

Check the subscriptions for the account.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName $Sub5
```

3. Create a new resource group.

```
New-AzResourceGroup -Name $RG5 -Location $Location5
```

4. Create the subnet configurations for TestVNet5.

```
$fesub5 = New-AzVirtualNetworkSubnetConfig -Name $FESubName5 -AddressPrefix $FESubPrefix5
$besub5 = New-AzVirtualNetworkSubnetConfig -Name $BESubName5 -AddressPrefix $BESubPrefix5
$gwsb5 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName5 -AddressPrefix $GWSubPrefix5
```

5. Create TestVNet5.

```
New-AzVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5 -Location $Location5 ` 
-AddressPrefix $VnetPrefix51,$VnetPrefix52 -Subnet $fesub5,$besub5,$gwsb5
```

6. Request a public IP address.

```
$gwip5 = New-AzPublicIpAddress -Name $GWIPName5 -ResourceGroupName $RG5 ` 
-Location $Location5 -AllocationMethod Dynamic
```

7. Create the gateway configuration.

```
$vnet5 = Get-AzVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5
$subnet5 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet5
$gwipconf5 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName5 -Subnet $subnet5 -PublicIpAddress
$gwip5
```

8. Create the TestVNet5 gateway.

```
New-AzVirtualNetworkGateway -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. Sign in and connect to Subscription 1 before running the following example:

```
$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
```

Copy the output of the following elements and send these to the administrator of Subscription 5 via email or another method.

```
$vnet1gw.Name
$vnet1gw.Id
```

These two elements will have values similar to the following example output:

```
PS D:\> $vnet1gw.Name
VNet1GW
PS D:\> $vnet1gw.Id
/subscriptions/b636ca99-6f88-4df4-a7c3-
2f8dc4545509/resourceGroupsTestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW
```

2. **[Subscription 5]** Get the virtual network gateway for Subscription 5. Sign in and connect to Subscription 5 before running the following example:

```
$vnet5gw = Get-AzVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5
```

Copy the output of the following elements and send these to the administrator of Subscription 1 via email or another method.

```
$vnet5gw.Name
$vnet5gw.Id
```

These two elements will have values similar to the following example output:

```
PS C:\> $vnet5gw.Name
VNet5GW
PS C:\> $vnet5gw.Id
/subscriptions/66c8e4f1-ecd6-47ed-9de7-
7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW
```

3. **[Subscription 1]** Create the TestVNet1 to TestVNet5 connection. In this step, you create the connection from TestVNet1 to TestVNet5. The difference here is that \$vnet5gw cannot be obtained directly because it is in a different subscription. You will need to create a new PowerShell object with the values communicated from Subscription 1 in the steps above. Use the example below. Replace the Name, Id, and shared key with your own values. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

Connect to Subscription 1 before running the following example:

```
$vnet5gw = New-Object -TypeName Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet5gw.Name = "VNet5GW"
$vnet5gw.Id   = "/subscriptions/66c8e4f1-ecd6-47ed-9de7-
7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW"
$Connection15 = "VNet1toVNet5"
New-AzVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -
VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet5gw -Location $Location1 -ConnectionType
Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

4. **[Subscription 5]** Create the TestVNet5 to TestVNet1 connection. This step is similar to the one above, except you are creating the connection from TestVNet5 to TestVNet1. The same process of creating a PowerShell object based on the values obtained from Subscription 1 applies here as well. In this step, be sure that the shared keys match.

Connect to Subscription 5 before running the following example:

```
$vnet1gw = New-Object -TypeName Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet1gw.Name = "VNet1GW"
$vnet1gw.Id   = "/subscriptions/b636ca99-6f88-4df4-a7c3-
2f8dc4545509/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW "
$Connection51 = "VNet5toVNet1"
New-AzVirtualNetworkGatewayConnection -Name $Connection51 -ResourceGroupName $RG5 -
VirtualNetworkGateway1 $vnet5gw -VirtualNetworkGateway2 $vnet1gw -Location $Location5 -ConnectionType
Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

How to verify a connection

IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

You can verify that your connection succeeded by using the 'Get-AzVirtualNetworkGatewayConnection' cmdlet, with or without '-Debug'.

1. Use the following cmdlet example, configuring the values to match your own. If prompted, select 'A' in order to run 'All'. In the example, '-Name' refers to the name of the connection that you want to test.

```
Get-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. After the cmdlet has finished, view the values. In the example below, the connection status shows as 'Connected' and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

VNet-to-VNet FAQ

The VNet-to-VNet FAQ applies to VPN gateway connections. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. See the [Virtual Machines documentation](#) for more information.
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).

Configure a VNet-to-VNet VPN gateway connection using Azure CLI

1/9/2020 • 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.

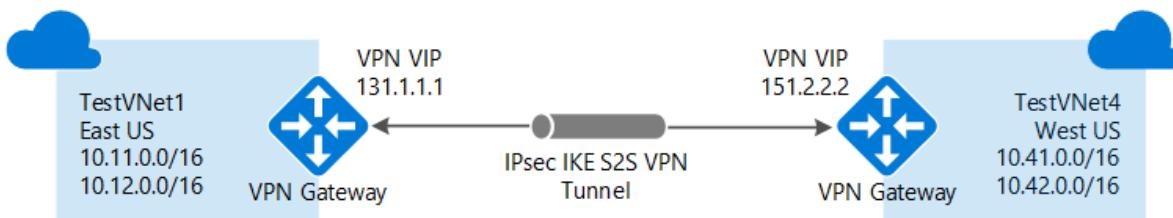
VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

Which VNet-to-VNet steps should I use?

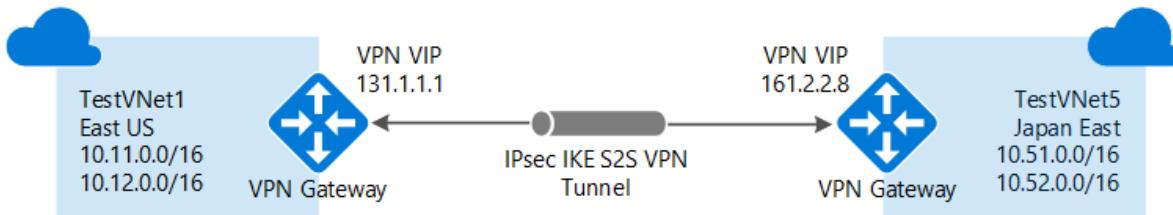
In this article, you see two different sets of VNet-to-VNet connection steps. One set of steps for [VNets that reside in the same subscription](#) and one for [VNets that reside in different subscriptions](#).

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 the Azure CLI](#).

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](#).

```
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 <subscriptionID>
```

 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 <subscriptionID>` 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 "eefffgg" --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 "eefffgg" --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 Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

You can verify that your connection succeeded by using the `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. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. 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

2/11/2020 • 21 minutes to read • [Edit Online](#)

This article shows you how to connect classic VNets to Resource Manager VNets to allow the resources located in the separate deployment models to communicate with each other. The steps in this article primarily use the Azure portal, but you can also create this configuration using the PowerShell by selecting the article from this list.

Connecting a classic VNet to a Resource Manager VNet is similar to connecting a VNet to an on-premises site location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE. You can create a connection between VNets that are in different subscriptions and in different regions. You can also connect VNets that already have connections to on-premises networks, as long as the gateway that they have been configured with is dynamic or route-based. For more information about VNet-to-VNet connections, see the [VNet-to-VNet FAQ](#) at the end of this article.

If you do not already have a virtual network gateway and do not want to create one, you may want to instead consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway. For more information, see [VNet peering](#).

Before you begin

- These steps assume that both VNets have already been created. If you are using this article as an exercise and don't have VNets, there are links in the steps to help you create them.
- Verify that the address ranges for the VNets do not overlap with each other, or overlap with any of the ranges for other connections that the gateways may be connected to.
- Install the latest PowerShell cmdlets for both Resource Manager and Service Management (classic). In this article, we use both the Azure portal and PowerShell. PowerShell is required to create the connection from the classic VNet to the Resource Manager VNet. For more information, see [How to install and configure Azure PowerShell](#).

Example settings

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

Classic VNet

VNet name = ClassicVNet

Address space = 10.0.0.0/24

Subnet name = Subnet-1

Subnet address range = 10.0.0.0/27

Subscription = the subscription you want to use

Resource Group = ClassicRG

Location = West US

GatewaySubnet = 10.0.0.32/28

Local site = RMVNetLocal

Resource Manager VNet

VNet name = RMVNet

Address space = 192.168.0.0/16

Resource Group = RG1

Location = East US
 Subnet name = Subnet-1
 Address range = 192.168.1.0/24
 GatewaySubnet = 192.168.0.0/26
 Virtual network gateway name = RMGateway
 Gateway type = VPN
 VPN type = Route-based
 SKU = VpnGw1
 Location = East US
 Virtual network = RMVNet
 (associate the VPN gateway to this VNet) First IP configuration = rmgwpip
 (gateway public IP address) Local network gateway = ClassicVNetLocal
 Connection name = RMtoClassic

Connection overview

For this configuration, you create a VPN gateway connection over an IPsec/IKE VPN tunnel between the virtual networks. Make sure that none of your VNet ranges overlap with each other, or with any of the local networks that they connect to.

The following table shows an example of how the example VNets and local sites are defined:

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE
ClassicVNet	(10.0.0.0/24)	West US	RMVNetLocal (192.168.0.0/16)
RMVNet	(192.168.0.0/16)	East US	ClassicVNetLocal (10.0.0.0/24)

Section 1 - Configure the classic VNet settings

In this section, you create the classic VNet, the local network (local site), and the virtual network gateway. Screenshots are provided as examples. Be sure to replace the values with your own, or use the [Example](#) values.

1. Create a classic VNet

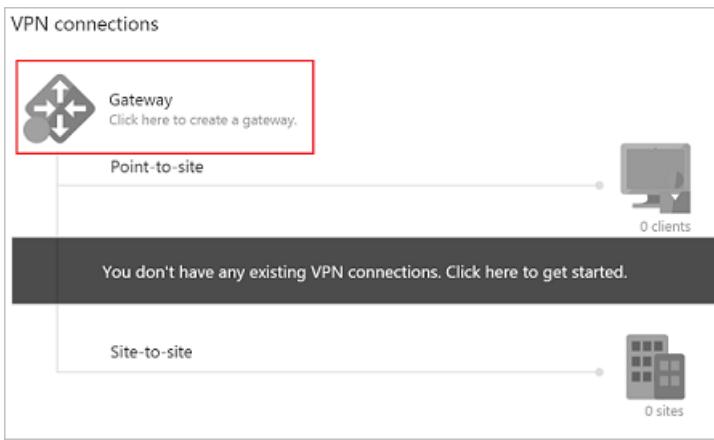
If you don't have a classic VNet and are running these steps as an exercise, you can create a VNet by using [this article](#) and the [Example](#) settings values from above.

If you already have a VNet with a VPN gateway, verify that the gateway is Dynamic. If it's Static, you must first delete the VPN gateway before you proceed to [Configure the local site](#).

1. Open the [Azure portal](#) and sign in with your Azure account.
2. Click **+ Create a resource** to open the 'New' page.
3. In the 'Search the marketplace' field, type 'Virtual Network'. If you instead, select Networking -> Virtual Network, you will not get the option to create a classic VNet.
4. Locate 'Virtual Network' from the returned list and click it to open the Virtual Network page.
5. On the virtual network page, select 'Classic' to create a classic VNet. If you take the default here, you will wind up with a Resource Manager VNet instead.

2. Configure the local site

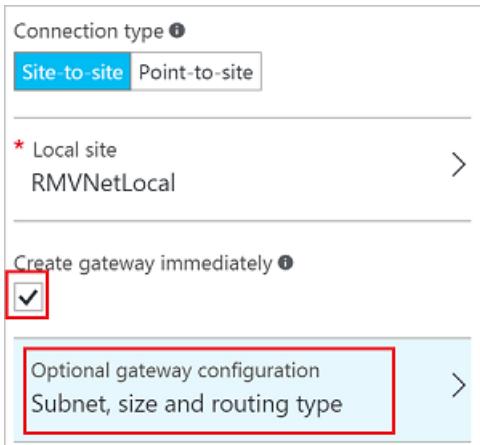
1. Navigate to **All resources** and locate the **ClassicVNet** in the list.
2. Click **Gateway** in the **Settings** section of the menu, and then click on the banner to create a gateway.



3. On the **New VPN Connection** page, for **Connection type**, select **Site-to-site**.
4. For **Local site**, click **Configure required settings**. This opens the **Local site** page.
5. On the **Local site** page, create a name to refer to the Resource Manager VNet. For example, 'RMVNetLocal'.
6. If the VPN gateway for the Resource Manager VNet already has a Public IP address, use the value for the **VPN gateway IP address** field. If you are doing these steps as an exercise, or don't yet have a virtual network gateway for your Resource Manager VNet, you can make up a placeholder IP address. Make sure that the placeholder IP address uses a valid format. Later, you replace the placeholder IP address with the Public IP address of the Resource Manager virtual network gateway.
7. For **Client Address Space**, use the **values** for the virtual network IP address spaces for the Resource Manager VNet. This setting is used to specify the address spaces to route to the Resource Manager virtual network. In the example, we use 192.168.0.0/16, the address range for the RMVNet.
8. Click **OK** to save the values and return to the **New VPN Connection** page.

3. Create the virtual network gateway

1. On the **New VPN Connection** page, select the **Create gateway immediately** checkbox.
2. Click **Optional gateway configuration** to open the **Gateway configuration** page.



3. Click **Subnet - Configure required settings** to open the **Add subnet** page. The **Name** is already configured with the required value: **GatewaySubnet**.
4. The **Address range** refers to the range for the gateway subnet. Although you can create a gateway subnet with a /29 address range (3 addresses), we recommend creating a gateway subnet that contains more IP addresses. This will accommodate future configurations that may require more available IP addresses. If possible, use /27 or /28. If you are using these steps as an exercise, you can refer to the **Example values**. For this example, we use '10.0.0.32/28'. Click **OK** to create the gateway subnet.
5. On the **Gateway configuration** page, **Size** refers to the gateway SKU. Select the gateway SKU for your VPN gateway.
6. Verify the **Routing Type** is **Dynamic**, then click **OK** to return to the **New VPN Connection** page.

7. On the **New VPN Connection** page, click **OK** to begin creating your VPN gateway. Creating a VPN gateway can take up to 45 minutes to complete.

4. Copy the virtual network gateway Public IP address

After the virtual network gateway has been created, you can view the gateway IP address.

1. Navigate to your classic VNet, and click **Overview**.
2. Click **VPN connections** to open the VPN connections page. On the VPN connections page, you can view the Public IP address. This is the Public IP address assigned to your virtual network gateway. Make a note of the IP address. You use it in later steps when you work with your Resource Manager local network gateway configuration settings.
3. You can view the status of your gateway connections. Notice the local network site you created is listed as 'Connecting'. The status will change after you have created your connections. You can close this page when you are finished viewing the status.

Section 2 - Configure the Resource Manager VNet settings

In this section, you create the virtual network gateway and the local network gateway for your Resource Manager VNet. Screenshots are provided as examples. Be sure to replace the values with your own, or use the [Example](#) values.

1. Create a virtual network

Example values:

- VNet name = RMVNet
- Address space = 192.168.0.0/16
- Resource Group = RG1
- Location = East US
- Subnet name = Subnet-1
- Address range = 192.168.1.0/24

If you don't have a Resource Manager VNet and are running these steps as an exercise, create a virtual network with the steps in [Create a virtual network](#), using the example values.

2. Create a virtual network gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

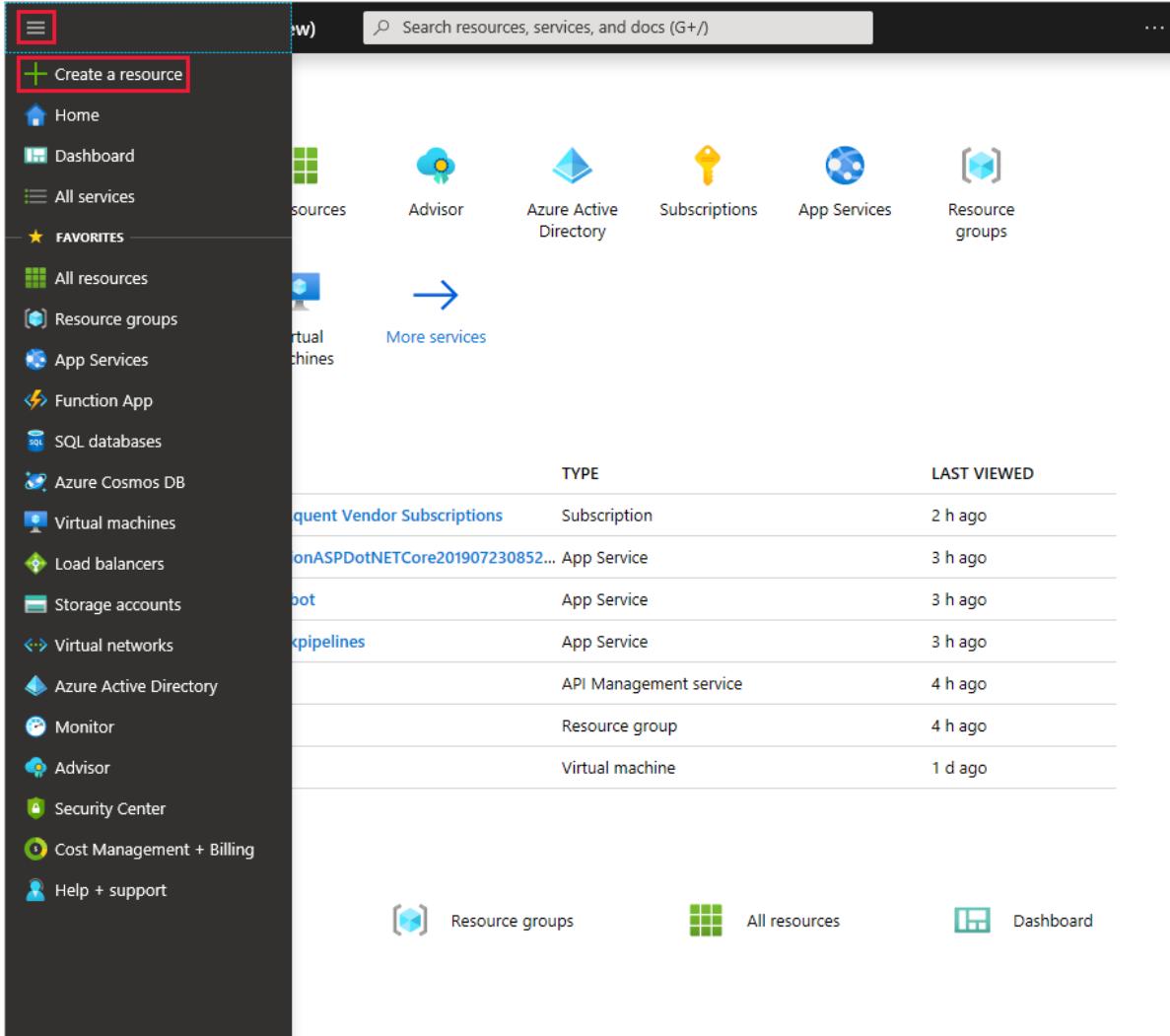
When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

Example values:

- Virtual network gateway name = RMGateway
- Gateway type = VPN
- VPN type = Route-based
- SKU = VpnGw1
- Location = East US
- Virtual network = RMVNet
- GatewaySubnet = 192.168.0.0/26
- First IP configuration = rmgwpip

1. From the [Azure portal](#) menu, select **Create a resource**.



The screenshot shows the Azure portal interface. On the left, a dark sidebar lists various services and links. A red box highlights the '+ Create a resource' button at the top of this sidebar. The main area features a search bar at the top right and several service icons: Advisor, Azure Active Directory, Subscriptions, App Services, and Resource groups. Below these are links for 'More services' and 'Virtual machines'. The central part of the screen displays a table of recently viewed resources, with columns for 'TYPE', 'LAST VIEWED', and resource names like 'Frequent Vendor Subscriptions', 'onASPDotNETCore201907230852...', 'root', 'pipelines', 'Azure Active Directory', 'Monitor', 'Advisor', 'Security Center', 'Cost Management + Billing', and 'Help + support'. At the bottom, there are three navigation buttons: 'Resource groups', 'All resources', and 'Dashboard'.

2. In the **Search the Marketplace** field, type 'Virtual Network Gateway'. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** page, click **Create**. This opens the **Create virtual network gateway** page.

Create virtual network gateway

Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)

Project details

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

Subscription *

Resource group ⓘ

TestRG1 (derived from virtual network's resource group)

Instance details

Name *

 VNet1GW ✓

Region *

 (US) East US ✓

Gateway type * ⓘ

VPN ExpressRoute

VPN type * ⓘ

Route-based Policy-based

SKU * ⓘ

 VpnGw1 ✓

Generation ⓘ

 Generation1 ✓

Virtual network * ⓘ

 VNet1 ✓

[Create virtual network](#)

Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range * ⓘ

 10.11.255.0/27 ✓

10.11.255.0 - 10.11.255.31 (32 addresses)

Public IP address

Public IP address * ⓘ

Create new Use existing

Public IP address name *

 VNet1GWpip ✓

Public IP address SKU

Basic

Assignment

Dynamic Static

Enable active-active mode * ⓘ

Enabled Disabled

Configure BGP ASN * ⓘ

Enabled Disabled

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

Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway must be the same as the virtual network.
- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.

- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).

Virtual network: Choose the virtual network to which you want to add this gateway.

Gateway subnet address range: This field only appears if your VNet doesn't have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.). We don't recommend creating a range any smaller than /28. If you already have a gateway subnet, you can view GatewaySubnet details by navigating to your virtual network. Click **Subnets** to view the range. If you want to change the range, you can delete and recreate the GatewaySubnet.

Public IP address: This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only Dynamic.

Active-Active mode: Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting unselected.

Leave **Configure BGP ASN** deselected, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

4. Click **Review + create** to run validation. Once validation passes, click **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

IMPORTANT

When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

3. Create a local network gateway

Example values: Local network gateway = ClassicVNetLocal

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE	GATEWAY PUBLIC IP ADDRESS
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, 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 gate...

* Name

* IP address

Address space

...

Configure BGP settings

* Subscription

* Resource group [Create new](#)

* Location

[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. 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.*

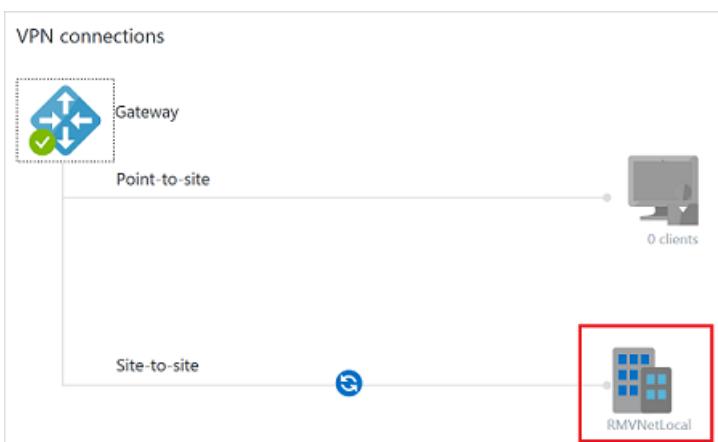
- **Configure BGP settings:** Use only when configuring BGP. Otherwise, don't select this.
- **Subscription:** Verify that the correct subscription is showing.
- **Resource Group:** Select the resource group that you want to use. You can either create a new resource group, or select one that you have 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 to create the gateway.

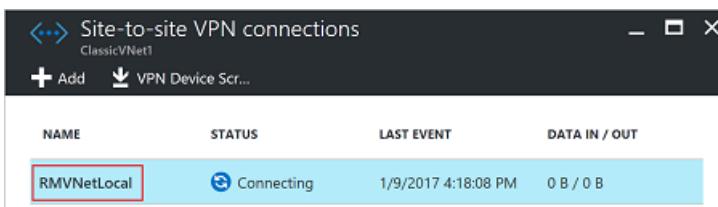
Section 3 - Modify the classic VNet local site settings

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.

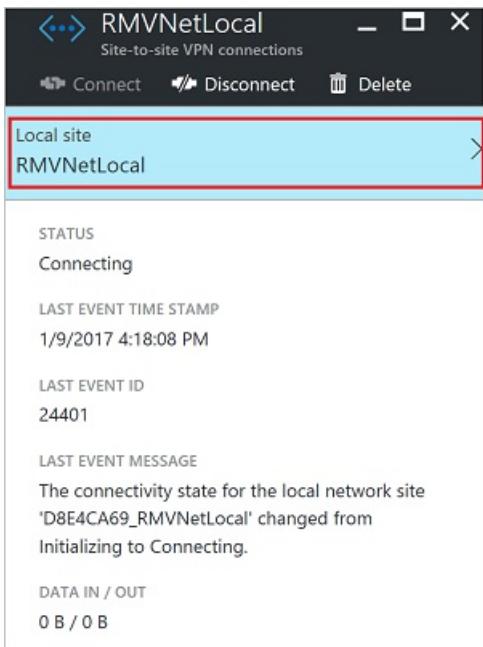
1. In the Azure portal, navigate to the classic virtual network.
2. On the page for your virtual network, click **Overview**.
3. In the **VPN connections** section, click the name of your local site in the graphic.



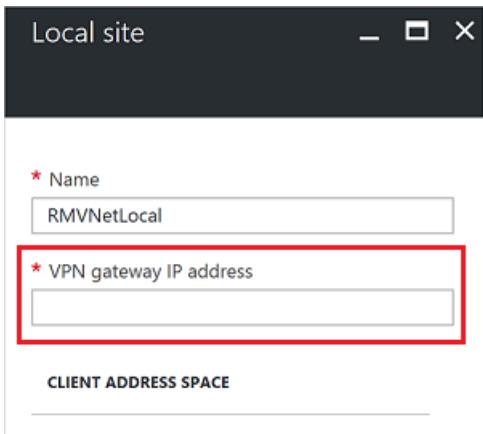
4. On the **Site-to-site VPN connections** page, click the name of the site.



5. On the connection page for your local site, click the name of the local site to open the **Local site** page.



6. On the **Local site** page, replace the **VPN gateway IP address** with the IP address of the Resource Manager gateway.



7. Click **OK** to update the IP address.

Section 4 - Create Resource Manager to classic connection

In these steps, you configure the connection from the Resource Manager VNet to the classic VNet using the Azure portal.

1. In **All resources**, locate the local network gateway. In our example, the local network gateway is **ClassicVNetLocal**.
2. 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.
3. In **All resources**, click the local network gateway.
4. Click **Connections** to open the Connections page.
5. On the **Connections** page, click **+** to add a connection.
6. On the **Add connection** page, name the connection. For example, 'RMtoClassic'.
7. **Site-to-Site** is already selected on this page.
8. Select the virtual network gateway that you want to associate with this site.
9. Create a **shared key**. This key is also used in the connection that you create from the classic VNet to the Resource Manager VNet. You can generate the key or make one up. In our example, we use 'abc123', but you can (and should) use something more complex.
10. Click **OK** to create the connection.

Section 5 - Create classic to Resource Manager connection

In these steps, you configure the connection from the classic VNet to the Resource Manager VNet. These steps require PowerShell. You can't create this connection in the portal. Make sure you have downloaded and installed both the classic (SM) and Resource Manager (RM) PowerShell cmdlets.

1. Connect to your Azure account

Open the PowerShell console with elevated rights and log in to your Azure account. After logging in, your account settings are downloaded so that they are available to Azure PowerShell. The following cmdlet prompts you for the login credentials for your Azure Account for the Resource Manager deployment model:

```
Connect-AzAccount
```

Get a list of your Azure subscriptions.

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Name of subscription"
```

Next, log in to use the classic PowerShell cmdlets (Service Management). Use the following command to add your Azure account for the classic deployment model:

```
Add-AzureAccount
```

Get a list of your subscriptions. This step may be necessary when adding the Service Management cmdlets, depending on your Azure module install.

```
Get-AzureSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionName "Name of subscription"
```

2. View the network configuration file values

When you create a VNet in the Azure portal, the full name that Azure uses is not visible in the Azure portal. For example, a VNet that appears to be named 'ClassicVNet' in the Azure portal may have a much longer name in the network configuration file. The name might look something like: 'Group ClassicRG ClassicVNet'. In these steps, you download the network configuration file and view the values.

Create a directory on your computer and then export the network configuration file to the directory. In this example, the network configuration file is exported to C:\AzureNet.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

Open the file with a text editor and view the name for your classic VNet. Use the names in the network configuration file when running your PowerShell cmdlets.

- VNet names are listed as **VirtualNetworkSite name** =
- Site names are listed as **LocalNetworkSite name**=

3. Create the connection

Set the shared key and create the connection from the classic VNet to the Resource Manager VNet. You cannot set the shared key using the portal. Make sure you run these steps while logged in using the classic version of the PowerShell cmdlets. To do so, use **Add-AzureAccount**. Otherwise, you will not be able to set the '`-AzureVNetGatewayKey`'.

- In this example, **-VNetName** is the name of the classic VNet as found in your network configuration file.
- The **-LocalNetworkSiteName** is the name you specified for the local site, as found in your network configuration file.
- The **-SharedKey** is a value that you generate and specify. For this example, we used *abc123*, but you can generate something more complex. The important thing is that the value you specify here must be the same value that you specified when creating your Resource Manager to classic connection.

```
Set-AzureVNetGatewayKey -VNetName "Group ClassicRG ClassicVNet" ` 
-LocalNetworkSiteName "172B9E16_RMVNetLocal" -SharedKey abc123
```

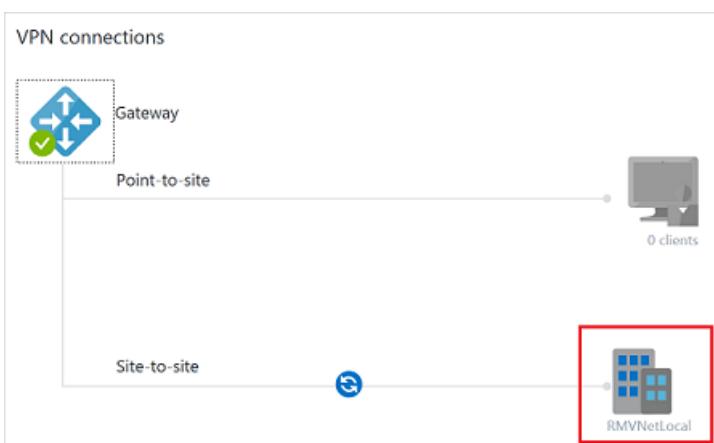
Section 6 - Verify your connections

You can verify your connections by using the Azure portal or PowerShell. When verifying, you may need to wait a minute or two as the connection is being created. When a connection is successful, the connectivity state changes from 'Connecting' to 'Connected'.

To verify the connection from your classic VNet to your Resource Manager VNet

In the Azure portal, you can view the connection status for a classic VNet VPN Gateway by navigating to the connection. The following steps show one way to navigate to your connection and verify.

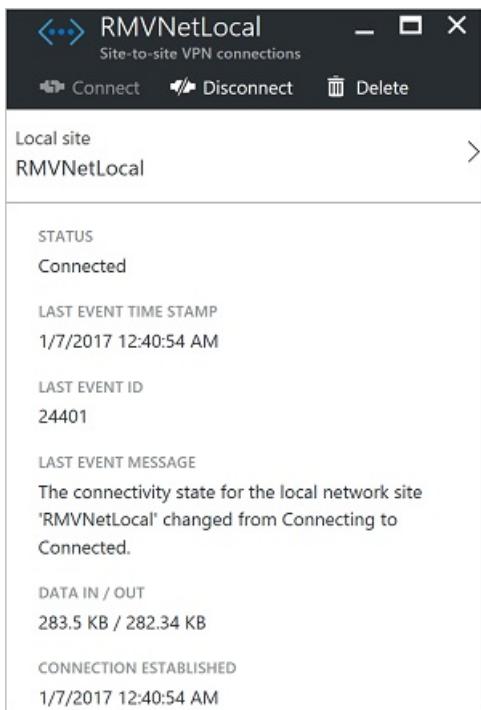
1. In the [Azure portal](#), click **All resources** and navigate to your classic virtual network.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.
3. On the VPN connections graphic, click the site.



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

- 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.

- In the [Azure portal](#), click **All resources** and navigate to your virtual network gateway.
- On the blade for your virtual network gateway, click **Connections**. You can see the status of each connection.
- 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. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by

using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

Connect virtual networks from different deployment models using PowerShell

2/11/2020 • 15 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. If you need to create a resource manager virtual network, see [Create a resource group and a virtual network](#). To create a classic virtual network, see [Create a classic VNet](#).
- 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 Virtual Network gateway(VPN, Express Route gateway) to stop functioning as expected. For more information about network security groups, see [What is a network security group?](#)

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

The prerequisites assume that you already have created an RM VNet. In this step, you create a VPN gateway for the RM VNet. Don't start these steps until after you have retrieved the public IP address for the classic VNet's gateway.

1. Sign in to your Azure account in the PowerShell console. The following cmdlet prompts you for the login credentials for your Azure Account. After signing in, your account settings are downloaded so that they are available to Azure PowerShell. You can optionally use the "Try It" feature to launch Azure Cloud Shell in the browser.

If you use Azure Cloud Shell, skip the following cmdlet:

```
Connect-AzAccount
```

To verify that you are using the right subscription, run the following cmdlet:

```
Get-AzSubscription
```

If you have more than one subscription, specify the subscription that you want to use.

```
Select-AzSubscription -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 text "n.n.n.n" to reflect the correct IP address.

```
New-AzLocalNetworkGateway -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-AzPublicIpAddress -Name gwipip `  
-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-AzVirtualNetworkSubnetConfig -Name GatewaySubnet `  
-VirtualNetwork (Get-AzVirtualNetwork -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.

```
$gwipconfig = New-AzVirtualNetworkGatewayIpConfig `  
-Name gwipconfig -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-AzVirtualNetworkGateway -Name RMGateway -ResourceGroupName RG1 `  
-Location "EastUS" -GatewaySKU Standard -GatewayType Vpn `  
-IpConfigurations $gwipconfig `  
-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-AzPublicIpAddress -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. Because you are working with the classic VNet, use PowerShell installed locally to your computer, not the Azure Cloud Shell TryIt.

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-AzLocalNetworkGateway -Name ClassicVNetLocal -ResourceGroupName RG1  
$vnet02gateway = Get-AzVirtualNetworkGateway -Name RMGateway -ResourceGroupName RG1
```

Create the connection. Notice that the **-ConnectionType** is IPsec, not Vnet2Vnet.

```
New-AzVirtualNetworkGatewayConnection -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"
```

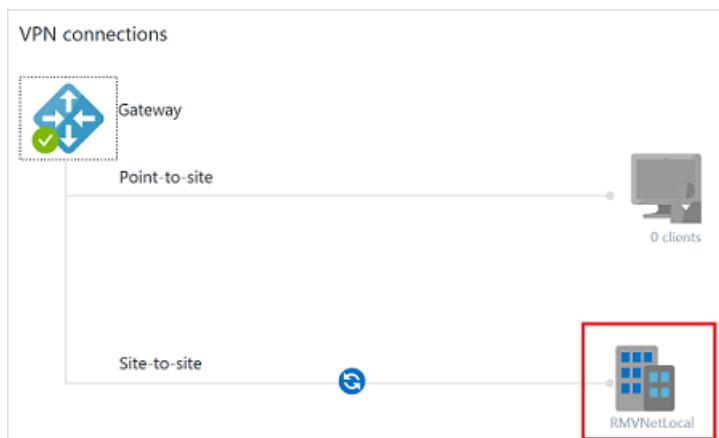
2. After the cmdlet has finished, view the values. In the example below, the Connectivity State shows as 'Connected' and you can see ingress and egress bytes.

```
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.

Site-to-site VPN connections			
ClassicVNet			
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.

The screenshot shows the 'RMVNetLocal' connection details:

- 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

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-AzVirtualNetworkGatewayConnection' cmdlet, with or without '-Debug'.

1. Use the following cmdlet example, configuring the values to match your own. If prompted, select 'A' in order to run 'All'. In the example, '-Name' refers to the name of the connection that you want to test.

```
Get-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

2. After the cmdlet has finished, view the values. In the example below, the connection status shows as 'Connected' and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

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. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

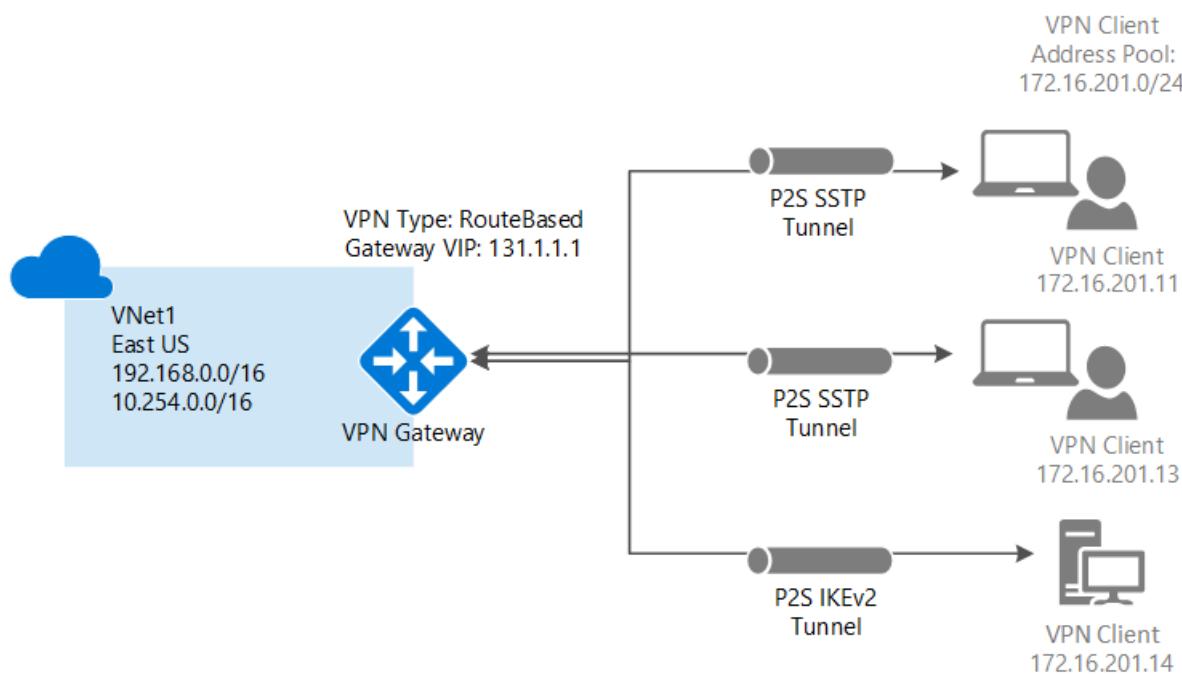
Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

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

1/10/2020 • 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
- **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. Sign in to the [Azure portal](#). On the Azure portal menu or from the **Home** page, and select **Create a resource**. The **New** page opens.
2. In **Search the marketplace**, enter *virtual network* and select **Virtual Network** from the results.

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:

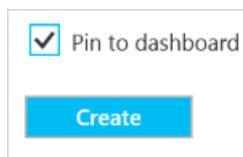
Name	VNet1				
Address space	192.168.0.0/16 192.168.0.0 - 192.168.255.255 (65536 addresses)				
Subscription	Windows Azure Internal Consumption				
Resource group	<input checked="" type="radio"/> Create new <input type="radio"/> Use existing TestRG				
Location	East US				
Subnet	<table border="1"> <tr> <td>Name</td> <td>FrontEnd</td> </tr> <tr> <td>Address range</td> <td>192.168.1.0/24 192.168.1.0 - 192.168.1.255 (256 addresses)</td> </tr> </table>	Name	FrontEnd	Address range	192.168.1.0/24 192.168.1.0 - 192.168.1.255 (256 addresses)
Name	FrontEnd				
Address range	192.168.1.0/24 192.168.1.0 - 192.168.1.255 (256 addresses)				

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**.



12. 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. Create a virtual network gateway

In this step, you create the virtual network gateway for your VNet. Creating a gateway can often take 45 minutes or more, depending on the selected gateway SKU.

The virtual network gateway uses specific subnet called the gateway subnet. The gateway subnet is part of the virtual network IP address range that you specify when configuring your virtual network. It contains the IP addresses that the virtual network gateway resources and services use.

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The number of IP addresses needed depends on the VPN gateway configuration that you want to create. Some configurations require more IP addresses than others. We recommend that you create a gateway subnet that uses a /27 or /28.

If you see an error that specifies that the address space overlaps with a subnet, or that the subnet is not contained within the address space for your virtual network, check your VNet address range. You may not have enough IP addresses available in the address range you created for your virtual network. For example, if your default subnet encompasses the entire address range, there are no IP addresses left to create additional subnets. You can either adjust your subnets within the existing address space to free up IP addresses, or specify an additional address range and create the gateway subnet there.

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**. This opens the **Create virtual network gateway** page.

Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)

Project details

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

* Subscription

Resource group [i](#)

Select a virtual network to get resource group

Instance details

* Name

 VNet1GW ✓

* Region

 (US) East US ✓

* Gateway type [i](#)

VPN ExpressRoute

* VPN type [i](#)

Route-based Policy-based

* SKU [i](#)

 VpnGw1 ✓

[i](#) Only virtual networks in the currently selected subscription and region are listed.

VIRTUAL NETWORK

* Virtual network [i](#)

 VNet1 ✓

Gateway subnet address range

10.1.255.0/27



Public IP address

* Public IP address [i](#)

Create new Use existing

* Public IP address name

 VNet1GWpip ✓

Public IP address SKU

Basic

* Assignment

Dynamic Static

* Enable active-active mode [i](#)

Enabled Disabled

* Configure BGP ASN [i](#)

Enabled Disabled

Azure recommends using a validated VPN device with your virtual network gateway. To view a list of validated devices and instructions for configuration, refer to Azure's [documentation](#) regarding validated VPN devices.

[Review + create](#)

< Previous

Next : Tags >

[Download a template for automation](#)

2. On the **Create virtual network gateway** page, fill in the values for your virtual network gateway.

Project details

- **Subscription:** Select the subscription you want to use from the dropdown.
- **Resource Group:** This setting is autofilled when you select your virtual network on this page.

Instance details

- **Name:** Name your gateway. Naming your gateway not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
- **Region:** Select the region in which you want to create this resource. The region for the gateway

must be the same as the virtual network.

- **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
- **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
- **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select. For more information about gateway SKUs, see [Gateway SKUs](#).

Virtual network: Choose the virtual network to which you want to add this gateway.

Gateway subnet address range: This field only appears if the virtual network you selected does not have a gateway subnet. Fill in the range if you don't already have a gateway subnet. If possible, make the range /27 or larger (/26,/25 etc.)

Public IP address: This setting specifies the public IP address object that gets associated to the VPN gateway. The public IP address is dynamically assigned to this object when the VPN gateway is created. The only time the Public IP address changes is when the gateway is deleted and re-created. It doesn't change across resizing, resetting, or other internal maintenance/upgrades of your VPN gateway.

- **Public IP address:** Leave **Create new** selected.
- **Public IP address name:** In the text box, type a name for your public IP address instance.
- **Assignment:** VPN gateway supports only Dynamic.

Active-Active mode: Only select **Enable active-active mode** if you are creating an active-active gateway configuration. Otherwise, leave this setting unselected.

Leave **Configure BGP ASN** deselected, unless your configuration specifically requires this setting. If you do require this setting, the default ASN is 65515, although this can be changed.

3. Click **Review + Create** to run validation. Once validation passes, click **Create** to deploy the VPN gateway. A gateway can take up to 45 minutes to fully create and deploy. You can see the deployment status on the Overview page for your gateway.

After the gateway is created, you can view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway appears as a connected device.

NOTE

The Basic gateway SKU does not support IKEv2 or RADIUS authentication. If you plan on having Mac clients connect to your virtual network, do not use the Basic SKU.

3. 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

Use either a root certificate that was generated with an enterprise solution (recommended), or generate a self-signed certificate. After you create the root certificate, export the public certificate data (not the private key) as a Base64 encoded X.509 .cer file. Then, upload the public certificate data to the Azure server.

- **Enterprise certificate:** If you're using an enterprise solution, you can use your existing certificate chain.

Acquire 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, create a self-signed root certificate. Otherwise, the certificates you create won't be compatible with your P2S connections and clients will receive a connection error when they try to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the following articles describe how to generate a compatible self-signed root 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. Although MakeCert is deprecated, you can still use it to generate certificates. Client certificates that you generate from the root certificate can be installed on any supported P2S client.
- [Linux instructions](#)

2. Generate a client certificate

Each client computer that you connect to a VNet with a Point-to-Site connection must have a client certificate installed. You generate it from the root certificate and install it on each client computer. If you don't install a valid client certificate, authentication will fail when the client tries to connect to the VNet.

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 use the same client certificate to authenticate and you revoke it, you'll need to generate and install new certificates for every client that uses that certificate.

You can generate client certificates by using the following methods:

- **Enterprise certificate:**
 - If you're using an enterprise certificate solution, generate a client certificate with the common name value format *name@yourdomain.com*. Use this format instead of the *domain name\username* format.
 - Make sure the client certificate is based on a user certificate template that has *Client Authentication* listed as the first item in the user list. Check the certificate by double-clicking it and viewing **Enhanced Key Usage** in the **Details** tab.
- **Self-signed root certificate:** Follow the steps in one of the following P2S certificate articles so that the client certificates you create will be compatible with your P2S connections. The steps in these articles generate a compatible client certificate:
 - [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. The generated certificates can be installed on any supported P2S client.
 - [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer for generating certificates. Although MakeCert is deprecated, you can still use it to generate certificates. You can install the generated certificates on any supported P2S client.
 - [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate, 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, export it as a .pfx file, along with the entire certificate chain. Doing so will create a .pfx file that contains the root certificate information required for the client to authenticate.

To export the certificate

For steps to export a certificate, see [Generate and export certificates for Point-to-Site using PowerShell](#).

4. 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. If you configure multiple protocols and SSTP is one of the protocols, then the configured address pool is split between the configured protocols equally.

- 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**.

A screenshot of the Azure portal showing the 'Point-to-site configuration' section. The left sidebar shows various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Connections, and Point-to-site configuration. The 'Point-to-site configuration' option is highlighted with a blue background.

- Click **Configure now** to open the configuration page.

A screenshot of the 'Point-to-site is not configured' configuration page. It features a large red 'Configure now' button at the bottom.

- 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. The minimum subnet mask is 29 bit for active/passive and 28 bit for active/active configuration. Click **Save** to validate and save the setting.

A screenshot of the 'Point-to-site configuration' settings page. The 'Address pool' field contains '172.16.0.0/24'. The 'Tunnel type' dropdown menu lists several options: OpenVPN (SSL), OpenVPN (SSL) (selected), SSTP (SSL), IKEv2, IKEv2 and OpenVPN (SSL), and IKEv2 and SSTP (SSL).

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.

5. Configure tunnel type

You can select the tunnel type. The tunnel options are OpenVPN, 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 use the OpenVPN client to connect to the OpenVPN tunnel type.

The screenshot shows a configuration interface for a tunnel type. At the top, there is a section labeled "Address pool" with the value "172.16.0.0/24". Below this is a section labeled "Tunnel type" containing a dropdown menu. The menu items are: "OpenVPN (SSL)" (which is highlighted with a blue background), "OpenVPN (SSL)", "SSTP (SSL)", "IKEv2", "IKEv2 and OpenVPN (SSL)", and "IKEv2 and SSTP (SSL)".

6. Configure authentication type

Select **Azure certificate**.

The screenshot shows a configuration interface for authentication type. It includes sections for "Address pool" (value: "172.16.0.0/24") and "Tunnel type" (value: "IKEv2 and SSTP (SSL)"). At the bottom, there is a section labeled "Authentication type" containing two radio buttons: one for "Azure certificate" (which is selected and highlighted with a red border) and one for "RADIUS authentication".

7. 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
MRVwFAYDVQQDDA1QM1NSb290Q2vydEWMB4XDTE3MDgwn2IxNTg0N1oDTE4MDgw
VzIyMTg0N1owGDEwMBQGA1UEAwNUJDUTm9vdENlcnQxMDCCASIwDQYKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANW4PjxpJKPnYHbToxn4+YEi78cP8HzIsZqvzqwv
JvgoeVhQ2lQnxweU127arHaZP9tja39ACOUg7XKC2qn3mdej42cdPzg7Hgpfe
mVZzUaUdaEuH1D9nqnxpsvCuCrRIuHyoT9Kyh9zWRDHQaL2/tatJb3fP7cxPj1
KSpvdvmSesZpwyPpNVBN3KAHuWGWK4eCX2kS9FRGte3iR9RjGo/Ueqj/I/pVmUN
aIE4AEKmmjD8Lg6rdqd+hlewy9u3fxZTPCwqqTE4TzL69JZmDzUip1lyV8qSL
nXbmLQPUXaKmNGjIvZ6Tk14xqc5+0z8pRa8qjIImzK93N10ECauEAAamMcBwOgYD
VR0PAQH/BAQDAgIEM80GAIuDgQWBREyrqYxzhULzGcfna3QbPoKSSTANBgkq
h1kIG9w0BAQsFAAOCAGEAf1qxeuzsxEU24p0rPyq899QfyFJHAZ3n3kawIxHTQ
+hu6tDoemScv9u+aYRRj8j2CrkDec6Seud3Daptw+PvTUEw7hQpihVpy1iihphL
FpyoUCqhK7X3lzYwazIAFP90/+Cn0WZ1b1RgagY7x4pViaghWhCvJVHTtB0fczX
oCk2j jpjehBec38KfhD1NxByJEFXkf/VaihuqOkPGV03L21oNVGLyuG7xb6b
1kQoKTCRTvHYA9w09vCERSmhHBC5jbaQ0jTm7jgSeciLC11KyMC7LRZQkC0NyB
+SPkthQa3ky0Keb3DGTdzgdr3Ic0Zuj6E1D1Ejhpg=#
-----END CERTIFICATE-----

```

- 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	
NAME	PUBLIC CERTIFICATE DATA
P2SRootCert	MIIc6zCCAdOgAwIBAgIQUvU0/H9T3qJGMbd6rc9zCTANBgkqhkiG9w0BAQsFADAY MRYwFAyDVQQDDA1QM1NSb290Q2\ ...

- Click **Save** at the top of the page to save all of the configuration settings.

Save (highlighted with a red box) | **Discard** | **Download VPN client**

Address pool
172.16.0.0/24

Tunnel type
IKEv2 and SSTP (SSL)

Authentication type
 Azure certificate RADIUS authentication

Root certificates

NAME
P2SRootCert

8. 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](#).

9. 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](#).

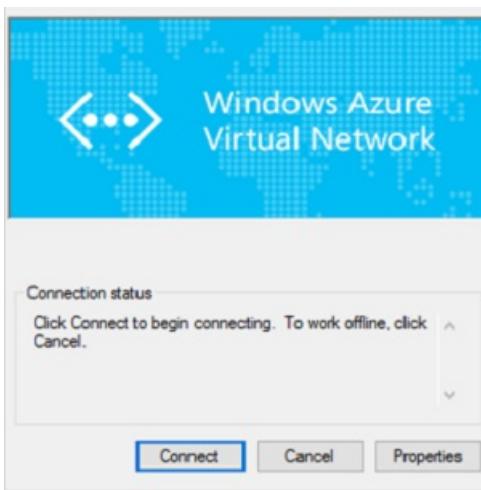
10. Connect to Azure

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 have trouble connecting, check the following items:

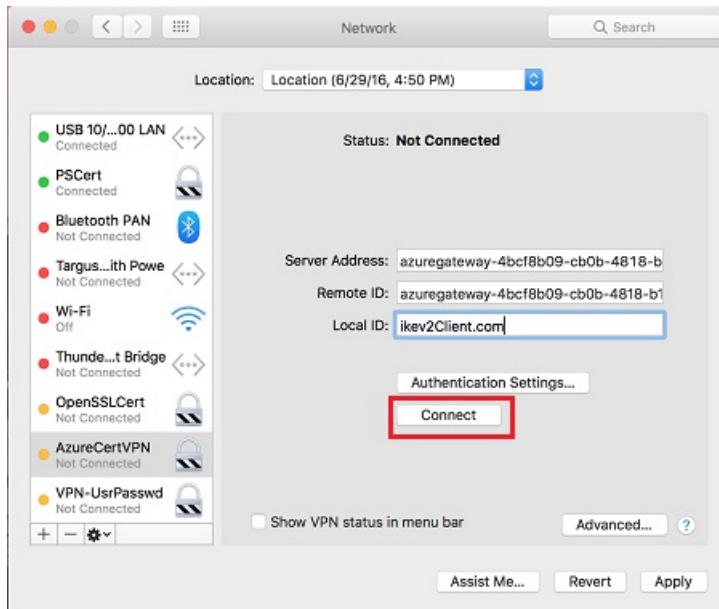
- If you exported a client certificate with **Certificate Export Wizard**, make sure that you exported it as a .pfx file and selected **Include all certificates in the certification path if possible**. When you export it with this value, the root certificate information is also exported. After you install the certificate on the client computer, the root certificate in the .pfx file is also installed. To verify that the root certificate is installed, open **Manage user certificates** and select **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed, which must be present for authentication to work.
- If you used a certificate that was issued by an Enterprise CA solution and you can't authenticate, verify the authentication order on the client certificate. Check the authentication list order by double-clicking the client certificate, selecting the **Details** tab, and then selecting **Enhanced Key Usage**. Make sure *Client Authentication* is the first item in the list. If it isn't, 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**.

Check [Install - Mac \(OS X\)](#) for detailed instructions. If you are having trouble connecting, verify that the virtual network gateway is not using a Basic SKU. Basic SKU is not supported for Mac clients.



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 group.

groups. You don't need to modify this example before using it.

```
$VMs = Get-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}
```

2. Verify that you are connected to your VNet using the Point-to-Site VPN connection.
3. Open **Remote Desktop Connection** by typing "RDP" or "Remote Desktop Connection" in the search box on the taskbar, then select Remote Desktop Connection. You can also open Remote Desktop Connection using the 'mstsc' command in PowerShell.
4. In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

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 VPNCIPAddressPool, 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?

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

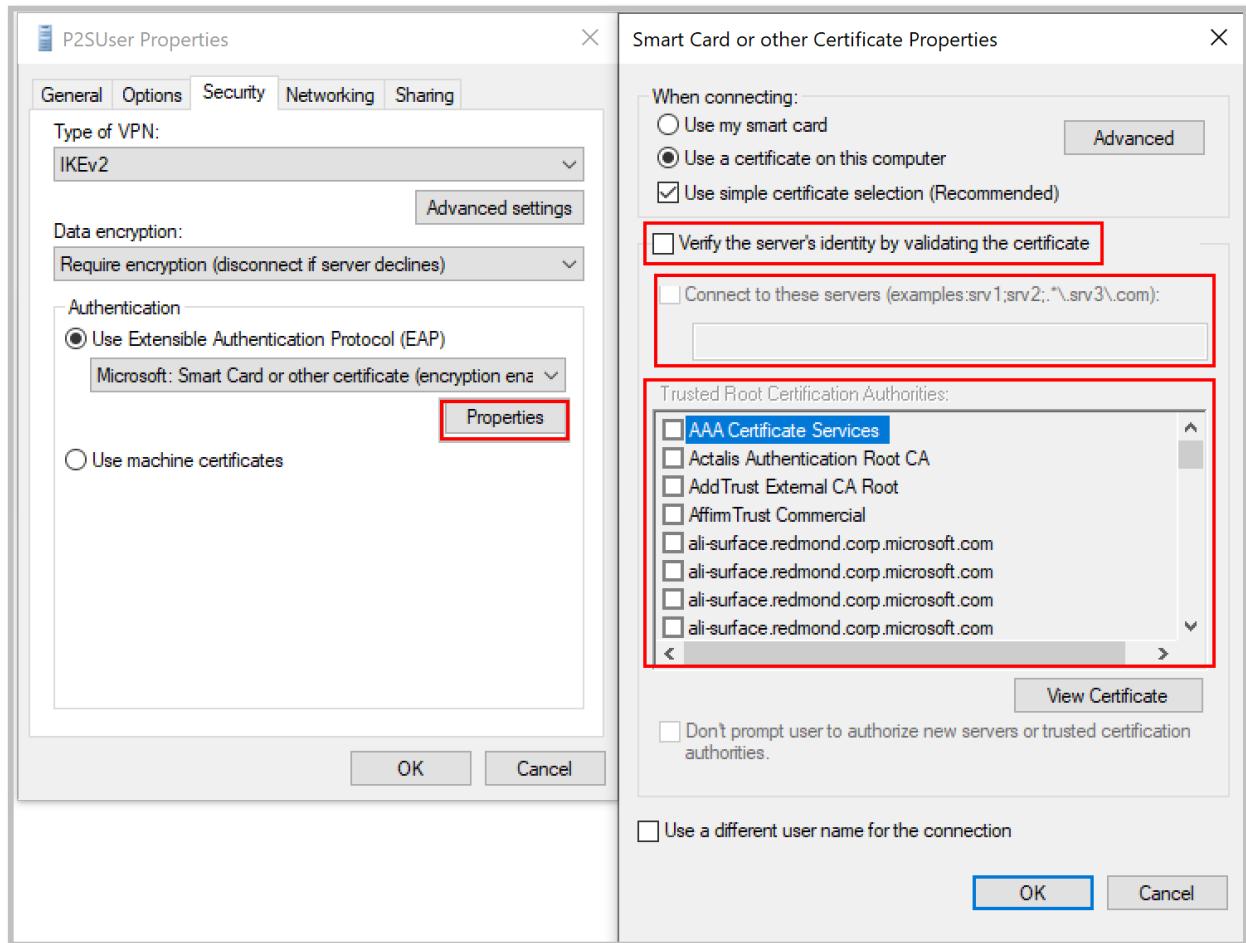
Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

What should I do if I'm getting a certificate mismatch when connecting using certificate authentication?

Uncheck "**Verify the server's identity by validating the certificate**" or **add the server FQDN along with the certificate** when creating a profile manually. You can do this by running **rasphone** from a command prompt and picking the profile from the drop-down list.

Bypassing server identity validation is not recommended in general, but with Azure certificate authentication, the same certificate is being used for server validation in the VPN tunneling protocol (IKEv2/SSTP) and the EAP protocol. Since the server certificate and FQDN is already validated by the VPN tunneling protocol, it is redundant to validate the same again in EAP.



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

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

Can I use certificates from Azure Key Vault?

No.

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

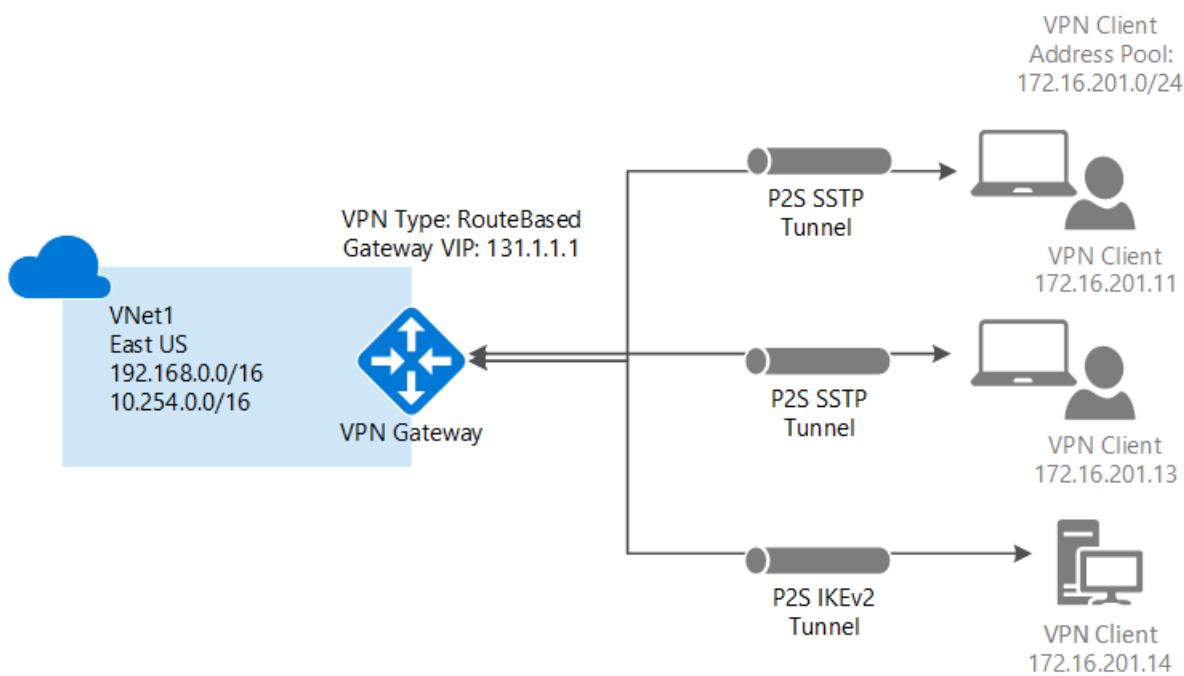
Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#). To understand more about networking and virtual machines, see [Azure and Linux VM network overview](#).

For P2S troubleshooting information, [Troubleshooting Azure point-to-site connections](#).

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

1/16/2020 • 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.

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](#).

Azure PowerShell

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

NOTE

Most of the steps in this article can use Azure Cloud Shell. However, to upload the root certificate public key, you must either use PowerShell locally, or the Azure portal.

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. Sign in and set variables

In this section, you sign 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.

Sign in

Open your PowerShell console with elevated privileges.

If you are running Azure PowerShell locally, connect to your Azure account. The *Connect-AzAccount* cmdlet prompts you for credentials. After authenticating, it downloads your account settings so that they are available to Azure PowerShell. If you are using Azure Cloud Shell instead, you do not need to run *Connect-AzAccount*. Azure Cloud Shell connects to your Azure account automatically.

```
Connect-AzAccount
```

If you have more than one subscription, get a list of your Azure subscriptions.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Name of subscription"
```

Declare variables

Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary. If you close your PowerShell/Cloud Shell session at any point during the exercise, just copy and paste the values again to re-declare the variables.

```
$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-AzResourceGroup -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-AzVirtualNetworkSubnetConfig -Name $FESubName -AddressPrefix $FESubPrefix  
$besub = New-AzVirtualNetworkSubnetConfig -Name $BESubName -AddressPrefix $BESubPrefix  
$gwsub = New-AzVirtualNetworkSubnetConfig -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-AzVirtualNetwork -Name $VNetName -ResourceGroupName $RG -Location $Location -AddressPrefix  
$VNetPrefix1,$VNetPrefix2 -Subnet $fesub, $besub, $gwsub -DnsServer 10.2.1.3
```

4. Specify the variables for the virtual network you created.

```
$vnet = Get-AzVirtualNetwork -Name $VNetName -ResourceGroupName $RG  
$subnet = Get-AzVirtualNetworkSubnetConfig -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-AzPublicIpAddress -Name $GWIPName -ResourceGroupName $RG -Location $Location -  
AllocationMethod Dynamic  
$ipconf = New-AzVirtualNetworkGatewayIpConfig -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 tunnel options are **OpenVPN**, **SSTP** and **IKEv2**. You can choose to enable one of them or any supported combination. If you want to enable multiple types, then specify the names separated by a comma. OpenVPN and SSTP cannot be enabled together. 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. You can use the OpenVPN client to connect to OpenVPN tunnel type.

- The virtual network gateway 'Basic' SKU does not support IKEv2, OpenVPN or RADIUS authentication. If you are planning on having Mac clients connect to your virtual network, do not use the Basic SKU.
- A VPN gateway can take up to 45 minutes to complete, depending on the [gateway sku](#) you select. This example uses IKEv2.

```
New-AzVirtualNetworkGateway -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-AzVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzVirtualNetworkGateway -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

Use either a root certificate that was generated with an enterprise solution (recommended), or generate a self-signed certificate. After you create the root certificate, export the public certificate data (not the private key) as a Base64 encoded X.509 .cer file. Then, upload the public certificate data to the Azure server.

- **Enterprise certificate:** If you're using an enterprise solution, you can use your existing certificate chain. Acquire 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, create a self-signed root certificate. Otherwise, the certificates you create won't be compatible with your P2S connections and clients will receive a connection error when they try to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the following articles describe how to generate a compatible self-signed root 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. Although MakeCert is deprecated, you can still use it to generate certificates.

Client certificates that you generate from the root certificate can be installed on any supported P2S client.

- [Linux instructions](#)

2. Generate a client certificate

Each client computer that you connect to a VNet with a Point-to-Site connection must have a client certificate installed. You generate it from the root certificate and install it on each client computer. If you don't install a valid client certificate, authentication will fail when the client tries to connect to the VNet.

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 use the same client certificate to authenticate and you revoke it, you'll need to generate and install new certificates for every client that uses that certificate.

You can generate client certificates by using the following methods:

- **Enterprise certificate:**

- If you're using an enterprise certificate solution, generate a client certificate with the common name value format *name@yourdomain.com*. Use this format instead of the *domain name\username* format.
- Make sure the client certificate is based on a user certificate template that has *Client Authentication* listed as the first item in the user list. Check the certificate by double-clicking it and viewing **Enhanced Key Usage** in the **Details** tab.

- **Self-signed root certificate:** Follow the steps in one of the following P2S certificate articles so that the client certificates you create will be compatible with your P2S connections. The steps in these articles generate a compatible client certificate:

- [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. The generated certificates can be installed on any supported P2S client.
- [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer for generating certificates. Although MakeCert is deprecated, you can still use it to generate certificates. You can install the generated certificates on any supported P2S client.
- [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate, 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, export it as a .pfx file, along with the entire certificate chain. Doing so will create a .pfx file that contains the root certificate information required for the client to authenticate.

To export the certificate

For steps to export a certificate, see [Generate and export certificates for Point-to-Site using PowerShell](#).

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.

NOTE

You can't upload the .cer file using Azure Cloud Shell. You can either use PowerShell locally on your computer, or you can use the [Azure portal steps](#).

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-AzVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $CertBase64
```

3. Upload the public key information to Azure. Once the certificate information is uploaded, Azure considers it to be a trusted root certificate. When uploading, make sure you are running PowerShell locally on your computer, or instead, you can use the [Azure portal steps](#). You can't upload using Azure Cloud Shell.

```
Add-AzVpnClientRootCertificate -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](#).

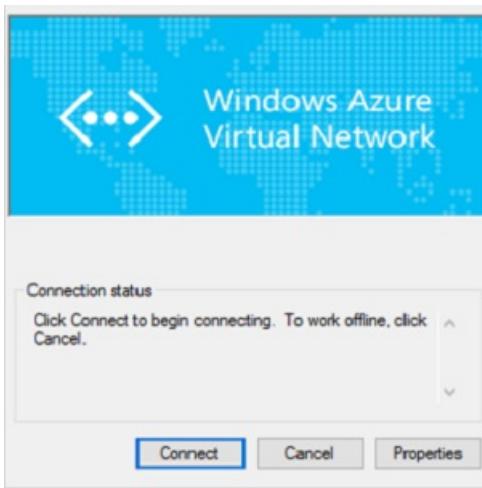
9. Connect to Azure

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.



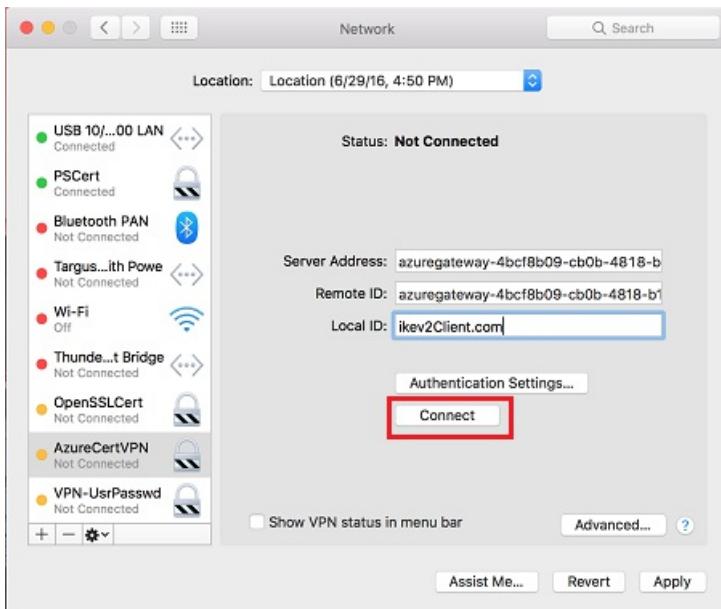
Troubleshooting Windows client P2S connections

If you have trouble connecting, check the following items:

- If you exported a client certificate with **Certificate Export Wizard**, make sure that you exported it as a .pfx file and selected **Include all certificates in the certification path if possible**. When you export it with this value, the root certificate information is also exported. After you install the certificate on the client computer, the root certificate in the .pfx file is also installed. To verify that the root certificate is installed, open **Manage user certificates** and select **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed, which must be present for authentication to work.
- If you used a certificate that was issued by an Enterprise CA solution and you can't authenticate, verify the authentication order on the client certificate. Check the authentication list order by double-clicking the client certificate, selecting the **Details** tab, and then selecting **Enhanced Key Usage**. Make sure *Client Authentication* is the first item in the list. If it isn't, 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**. Check [Install - Mac \(OS X\)](#) for detailed instructions. If you are having trouble connecting, verify that the virtual network gateway is not using a Basic SKU. Basic SKU is not supported for Mac clients.



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-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}

```

2. Verify that you are connected to your VNet using the Point-to-Site VPN connection.
3. Open **Remote Desktop Connection** by typing "RDP" or "Remote Desktop Connection" in the search box on the taskbar, then select Remote Desktop Connection. You can also open Remote Desktop Connection using the 'mstsc' command in PowerShell.
4. In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

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 method is the most efficient way to upload a root certificate. It requires Azure PowerShell cmdlets installed locally on your computer (not Azure Cloud Shell).

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-AzVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $certBase64_3
```

- Upload the file. You can only upload one file at a time.

```
Add-AzVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName -
VirtualNetworkGatewayName "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData $certBase64_3
```

- To verify that the certificate file uploaded:

```
Get-AzVpnClientRootCertificate -ResourceGroupName "TestRG" `
```

```
-VirtualNetworkGatewayName "VNet1GW"
```

Method 2 - Azure portal

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. It requires Azure PowerShell cmdlets installed locally on your computer (not Azure Cloud Shell).

- 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:



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/zCCAeugAwIBAgIQKazxzFjMkp9JRiX+tkTfSzAJBgUrDgMCHQUAMBgxFjAUBgNVBAMTDU15UDJTUm9vdENlcnQwHhcNMTUxMjE5MDI1MTIxWhcNMzkxMjMzMjM10TU5WjAYMRYwFAYDVQQDEw1NeVAyU1Jvb3RDZXJ0MIIBiGANBgkqhkiG9w0BAQEFAOCAQ8AMIIBCgKCAQEAYjIXoWY8xE/GF10SIvUaA0bxBjZ1PJfcXkMwsHPzvhWc2esOKrVQtgFgDz4ggAnOUFEkFaszjiHdnXv3mjzE2SpmAVIZPf2/yPwqkoHwkmrp6Bp0vNVOpKxaGPOuK8+dql1xcL0eCkt69g4lxy0FGRFkBcSJgVTViS9wjuuS7LPo5+OxyFkAY3pSDiMzQCKRGNEgw5WGMRDAiruDQF1ciLNjoAQCsDdLnI3pDYsvRW73HZEHm0qRRnJQe6VekvBYKlvnKaxUTKhFIYwymHB896nMFdRUKCZIiWRIy8Hc+sQEaML2EItAjQv4+fqgYiFdSwqnQCpF/7IZbotgQIDAQABo00wSzBJBgvNHQEEQjBAgBAkuVrWvFsCJAk5pb/eoCnRowGDEWMBQGA1UEAxMNTX1QM1NSb290Q2VydIIQKazxzFjMkp9JRiX+tkTfSzAJBgUrDgMCHQUAA4IBAQAA223veAZEiar9N12ubNH2+HwZASNzDVNsapkPKD97TFKH1PlIcS43TaYkTz38eVrwI6E0yDk4jAuPaKnPuPYFRj9w540SvY6Pd0UwDoEqpIcAvP+b4VYwxPL6oyEQ8wnOYuoAK1hh201Cbo8h9mMy9ofu+RP6HJ71TquplFxDID/XevI8tW6Dm+C/wCeV3EmI109KUob1D/e24zlo3Yz0tbyXwTIh34T0f0/zQvuBqZMcIPfM1cDvqcqiEFLWvWKoAnxbzckye2uk1gH052d8AVL3mGiX8wBJkjc/pMdxrEvvCzJkltBmqxTM6XjdJALuVh16qFlqgTWCICb7ju"
```

- Add the new root certificate. You can only add one certificate at a time.

```
Add-AzVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -  
VirtualNetworkGatewayname "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData  
$MyP2SCertPubKeyBase64_2
```

4. You can verify that the new certificate was added correctly by using the following example:

```
Get-AzVpnClientRootCertificate -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+tkTfSzAJBgUrDgMCHQUAMBgxFjAUBgNVBAMTDU15UDJTUm9vdENlcnQwHhcNMTUxMj  
E5MDI1MTIxWhcNMzkxMjMxMjM10TU5WjAYMRYwFAYDVQQDEw1NeVAyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIIBC  
gKCAQEAYjIXoWy8xE/GF10SIvUaA0bxBjZ1PJfcXkMwsHPzvhWc2esOKrVQtgFgDz4ggAnOUFEkFaszjiHdnXv3mjzE2SpmAVIZPf2/  
yPwqkoHwkmp6Bp0vNVOpKxaGPOuK8+dql1xcL0eCkt69g41xy0FGFRFkBcSIgVTViS9wjuuS7LPo5+OXgyFkAY3pSDiMzQCKrgNFgw5  
WGMHRDAiruDQF1ciLNQaQCsDdLnI3pDYsvRW73HZEHm0qRRnJQe6VekvBYKLvnKaxUTKhFIYwumHB96nMFdRUKCZIiWRIy8Hc+s  
QEaML2EItAjQv4+fgyIfdSwqnQCPf/7IZbotqQIDAQABo00wSzBJBgnVHQEEQjBAGBAkuVrWvFsCJAdK5pb/eoCNoRowGDEWMQBGA  
1UEAxMNTX1QM1NSb29Q02VydIIQKazxzFjMkp9JRIx+tkTfSzAJBgUrDgMCHQUAA4IBAQAA23veAZEiar9N12ubNH2+HwZASNzDVNs  
pkPKD97TXFKH1P1IcS43TaYkTz38eVrwI6E0yDk4jAuPaKnPuPYFRj9w540SvY6Pd0UwDoEqpIcAvp+b4VYwxPL6oyEQ8wnOYuoAK1h  
hh201Cbo8h9mMy9ofU+RP6HJ71TquplFxDID/XevI8tW6Dm+C/wCeV3EmI109KUob1D/e24z1o3Yz0tbyXwTIh34T0f0/zQvUuBqzMc  
IPfM1cDvcqciEFLWvWKoAnxbzckye2uk1gH052d8AVL3mGiX8wBJkjc/pMdxrEvvCzJkltBmqxtM6XjdJALuVh16qFlqgTWCicb7ju"
```

2. Remove the certificate.

```
Remove-AzVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -  
VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -PublicCertData $MyP2SCertPubKeyBase64_2
```

3. Use the following example to verify that the certificate was removed successfully.

```
Get-AzVpnClientRootCertificate -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-AzVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 ` 
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG ` 
-Thumbprint $RevokedThumbprint1
```

5. Verify that the thumbprint was added to the certificate revocation list.

```
Get-AzVpnClientRevokedCertificate -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-AzVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 ` 
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -Thumbprint $RevokedThumbprint1
```

3. Check if the thumbprint is removed from the revoked list.

```
Get-AzVpnClientRevokedCertificate -VirtualNetworkGatewayName $GWName -ResourceGroupName $RG
```

Point-to-Site FAQ

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

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.

- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

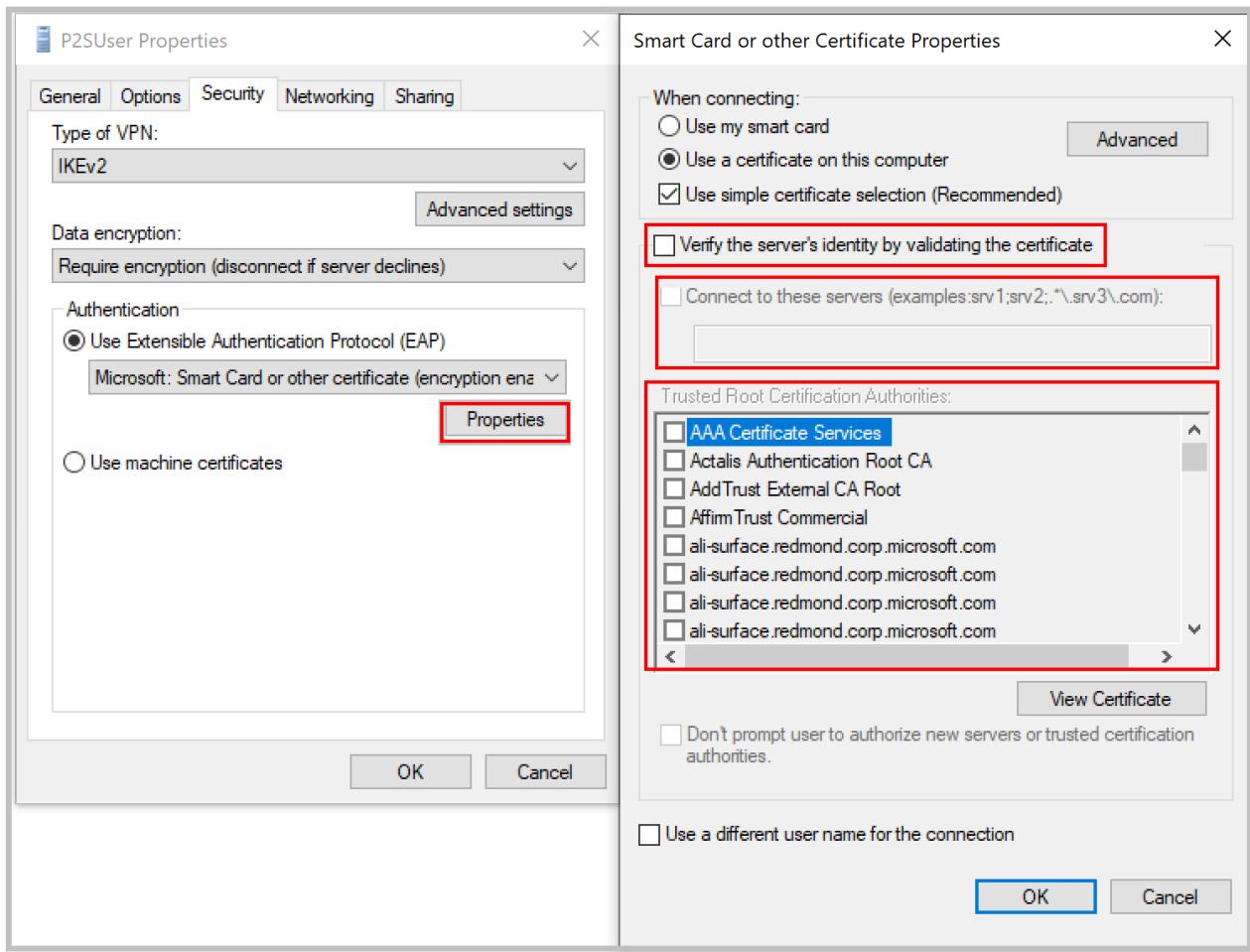
Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

What should I do if I'm getting a certificate mismatch when connecting using certificate authentication?

Uncheck "**Verify the server's identity by validating the certificate**" or **add the server FQDN along with the certificate** when creating a profile manually. You can do this by running **rasphone** from a command prompt and picking the profile from the drop-down list.

Bypassing server identity validation is not recommended in general, but with Azure certificate authentication, the same certificate is being used for server validation in the VPN tunneling protocol (IKEv2/SSTP) and the EAP protocol. Since the server certificate and FQDN is already validated by the VPN tunneling protocol, it is redundant to validate the same again in EAP.



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

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

Can I use certificates from Azure Key Vault?

No.

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

11/17/2019 • 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.

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. These examples do not work in the Azure Cloud Shell "Try It". You must run these examples locally.
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
```

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

Use this example if you have not closed your PowerShell console after creating the self-signed root certificate. This example continues from the previous section and uses the declared '\$cert' variable. 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.

```

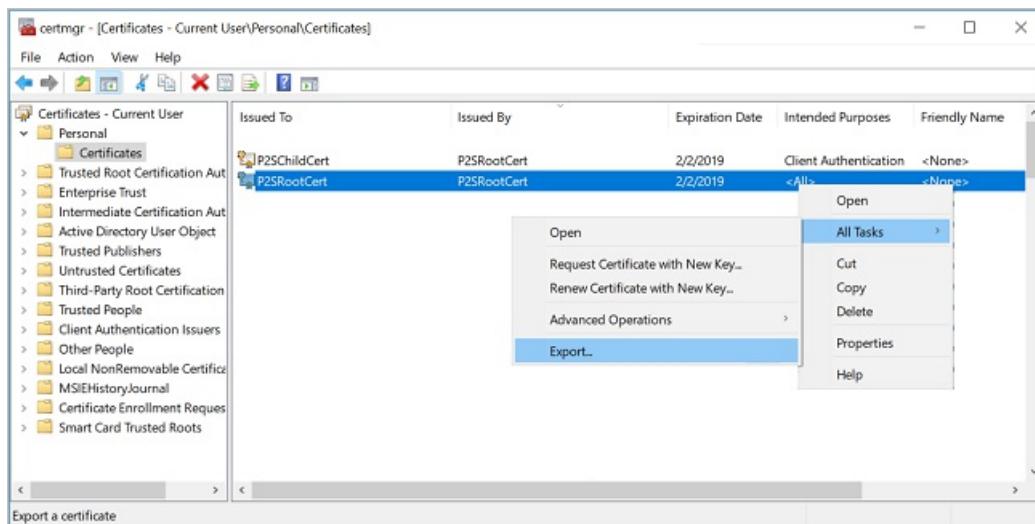
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec Signature ` 
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable ` 
-HashAlgorithm sha256 -KeyLength 2048 ` 
-CertStoreLocation "Cert:\CurrentUser\My" ` 
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")

```

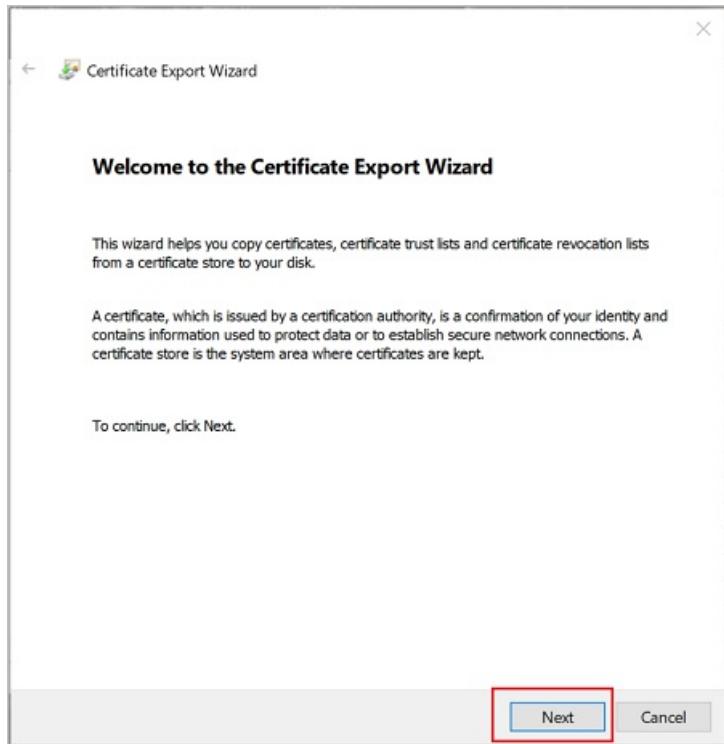
Export the root certificate public key (.cer)

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:

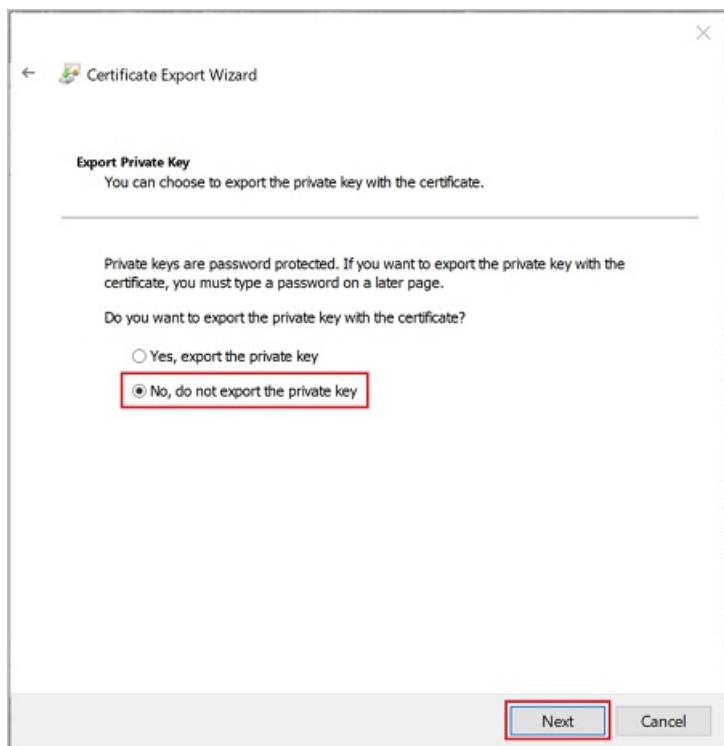
1. To obtain a .cer file from the certificate, open **Manage user certificates**. Locate the self-signed root certificate, typically in 'Certificates - Current User\Personal\Certificates', and right-click. Click **All Tasks**, and then click **Export**. This opens the **Certificate Export Wizard**. If you can't find the certificate under Current User\Personal\Certificates, you may have accidentally opened "Certificates - Local Computer", rather than "Certificates - Current User"). If you want to open Certificate Manager in current user scope using PowerShell, you type *certmgr* in the console window.



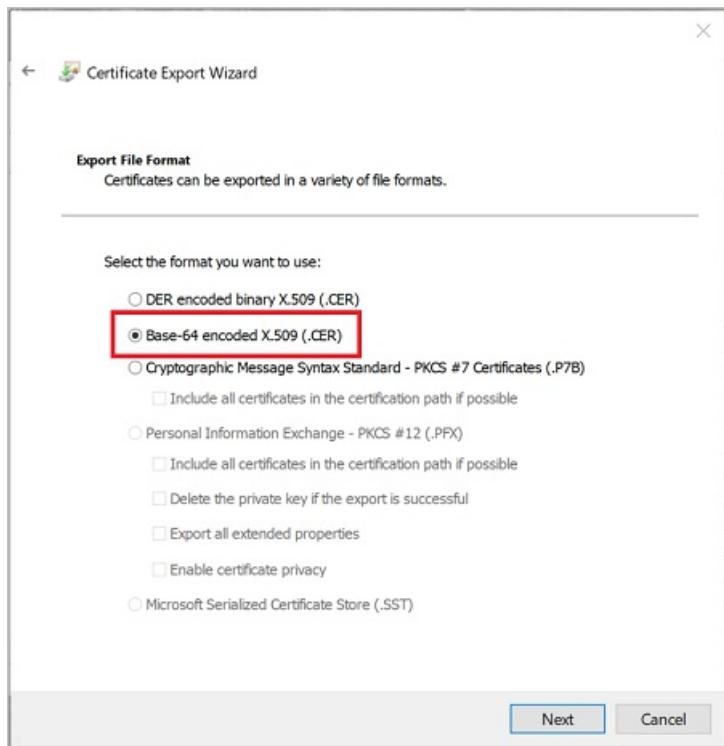
2. In the Wizard, click **Next**.



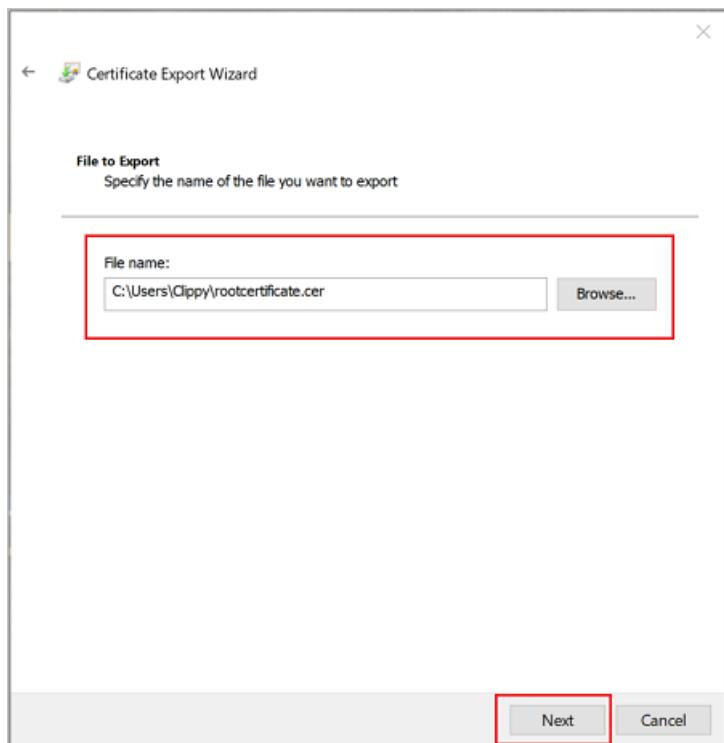
3. Select **No, do not export the private key**, and then click **Next**.



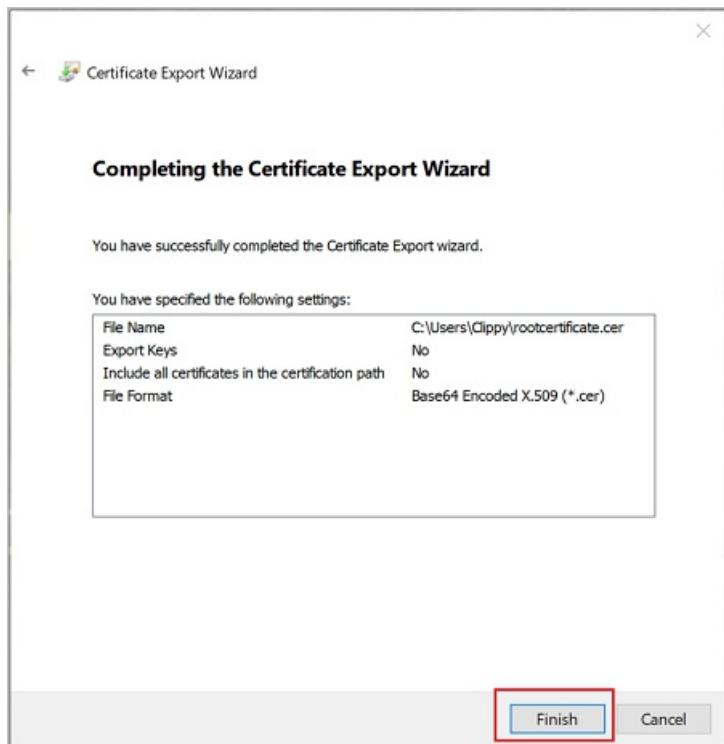
4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**, and then click **Next**.



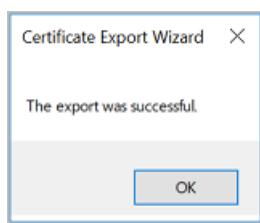
5. For **File to Export**, **Browse** to the location to which you want to export the certificate. For **File name**, name the certificate file. Then, click **Next**.



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/F6l+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+BgzxEGB8v4oL1Qzzbvj+xAb1WKpYqXFbp/ulhd6Ao2qn9sIVuKRkJjBgJ78
o45M2omAFZZaAVQrUa/fprKr3es/6IYrPT8J
-----END CERTIFICATE-----

```

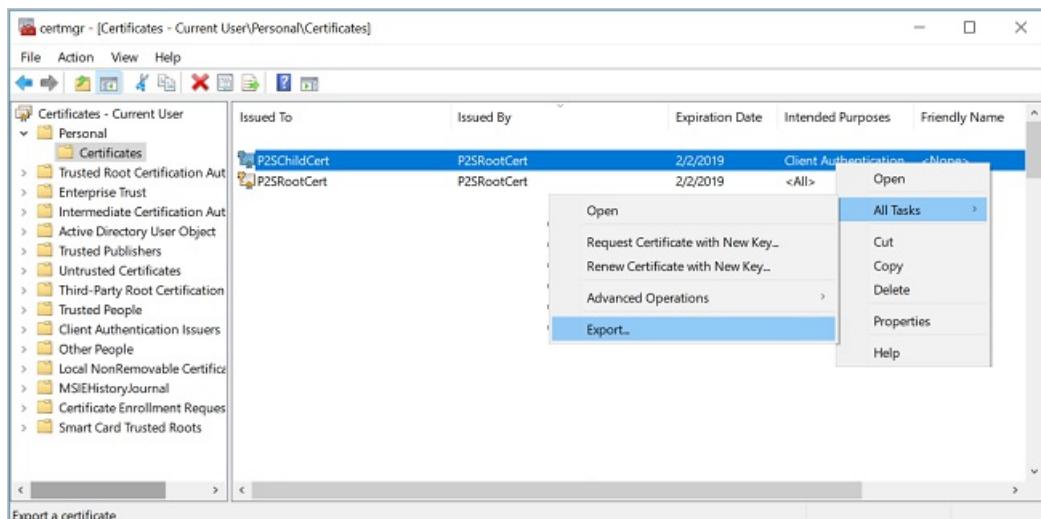
Export the self-signed root certificate and private key to store it (optional)

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](#).

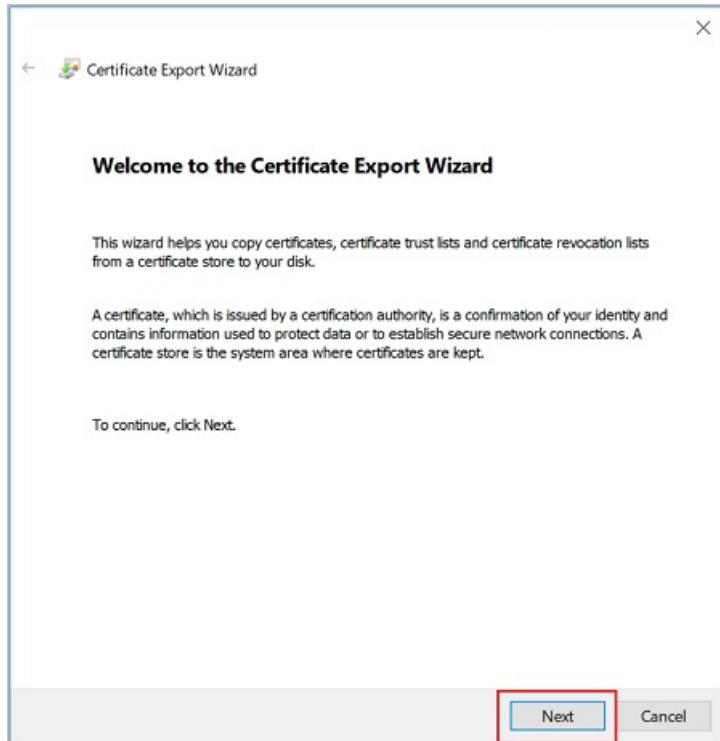
Export the client certificate

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.

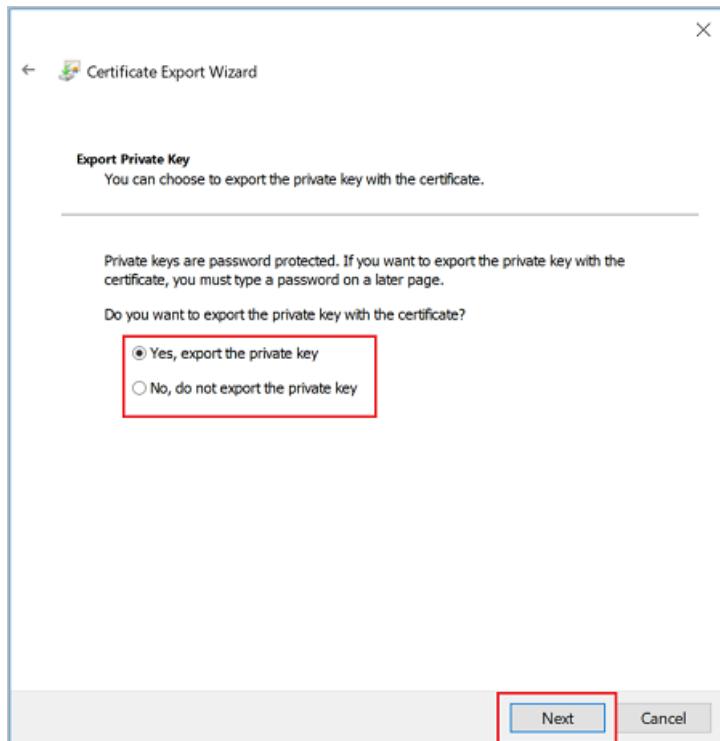
- To export a client certificate, open **Manage user certificates**. The client certificates that you generated are, by default, located in 'Certificates - Current User\Personal\Certificates'. Right-click the client certificate that you want to export, click **all tasks**, and then click **Export** to open the **Certificate Export Wizard**.



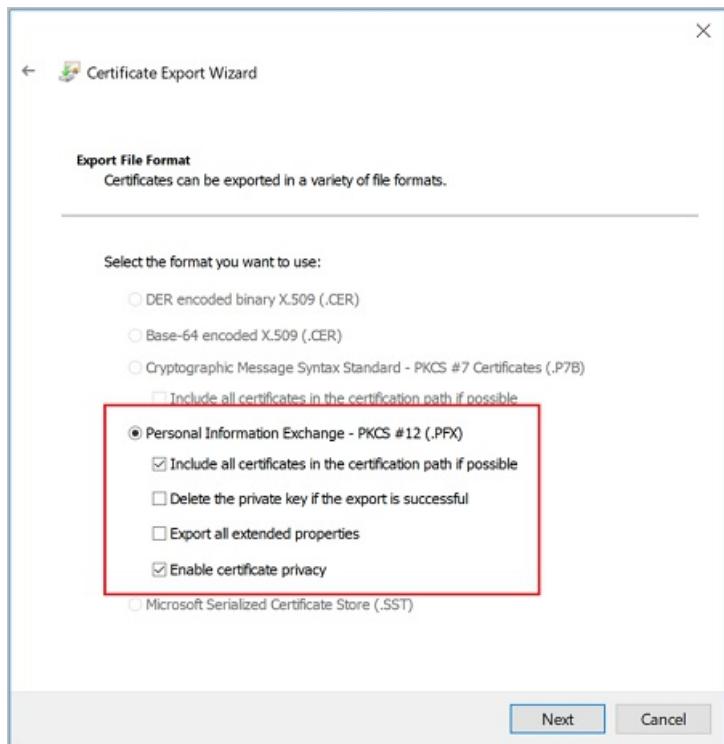
2. In the Certificate Export Wizard, click **Next** to continue.



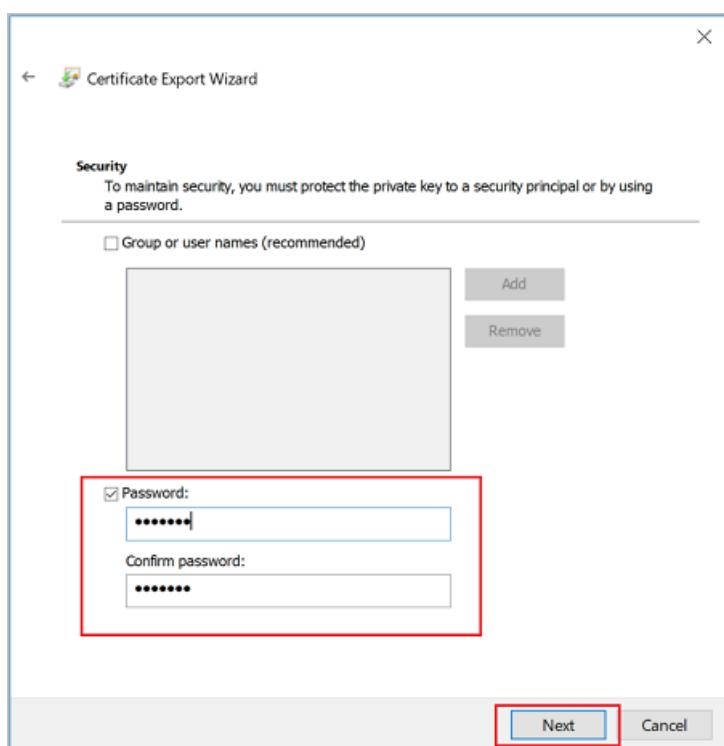
3. Select **Yes, export the private key**, and then click **Next**.



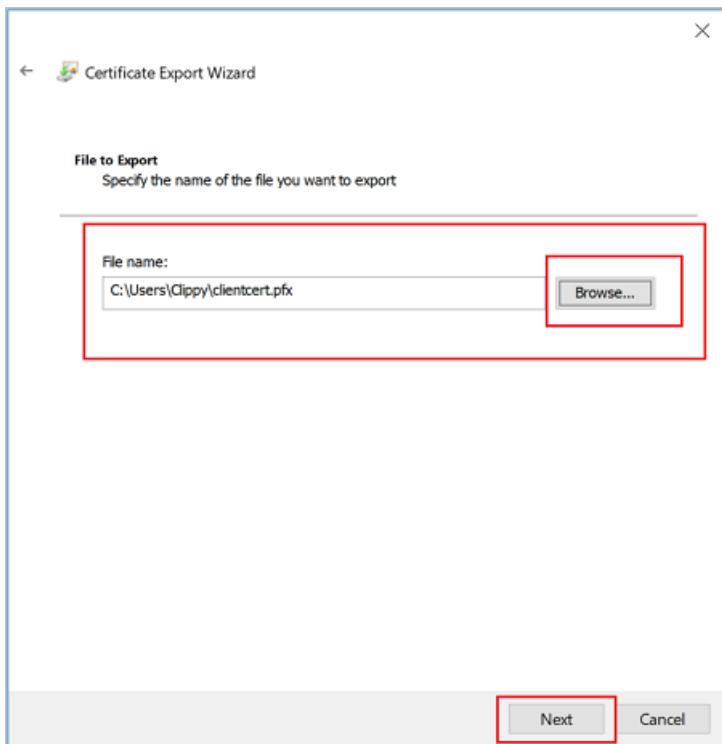
4. On the **Export File Format** page, leave the defaults selected. Make sure that **Include all certificates in the certification path if possible** is selected. This setting additionally exports the root certificate information that is required for successful client authentication. Without it, client authentication fails because the client doesn't have the trusted root certificate. Then, click **Next**.



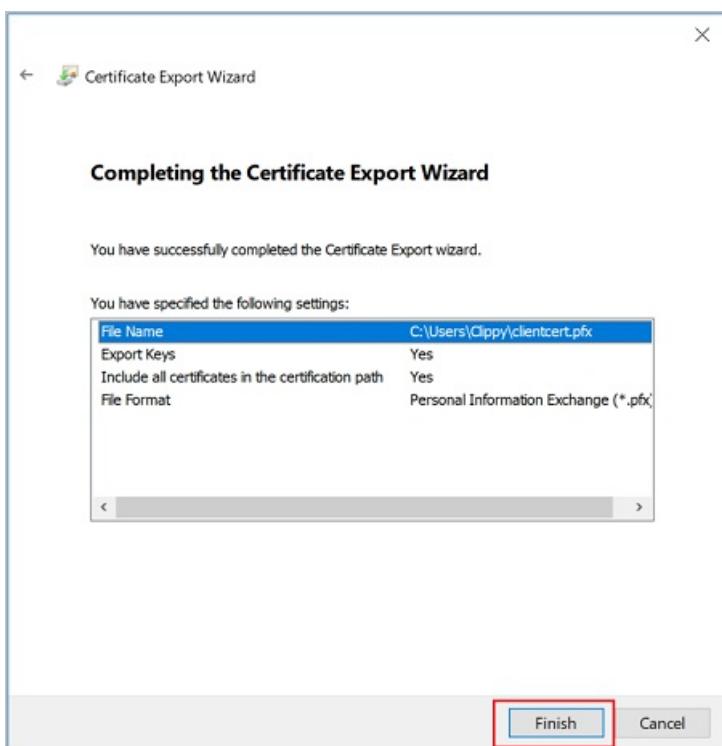
5. On the **Security** page, you must protect the private key. If you select to use a password, make sure to record or remember the password that you set for this certificate. Then, click **Next**.



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.



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](#).

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 \(classic\)](#).

Generate and export certificates for Point-to-Site connections using MakeCert

1/9/2020 • 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
```

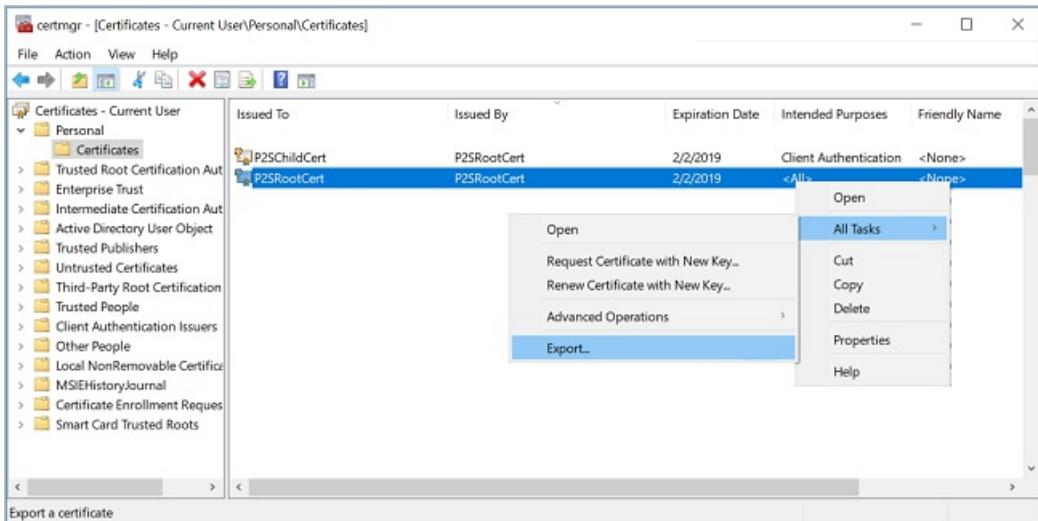
3. Create and install a certificate in the Personal certificate store on your computer. The following example creates a corresponding .cer file that you upload to Azure when configuring P2S. Replace 'P2SRootCert' and 'P2SRootCert.cer' with the name that you want to use for the certificate. The certificate is located in your 'Certificates - Current User\Personal\Certificates'.

```
makecert -sky exchange -r -n "CN=P2SRootCert" -pe -a sha256 -len 2048 -ss My
```

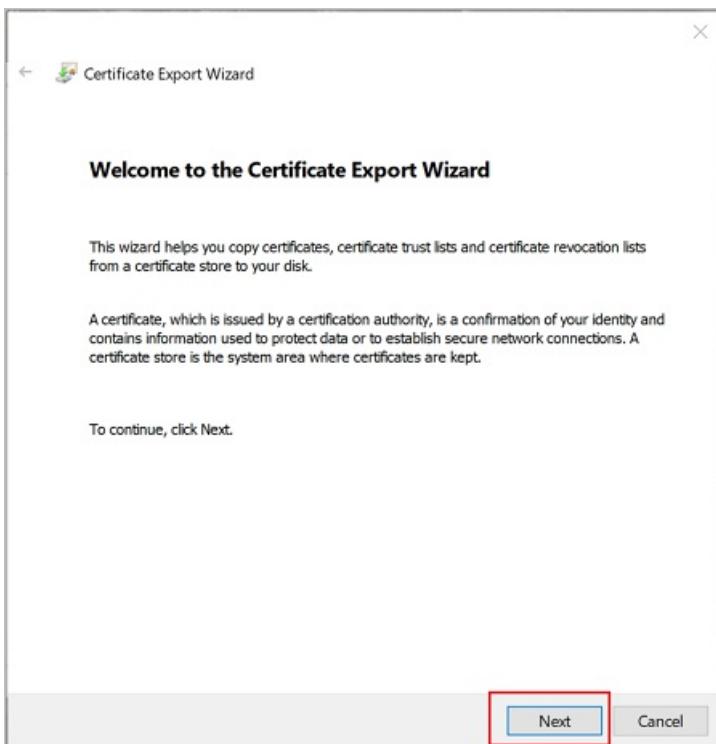
Export the public key (.cer)

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:

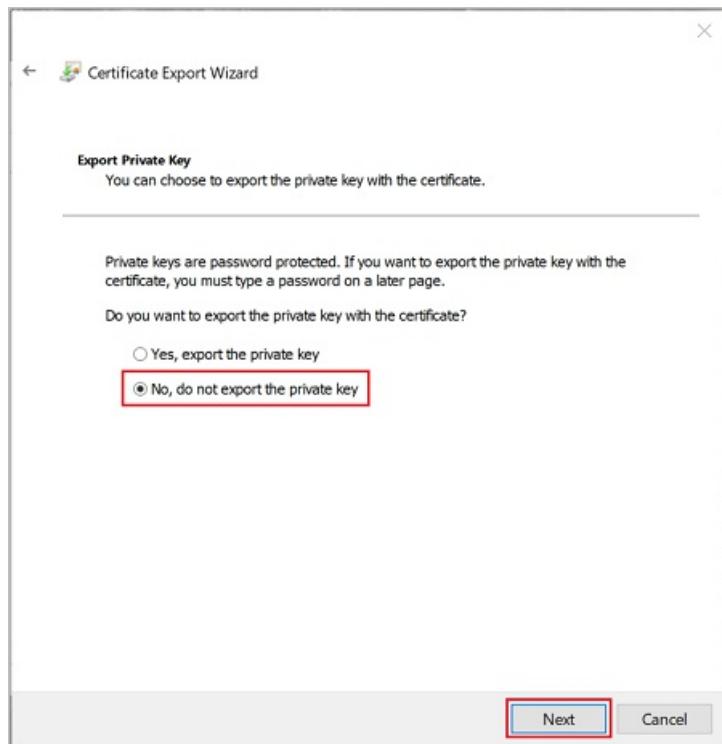
1. To obtain a .cer file from the certificate, open **Manage user certificates**. Locate the self-signed root certificate, typically in 'Certificates - Current User\Personal\Certificates', and right-click. Click **All Tasks**, and then click **Export**. This opens the **Certificate Export Wizard**. If you can't find the certificate under Current User\Personal\Certificates, you may have accidentally opened "Certificates - Local Computer", rather than "Certificates - Current User"). If you want to open Certificate Manager in current user scope using PowerShell, you type certmgr in the console window.



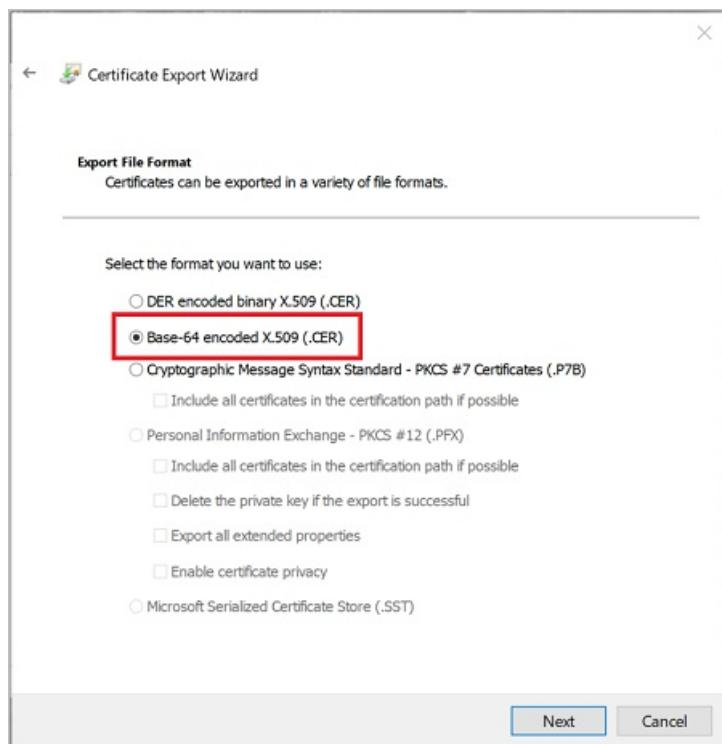
2. In the Wizard, click **Next**.



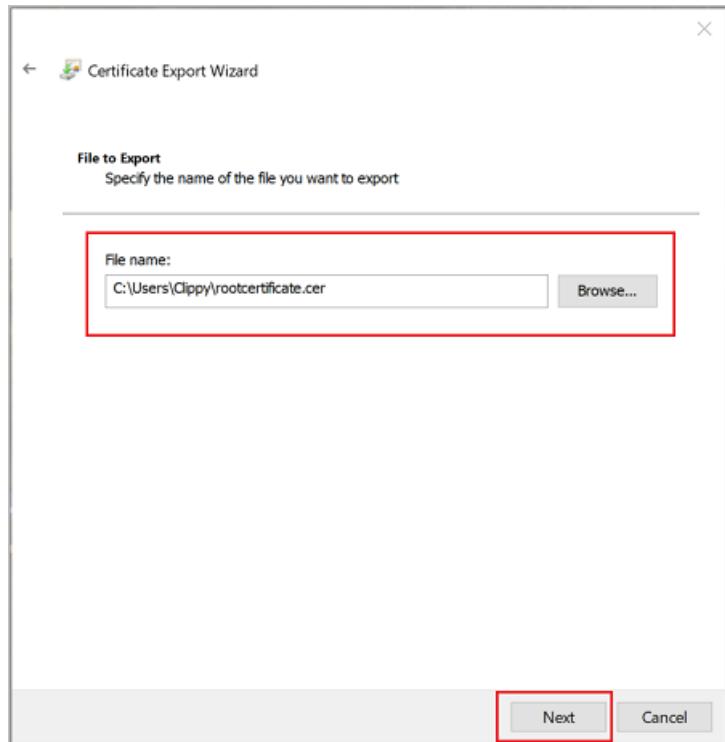
3. Select **No, do not export the private key**, and then click **Next**.



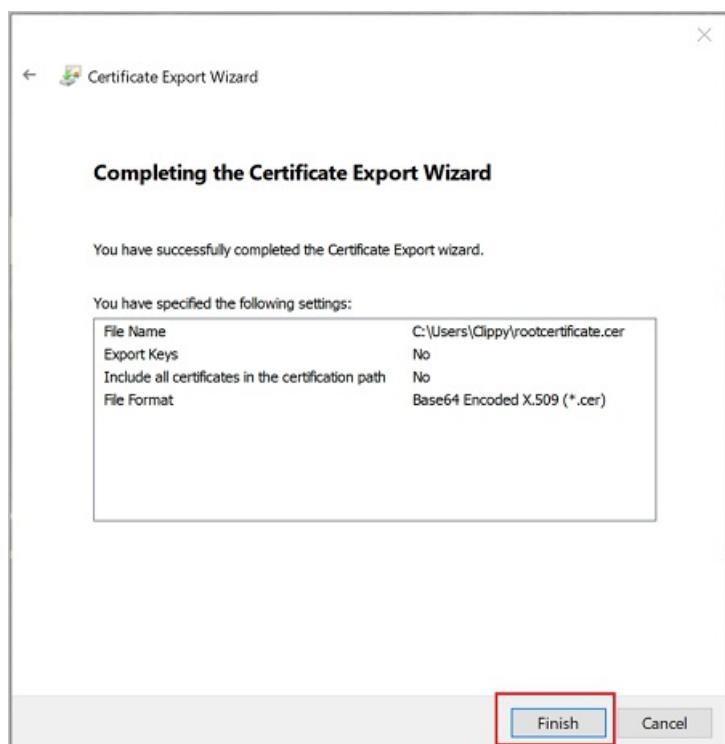
4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**, and then click **Next**.



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.

A screenshot of a Windows Notepad window titled "rootcertificate.cer - Notepad". The window shows the certificate content in plain text. The text starts with "-----BEGIN CERTIFICATE-----" and ends with "-----END CERTIFICATE-----". The main body of the text is highlighted in blue, representing the base-64 encoded certificate data.

```
-----BEGIN CERTIFICATE-----
MIIC5zCCAc+gAwIBAgIQRtNVXFGwtIhK4R7RaOak0jANBgkqhkiG9w0BAQsFADAW
MRQwEgYDVQQDDATQM1NSb290Q2VydDAeFw0xODAyMDIxOTM5MThaFw0xOTAyMDIx
OTU5MThaMBYxFDASBgvNBAMMC1AyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEAvrvt56dNhYGxf01/NUS2GcCCQ9mSKQeMCsoMZgyEC8nP
1ZioGeFwd23Thb9+k00k08sPbM4Jm42riyNsmtVNyCviP5pC/V2NyQr/F6l+K5X
0GurFxSm/mv6wf0xf/FHvu5PojX7Z5/oEbeYB11GVVPgq6QWSrx331W1zmD2FeuA
QE46dMPSPHFhnc6P2hfthzs3+tv1R4dg02wr5drVNvr1cVOHqoSbf/dqjV2thzz
ZSSN5kew2G/3H7Mc2ScZD+AYXWRzuiIrKrJSZiuIfJxLpQJaLQTByEU+wT8R8Rnq
GKmyKuUCPoXGYY3TRPmBXgA0800RsKFrXPWp5SiuuQIDAQABozEwLzAOBgNVHQ8B
Af8EBAMCAgQwHQYDVR0OBByEFCk6GjsuuTSfxMNz4ATy77DEbyCwMA0GCSqGSIb3
DQEBCwUA4IBAQBzzQCC4SEXqDgR2BL3uj17XD0scR/52U9rVxLorW1Z4Wu4kRA0
EA4IppBNmQep9eaCCqNfc6sbXf4QWjkBv1TBja20rFzt5cTAkwYG6Y0aWT1L//fw
u9goi2RihBs6IeBwc621u1Lo0Lw5htCoV0XPSS01mAm5R6//IqyHZRcAK/TffitC
EIYTFcKdavxe9CgY/TtzEMCS7gLARDpHh/nDrxtIeKxlvvUfnOoeXeaSsQwHtumq
GFH3+BgzxEGB8v4oLlQzzbj+j+xb1WKpYqXFbp/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.

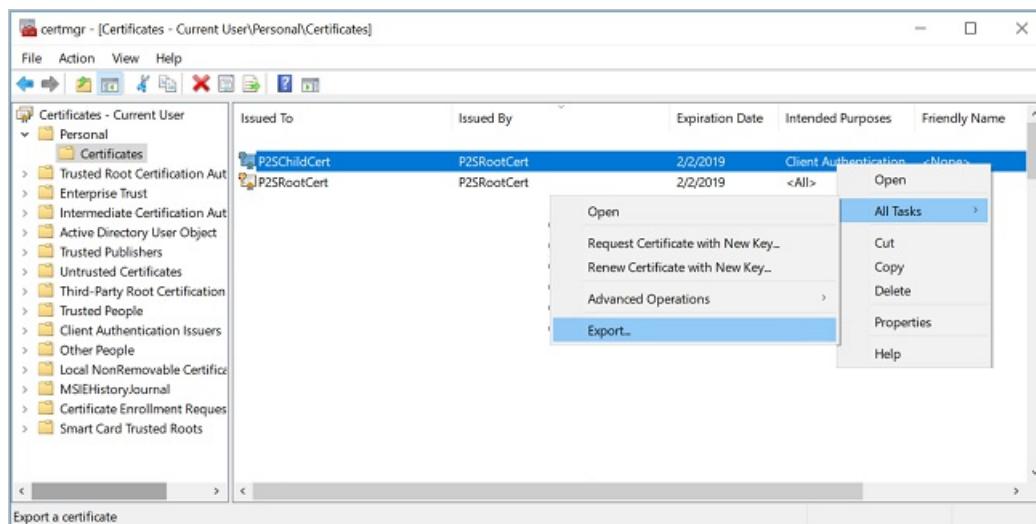
If you run the following example without modifying it, the result is a client certificate named P2SChildcert in your Personal certificate store that was generated from root certificate P2SRootCert.

```
makecert.exe -n "CN=P2SChildCert" -pe -sky exchange -m 96 -ss My -in "P2SRootCert" -is my -a sha256
```

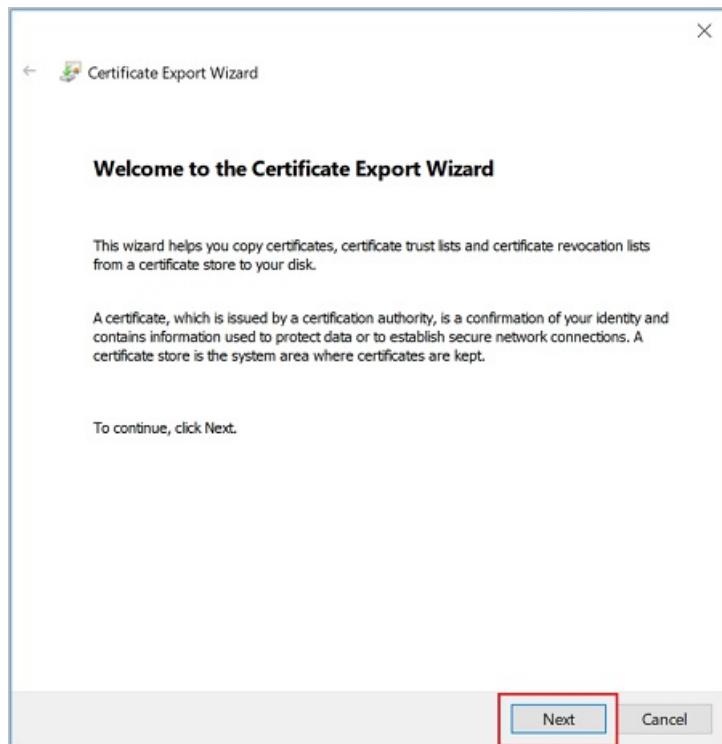
Export a client certificate

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.

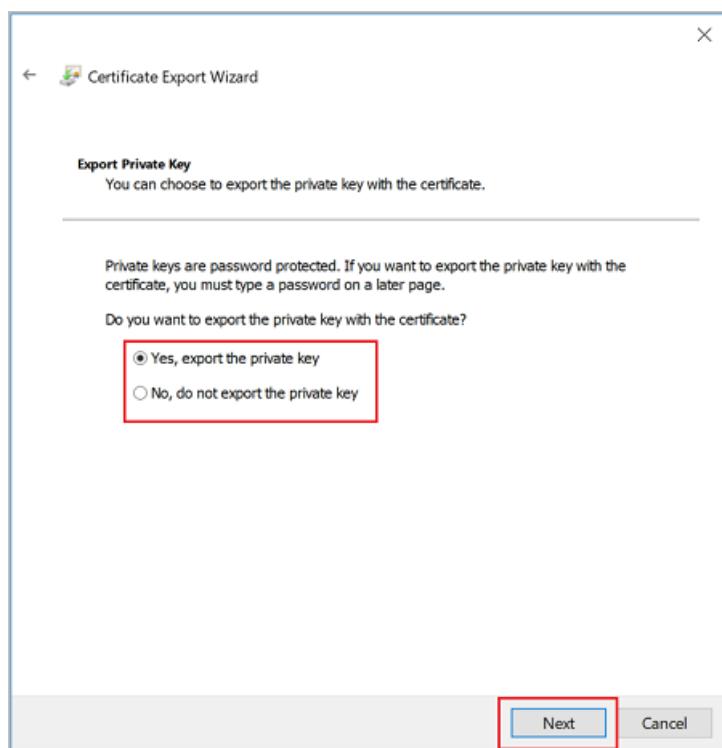
1. To export a client certificate, open **Manage user certificates**. The client certificates that you generated are, by default, located in 'Certificates - Current User\Personal\Certificates'. Right-click the client certificate that you want to export, click **all tasks**, and then click **Export** to open the **Certificate Export Wizard**.



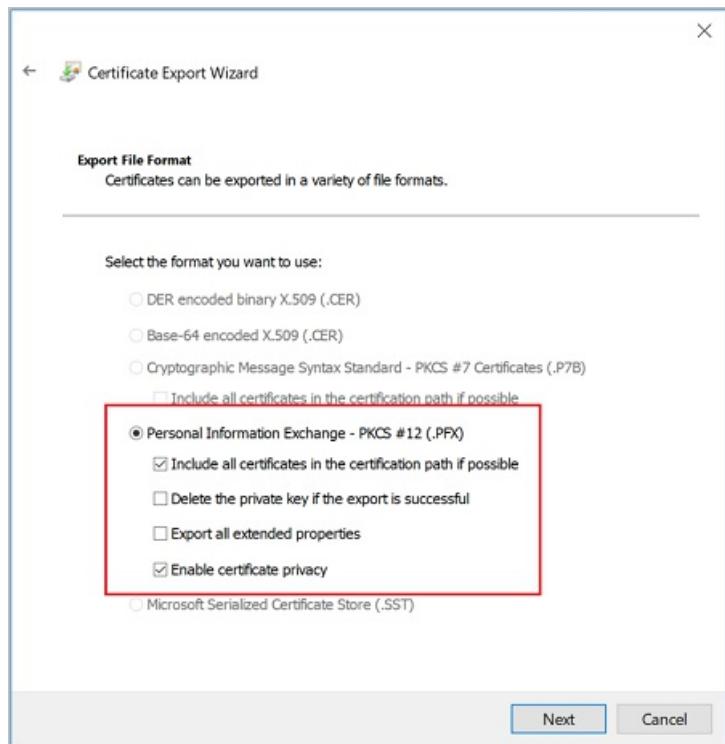
2. In the Certificate Export Wizard, click **Next** to continue.



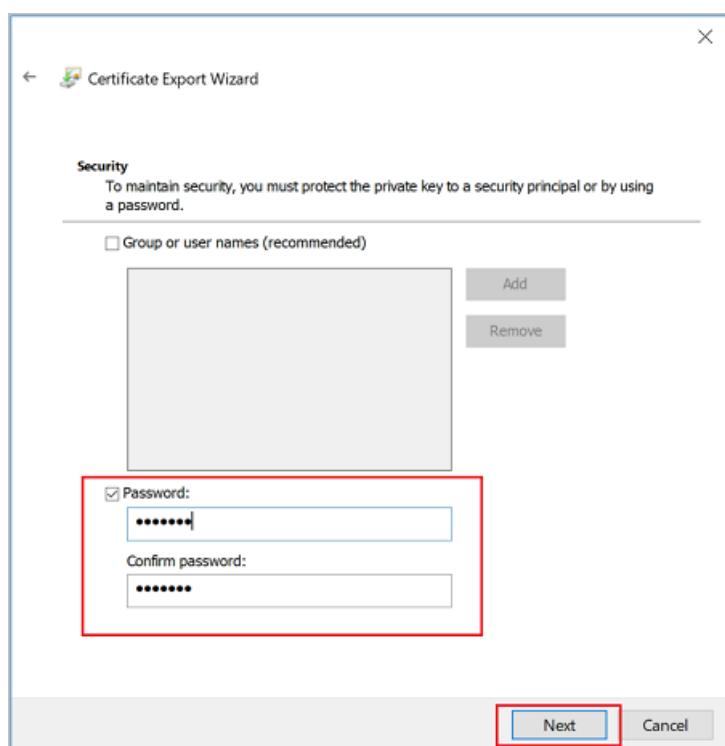
3. Select **Yes, export the private key**, and then click **Next**.



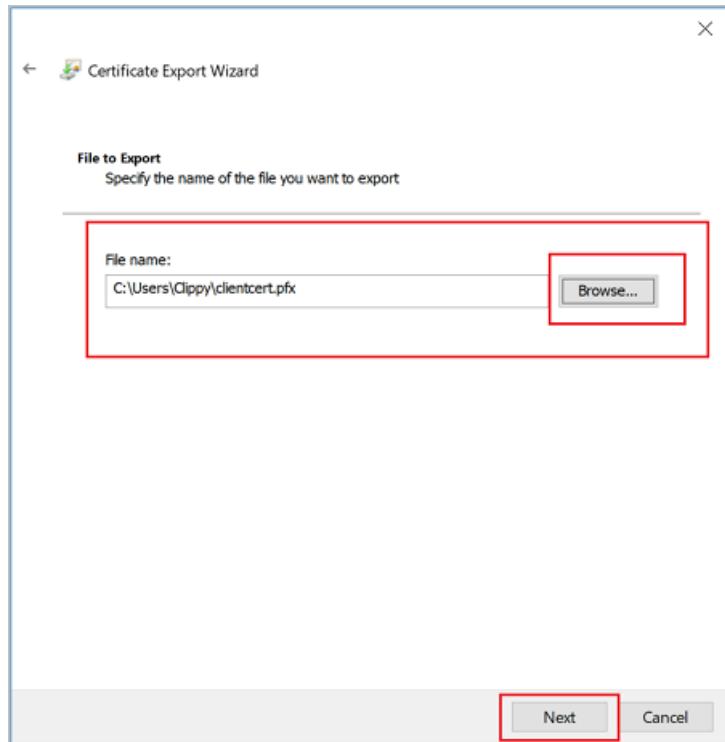
4. On the **Export File Format** page, leave the defaults selected. Make sure that **Include all certificates in the certification path if possible** is selected. This setting additionally exports the root certificate information that is required for successful client authentication. Without it, client authentication fails because the client doesn't have the trusted root certificate. Then, click **Next**.



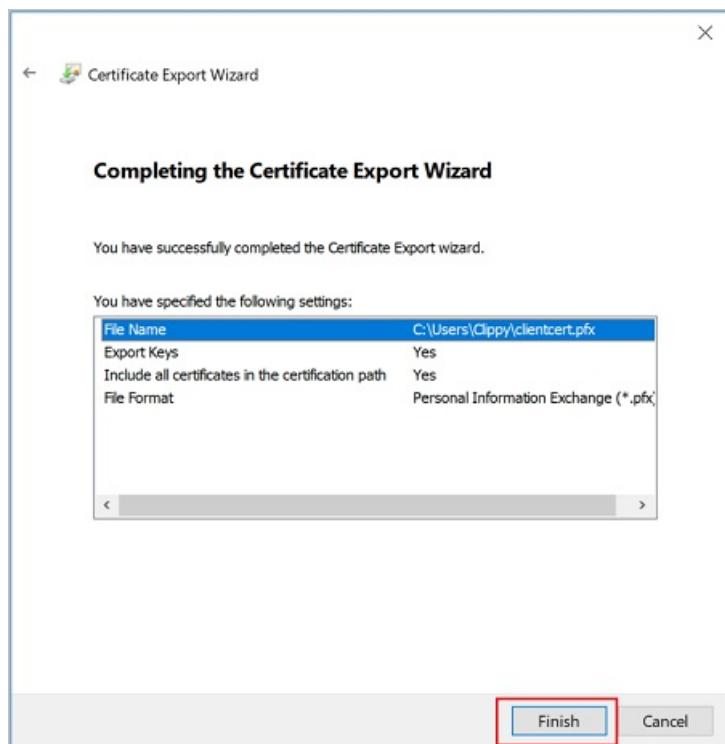
5. On the **Security** page, you must protect the private key. If you select to use a password, make sure to record or remember the password that you set for this certificate. Then, click **Next**.



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

1/9/2020 • 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. For information about how to install strongSwan using the GUI instead of CLI, see the steps in the [Client configuration](#) article.

Install strongSwan

The following configuration was used for the steps below:

Computer	Ubuntu Server 18.04
Dependencies	strongSwan

Use the following commands to install the required strongSwan configuration:

```
sudo apt install strongswan
```

```
sudo apt install strongswan-pki
```

```
sudo apt install libstrongswan-extra-plugins
```

Use the following command to install the Azure command-line interface:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

[Additional instructions on how to install the Azure CLI](#)

Generate and export certificates

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
```

Print the CA certificate in base64 format. This is the format that is supported by Azure. You upload this certificate to Azure as part of the [P2S configuration steps](#).

```
openssl x509 -in caCert.pem -outform der | base64 -w0 ; echo
```

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"
```

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

1/11/2020 • 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.

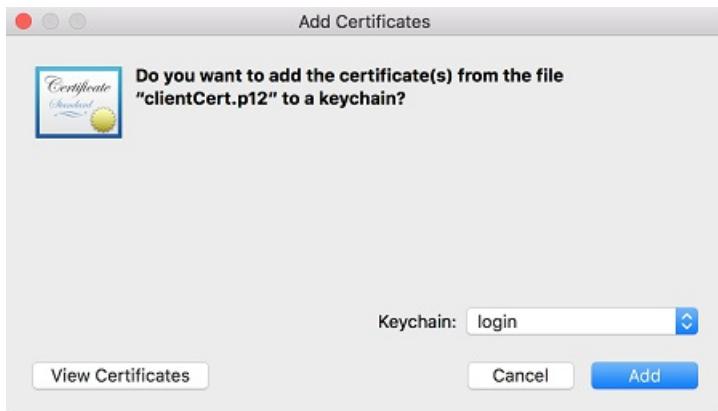
Mac

NOTE

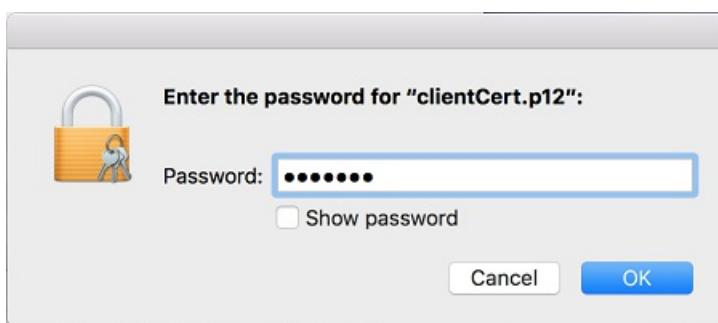
Mac VPN clients are supported for the Resource Manager deployment model only. They are not supported for the classic deployment model.

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, the 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

2/11/2020 • 9 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.

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.

- For more information about Point-to-Site connections, see [About Point-to-Site VPN](#).
- For OpenVPN instructions, see [Configure OpenVPN for P2S](#) and [Configure OpenVPN clients](#).

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.

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-AzVpnClientConfiguration -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)

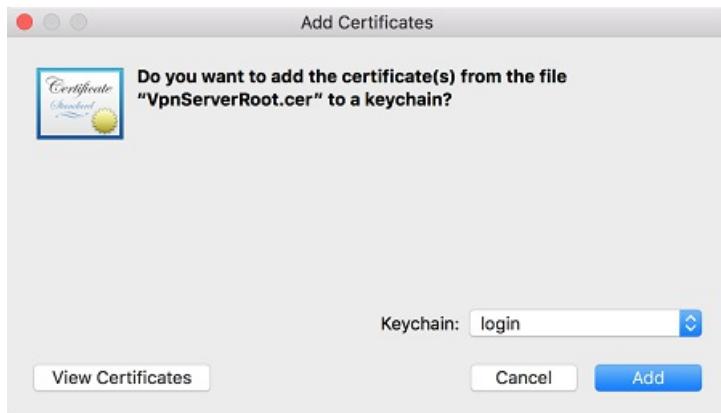
You have to manually configure the native IKEv2 VPN client on every Mac that will connect to Azure. Azure does not provide mobileconfig file for native Azure certificate authentication. The **Generic** contains all of the information that you need for configuration. If you don't see the Generic folder in your download, it's likely that IKEv2 was not selected as a tunnel type. Note that the VPN gateway Basic SKU does not support IKEv2. 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.

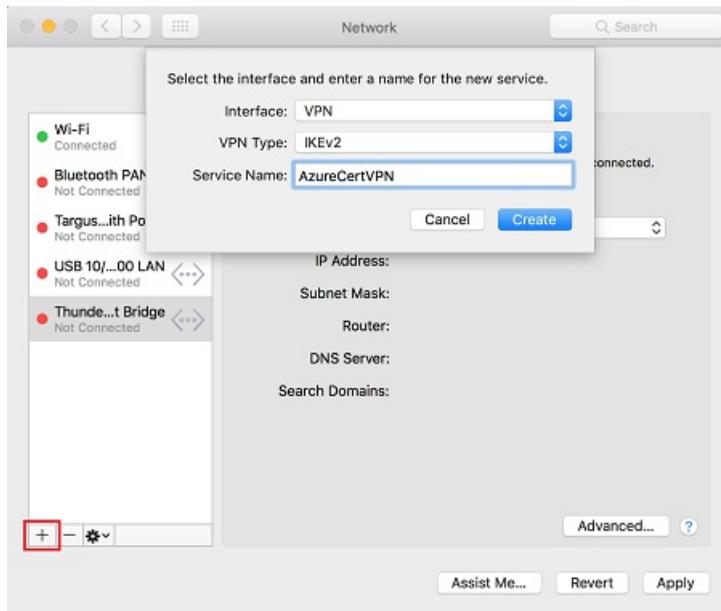


NOTE

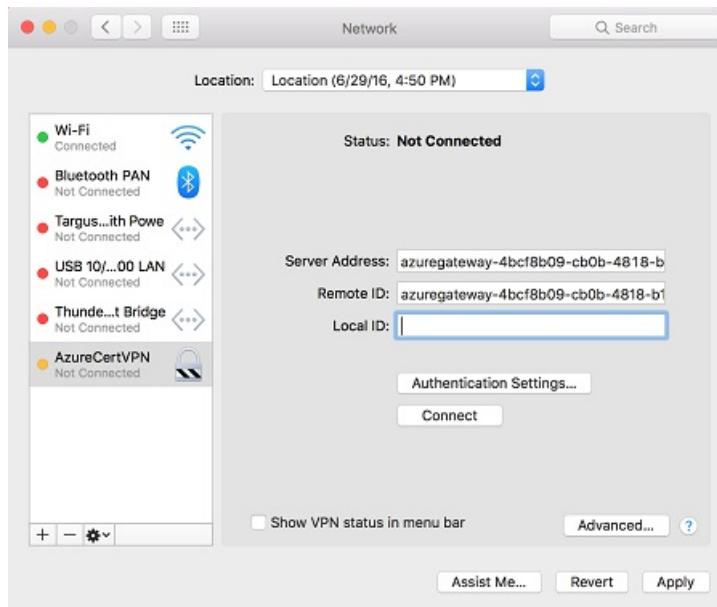
Double-clicking on the certificate may not display the **Add** dialog, but the certificate is installed in the correct store. You can check for the certificate in the login keychain under the certificates category.

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.

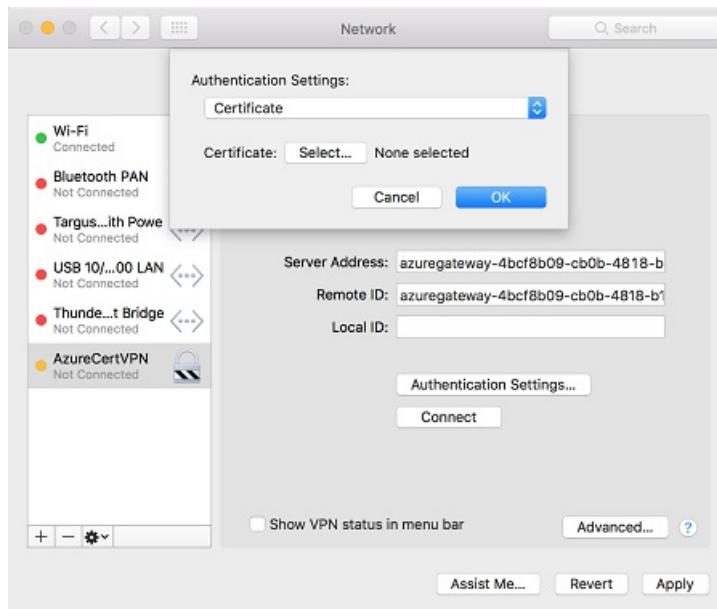
The **Interface** value is 'VPN' and **VPN Type** value is 'IKEv2'. Specify a name for the profile in the **Service Name** field, then click **Create** to create the VPN client connection profile.



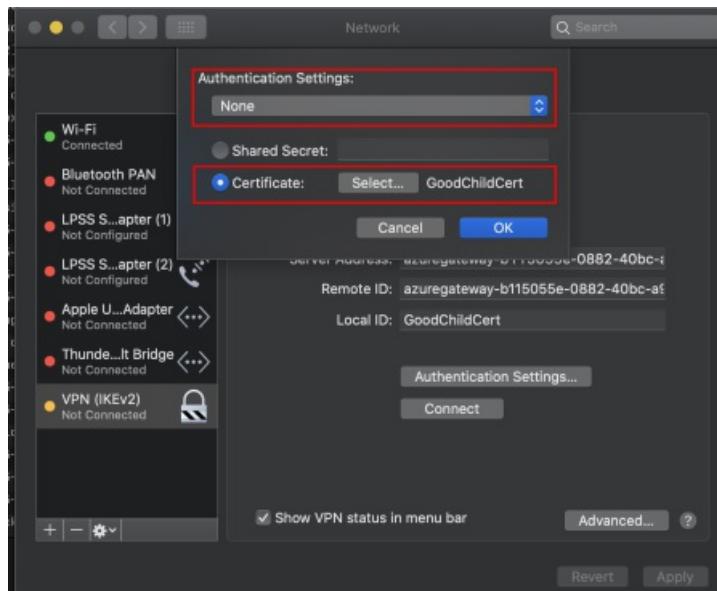
4. In the **Generic** folder, from the **VpnSettings.xml** file, copy the **VpnServer** tag value. Paste this value in the **Server Address** and **Remote ID** fields of the profile.



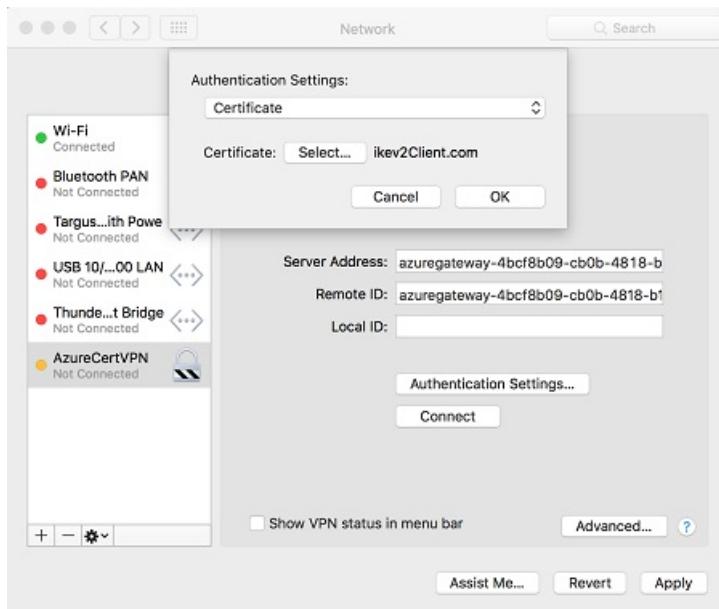
5. Click **Authentication Settings** and select **Certificate**. For **Catalina**, click **None** and then **certificate**



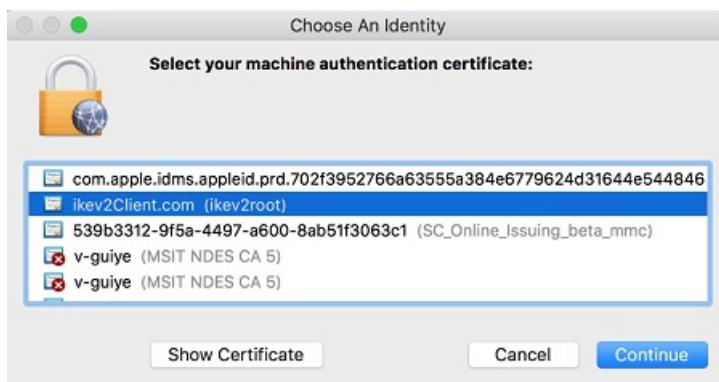
- For Catalina, select **None** and then **Certificate**. Select the correct certificate:



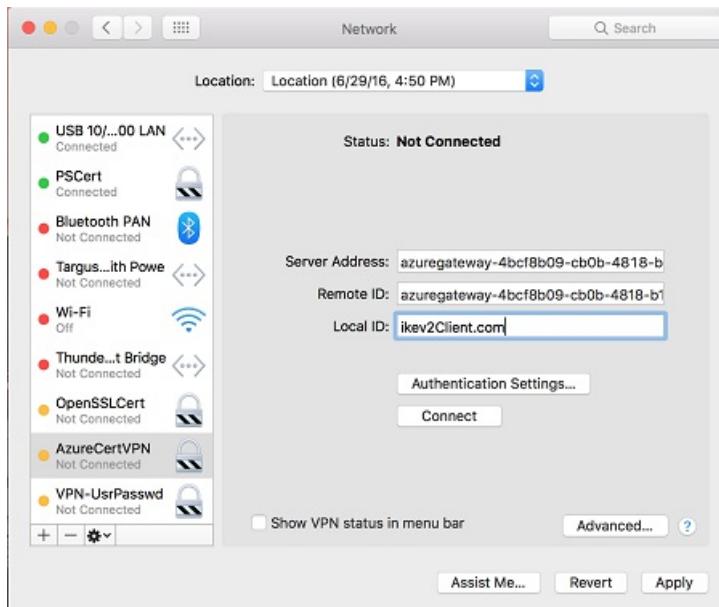
6. Click **Select...** to choose the client certificate that you want to use for authentication. This is the certificate that you installed in Step 2.



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)

Install strongSwan

The following configuration was used for the steps below:

Computer	Ubuntu Server 18.04
Dependencies	strongSwan

Use the following commands to install the required strongSwan configuration:

```
sudo apt install strongswan
```

```
sudo apt install strongswan-pki
```

```
sudo apt install libstrongswan-extra-plugins
```

Use the following command to install the Azure command-line interface:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Additional instructions on how to install the Azure CLI

Generate certificates

If you have not already generated certificates, use the following steps:

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
```

Print the CA certificate in base64 format. This is the format that is supported by Azure. You upload this certificate to Azure as part of the [P2S configuration steps](#).

```
openssl x509 -in caCert.pem -outform der | base64 -w0 ; echo
```

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"
```

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}"
```

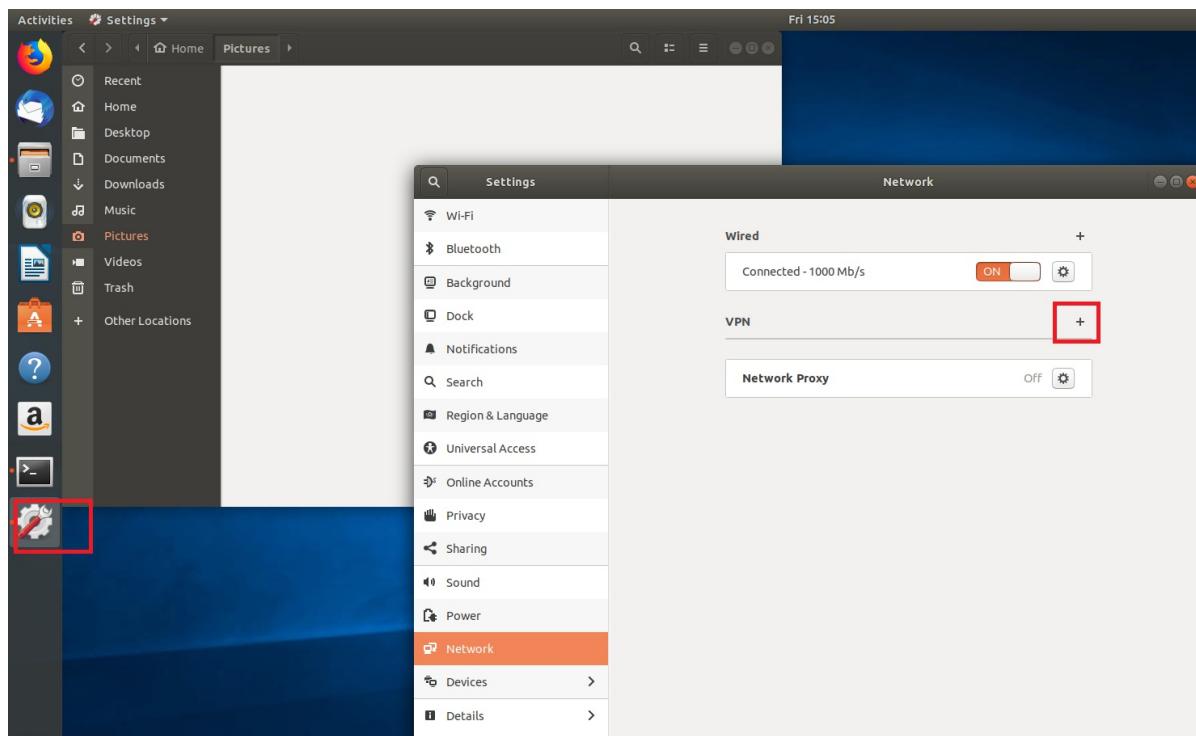
Install and configure

The following instructions were created on Ubuntu 18.0.4. 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.

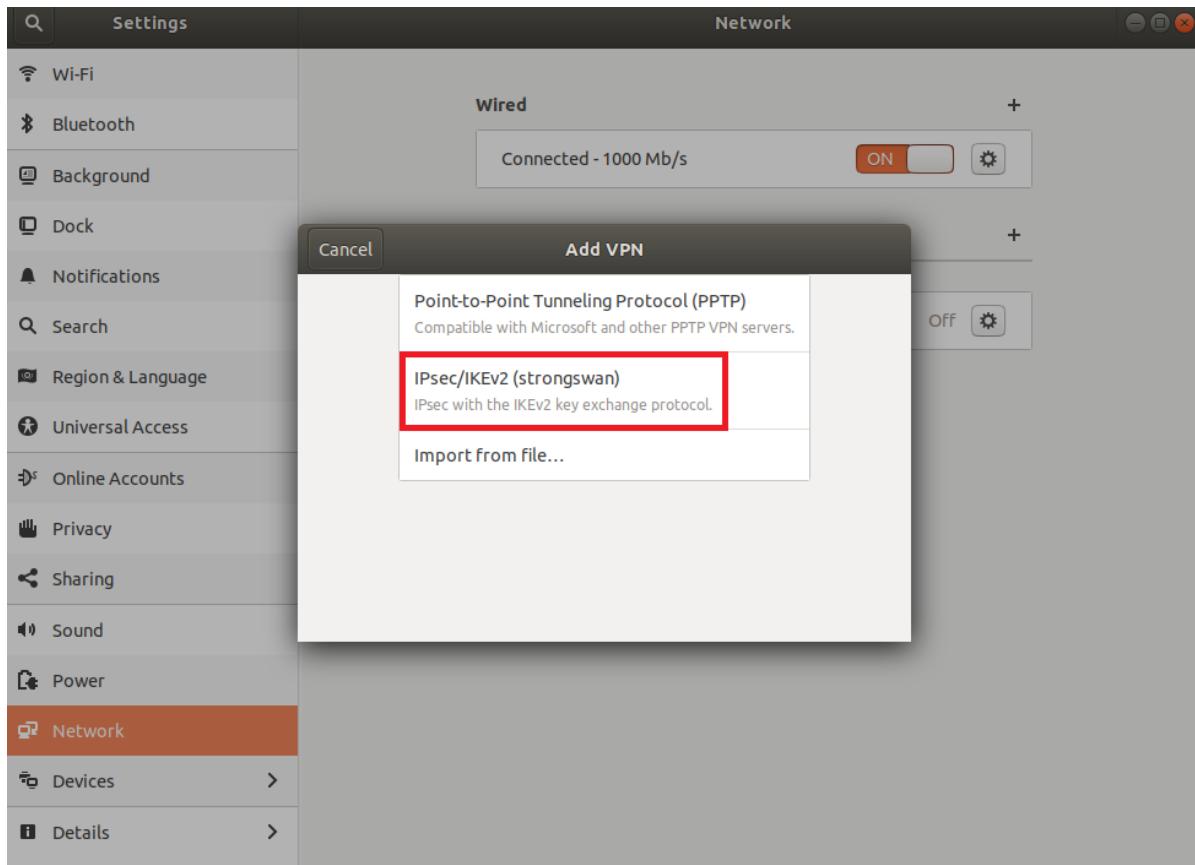
1. Open the **Terminal** to install **strongSwan** and its Network Manager by running the command in the example.

```
sudo apt install network-manager-strongswan
```

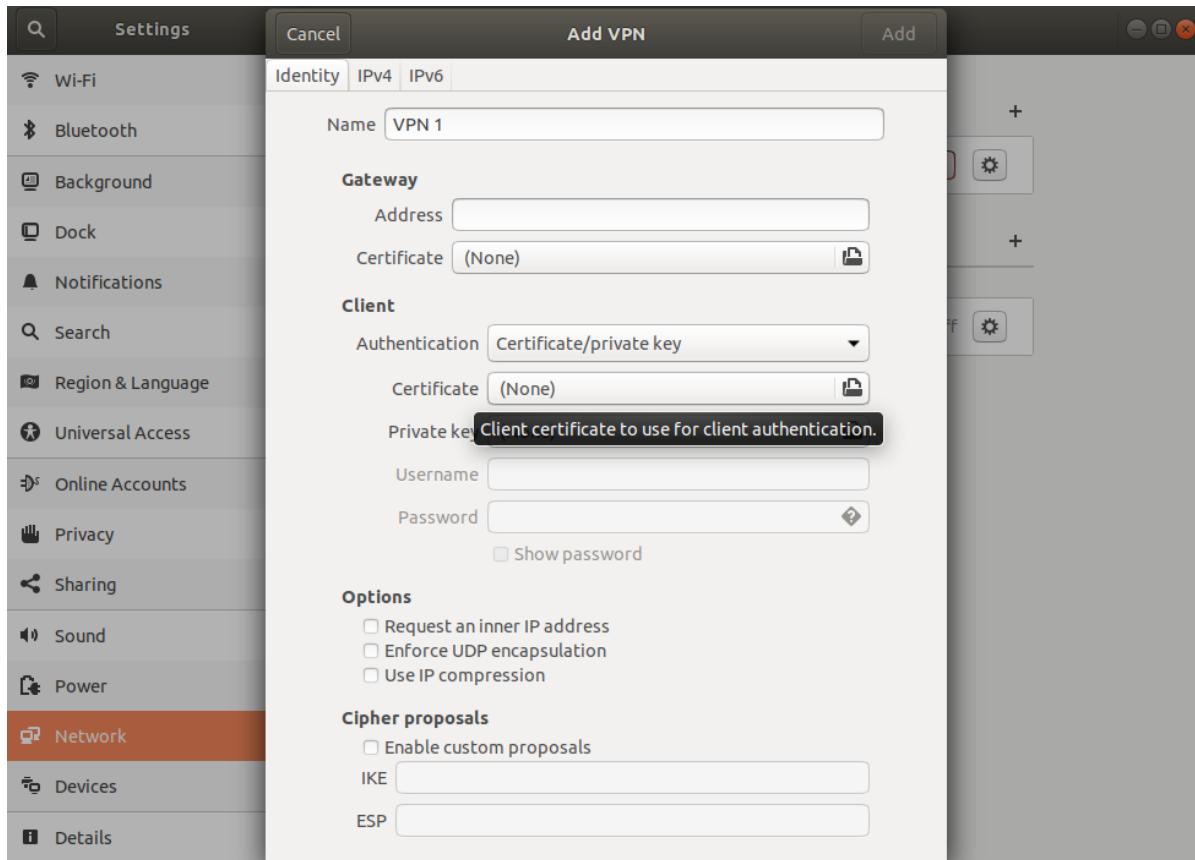
2. Select **Settings**, then select **Network**.



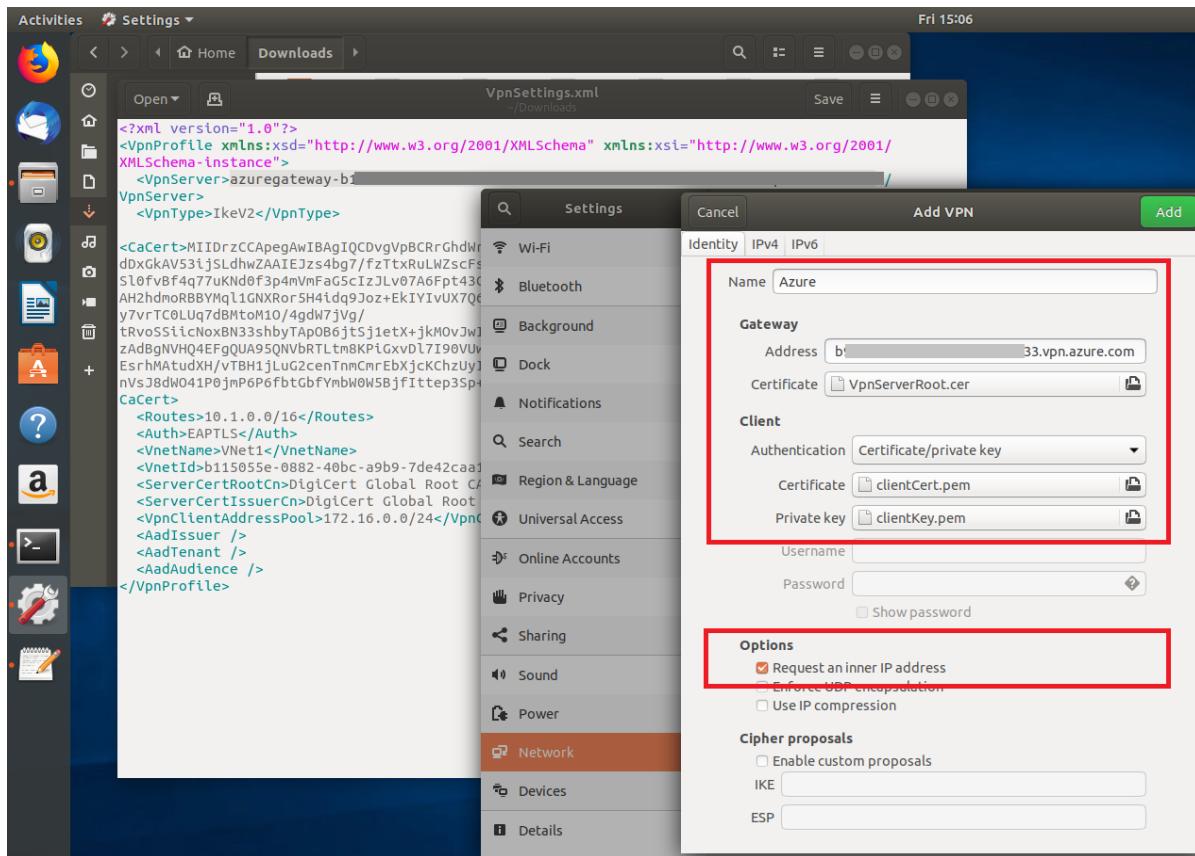
3. Click the + button to create a new connection.



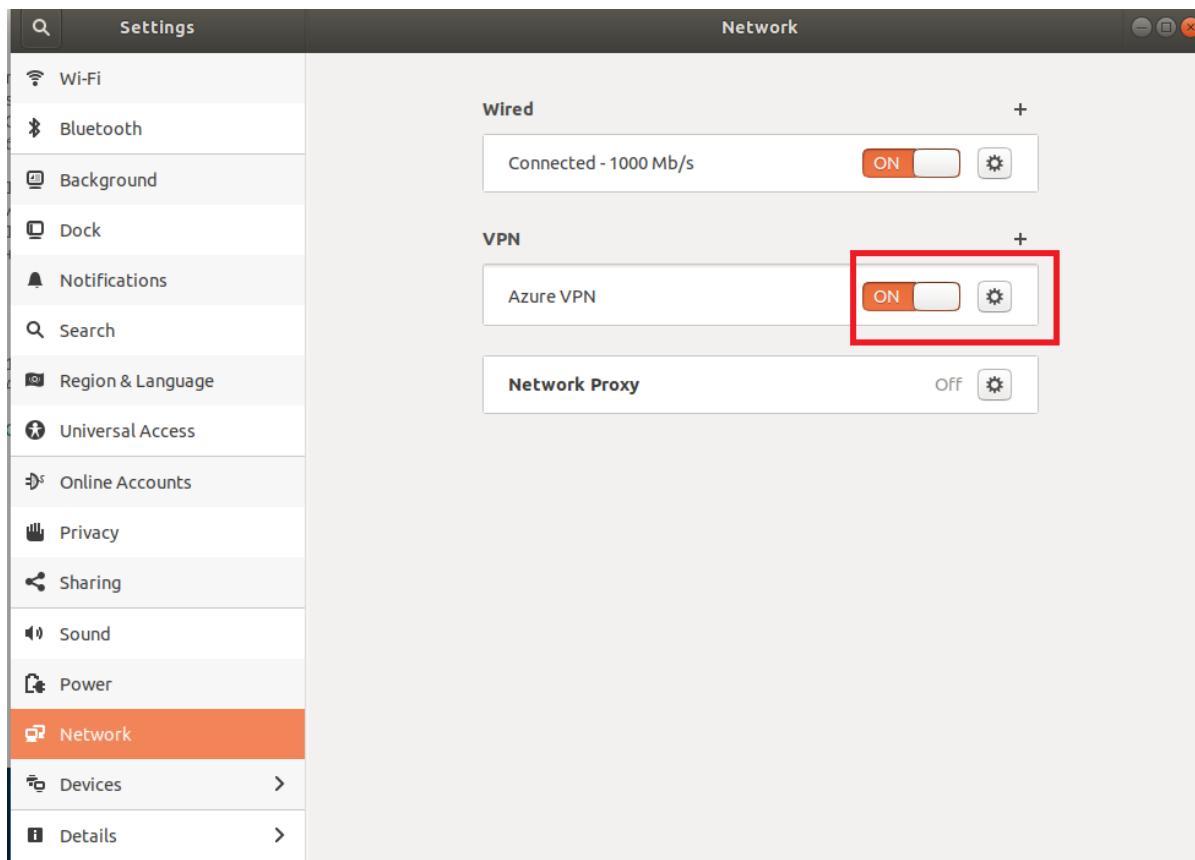
4. Select **IPsec/IKEv2 (strongSwan)** from the menu, and double-click. You can name your connection in this step.



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'.



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**.



8. Turn the connection **On**.

Linux (strongSwan CLI)

Install strongSwan

The following configuration was used for the steps below:

Computer	Ubuntu Server 18.04
Dependencies	strongSwan

Use the following commands to install the required strongSwan configuration:

```
sudo apt install strongswan
```

```
sudo apt install strongswan-pki
```

```
sudo apt install libstrongswan-extra-plugins
```

Use the following command to install the Azure command-line interface:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Additional instructions on how to install the Azure CLI

Generate certificates

If you have not already generated certificates, use the following steps:

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
```

Print the CA certificate in base64 format. This is the format that is supported by Azure. You upload this certificate to Azure as part of the [P2S configuration steps](#).

```
openssl x509 -in caCert.pem -outform der | base64 -w0 ; echo
```

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"
```

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}"
```

Install and configure

1. Download the VPNClient package from Azure portal.
2. Extract the File.
3. From the **Generic** folder, copy or move the VpnServerRoot.cer to /etc/ipsec.d/cacerts.
4. Copy or move cp client.p12 to /etc/ipsec.d/private/. This file is client certificate for Azure VPN Gateway.
5. Open VpnSettings.xml file and copy the `<vpnServer>` value. You will use this value in the next step.
6. 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 FQDN 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

2/11/2020 • 19 minutes to read • [Edit Online](#)

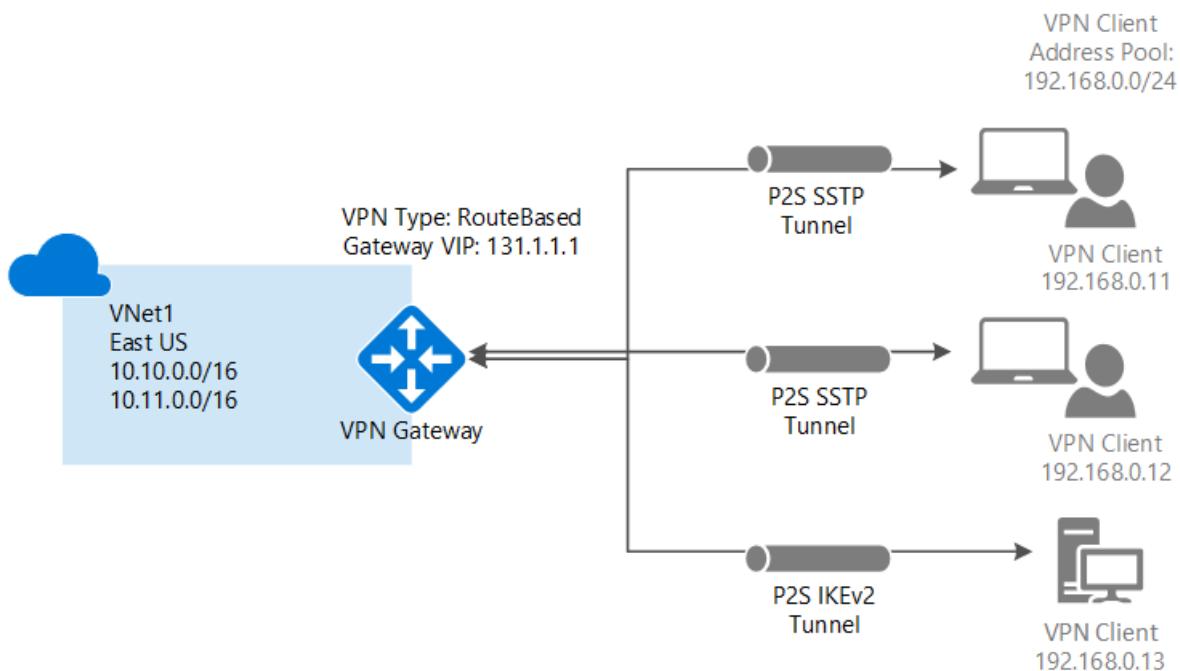
This article shows you how to create a VNet with a Point-to-Site connection that uses RADIUS authentication. This configuration is only available for the Resource Manager deployment model.

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), OpenVPN 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.
- OpenVPN® Protocol, an SSL/TLS based VPN protocol. An SSL VPN solution can penetrate firewalls, since most firewalls open TCP port 443 outbound, which SSL uses. OpenVPN can be used to connect from

Android, iOS (versions 11.0 and above), Windows, Linux and Mac devices (OSX versions 10.13 and above).

- 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:

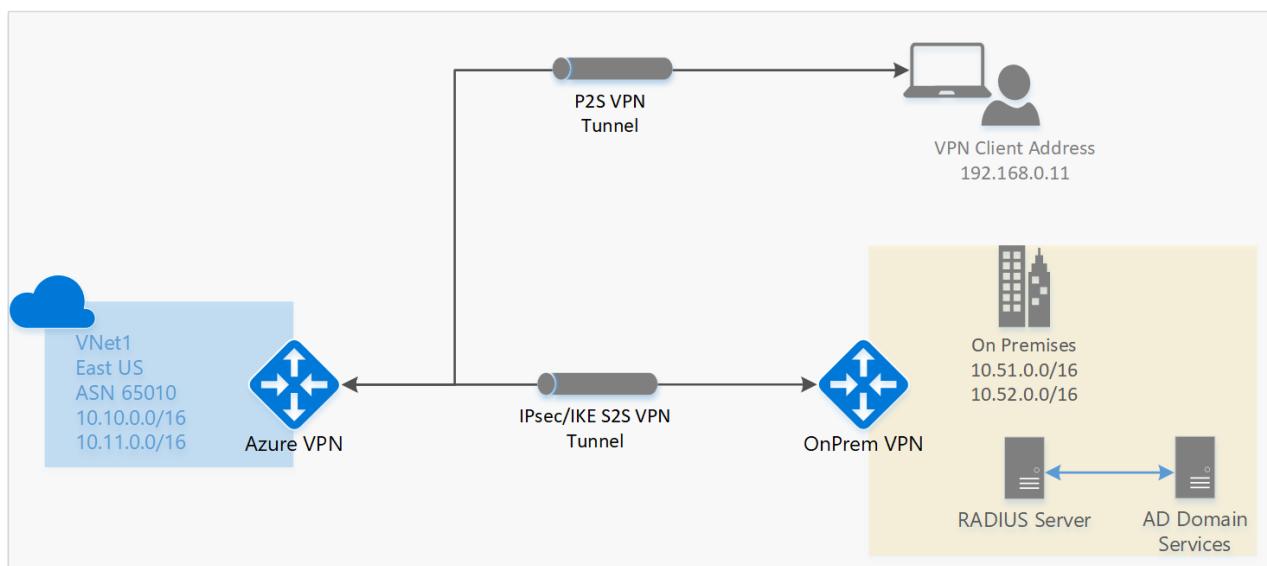
- A RouteBased VPN gateway.
- A RADIUS server to handle user authentication. The RADIUS server can be deployed on-premises, or in the Azure VNet.
- A VPN client configuration package for the Windows devices that will connect to the VNet. A VPN client configuration package provides the settings required for a VPN client to connect over P2S.

About Active Directory (AD) Domain Authentication for P2S VPNs

AD Domain authentication allows users to sign 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](#).

Working with Azure PowerShell

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

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. Set the variables

Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary. If you close your PowerShell/Cloud Shell session at any point during the exercise, just copy and paste the values again to re-declare the variables.

```
$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"
$VPNCClientAddressPool = "172.16.201.0/24"
$RG = "TestRG"
$Location = "East US"
$GWName = "VNet1GW"
$GWIPName = "VNet1GWPIP"
$GWIPconfName = "gwipconf"
```

2. 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-AzResourceGroup -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-AzVirtualNetworkSubnetConfig -Name "FrontEnd" -AddressPrefix "192.168.1.0/24"
$besub = New-AzVirtualNetworkSubnetConfig -Name "Backend" -AddressPrefix "10.254.1.0/24"
$gwsu = New-AzVirtualNetworkSubnetConfig -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-AzVirtualNetwork -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-AzVirtualNetwork -Name "VNet1" -ResourceGroupName "TestRG"
$subnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$PIP = New-AzPublicIpAddress -Name "VNet1GWPPIP" -ResourceGroupName "TestRG" -Location "East US" -
AllocationMethod Dynamic
$ipconf = New-AzVirtualNetworkGatewayIpConfig -Name "gwipconf" -Subnet $subnet -PublicIpAddress $PIP
```

3. 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.

4. 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-AzVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG ` 
-Location $Location -IpConfigurations $ipconf -GatewayType Vpn ` 
-VpnType RouteBased -EnableBgp $false -GatewaySku VpnGw1
```

5. 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-AzVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol "SSTP" `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

For OpenVPN® configurations:

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientRootCertificates @()  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol "OpenVPN" `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

For IKEv2 configurations:

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol "IKEv2" `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

For SSTP + IKEv2

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName $RG -Name $GWName  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway `  
-VpnClientAddressPool "172.16.201.0/24" -VpnClientProtocol @("SSTP", "IKEv2") `  
-RadiusServerAddress "10.51.0.15" -RadiusServerSecret $Secure_Secret
```

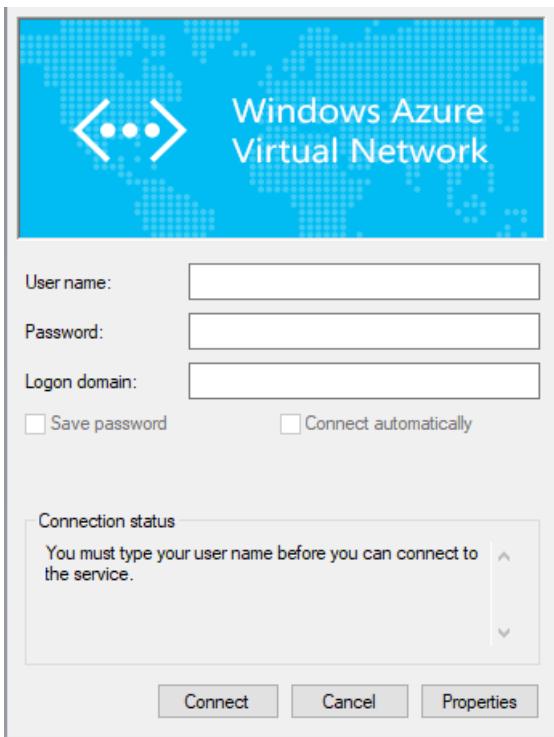
6. Download the VPN client configuration package and set up the VPN client

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](#).

7. 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.

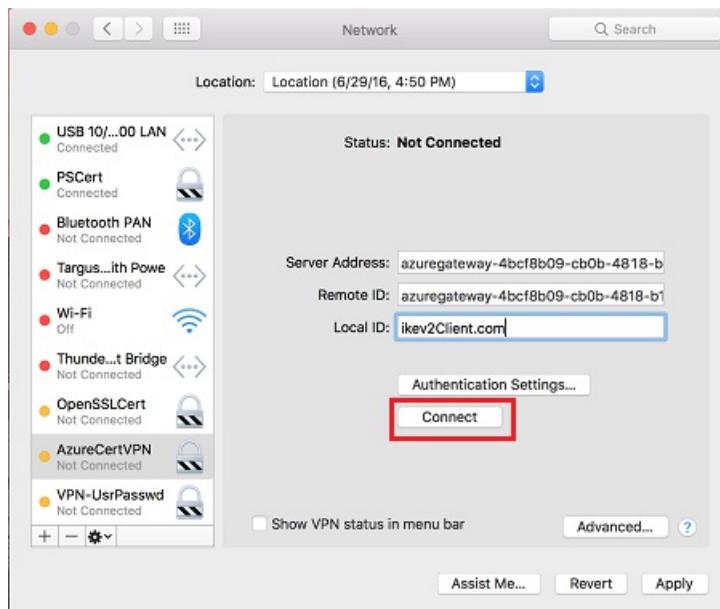


2. Your connection is established.



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-AzVM
$Nics = Get-AzNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)
{
    $VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id
    $Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress
    $Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod
    Write-Output "$($VM.Name): $Prv,$Alloc"
}

```

2. Verify that you are connected to your VNet using the Point-to-Site VPN connection.
3. Open **Remote Desktop Connection** by typing "RDP" or "Remote Desktop Connection" in the search box on the taskbar, then select Remote Desktop Connection. You can also open Remote Desktop Connection using the 'mstsc' command in PowerShell.
4. In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

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](#).

FAQ

This FAQ applies to P2S using RADIUS authentication

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

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

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 for SSTP, but depends on the gateway SKU for IKEv2. For more information on the number of connections supported, see [Gateway SKUs](#).

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.

Does RADIUS authentication work with the OpenVPN client?

RADIUS authentication is supported for the OpenVPN protocol only through PowerShell.

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

2/11/2020 • 12 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

You can generate the VPN client configuration files by using the Azure portal, or by using Azure PowerShell.

Azure portal

1. Navigate to the virtual network gateway.
2. Click **Point-to-Site configuration**.
3. Click **Download VPN client**.
4. Select the client and fill out any information that is requested.
5. Click **Download** to generate the .zip file.
6. The .zip file will download, typically to your Downloads folder.

Azure PowerShell

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-AzVpnClientConfiguration -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-AzVpnClientConfiguration` 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-AzVpnClientConfiguration` cmdlet to create a new configuration download.

To retrieve previously generated client configuration files, use the following command:

```
Get-AzVpnClientConfiguration -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

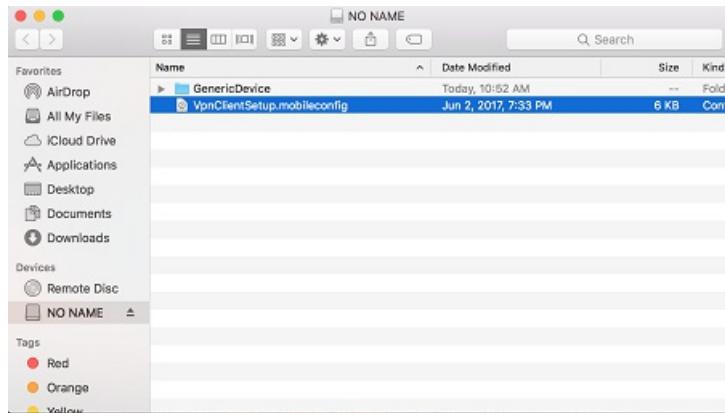
You can use the same VPN client configuration package on each Windows client computer, as long as the version matches the architecture for the client. For the list of client operating systems that are supported, see the [FAQ](#).

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.



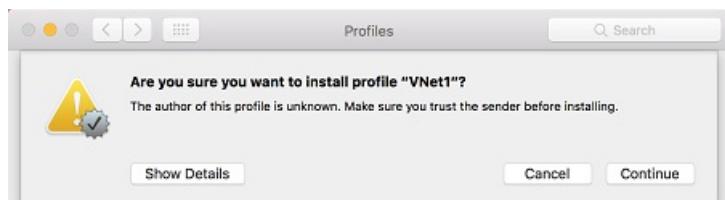
3. Optional Step - If you want to specify a custom DNS, add the following lines to the **mobileconfig** file:

```
<key>DNS</key>
<dict>
    <key>ServerAddresses</key>
        <array>
            <string>10.0.0.132</string>
        <array>
            <key>SupplementalMatchDomains</key>
                <array>
                    <string>TestDomain.com</string>
                </array>
            </key>
        </array>
    </dict>
```

4. Double-click the profile to install it, and select **Continue**. The profile name is the same as the name of your virtual network.



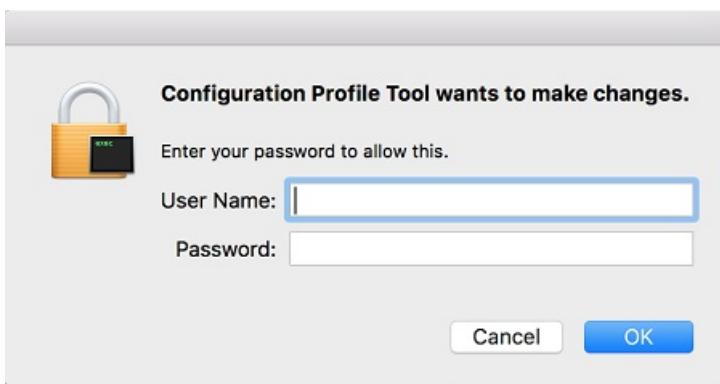
5. Select **Continue** to trust the sender of the profile and proceed with the installation.



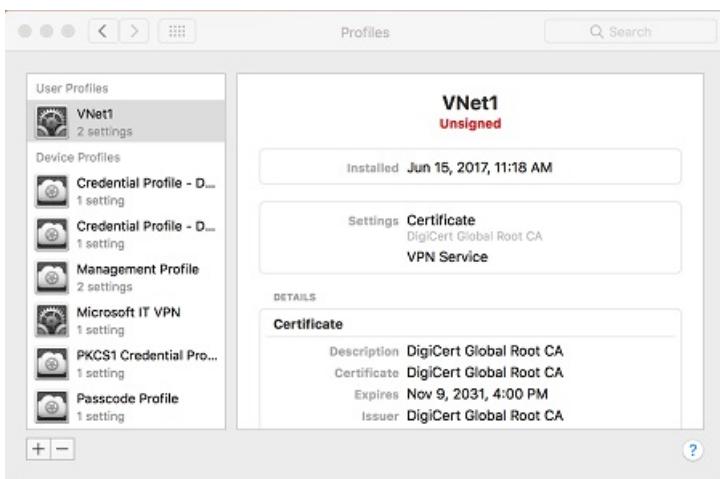
6. During profile installation, you have the option to specify the username and password for VPN authentication. It's not mandatory to enter this information. If you do, the information is saved and automatically used when you initiate a connection. Select **Install** to proceed.



7. Enter a username and password for the privileges that are required to install the profile on your computer. Select **OK**.



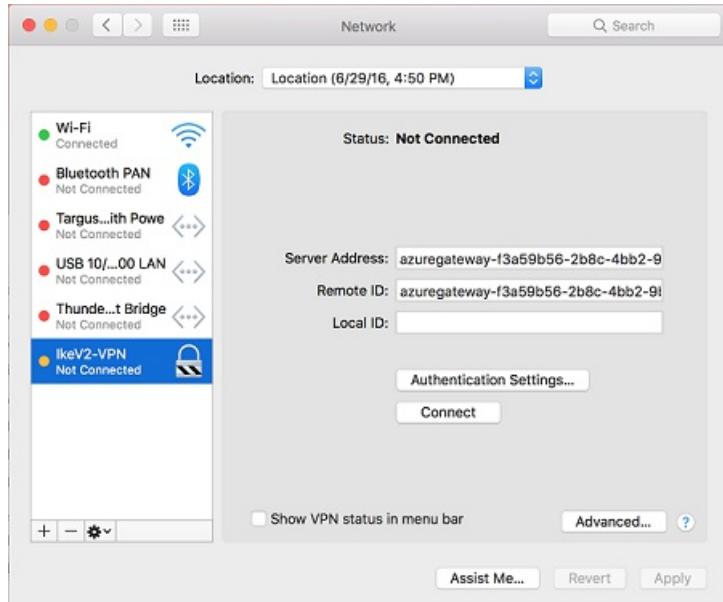
8. After the profile is installed, it's visible in the **Profiles** dialog box. You can also open this dialog box later from **System Preferences**.



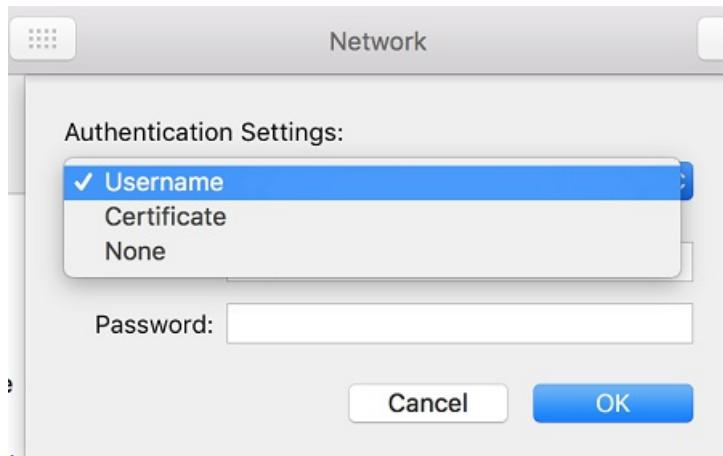
9. To access the VPN connection, open the **Network** dialog box from **System Preferences**.



10. The VPN connection appears as **IkeV2-VPN**. You can change the name by updating the **mobileconfig** file.



11. 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.



12. Back in the **Network** dialog box, select **Apply** to save the changes. To initiate the connection, select **Connect**.

Linux VPN client setup through strongSwan

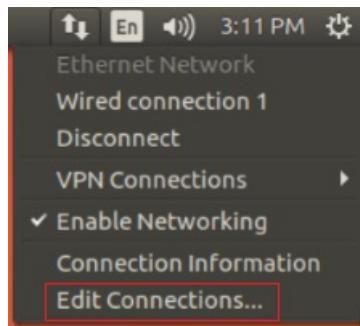
The following instructions were created through strongSwan 5.5.1 on Ubuntu 17.0.4. Actual screens might be

different, depending on your version of Linux and strongSwan.

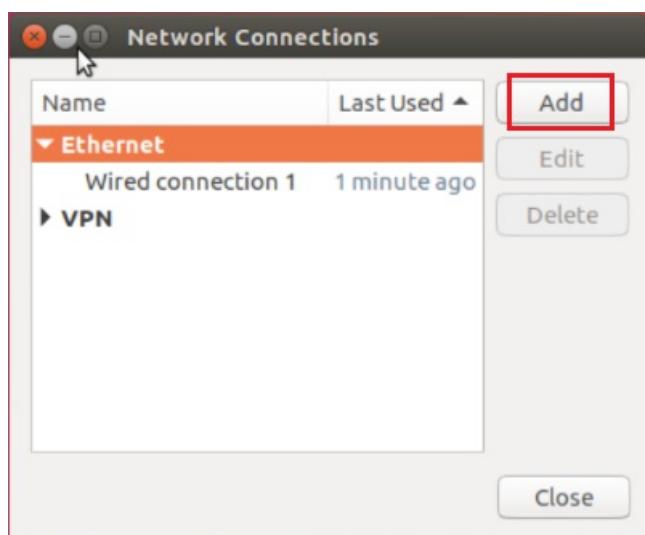
1. Open the **Terminal** to install **strongSwan** and its Network Manager by running the command in the example. If you receive an error that's related to `libcharon-extra-plugins`, replace it with `strongswan-plugin-eap-mschapv2`.

```
sudo apt-get install strongswan libcharon-extra-plugins moreutils iptables-persistent network-manager-strongswan
```

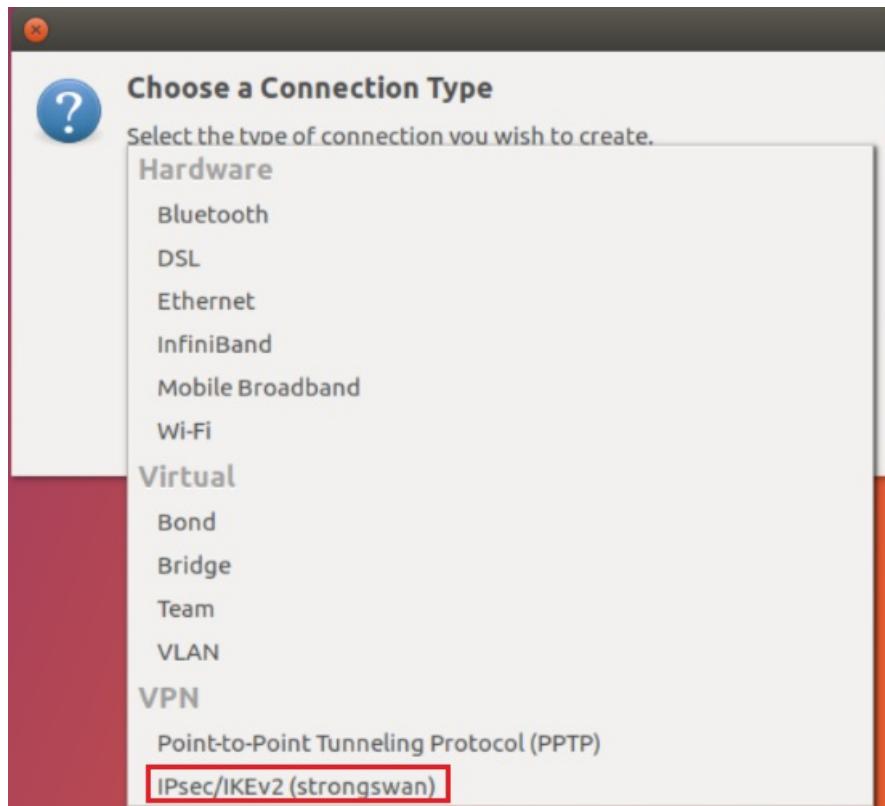
2. Select the **Network Manager** icon (up-arrow/down-arrow), and select **Edit Connections**.



3. Select the **Add** button to create a new connection.



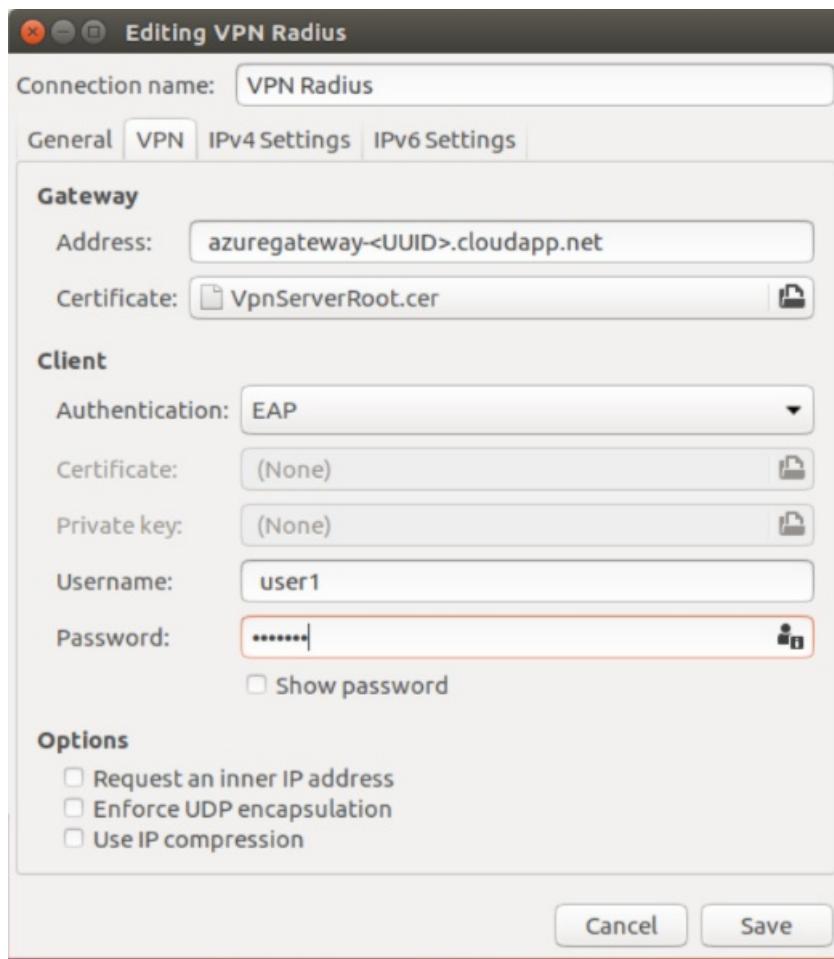
4. Select **IPsec/IKEv2 (strongswan)** from the drop-down menu, and then select **Create**. You can rename your connection in this step.



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-AzVpnClientConfiguration -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-AzVpnClientConfiguration` 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-AzVpnClientConfiguration` cmdlet to create a new configuration download.

To retrieve previously generated client configuration files, use the following command:

```
Get-AzVpnClientConfiguration -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.

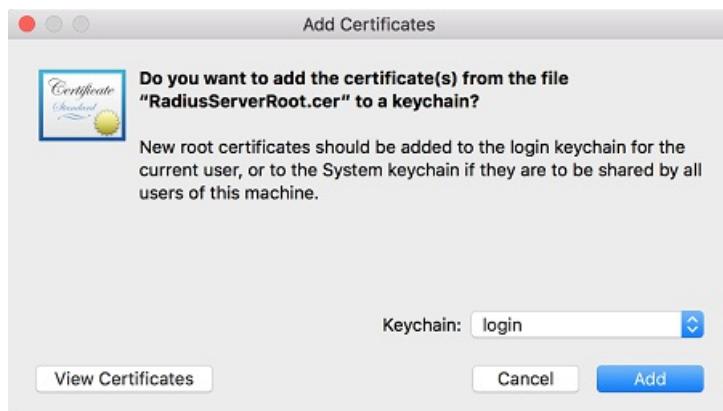
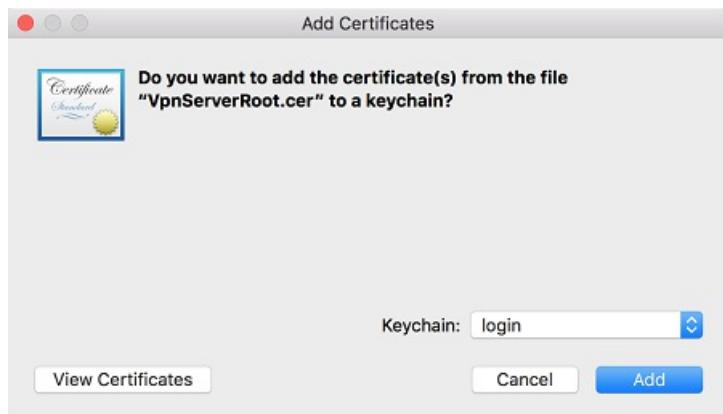
Mac (OS X) VPN client setup

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.

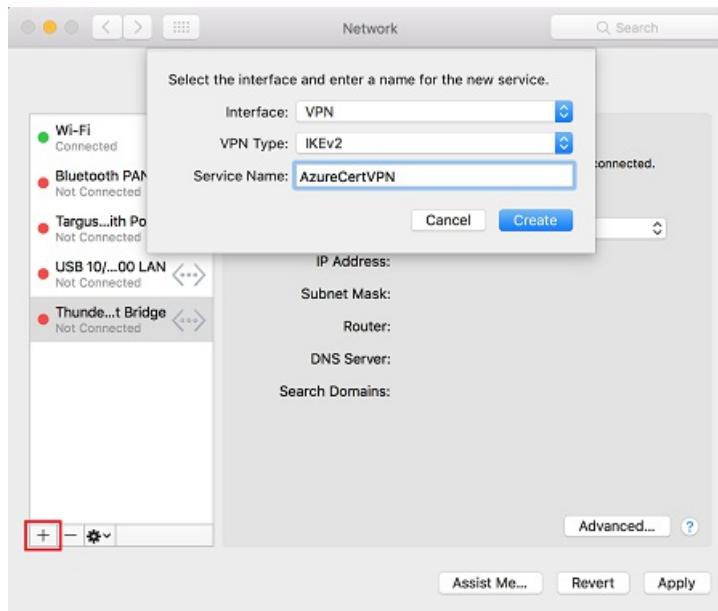
Use the following steps to configure the native VPN client on a Mac for certificate authentication:

1. Import the **VpnServerRoot** and **RadiusServerRoot** root certificates to your Mac. Copy each file to your Mac, double-click it, and then select **Add**.

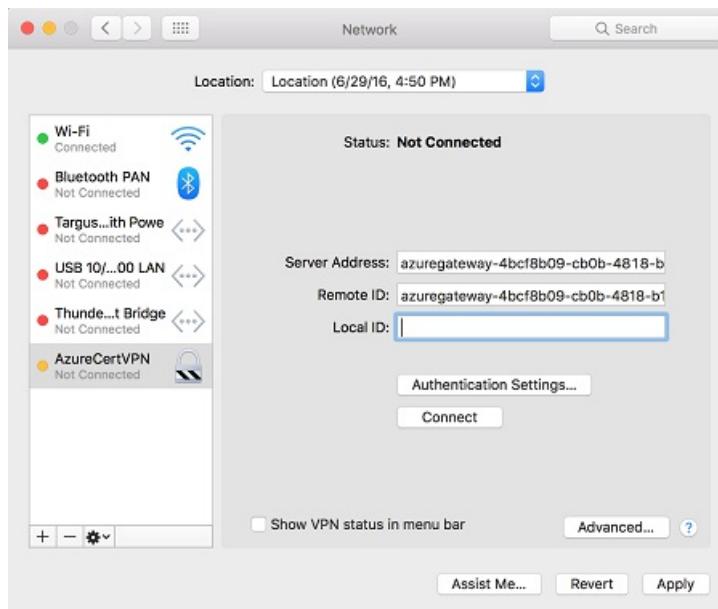


2. Each client requires a client certificate for authentication. Install the client certificate on the client device.
3. Open the **Network** dialog box under **Network Preferences**. Select + to create a new VPN client connection profile for a P2S connection to the Azure virtual network.

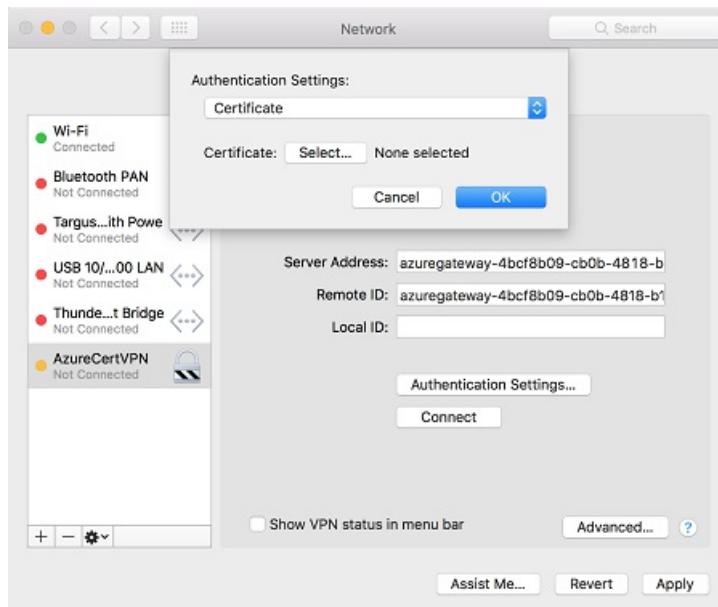
The **Interface** value is **VPN**, and the **VPN Type** value is **IKEv2**. Specify a name for the profile in the **Service Name** box, and then select **Create** to create the VPN client connection profile.



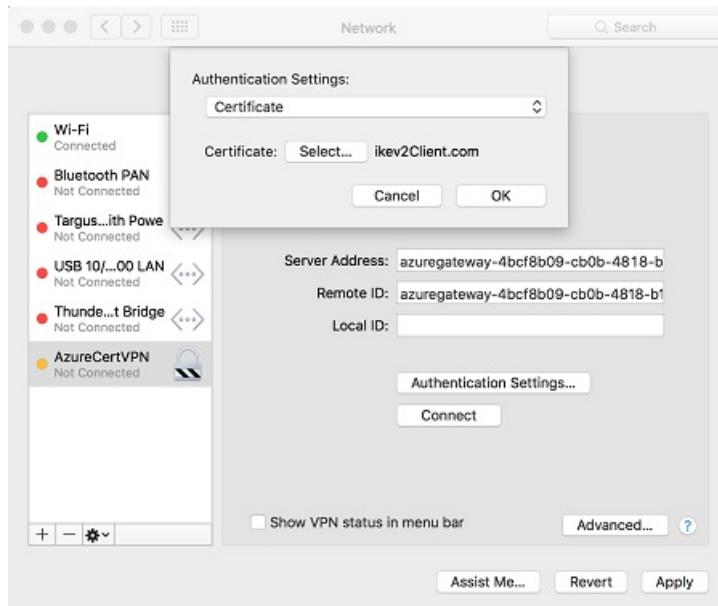
4. In the **Generic** folder, from the **VpnSettings.xml** file, copy the **VpnServer** tag value. Paste this value in the **Server Address** and **Remote ID** boxes of the profile. Leave the **Local ID** box blank.



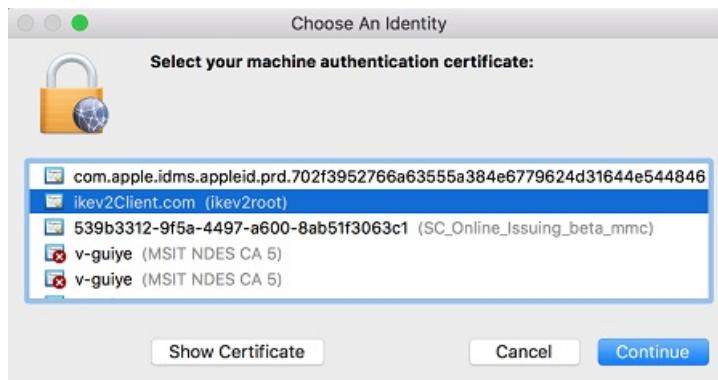
5. Select **Authentication Settings**, and select **Certificate**.



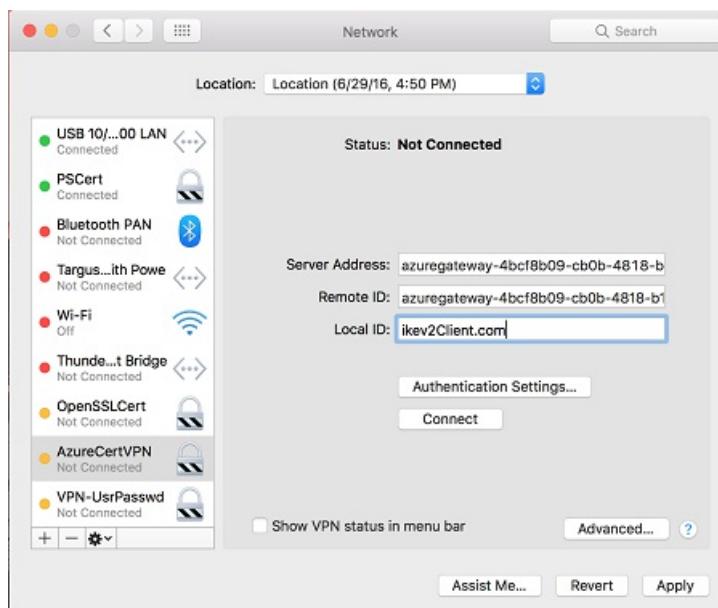
6. Click **Select** to choose the certificate that you want to use for authentication.



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-AzVpnClientConfiguration` cmdlet to generate the VPN client configuration for EapMSChapv2.
2. Unzip the `VpnClientConfiguration.zip` file and look for the **GenericDevice** folder. Ignore the folders that contain the Windows installers for 64-bit and 32-bit architectures.
3. The **GenericDevice** 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 **GenericDevice** 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

1/9/2020 • 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.

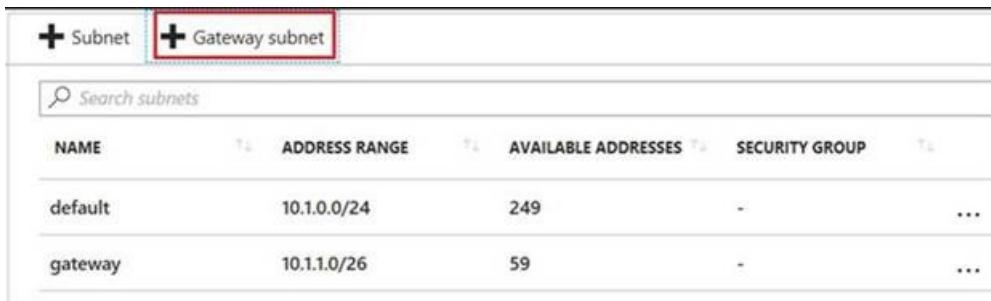
Prerequisite

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

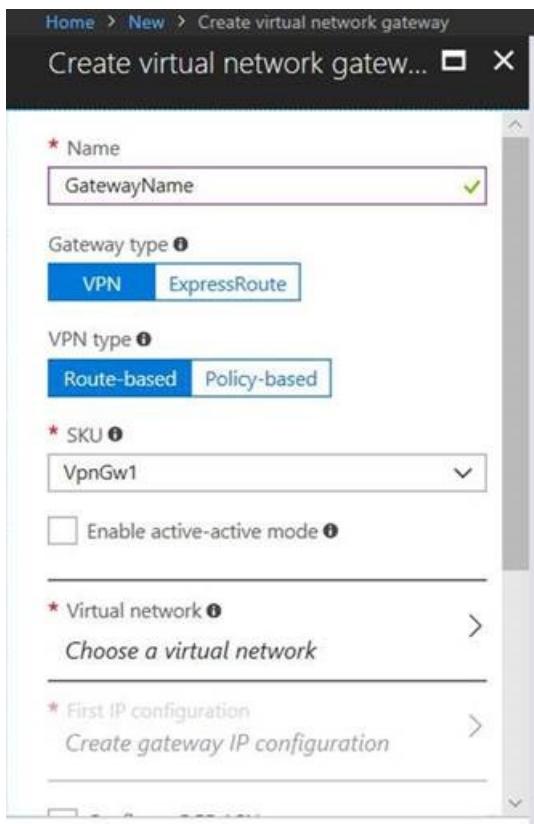
Step 1: Create a virtual network gateway

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.



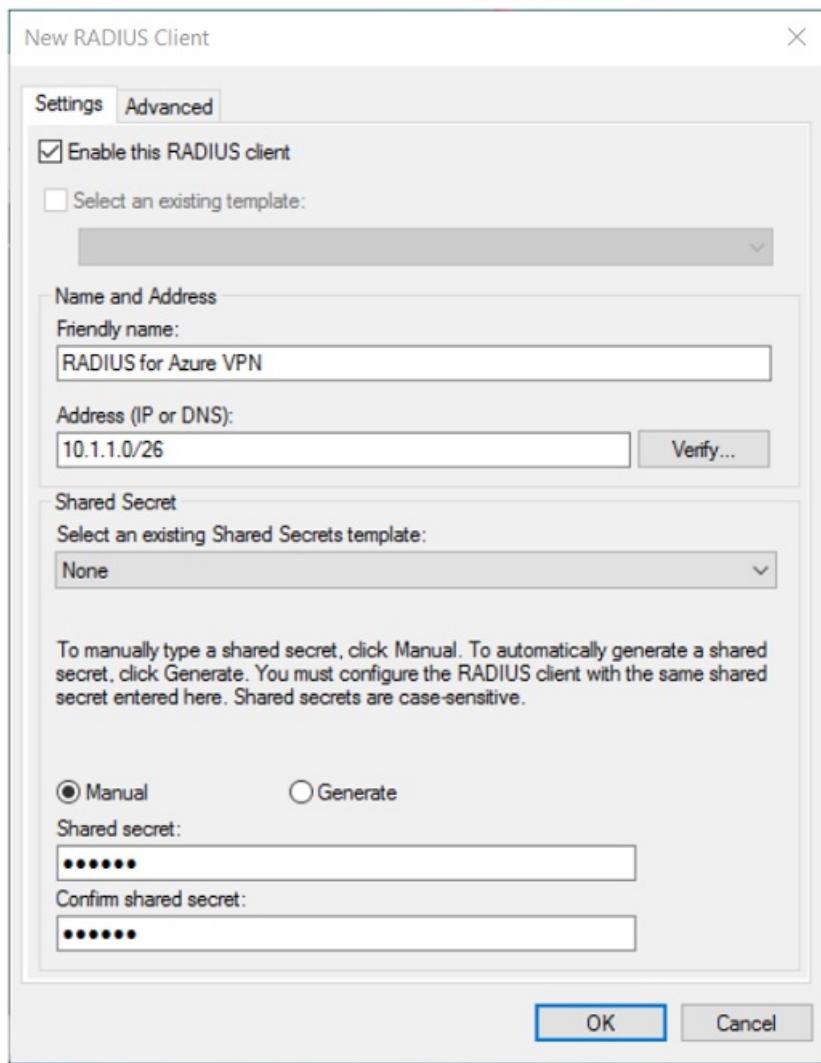
NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
default	10.1.0.0/24	249	-
gateway	10.1.1.0/26	59	-

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.

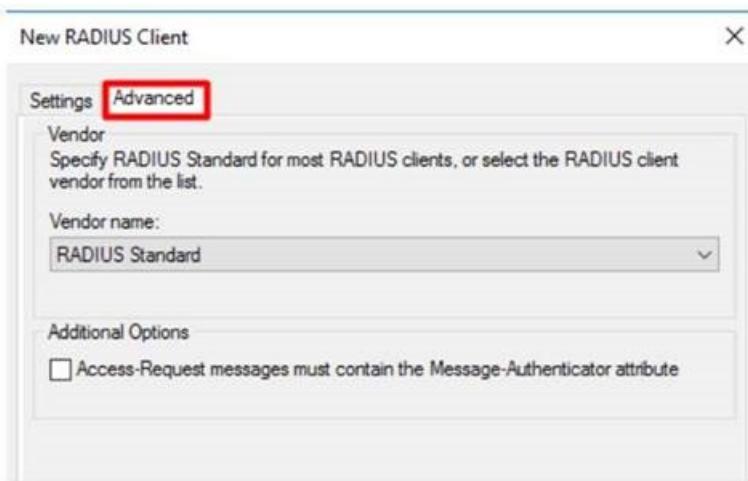


Step 2 Configure the NPS for Azure MFA

1. On the NPS server, [install the NPS extension for Azure MFA](#).
2. Open the NPS 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

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](#)

Create an Azure Active Directory tenant for P2S OpenVPN protocol connections

2/19/2020 • 2 minutes to read • [Edit Online](#)

When connecting to your VNet, you can use certificate-based authentication or RADIUS authentication. However, when you use the Open VPN protocol, you can also use Azure Active Directory authentication. This article helps you set up an Azure AD tenant for P2S Open VPN authentication.

NOTE

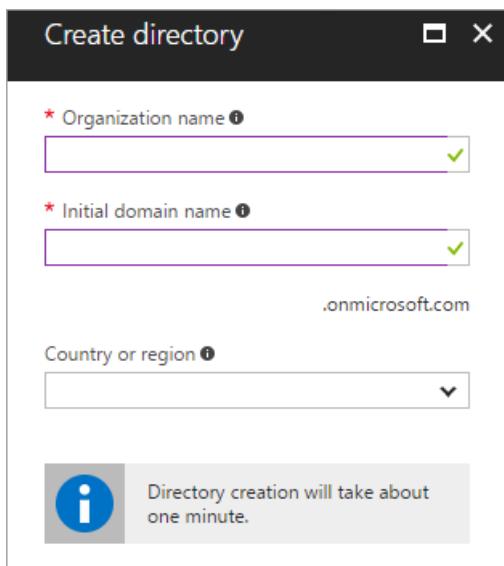
Azure AD authentication is supported only for OpenVPN® protocol connections.

1. Create the Azure AD tenant

Create an Azure AD tenant using the steps in the [Create a new tenant](#) article:

- Organizational name
- Initial domain name

Example:



2. Create Azure AD tenant users

Next, create two user accounts. Create one Global Admin account and one master user account. The master user account is used as your master embedding account (service account). When you create an Azure AD tenant user account, you adjust the Directory role for the type of user that you want to create.

Use the steps in [this article](#) to create at least two users for your Azure AD tenant. Be sure to change the **Directory Role** to create the account types:

- Global Admin
- User

3. Enable Azure AD authentication on the VPN gateway

- Locate the Directory ID of the directory that you want to use for authentication. It is listed in the properties section of the Active Directory page.

Contoso Corp - Properties

Directory properties

Name: Contoso Corp

Country or region: United States

Location: United States datacenters

Notification language: English

Directory ID: (highlighted with a red border)

Technical contact: am@microsoft.com

Global privacy contact:

Privacy statement URL:

Access management for Azure resources

can manage access to all Azure subscriptions and management groups in this directory. Learn more

- Copy the Directory ID.

- Sign in to the Azure portal as a user that is assigned the **Global administrator** role.

- Next, give admin consent. Copy and paste the URL that pertains to your deployment location in the address bar of your browser:

Public

```
https://login.microsoftonline.com/common/oauth2/authorize?client_id=41b23e61-6c1e-4545-b367-cd054e0ed4b4&response_type=code&redirect_uri=https://portal.azure.com&nonce=1234&prompt=admin_consent
```

Azure Government

```
https://login-us.microsoftonline.com/common/oauth2/authorize?client_id=51bb15d4-3a4f-4ebf-9dca-40096fe32426&response_type=code&redirect_uri=https://portal.azure.us&nonce=1234&prompt=admin_consent
```

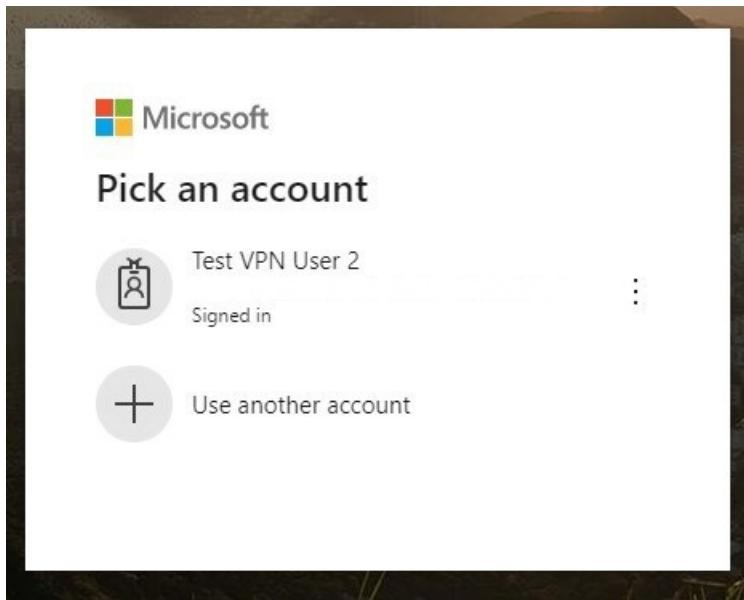
Microsoft Cloud Germany

```
https://login-us.microsoftonline.de/common/oauth2/authorize?client_id=538ee9e6-310a-468d-afef-ea97365856a9&response_type=code&redirect_uri=https://portal.microsoftazure.de&nonce=1234&prompt=admin_consent
```

Azure China 21Vianet

```
https://login.chinacloudapi.cn/common/oauth2/authorize?client_id=49f817b6-84ae-4cc0-928c-73f27289b3aa&response_type=code&redirect_uri=https://portal.azure.cn&nonce=1234&prompt=admin_consent
```

5. Select the **Global Admin** account if prompted.



6. Select **Accept** when prompted.



7. Under your Azure AD, in **Enterprise applications**, you see **Azure VPN** listed.

8. If you don't already have a functioning point-to-site environment, follow the instruction to create one. See [Create a point-to-site VPN](#) to create and configure a point-to-site VPN gateway with native Azure certificate authentication.

IMPORTANT

The Basic SKU is not supported for OpenVPN.

9. Enable Azure AD authentication on the VPN gateway by running the following commands, being sure to modify the command to reflect your own environment:

```
$gw = Get-AzVirtualNetworkGateway -Name <name of VPN gateway> -ResourceGroupName <Resource group>
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -VpnClientRootCertificates @()
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -AadTenantUri
"https://login.microsoftonline.com/<your Directory ID>" -AadAudienceId "41b23e61-6c1e-4545-b367-
cd054e0ed4b4" -AadIssuerUri "https://sts.windows.net/<your Directory ID>/" -VpnClientAddressPool
192.168.0.0/24 -VpnClientProtocol OpenVPN
```

NOTE

Make sure you include a trailing slash at the end of the `AadIssuerUri` value. Otherwise, the command will fail.

10. Create and download the profile by running the following commands. Change the `-ResourceGroupName` and `-Name` values to match your own.

```
$profile = New-AzVpnClientConfiguration -Name <name of VPN gateway> -ResourceGroupName <Resource group>
-AuthenticationMethod "EapTls"
$PROFILE.VpnProfileSASUrl
```

11. After running the commands, you see a result similar to the one below. Copy the result URL to your browser to download the profile zip file.

```
PS C:\WINDOWS\system32> $PROFILE.VpnProfileSASUrl
https://fvprodappmwh.blob.core.windows.net/vpnprofileimmutable/acfd0032-3d0e-4aa2-9586-dd15d89ea0f8/vpnprofile/e284f74d-ea23-4b79-985a-cdd8a9ca
qVdf7Rx2vwvy6M0%3D&st=2019-09-06T19%3A46%3A29Z&se=2019-09-06T20%3A46%3A29Z&sp=r&fileExtension=.zip
```

12. Extract the downloaded zip file.
13. Browse to the unzipped "AzureVPN" folder.
14. Make a note of the location of the "azurevpnconfig.xml" file. The azurevpnconfig.xml contains the setting for

the VPN connection and can be imported directly into the Azure VPN Client application. You can also distribute this file to all the users that need to connect via e-mail or other means. The user will need valid Azure AD credentials to connect successfully.

Next steps

In order to connect to your virtual network, you must create and configure a VPN client profile. See [Configure a VPN client for P2S VPN connections](#).

Create an Azure Active Directory tenant for P2S OpenVPN protocol connections

2/20/2020 • 4 minutes to read • [Edit Online](#)

When connecting to your VNet, you can use certificate-based authentication or RADIUS authentication. However, when you use the Open VPN protocol, you can also use Azure Active Directory authentication. If you want different set of users to be able to connect to different VPN gateways, you can register multiple apps in AD and link them to different VPN gateways. This article helps you set up an Azure AD tenant for P2S OpenVPN authentication and create and register multiple apps in Azure AD for allowing different access for different users and groups.

NOTE

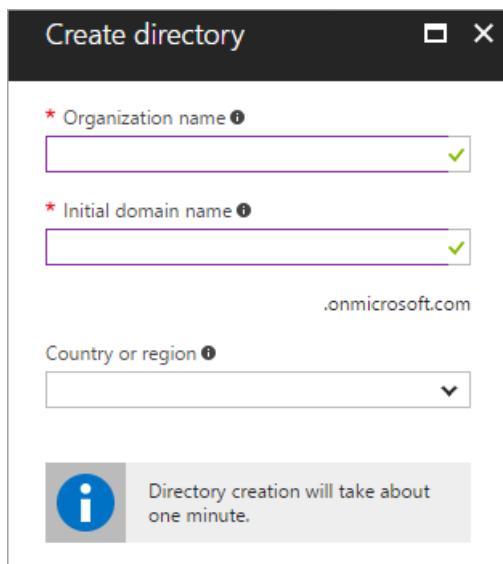
Azure AD authentication is supported only for OpenVPN® protocol connections.

1. Create the Azure AD tenant

Create an Azure AD tenant using the steps in the [Create a new tenant](#) article:

- Organizational name
- Initial domain name

Example:



2. Create tenant users

In this step, you create two Azure AD tenant users: One Global Admin account and one master user account. The master user account is used as your master embedding account (service account). When you create an Azure AD tenant user account, you adjust the Directory role for the type of user that you want to create. Use the steps in [this article](#) to create at least two users for your Azure AD tenant. Be sure to change the **Directory Role** to create the account types:

- Global Admin

- User

3. Register the VPN Client

Register the VPN client in the Azure AD tenant.

1. Locate the Directory ID of the directory that you want to use for authentication. It is listed in the properties section of the Active Directory page.

The screenshot shows the 'Contoso Corp - Properties' page in the Azure Active Directory portal. The left sidebar lists various management options like Overview, Getting started, Security, and Manage (Users, Groups, etc.). The main area displays 'Directory properties' with fields for Name (Contoso Corp), Country or region (United States), Location (United States datacenters), and Notification language (English). The 'Directory ID' field is highlighted with a red box. Below it are fields for Technical contact (am@microsoft.com), Global privacy contact, and Privacy statement URL. At the bottom, there's a section for Access management for Azure resources with a 'Yes' button for granting admin consent.

2. Copy the Directory ID.
3. Sign in to the Azure portal as a user that is assigned the **Global administrator** role.
4. Next, give admin consent. Copy and paste the URL that pertains to your deployment location in the address bar of your browser:

Public

```
https://login.microsoftonline.com/common/oauth2/authorize?client_id=41b23e61-6c1e-4545-b367-
cd054e0ed4b4&response_type=code&redirect_uri=https://portal.azure.com&nonce=1234&prompt=admin_consent
```

Azure Government

```
https://login-us.microsoftonline.com/common/oauth2/authorize?client_id=51bb15d4-3a4f-4ebf-9dca-
40096fe32426&response_type=code&redirect_uri=https://portal.azure.us&nonce=1234&prompt=admin_consent
```

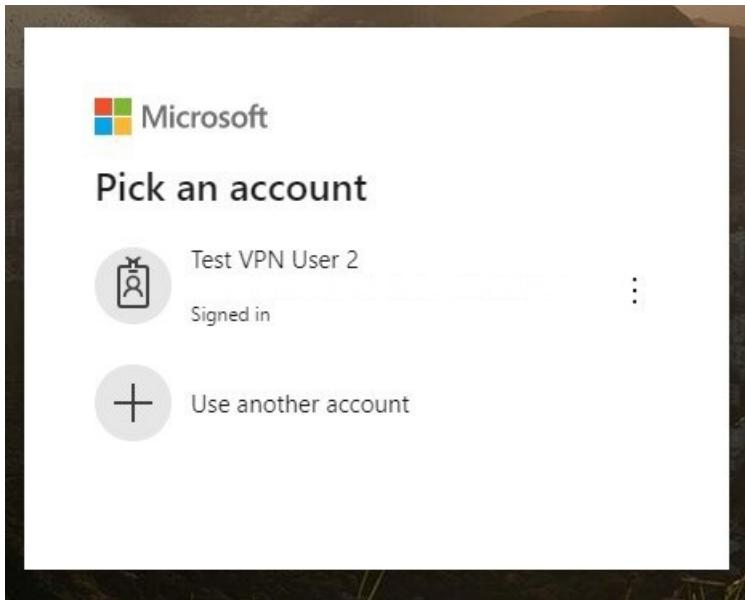
Microsoft Cloud Germany

https://login-us.microsoftonline.de/common/oauth2/authorize?client_id=538ee9e6-310a-468d-afef-ea97365856a9&response_type=code&redirect_uri=https://portal.microsoftazure.de&nonce=1234&prompt=admin_consent

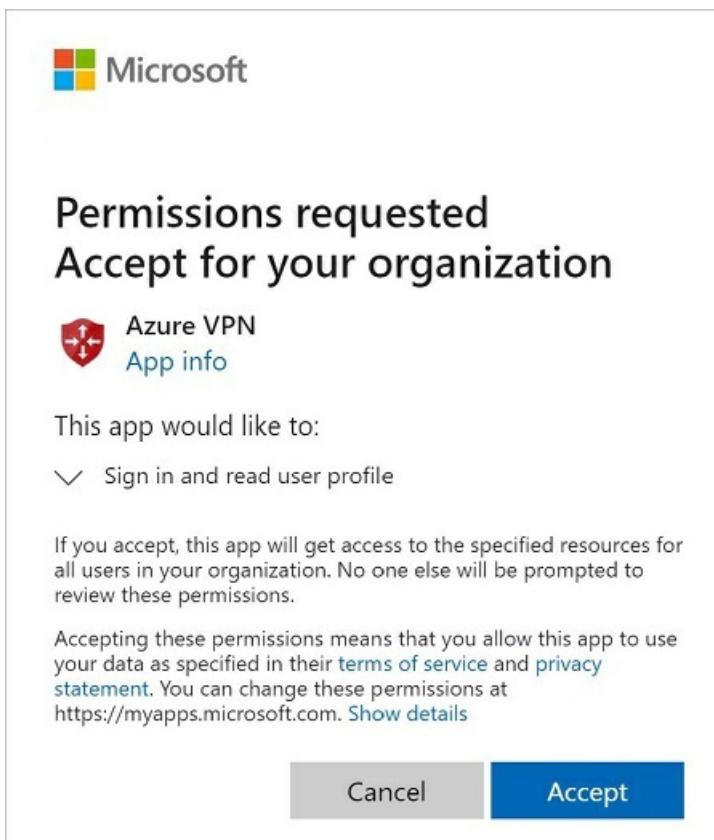
Azure China 21Vianet

https://login.chinacloudapi.cn/common/oauth2/authorize?client_id=49f817b6-84ae-4cc0-928c-73f27289b3aa&response_type=code&redirect_uri=https://portal.azure.cn&nonce=1234&prompt=admin_consent

5. Select the **Global Admin** account if prompted.



6. Select **Accept** when prompted.



7. Under your Azure AD, in **Enterprise applications**, you will see **Azure VPN** listed.

Enterprise applications - All applications

Contoso Corp - Azure Active Directory

Overview

New application | Columns

Application Type: Enterprise Applications | Applications status: Any | Application visibility: Any

Name: Azure VPN | Homepage URL: https://www.microsoft.com

4. Register additional applications

In this step, you register additional applications for various users and groups.

- Under your Azure Active Directory, click **App registrations** and then **+ New registration**.

The screenshot shows the Azure Active Directory - App registrations page. The left sidebar lists various management options, with 'App registrations' selected and highlighted by a red box. The top navigation bar includes a 'New registration' button, which is also highlighted by a red box. The main content area displays a table with two entries: 'HR' and 'HR VPN'. The 'All applications' tab is active, indicated by a blue dashed border around it. A search bar at the top allows filtering by display name.

2. On the **Register an application** page, enter the **Name**. Select the desired **Supported account types**, then click **Register**.

Microsoft Azure Search resources, service

Home > AliTestOrg - App registrations > Register an application

Register an application

* Name
The user-facing display name for this application (this can be changed later).

MarketingVPN ✓

Supported account types
Who can use this application or access this API?

Accounts in this organizational directory only (AliTestOrg only - Single tenant)

Accounts in any organizational directory (Any Azure AD directory - Multitenant)

Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web e.g. https://myapp.com/auth

By proceeding, you agree to the Microsoft Platform Policies [↗](#)

Register

3. Once the new app has been registered, click **Expose an API** under the app blade.
4. Click **+ Add a scope**.
5. Leave the default **Application ID URI**. Click **Save and continue**.

The screenshot shows the Microsoft Azure portal interface for managing an application. On the left, there's a sidebar with various navigation options like Overview, Quickstart, Manage, Authentication, Certificates & secrets, Token configuration (preview), API permissions, and the current selected option, Expose an API. The main content area has tabs for Overview, Application ID URI (which is currently set to 'api://03ecfa4-44d-441b-9e9-e5de0144e73'), Scopes, Who Can Consent, Admin Consent Display Name, User Consent Display Name, and State. The Scopes tab is active, showing a sub-section for Authorized client applications with a note about trusting the application. Below this, there are fields for Client Id and Scopes, both currently empty. At the bottom right of this section, there's a 'Save and continue' button. A red box highlights the 'Add a scope' button in the Scopes tab and the 'Application ID URI' input field.

6. Fill in the required fields and ensure that **State** is **Enabled**. Click **Add scope**.

Add a scope

Scope name * ⓘ
MarketingVPN
api://83ecfae4-f4dd-441b-9fe9-e6de68f44e73/MarketingVPN

Who can consent? ⓘ
Admins and users **Admins only**

Admin consent display name * ⓘ
Marketing Users

Admin consent description * ⓘ
Marketing department

User consent display name ⓘ
e.g. Read your files

User consent description ⓘ
e.g. Allows the app to read your files.

State ⓘ
Enabled Disabled

Add scope Cancel

7. Click **Expose an API** then + **Add a client application**. For **Client ID**, enter the following values depending on the cloud:

- Enter **41b23e61-6c1e-4545-b367-cd054e0ed4b4** for Azure **Public**
- Enter **51bb15d4-3a4f-4ebf-9dca-40096fe32426** for Azure **Government**
- Enter **538ee9e6-310a-468d-afef-ea97365856a9** for Azure **Germany**
- Enter **49f817b6-84ae-4cc0-928c-73f27289b3aa** for Azure **China 21Vianet**

8. Click **Add application**.

Add a client application

Client ID ⓘ
41b23e61-6c1e-4545-b367-cd054e0ed4b4

Authorized scopes ⓘ
 api://83ecfae4-f4dd-441b-9fe9-e6de68f44e73/MarketingVPN

Scopes defined by this API
Define custom scopes to restrict access to data API can request that a user or admin consent to

+ Add a scope

Scopes
api://83ecfae4-f4dd-441b-9fe9-e6de68f44e73

Authorized client applications
Authorizing a client application indicates that it this API.

+ Add a client application

Client Id
41b23e61-6c1e-4545-b367-cd054e0ed4b4

Add application Cancel

9. Copy the **Application (client) ID** from the **Overview** page. You will need this information to configure your VPN gateway(s).

Microsoft Azure | Search resources, services, and docs (G+/-)

Home > AliTestOrg - App registrations > MarketingVPN

MarketingVPN

Search (Ctrl+ /) Delete Endpoints

Overview Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name : MarketingVPN

Application (client) ID : 83ecfae4-f4dd-441b-9fe9-e6de68f44e73

Directory (tenant) ID : cf1f2e50-6414-4ee1-821b-29ad2fd82cf5

Object ID : 6836fd4a-96d2-4686-a57e-16adfd8be41

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

[View API permissions](#)

Sign in users in 5 minutes

Use our SDKs to sign in users and call APIs in a few steps

[View all quickstart guides](#)

10. Repeat the steps in this [register additional applications](#) section to create as many applications that are needed for your security requirement. Each application will be associated to a VPN gateway and can have a different set of users. Only one application can be associated to a gateway.

5. Assign users to applications

Assign the users to your applications.

- Under **Azure AD -> Enterprise applications**, select the newly registered application and click **Properties**. Ensure that **User assignment required?** is set to **yes**. Click **Save**.

MarketingVPN - Properties
Enterprise Application

Manage

- Properties** (highlighted)
- Owners
- Users and groups
- Provisioning
- Application proxy
- Self-service

Security

- Conditional Access
- Permissions
- Token encryption (Preview)

Activity

- Sign-ins
- Usage & insights (Preview)
- Audit logs
- Provisioning logs (Preview)
- Access reviews

Troubleshooting + Support

Enabled for users to sign-in? Yes (highlighted)

Name: MarketingVPN

Homepage URL: (empty)

Logo: (red square with MA logo)

Select a file

Application ID: 83ecfae4-f4dd-441b-9fe9-e6de68f44e73

Object ID: 52333356-911c-4ae9-9a36-6ca58074fed6

User assignment required? Yes (highlighted)

Visible to users? Yes

- On the app page, click **Users and groups**, and then click **+Add user**.

MarketingVPN - Users and groups
Enterprise Application

Manage

- Properties**
- Owners**
- Users and groups** (highlighted)
- Provisioning
- Application proxy
- Self-service

+ Add user

The application will appear on the Access Panel for assigned users. Set 'visible to users?' to Yes.

First 100 shown, to search all users & groups, enter a display name.

Display Name

No application assignments found

3. Under **Add Assignment**, click **Users and groups**. Select the users that you want to be able to access this VPN application. Click **Select**.

The screenshot shows the 'Add Assignment' dialog in the Azure portal. On the left, under 'Users and groups', 'None Selected' is listed. On the right, a list of users is shown with their icons and details. Two users are highlighted with a red box and labeled 'Selected': 'VPN User 1' (pink icon) and 'VPN User 2' (teal icon). At the bottom right of the dialog, the 'Select' button is also highlighted with a red box.

6. Enable authentication on the gateway

In this step, you enable Azure AD authentication on the VPN gateway.

1. Enable Azure AD authentication on the VPN gateway by running the following commands. Be sure to modify the commands to reflect your own environment:

```
$gw = Get-AzVirtualNetworkGateway -Name <name of VPN gateway> -ResourceGroupName <Resource group>
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -VpnClientRootCertificates @()
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -AadTenantUri
"https://login.microsoftonline.com/<your Directory ID>" -AadAudienceId "application ID from previous
section" -AadIssuerUri "https://sts.windows.net/<your Directory ID>/" -VpnClientAddressPool
192.168.0.0/24
```

NOTE

Do not use the Azure VPN client's application ID in the commands above: It will grant all users access to the VPN gateway. Use the ID of the application(s) you registered.

2. Create and download the profile by running the following commands. Change the -ResourcGroupName and -Name values to match your own.

```
$profile = New-AzVpnClientConfiguration -Name <name of VPN gateway> -ResourceGroupName <Resource group>
-AuthenticationMethod "EapTls"
$PROFILE.VpnProfileSASUrl
```

3. After running the commands, you see a result similar to the one below. Copy the result URL to your browser to download the profile zip file.

```
PS C:\WINDOWS\system32> $PROFILE.VpnProfileSASUrl  
https://nfvprodsspmwh.blob.core.windows.net/vpnprofileimmutable/acfd0032-3d0e-4aa2-9586-dd15d89ea0f8/vpnprofile/e284f74d-ea23-4b79-985a-cdd8a9ca  
qVdf7Rx2vwvy6MQ%3D&st=2019-09-06T19%3A46%3A29Z&se=2019-09-06T20%3A46%3A29Z&sp=r&fileExtension=.zip
```

4. Extract the downloaded zip file.
5. Browse to the unzipped “AzureVPN” folder.
6. Make a note of the location of the “azurevpnconfig.xml” file. The azurevpnconfig.xml contains the setting for the VPN connection and can be imported directly into the Azure VPN Client application. You can also distribute this file to all the users that need to connect via e-mail or other means. The user will need valid Azure AD credentials to connect successfully.

Next steps

In order to connect to your virtual network, you must create and configure a VPN client profile. See [Configure a VPN client for P2S VPN connections](#).

Enable Azure Multi-Factor Authentication (MFA) for VPN users

2/19/2020 • 2 minutes to read • [Edit Online](#)

If you want users to be prompted for a second factor of authentication before granting access, you can configure Azure Multi-Factor Authentication (MFA). You can configure MFA on a per user basis, or you can leverage MFA via [Conditional Access](#).

- MFA per user can be enabled at no-additional cost. When enabling MFA per user, the user will be prompted for second factor authentication against all applications tied to the Azure AD tenant. See [Option 1](#) for steps.
- Conditional Access allows for finer-grained control over how a second factor should be promoted. It can allow assignment of MFA to only VPN, and exclude other applications tied to the Azure AD tenant. See [Option 2](#) for steps.

Enable authentication

1. Navigate to **Azure Active Directory -> Enterprise applications -> All applications**.

2. On the **Enterprise applications - All applications** page, select **Azure VPN**.

The screenshot shows the Azure portal interface. On the left, there's a sidebar with various service icons and a red box around the 'Azure Active Directory' icon. Below it, another red box surrounds the 'Enterprise applications' link under the 'Manage' section. The main content area is titled 'Enterprise applications - All applications'. It has sections for Overview, Manage, Security, Activity, and Troubleshooting + Support. Under 'Manage', the 'All applications' link is also highlighted with a red box. In the main table, there's one entry for 'Azure VPN', which is also highlighted with a red box. The table columns include Name, Application Type (Enterprise Applications), Applications status (Any), Application visibility (Any), and Homepage URL (https://www.microsoft.com).

Configure sign-in settings

On the **Azure VPN - Properties** page, configure sign-in settings.

1. Set **Enabled for users to sign-in?** to **Yes**. This setting allows all users in the AD tenant to connect to the VPN successfully.
2. Set **User assignment required?** to **Yes** if you want to limit sign-in to only users that have permissions to the Azure VPN.
3. Save your changes.

Azure VPN - Properties

Enterprise Application

Save Discard Delete

Enabled for users to sign-in? Yes No

Name: Azure VPN

Homepage URL: https://www.microsoft.com

Logo: A red shield with four white arrows pointing outwards.

Application ID: 41b23e61-6c1e-4545-b367-cd054e0ed4b4

Object ID: fdef36b3-187d-47e7-a774-8d16554dea15

User assignment required? Yes No

Visible to users? Yes No

Option 1 - Per User access

Open the MFA page

1. Sign in to the Azure portal.
2. Navigate to **Azure Active Directory** -> **All users**.
3. Select **Multi-Factor Authentication** to open the multi-factor authentication page.

Home > AllTestOrg > Users - All users

Users - All users

All users

Multi-Factor Authentication

Name	User type	Source
AD Admin	Member	Azure Active Directory
AF	Member	External Azure Active D
VU VPN User 1	Member	Azure Active Directory
VU VPN User 2	Member	Azure Active Directory

Select users

1. On the **multi-factor authentication** page, select the user(s) for whom you want to enable MFA.
2. Select **Enable**.

The screenshot shows a table of users with their multi-factor authentication status. The columns are 'DISPLAY NAME', 'USER NAME', and 'MULTI-FACTOR AUTH STATUS'. The rows include 'Admin' (Disabled), 'VPN User 1' (Disabled, highlighted with a red box around the checkbox), and 'VPN User 2' (Enforced). To the right of the table, there is a summary for 'VPN User 1' with fields for 'quick steps' (containing 'Enable' which is also highlighted with a red box) and 'Manage user settings'.

DISPLAY NAME	USER NAME	MULTI-FACTOR AUTH STATUS
Admin		Disabled
VPN User 1	vpn1@alitestorg.onmicrosoft.com	Disabled
VPN User 2		Enforced

Option 2 - Conditional Access

Conditional Access allows for fine-grained access control on a per-application basis. In order to use Conditional Access, you should have Azure AD Premium 1 or greater licensing applied to the users that will be subject to the Conditional Access rules.

1. Navigate to the **Enterprise applications - All applications** page and click **Azure VPN**.
 - Click **Conditional Access**.
 - Click **New policy** to open the **New** pane.
2. On the **New** pane, navigate to **Assignments -> Users and groups**. On the **Users and groups -> Include** tab:
 - Click **Select users and groups**.
 - Check **Users and groups**.
 - Click **Select** to select a group or set of users to be affected by MFA.
 - Click **Done**.

The screenshot shows the Microsoft Azure portal interface for creating a new Azure VPN - Conditional Access policy. The left sidebar contains various service links like Home, Dashboard, All services, and Favorites. The main area has a breadcrumb path: Home > jackstromberg > Enterprise applications - All applications > Azure VPN - Conditional Access > New > Users and groups. The 'New' pane on the left has sections for Info, Name (set to 'VPN Policy'), Assignments (highlighted with a red box), Conditions, Access controls, and Enable policy (Report-only is On). The 'Users and groups' pane on the right shows options for Include or Exclude, with 'Select users and groups' selected (highlighted with a red box). Under 'Select', 'Users and groups' is checked (highlighted with a red box), and 'VPN Users' is selected (highlighted with a red box). A 'Done' button at the bottom right is also highlighted with a red box.

3. On the **New** pane, navigate to the **Access controls** -> **Grant** pane:

- Click **Grant access**.
- Click **Require multi-factor authentication**.
- Click **Require all the selected controls**.
- Click **Select**.

Microsoft Azure

Search resources, services, and docs (G+)

Home > jackstromberg > Enterprise applications - All applications > Azure VPN - Conditional Access > New > Grant

Create a resource

Home

Dashboard

All services

FAVORITES

Resource groups

Azure Active Directory

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Security Center

Subscriptions

Monitor

Help + support

Advisor

Cost Management + Billing

New

Info

Name * ✓

Assignments

Users and groups 1 >

Specific users included

Cloud apps or actions 1 >

1 app included

Conditions 0 >

0 conditions selected

Access controls

Grant 0 >

0 controls selected

Session 0 >

0 controls selected

Enable policy

Report-only On Off

Grant

Select the controls to be enforced.

Block access

Grant access

Require multi-factor authentication ⓘ

Require device to be marked as compliant ⓘ

Require Hybrid Azure AD joined device ⓘ

Require approved client app ⓘ
[See list of approved client apps](#)

Require app protection policy (Preview) ⓘ
[See list of policy protected client apps](#)

For multiple controls

Require all the selected controls

Require one of the selected controls

4. In the **Enable policy** section:

- Select **On**.
- Click **Create**.

Microsoft Azure

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

Home > jackstromberg > Enterprise applications - All applications > Azure VPN - Conditional Access > New

Create a resource

Home

Dashboard

All services

FAVORITES

Resource groups

Azure Active Directory

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Security Center

Subscriptions

Monitor

Help + support

Advisor

Cost Management + Billing

New

Info

Name *

VPN Policy

Assignments

Users and groups >

Specific users included

Cloud apps or actions >

1 app included

Conditions >

0 conditions selected

Access controls

Grant >

1 control selected

Session >

0 controls selected

Enable policy

Report-only **On** Off

Create

The screenshot shows the 'New' dialog for creating a VPN Policy in Microsoft Azure. The 'Name' field is filled with 'VPN Policy'. Under 'Assignments', there are sections for 'Users and groups' and 'Cloud apps or actions', each with one item selected. Under 'Access controls', there is a 'Grant' section with one control selected. In the 'Enable policy' section, the 'Report-only' radio button is selected. A red box highlights the 'Create' button at the bottom of the dialog.

Next steps

To connect to your virtual network, you must create and configure a VPN client profile. See [Configure a VPN client for P2S VPN connections](#).

Configure a VPN client for P2S OpenVPN protocol connections: Azure AD authentication

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

This article helps you configure a VPN client to connect to a virtual network using Point-to-Site VPN and Azure Active Directory authentication. Before you can connect and authenticate using Azure AD, you must first configure your Azure AD tenant. For more information, see [Configure an Azure AD tenant](#).

NOTE

Azure AD authentication is supported only for OpenVPN® protocol connections.

Working with client profiles

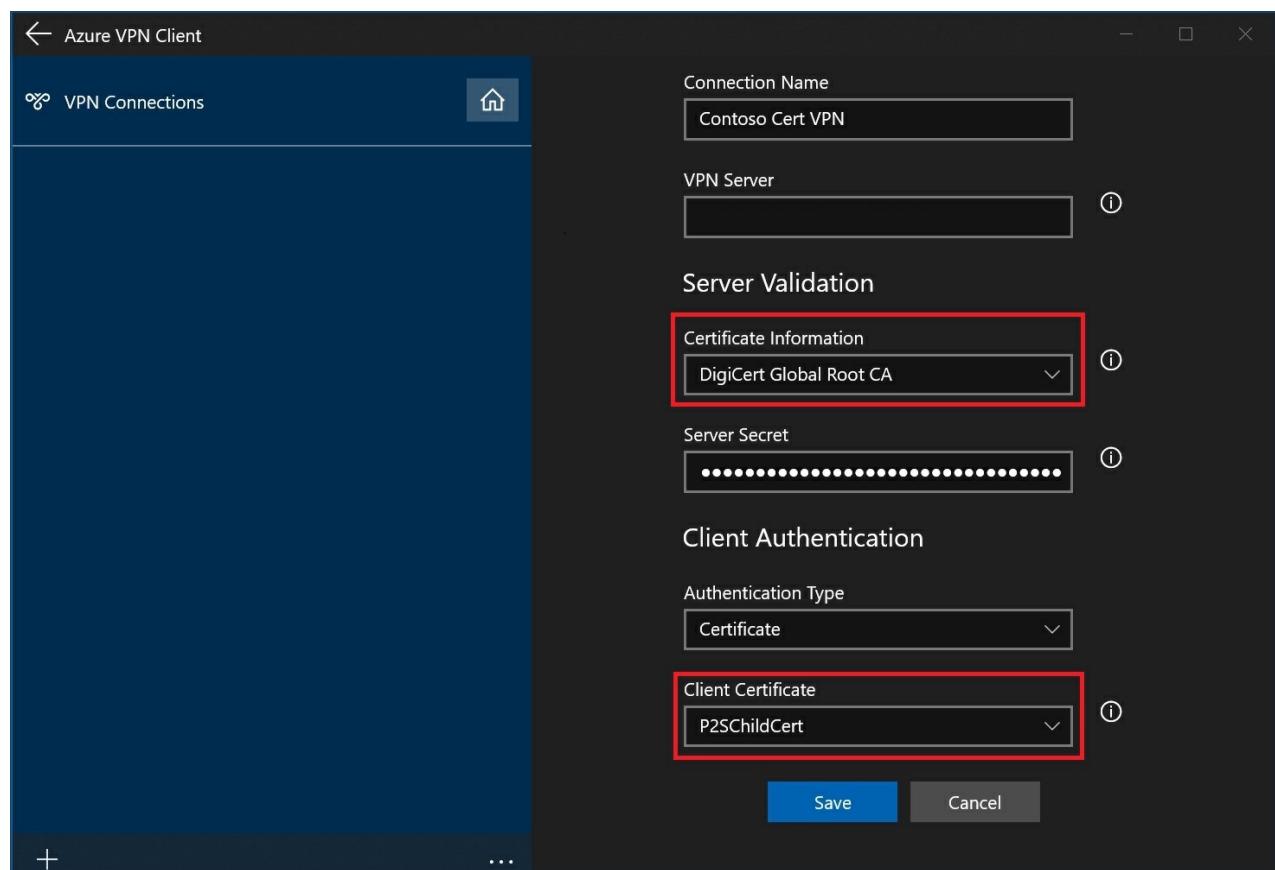
To connect, you need to download the Azure VPN Client and configure a VPN client profile on every computer that wants to connect to the VNet. You can create a client profile on a computer, export it, and then import it to additional computers.

To download the Azure VPN client

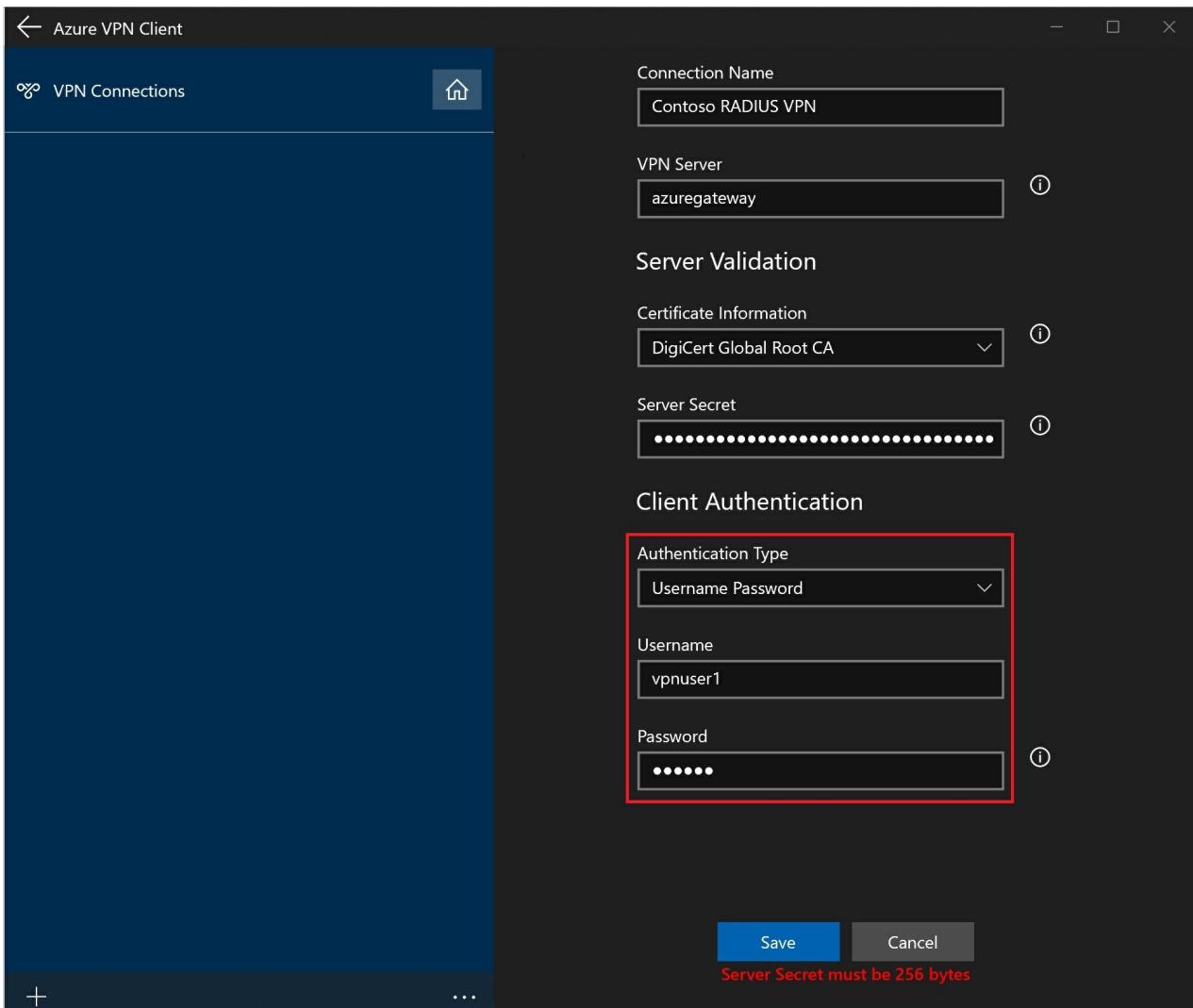
Use this [link](#) to download the Azure VPN Client.

To create a certificate-based client profile

When working with a certificate-based profile, make sure that the appropriate certificates are installed on the client computer. For more information about certificates, see [Install client certificates](#).



To create a RADIUS client profile



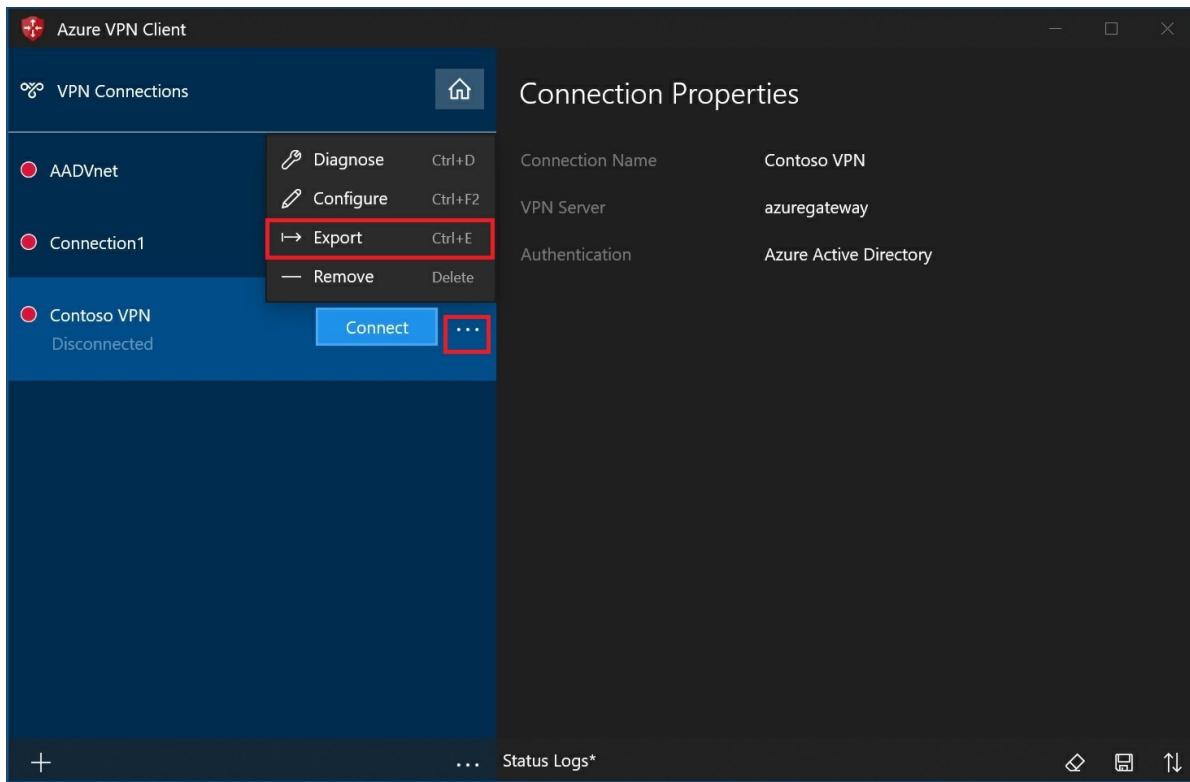
NOTE

The Server Secret can be exported in the P2S VPN client profile. Instructions on how to export a client profile can be found [here](#).

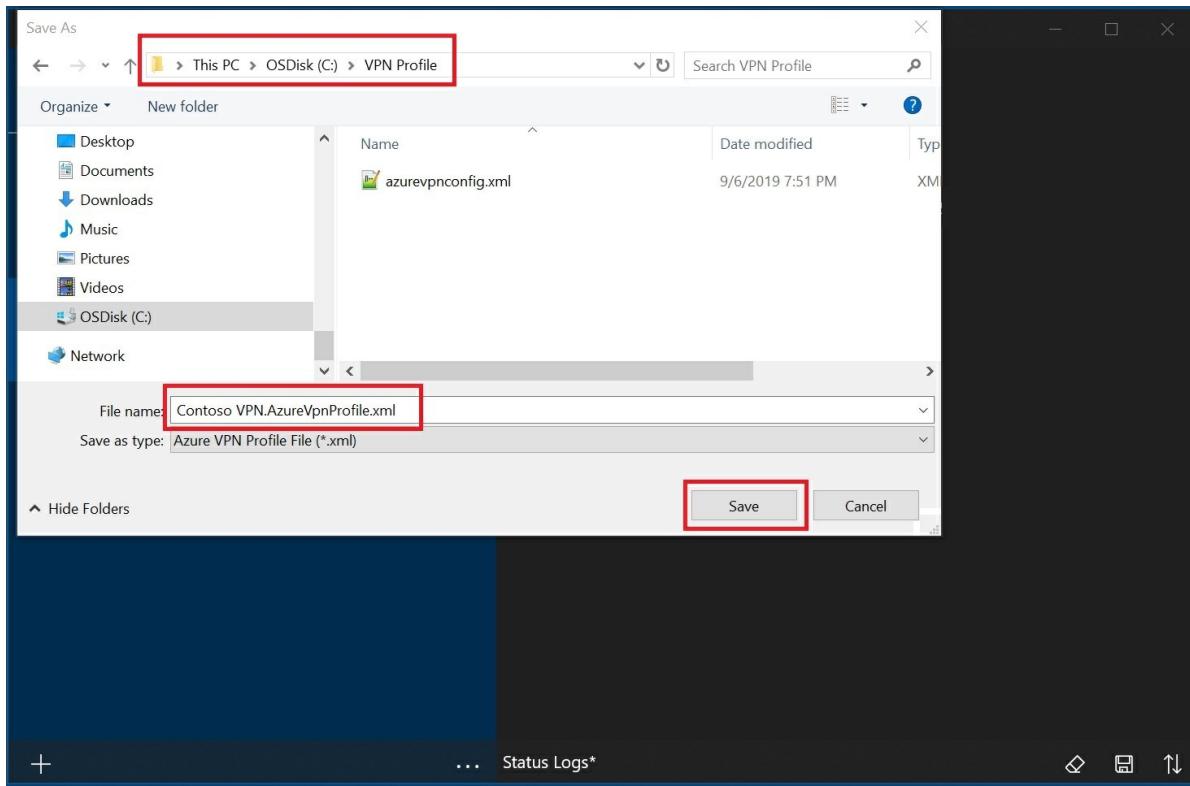
To export and distribute a client profile

Once you have a working profile and need to distribute it to other users, you can export it using the following steps:

1. Highlight the VPN client profile that you want to export, select the ..., then select **Export**.

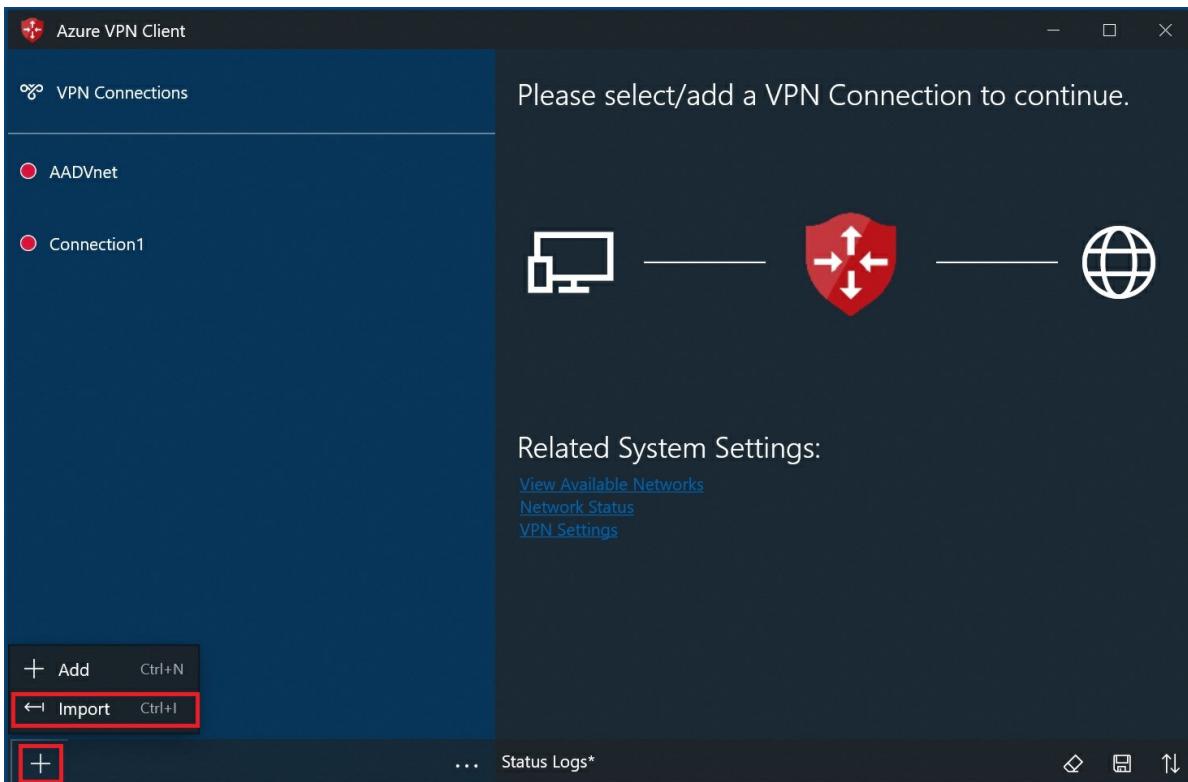


2. Select the location that you want to save this profile to, leave the file name as is, then select **Save** to save the xml file.

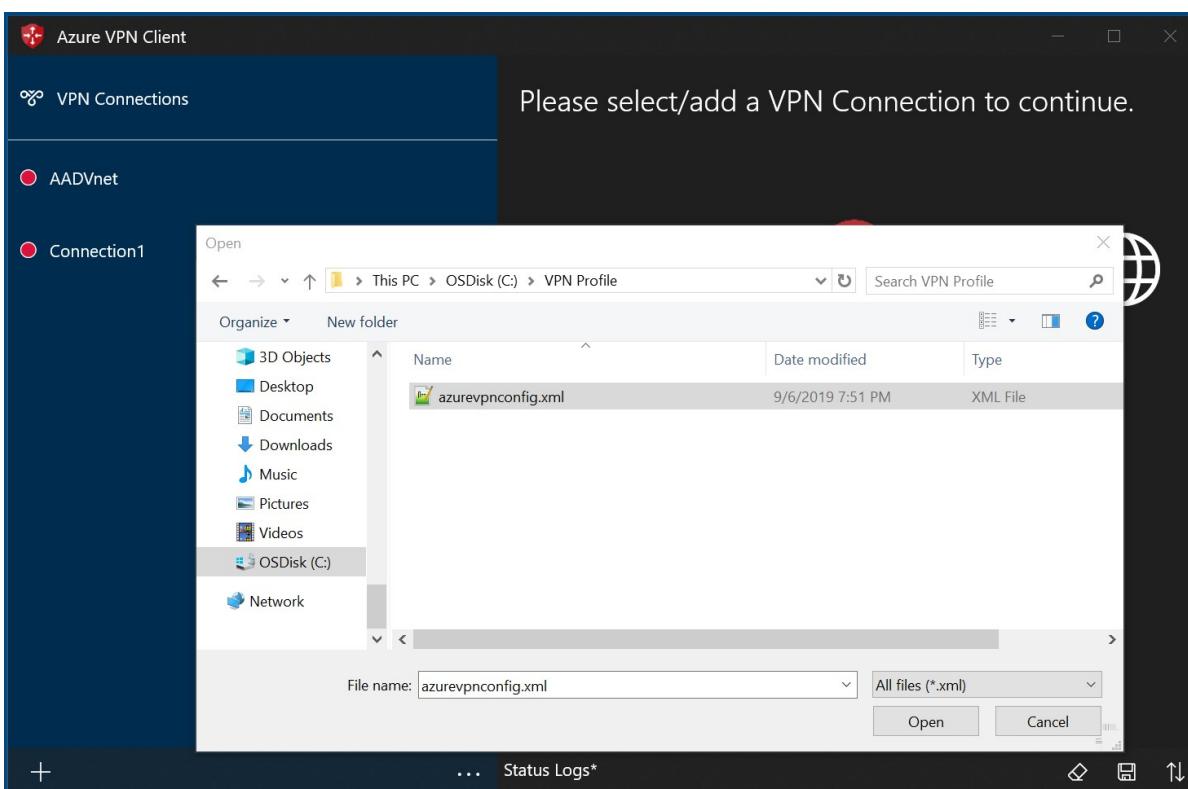


To import a client profile

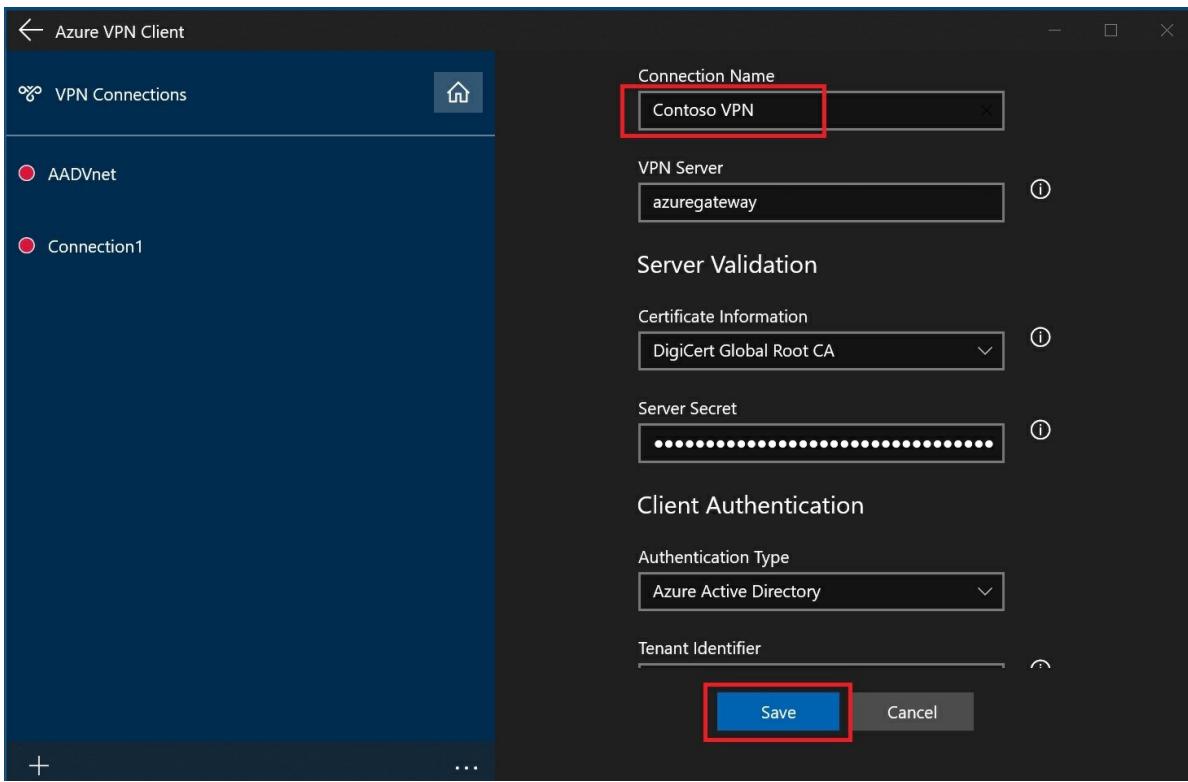
1. On the page, select **Import**.



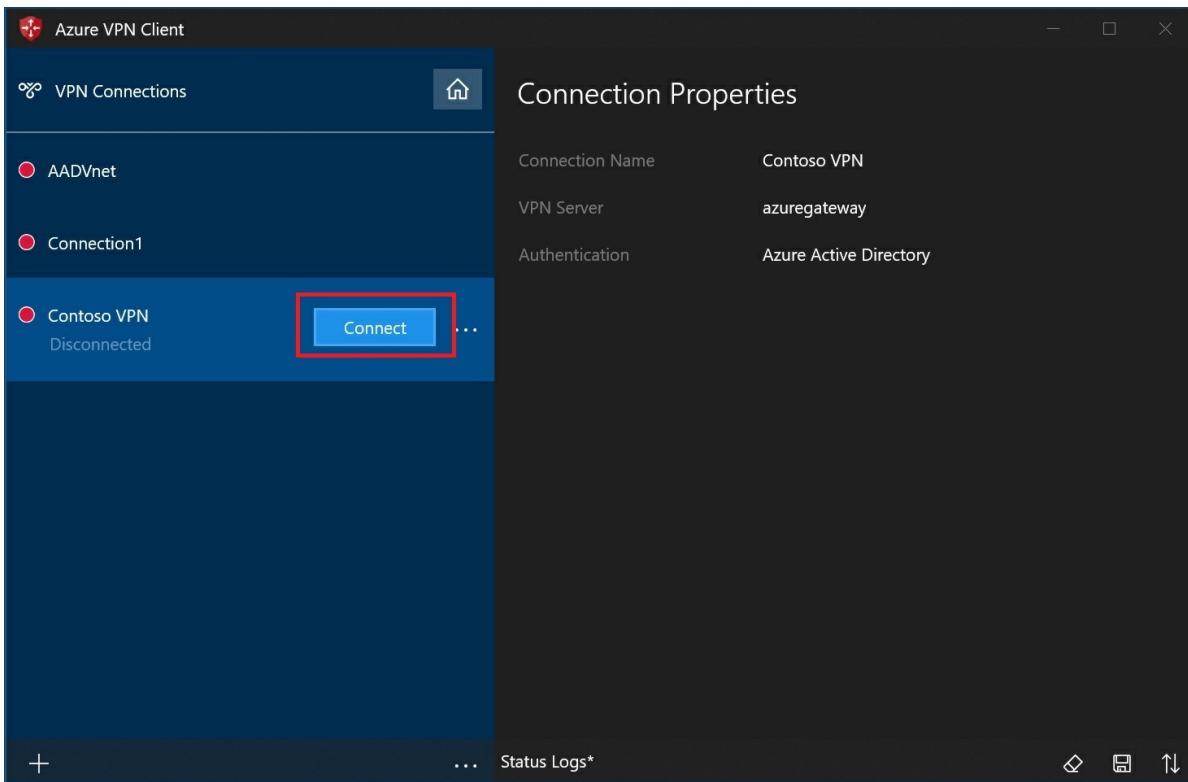
2. Browse to the profile xml file and select it. With the file selected, select **Open**.



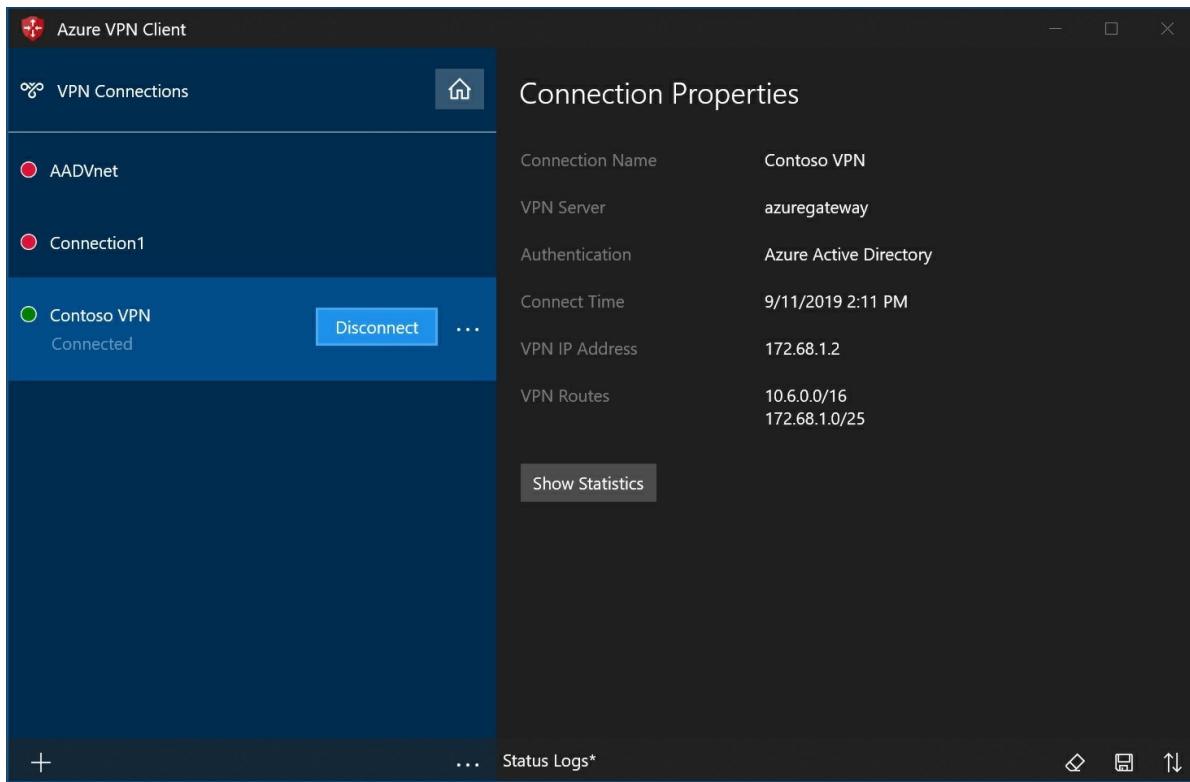
3. Specify the name of the profile and select **Save**.



4. Select **Connect** to connect to the VPN.

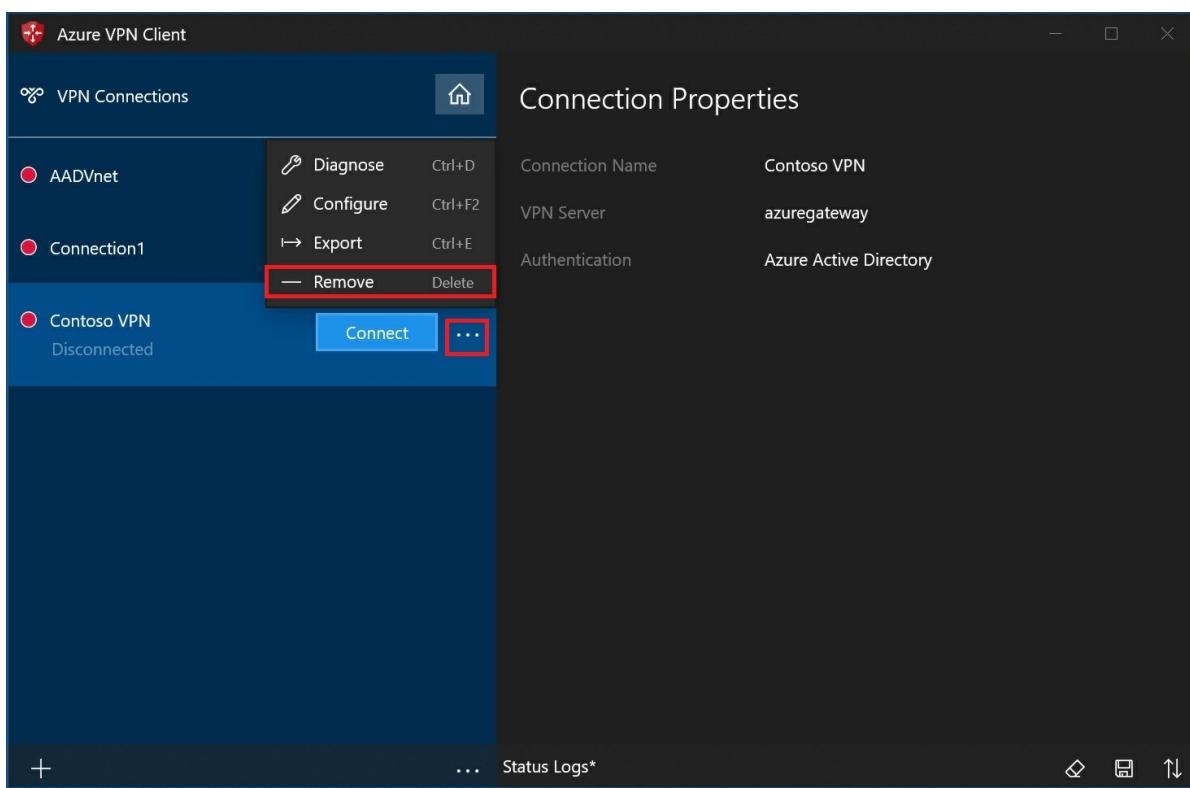


5. Once connected, the icon will turn green and say **Connected**.

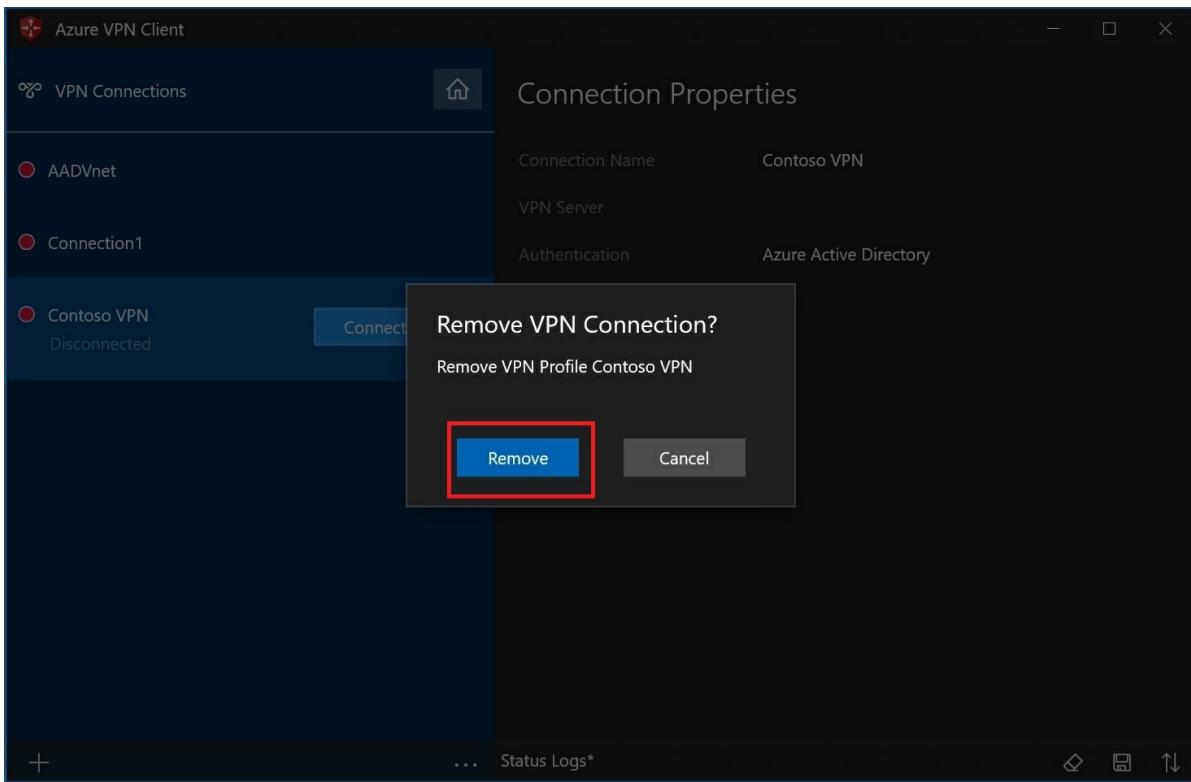


To delete a client profile

1. Select the ellipses next to the client profile that you want to delete. Then, select **Remove**.

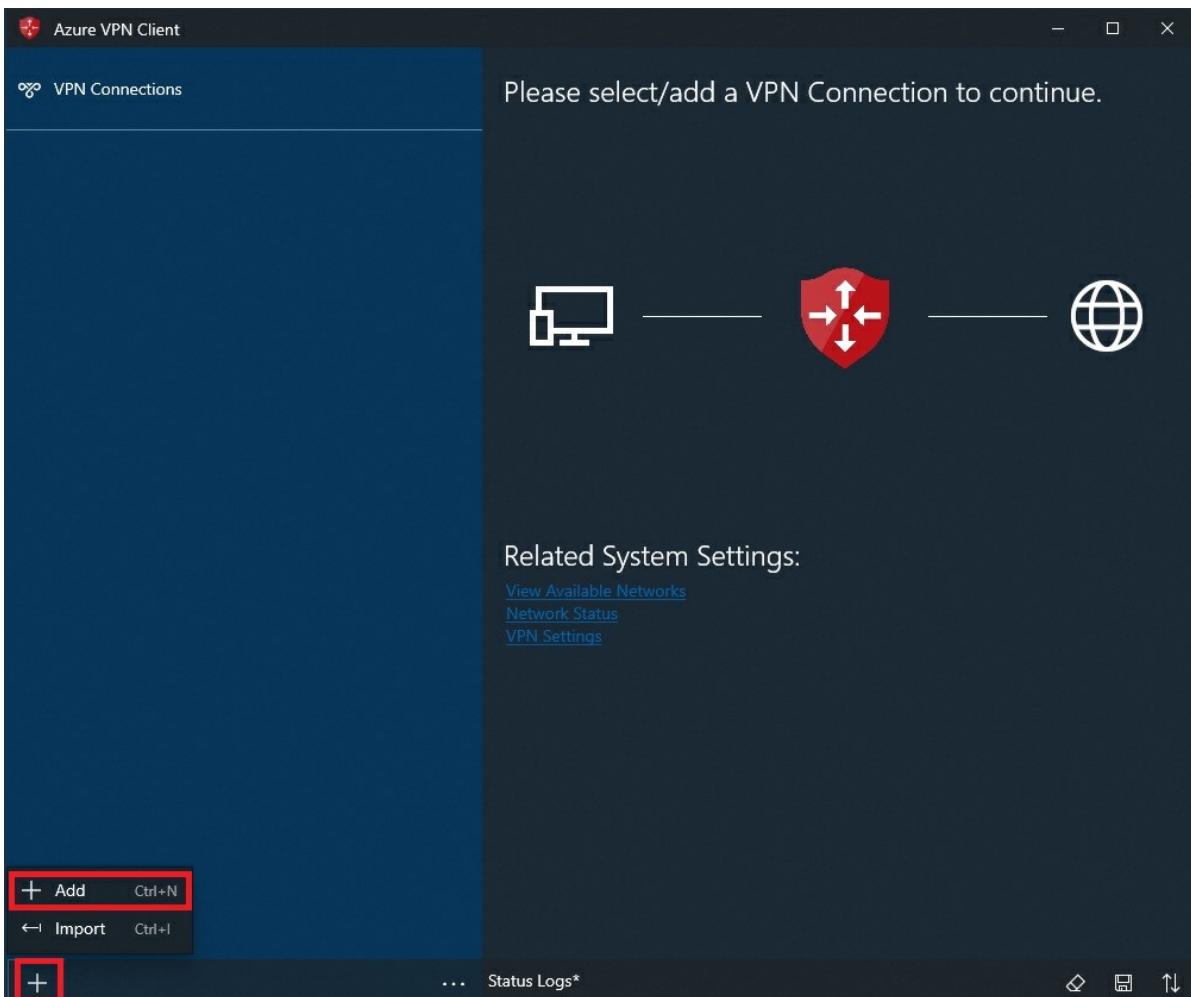


2. Select **Remove** to delete.

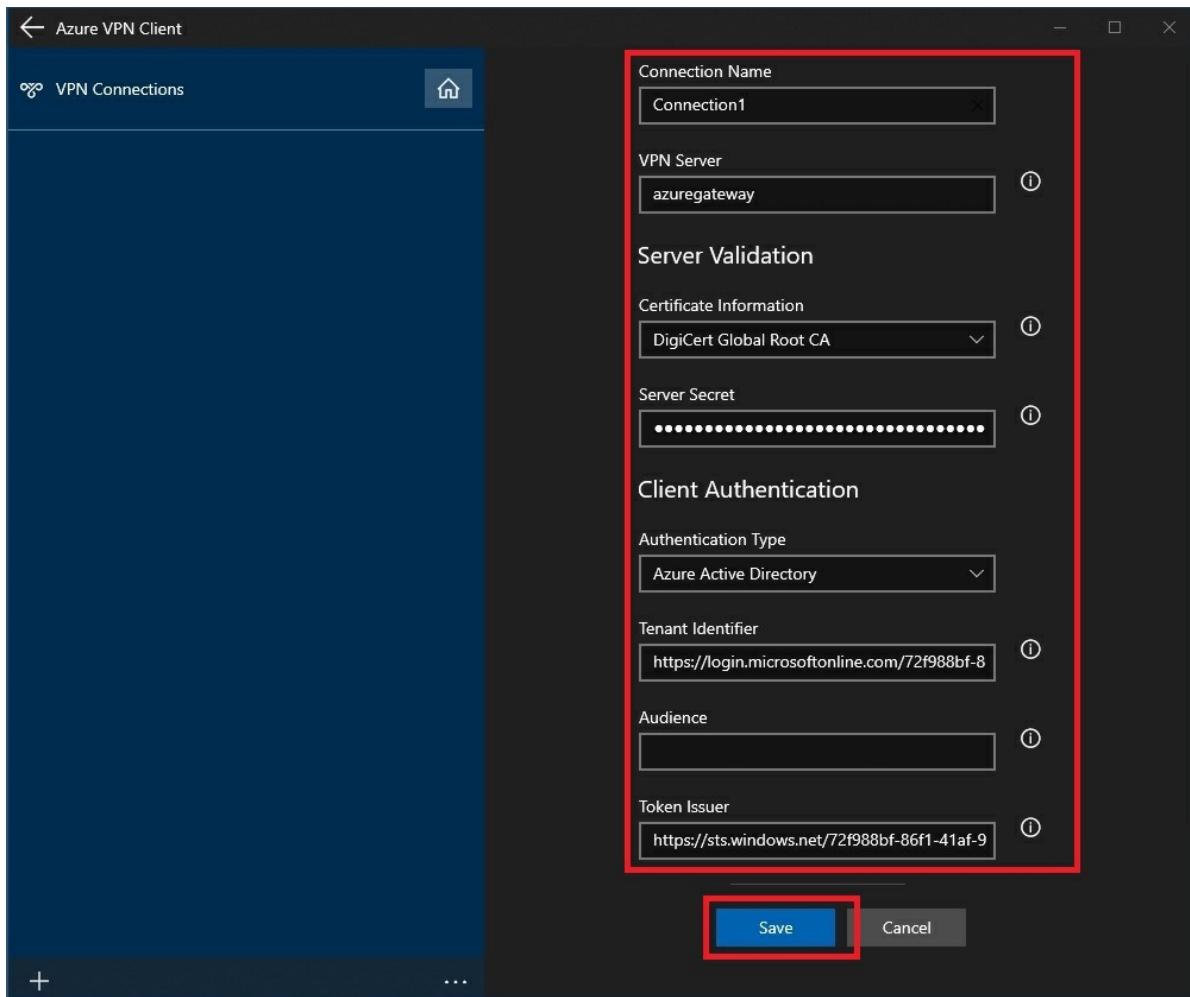


Create a connection

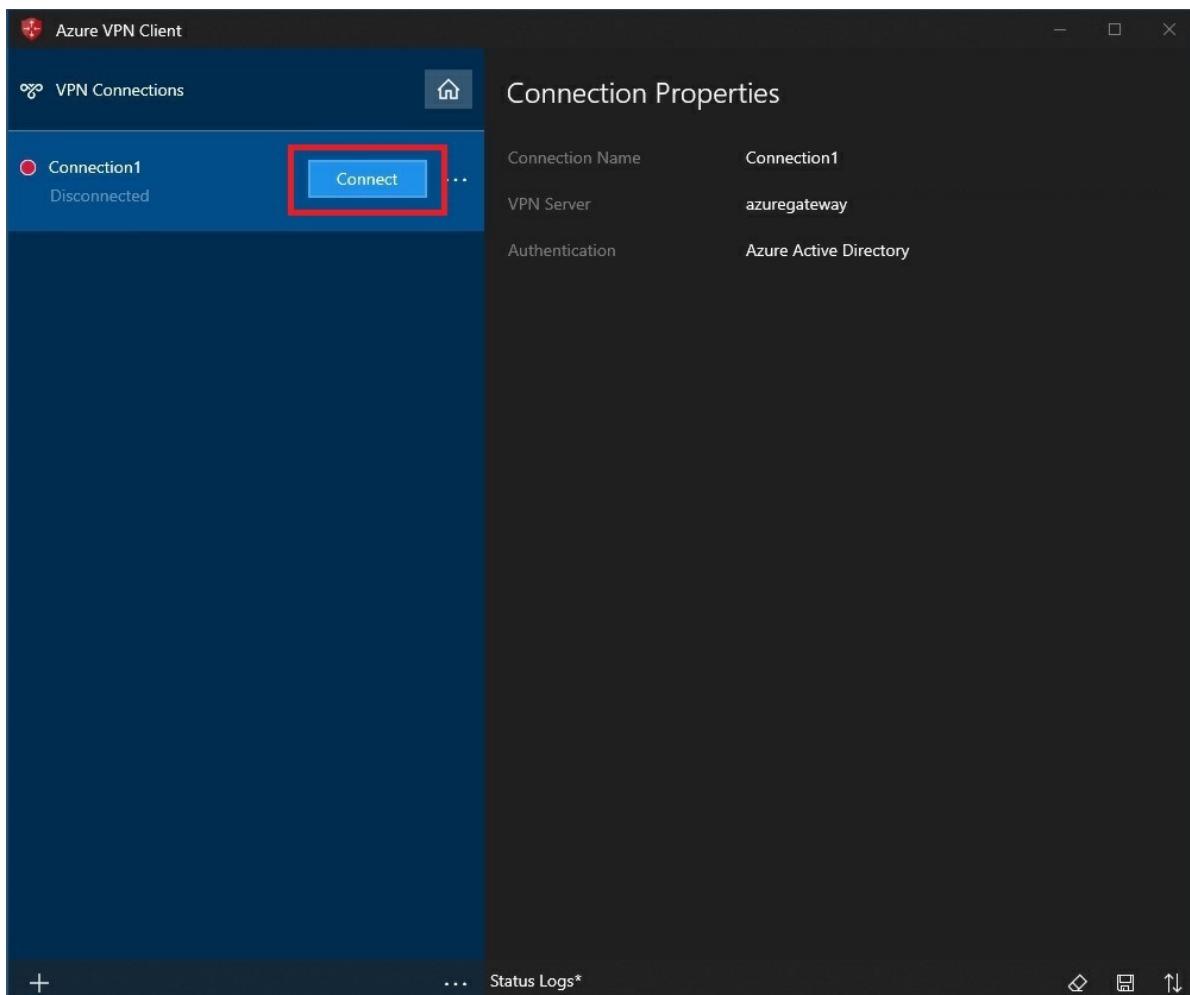
1. On the page, select +, then + **Add**.



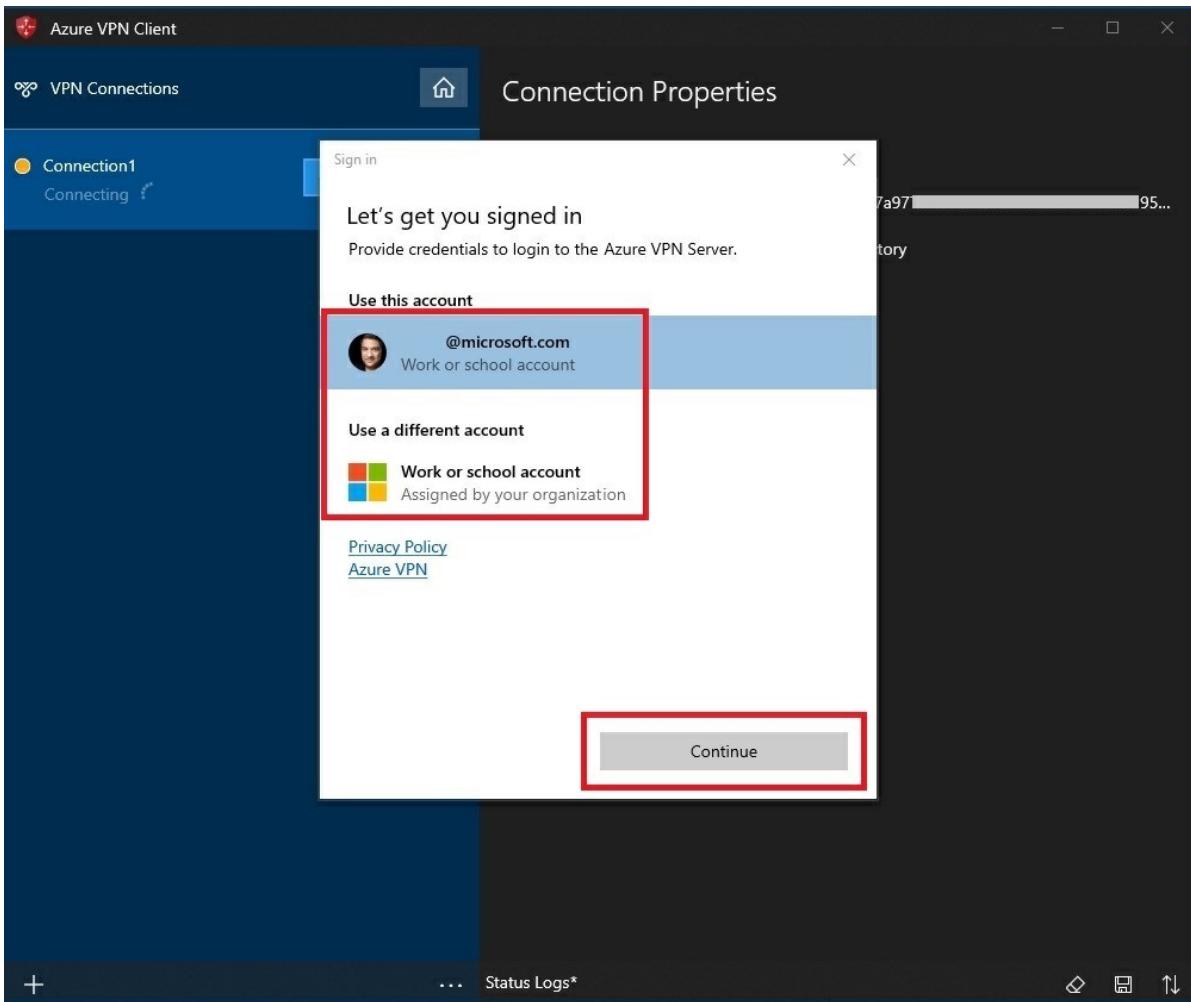
2. Fill out the connection information. If you are unsure of the values, contact your administrator. After filling out the values, select **Save**.



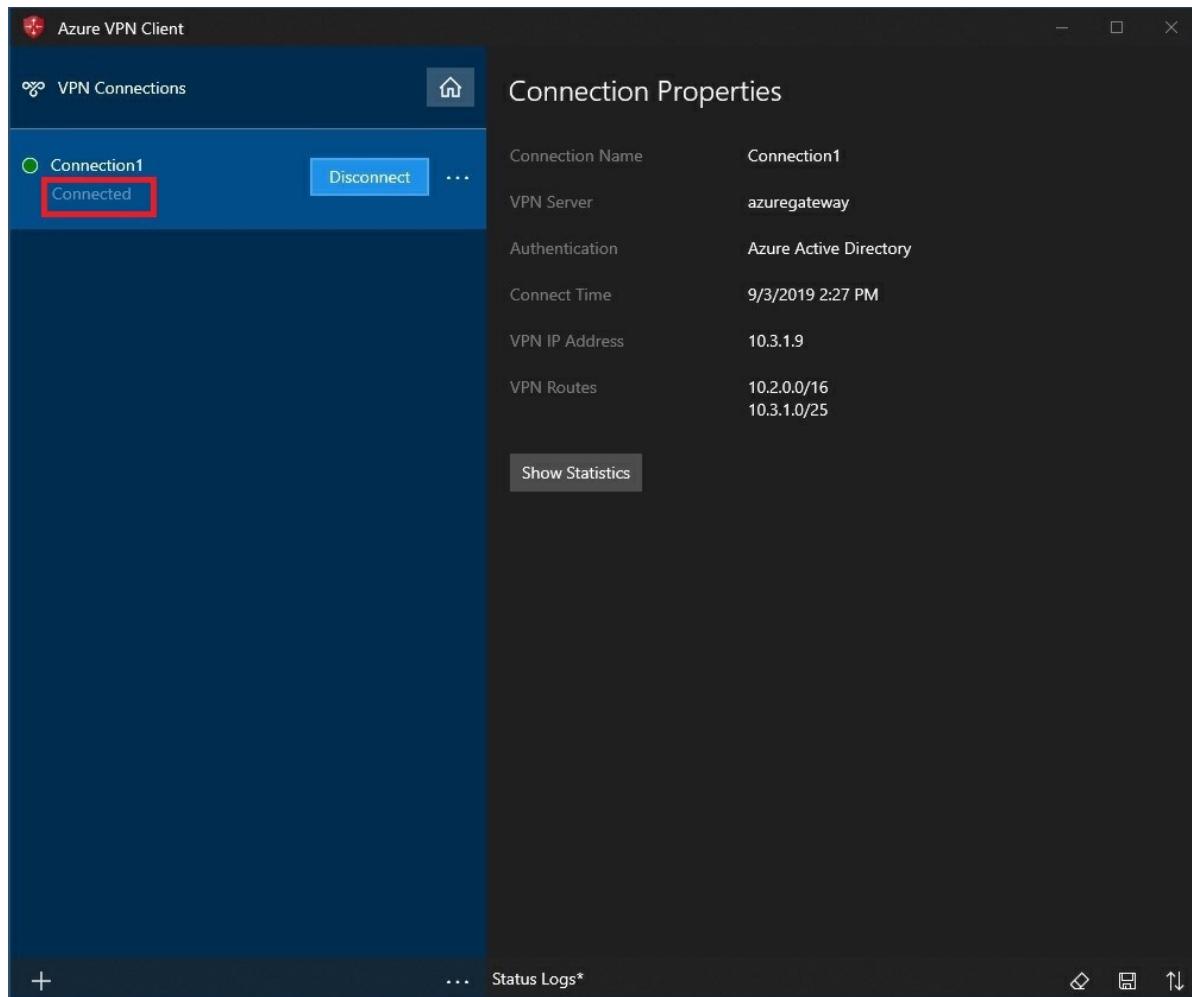
3. Select **Connect** to connect to the VPN.



4. Select the proper credentials, then select **Continue**.



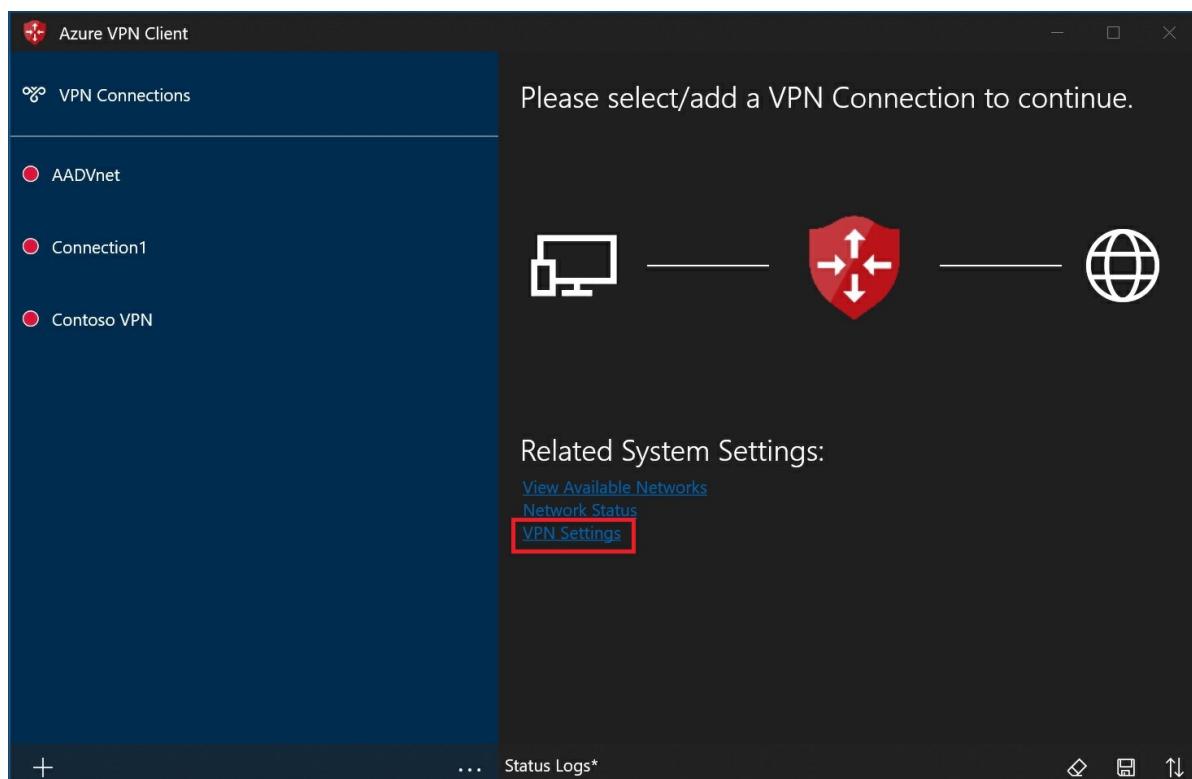
5. Once successfully connected, the icon will turn green and say **Connected**.



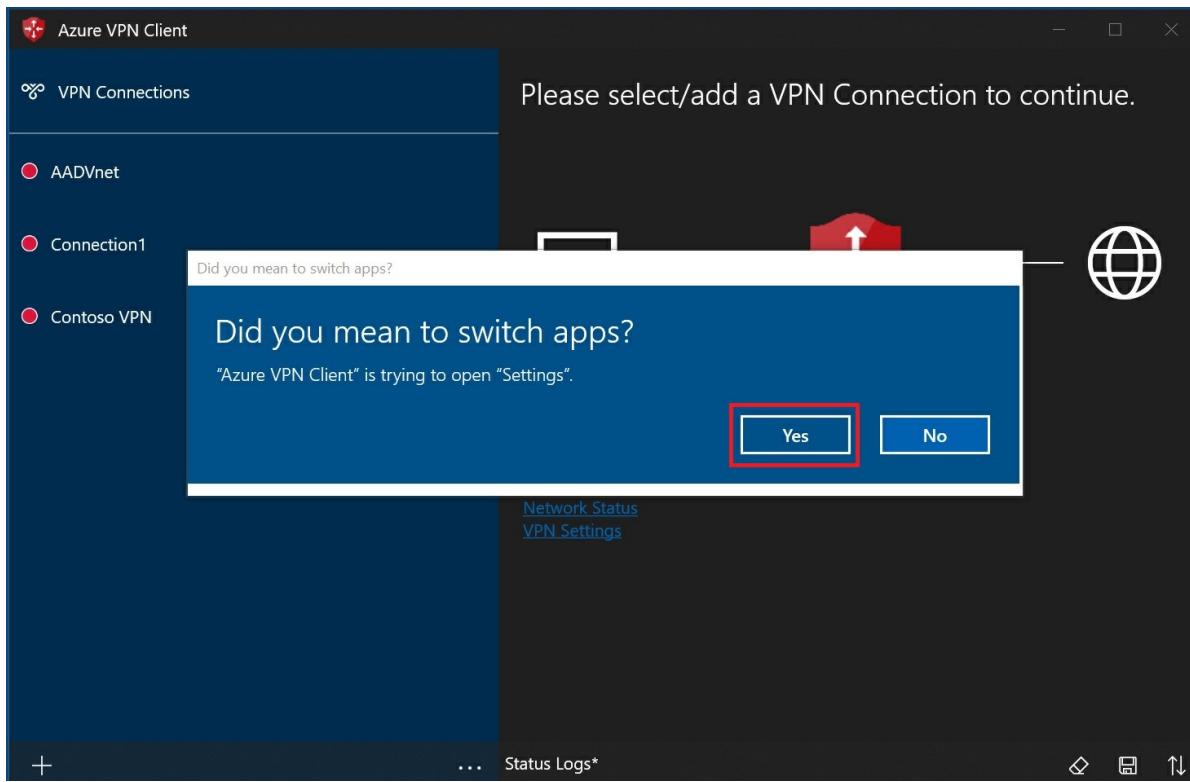
To connect automatically

These steps help you configure your connection to connect automatically with Always-on.

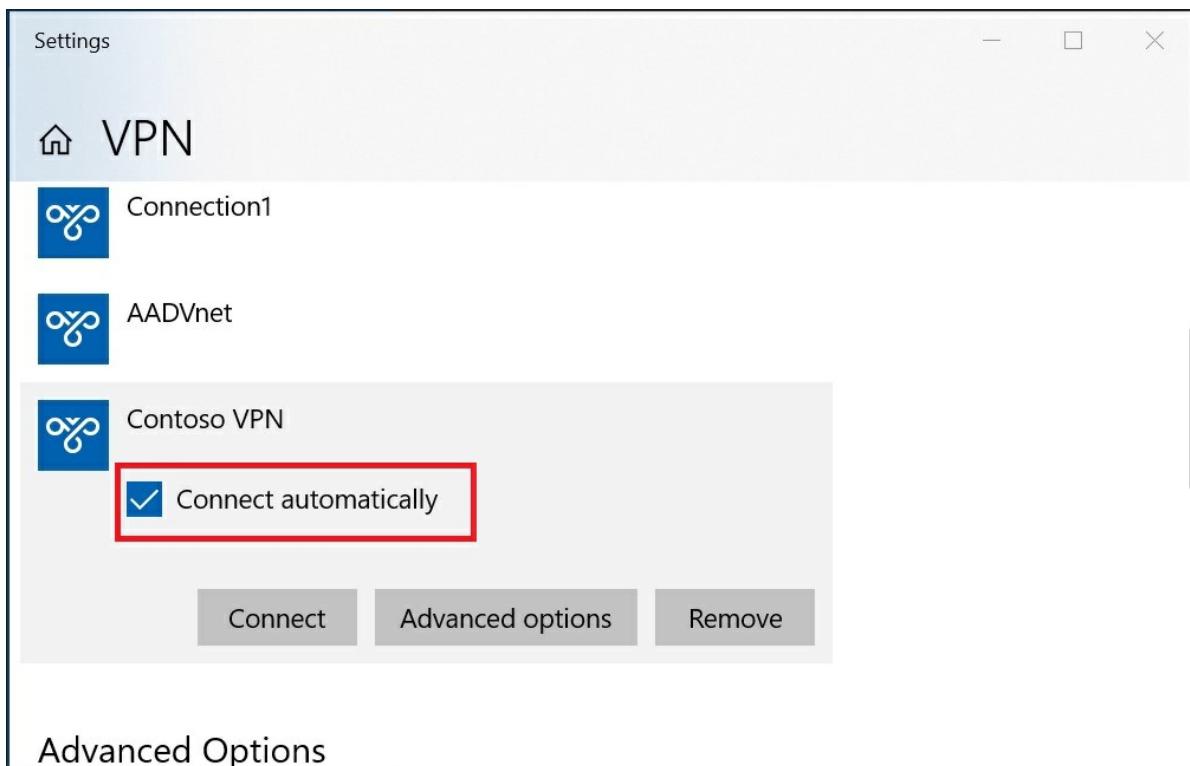
1. On the home page for your VPN client, select **VPN Settings**.



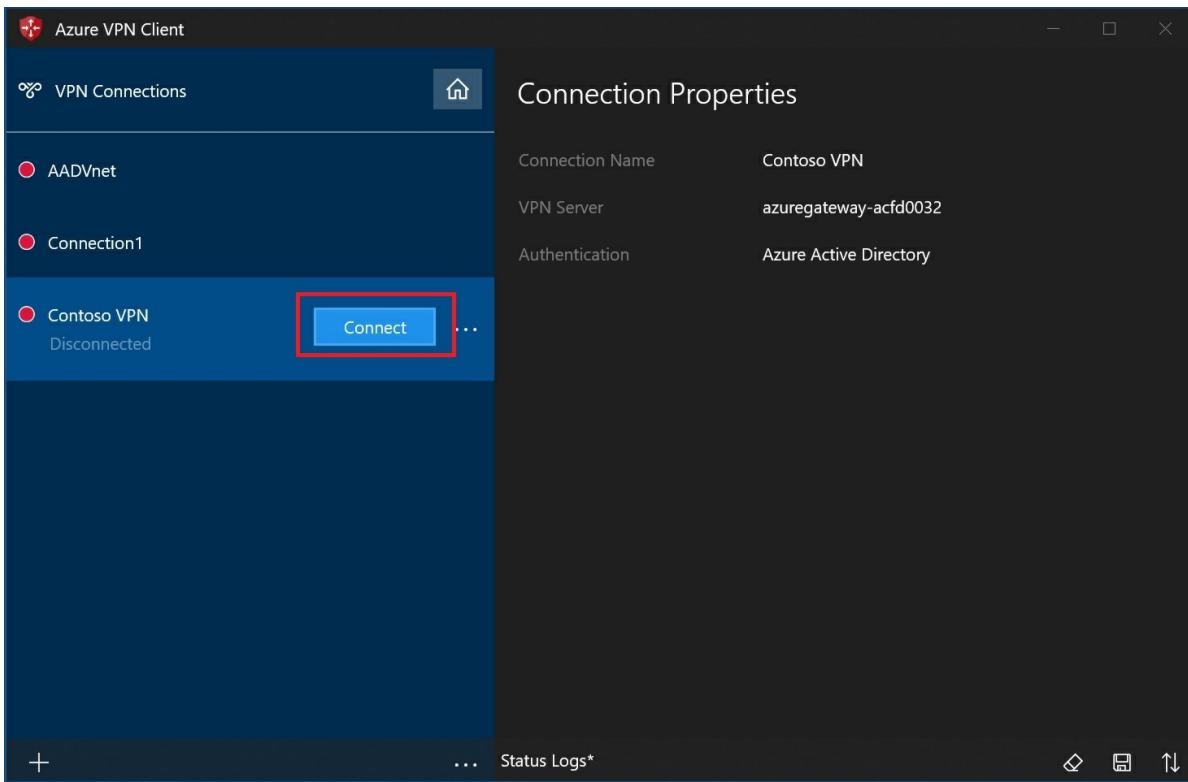
2. Select **Yes** on the switch apps dialogue box.



3. Make sure the connection that you want to set is not already connected, then highlight the profile and check the **Connect automatically** check box.

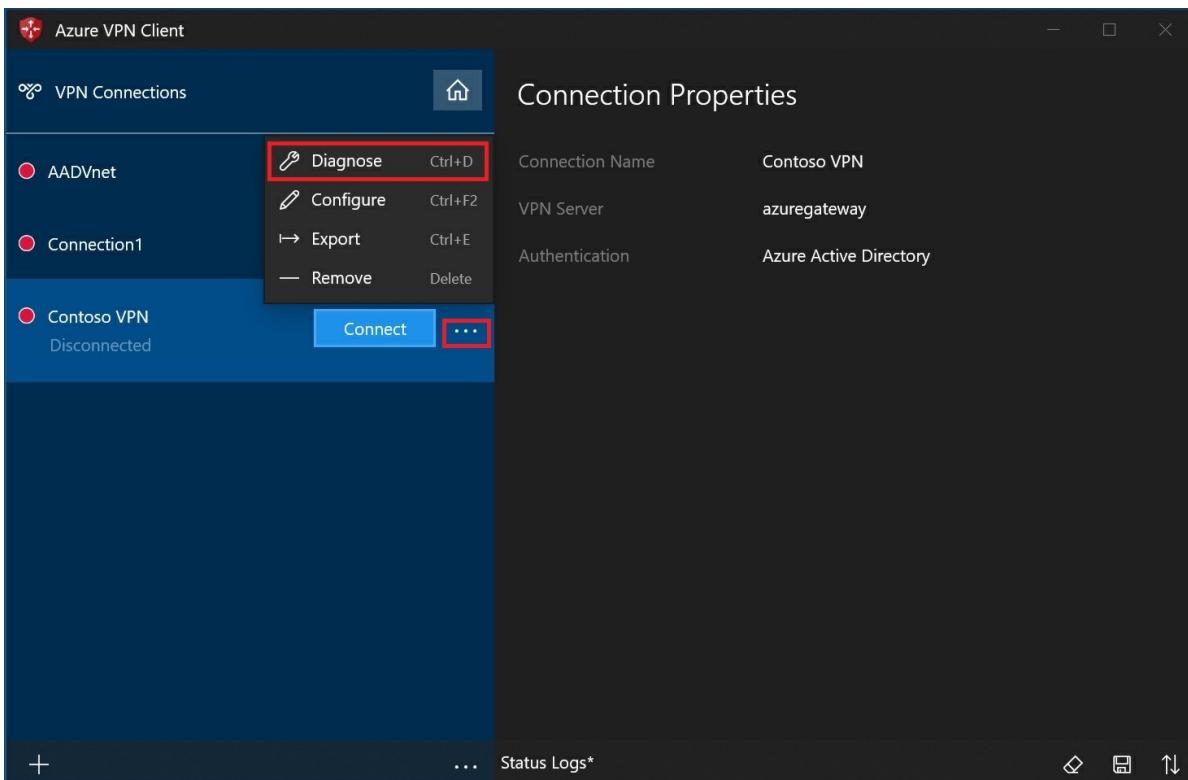


4. Select **Connect** to initiate the VPN connection.

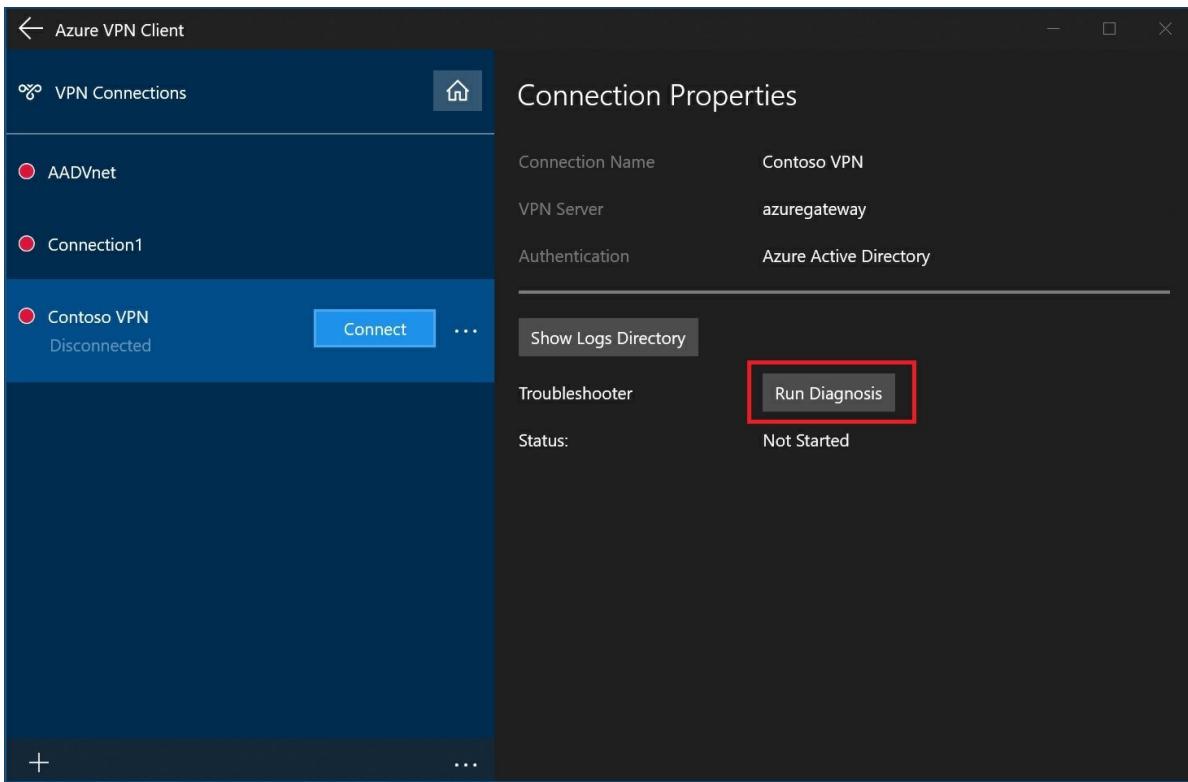


Diagnose connection issues

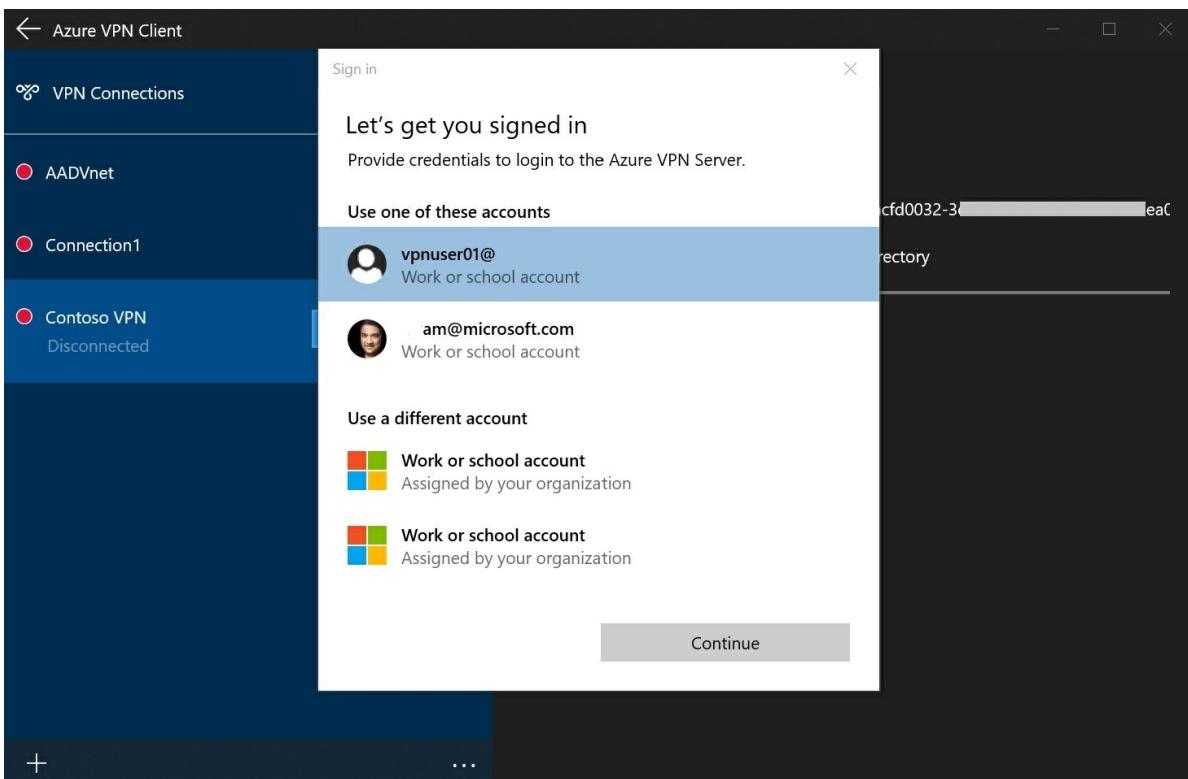
1. To diagnose connection issues, you can use the **Diagnose** tool. Select the ... next to the VPN connection that you want to diagnose to reveal the menu. Then select **Diagnose**.



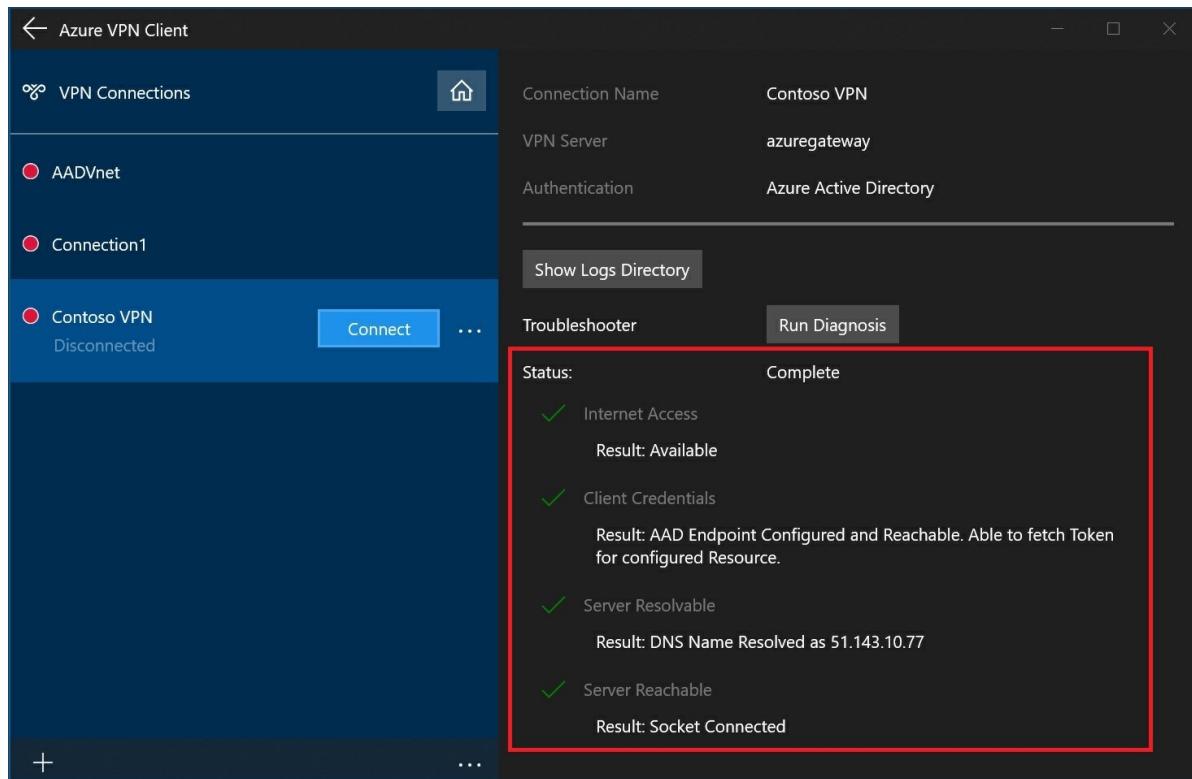
2. On the **Connection Properties** page, select **Run Diagnosis**.



3. Sign in with your credentials.



4. View the diagnosis results.



FAQ

How do I add DNS suffixes to the VPN client?

You can modify the downloaded profile XML file and add the `<dnssuffixes><dnssuffix> </dnssuffix>` `</dnssuffixes>` tags

```
<azvpnprofile>
<clientconfig>

    <dnssuffixes>
        <dnssuffix>.mycorp.com</dnssuffix>
        <dnssuffix>.xyz.com</dnssuffix>
        <dnssuffix>.etc.net</dnssuffix>
    </dnssuffixes>

</clientconfig>
</azvpnprofile>
```

How do I add custom routes to the VPN client?

You can modify the downloaded profile XML file and add the `<route><includeroutes><destination><mask>` `</route></includeroutes></destination></mask>` tags

```
<azvpnprofile>
<clientconfig>

    <includeroutes>
        <route>
            <destination>x.x.x.x</destination><mask>24</mask>
        </route>
    </includeroutes>

</clientconfig>
</azvpnprofile>
```

Next steps

For more information, see [Create an Azure Active Directory tenant for P2S Open VPN connections that use Azure AD authentication](#).

About P2S VPN client profiles

11/17/2019 • 2 minutes to read • [Edit Online](#)

The downloaded profile file contains information that is necessary to configure a VPN connection. This article will help you obtain and understand the information necessary for a VPN client profile.

1. Download the file

Run the following commands. Copy the result URL to your browser in order to download the profile zip file.

```
$profile = New-AzVpnClientConfiguration -ResourceGroupName AADAuth -Name AADauthGW -AuthenticationMethod "EapTls"  
$PROFILE.VpnProfileSASUrl
```

2. Extract the zip file

Extract the zip file. The file contains the following folders:

- AzureVPN
- Generic
- OpenVPN (If you enabled the OpenVPN and Azure AD authentication settings on the gateway. See [Create a tenant](#).)

3. Retrieve information

In the **AzureVPN** folder, navigate to the **azurevpnconfig.xml** file and open it with Notepad. Make a note of the text between the following tags.

```
<audience>          </audience>  
<issuer>           </issuer>  
<tenant>            </tenant>  
<fqdn>              </fqdn>  
<serversecret>      </serversecret>
```

Profile details

When you add a connection, use the information you collected in the previous step for the profile details page. The fields correspond to the following information:

- **Audience:** Identifies the recipient resource the token is intended for
- **Issuer:** Identifies the Security Token Service (STS) that emitted the token as well as the Azure AD tenant
- **Tenant:** Contains an immutable, unique identifier of the directory tenant that issued the token
- **FQDN:** The fully qualified domain name (FQDN) on the Azure VPN gateway
- **ServerSecret:** The VPN gateway preshared key

Folder contents

- The **OpenVPN folder** contains the *ovpn* profile that needs to be modified to include the key and the

certificate. For more information, see [Configure OpenVPN clients for Azure VPN Gateway](#).

- The **generic folder** contains the public server certificate and the VpnSettings.xml file. The VpnSettings.xml file contains information needed to configure a generic client.
- The downloaded zip file may also contain **WindowsAmd64** and **WindowsX86** folders. These folders contain the installer for SSTP and IKEv2 for Windows clients. You need admin rights on the client to install them.

Next steps

For more information about point-to-site, see [About point-to-site](#).

Configure OpenVPN for Azure point-to-site VPN Gateway

2/12/2020 • 2 minutes to read • [Edit Online](#)

This article helps you set up **OpenVPN® Protocol** on Azure VPN Gateway. The article assumes that you already have a working point-to-site environment. If you do not, use the instructions in step 1 to create a point-to-site VPN.

1. Create a point-to-site VPN

If you don't already have a functioning point-to-site environment, follow the instruction to create one. See [Create a point-to-site VPN](#) to create and configure a point-to-site VPN gateway with native Azure certificate authentication.

IMPORTANT

The Basic SKU is not supported for OpenVPN.

2. Enable OpenVPN on the gateway

Enable OpenVPN on your gateway. Make sure that the gateway is already configured for point-to-site (IKEv2 or SSTP) before running the following commands:

```
$gw = Get-AzVirtualNetworkGateway -ResourceGroupName $rgname -name $name  
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -VpnClientProtocol OpenVPN
```

Next steps

To configure clients for OpenVPN, see [Configure OpenVPN clients](#).

"OpenVPN" is a trademark of OpenVPN Inc.

Configure OpenVPN clients for Azure VPN Gateway

2/12/2020 • 6 minutes to read • [Edit Online](#)

This article helps you configure **OpenVPN ® Protocol** clients.

Before you begin

Verify that you have completed the steps to configure OpenVPN for your VPN gateway. For details, see [Configure OpenVPN for Azure VPN Gateway](#).

Windows clients

1. Download and install the OpenVPN client (version 2.4 or higher) from the official [OpenVPN website](#).
2. Download the VPN profile for the gateway. This can be done from the Point-to-site configuration tab in the Azure portal, or 'New-AzVpnClientConfiguration' in PowerShell.
3. Unzip the profile. Next, open the *vpnconfig.ovpn* configuration file from the OpenVPN folder using Notepad.
4. [Export](#) the P2S client certificate you created and uploaded to your P2S configuration on the gateway.
5. Extract the private key and the base64 thumbprint from the *.pfx*. There are multiple ways to do this. Using OpenSSL on your machine is one way. The *profileinfo.txt* file contains the private key and the thumbprint for the CA and the Client certificate. Be sure to use the thumbprint of the client certificate.

```
openssl pkcs12 -in "filename.pfx" -nodes -out "profileinfo.txt"
```

6. Open *profileinfo.txt* in Notepad. To get the thumbprint of the client (child) certificate, select the text (including and between) "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" for the child certificate and copy it. You can identify the child certificate by looking at the subject=/ line.
7. Switch to the *vpnconfig.ovpn* file you opened in Notepad from step 3. Find the section shown below and replace everything between "cert" and "/cert".

```
# P2S client certificate
# please fill this field with a PEM formatted cert
<cert>
$CLIENTCERTIFICATE
</cert>
```

8. Open the *profileinfo.txt* in Notepad. To get the private key, select the text (including and between) "----- BEGIN PRIVATE KEY-----" and "-----END PRIVATE KEY-----" and copy it.
9. Go back to the *vpnconfig.ovpn* file in Notepad and find this section. Paste the private key replacing everything between and "key" and "/key".

```
# P2S client root certificate private key
# please fill this field with a PEM formatted key
<key>
$PRIVATEKEY
</key>
```

10. Do not change any other fields. Use the filled in configuration in client input to connect to the VPN.
11. Copy the vpnconfig.ovpn file to C:\Program Files\OpenVPN\config folder.
12. Right-click the OpenVPN icon in the system tray and click connect.

Mac clients

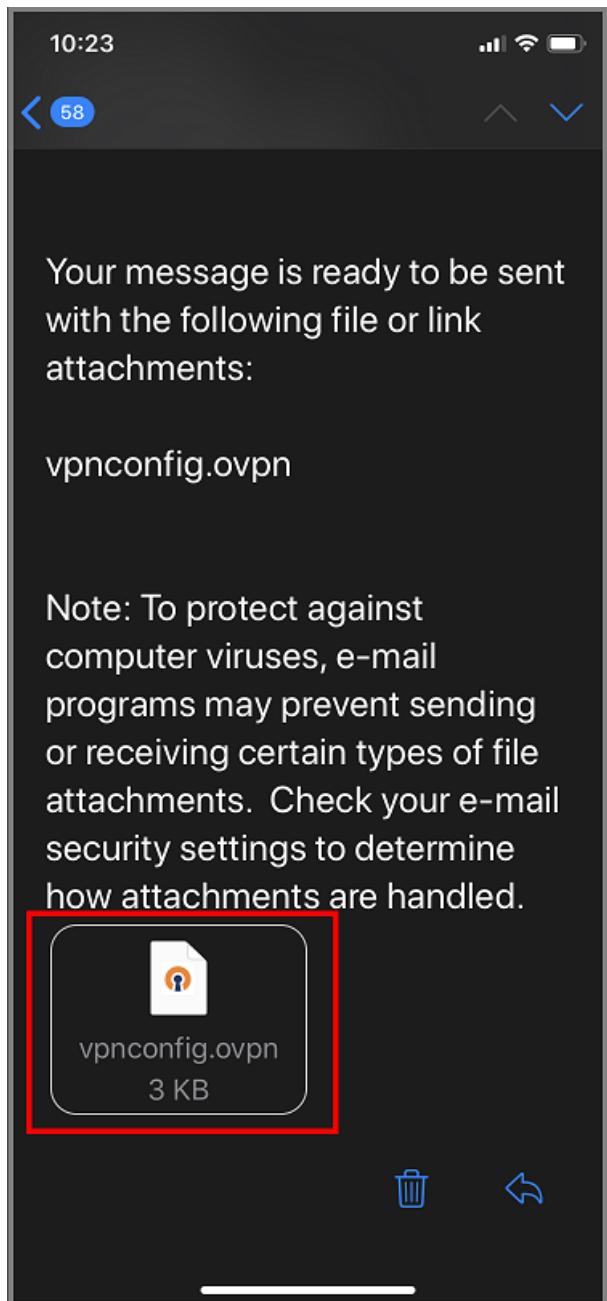
1. Download and install an OpenVPN client, such as [TunnelBlick](#).
2. Download the VPN profile for the gateway. This can be done from the point-to-site configuration tab in the Azure portal, or by using 'New-AzVpnClientConfiguration' in PowerShell.
3. Unzip the profile. Open the vpnconfig.ovpn configuration file from the OpenVPN folder in a text editor.
4. Fill in the P2S client certificate section with the P2S client certificate public key in base64. In a PEM formatted certificate, you can simply open the .cer file and copy over the base64 key between the certificate headers. See [Export the public key](#) for information about how to export a certificate to get the encoded public key.
5. Fill in the private key section with the P2S client certificate private key in base64. See [Export your private key](#) for information about how to extract a private key.
6. Do not change any other fields. Use the filled in configuration in client input to connect to the VPN.
7. Double-click the profile file to create the profile in Tunnelblick.
8. Launch Tunnelblick from the applications folder.
9. Click on the Tunnelblick icon in the system tray and pick connect.

IMPORTANT

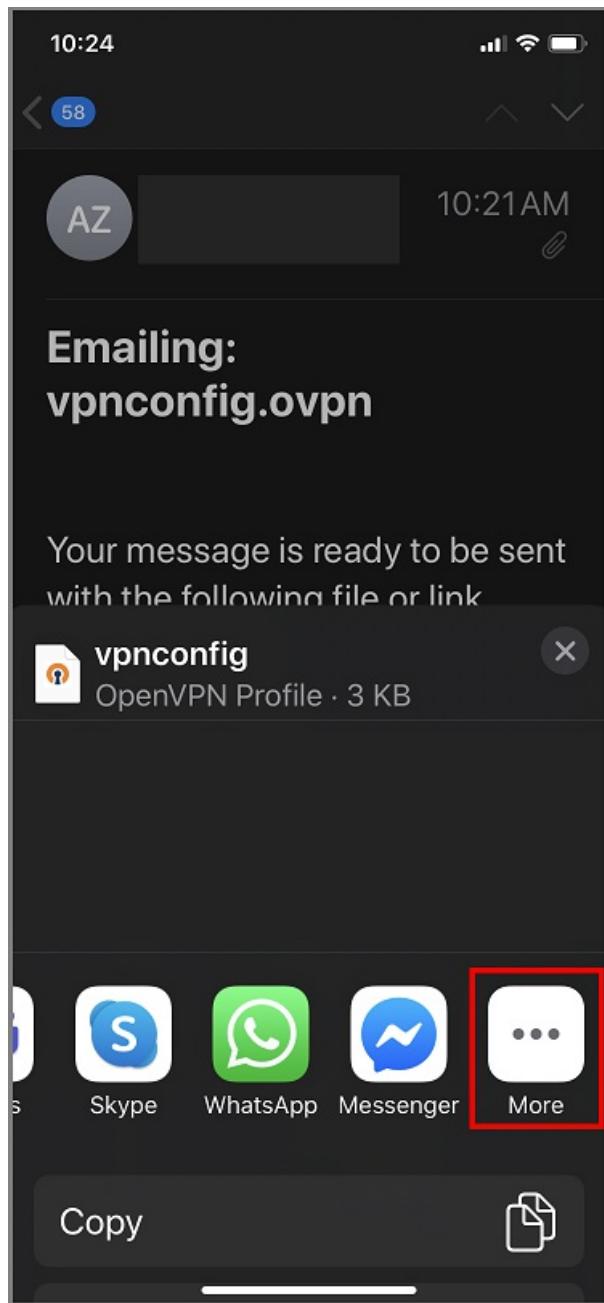
Only iOS 11.0 and above and MacOS 10.13 and above are supported with OpenVPN protocol.

iOS clients

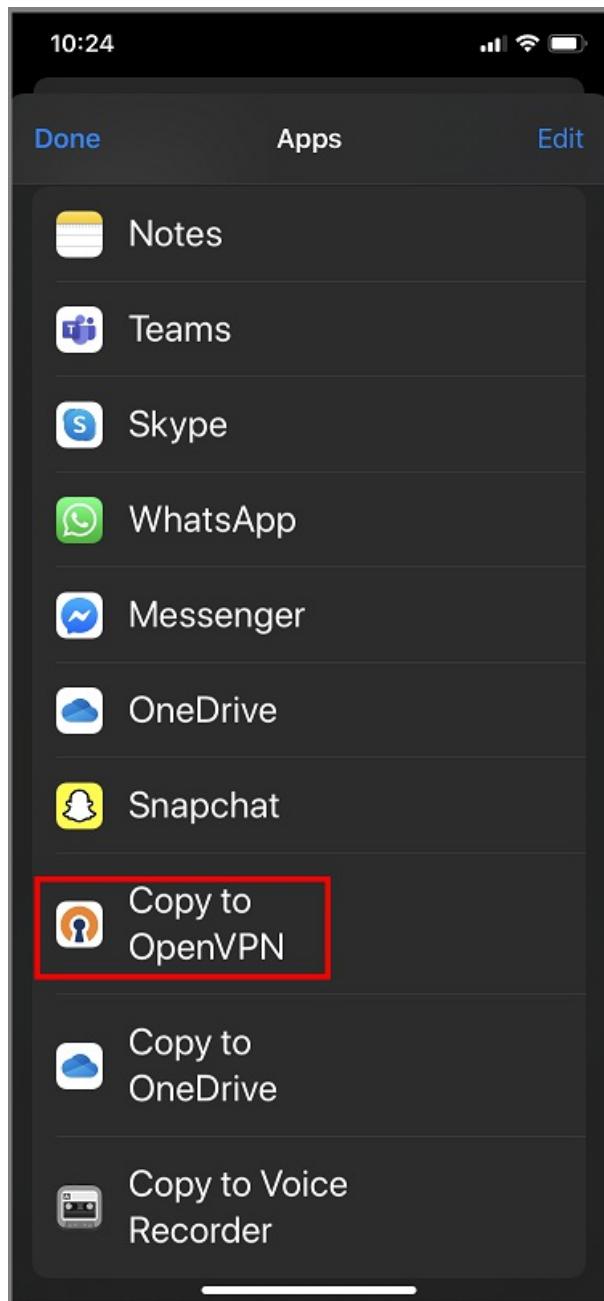
1. Install the OpenVPN client (version 2.4 or higher) from the App store.
2. Download the VPN profile for the gateway. This can be done from the point-to-site configuration tab in the Azure portal, or by using 'New-AzVpnClientConfiguration' in PowerShell.
3. Unzip the profile. Open the vpnconfig.ovpn configuration file from the OpenVPN folder in a text editor.
4. Fill in the P2S client certificate section with the P2S client certificate public key in base64. In a PEM formatted certificate, you can simply open the .cer file and copy over the base64 key between the certificate headers. See [Export the public key](#) for information about how to export a certificate to get the encoded public key.
5. Fill in the private key section with the P2S client certificate private key in base64. See [Export your private key](#) for information about how to extract a private key.
6. Do not change any other fields.
7. E-mail the profile file (.ovpn) to your email account that is configured in the mail app on your iPhone.
8. Open the e-mail in the mail app on the iPhone, and tap the attached file



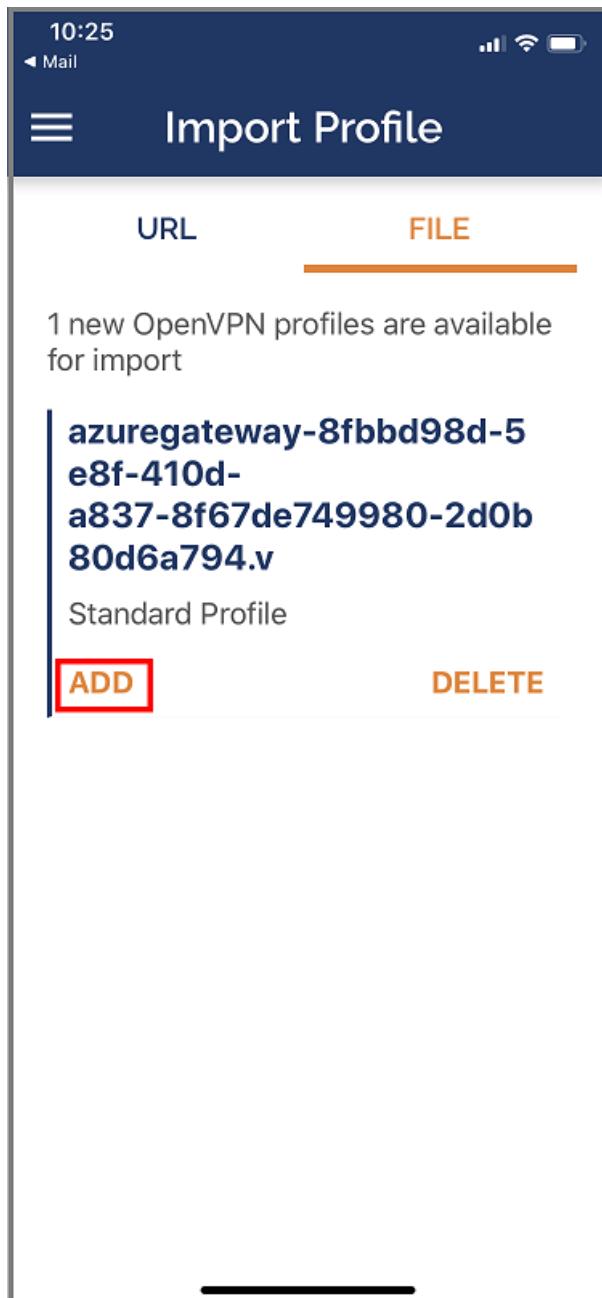
9. Tap on **More** if you do not see **Copy to OpenVPN** option



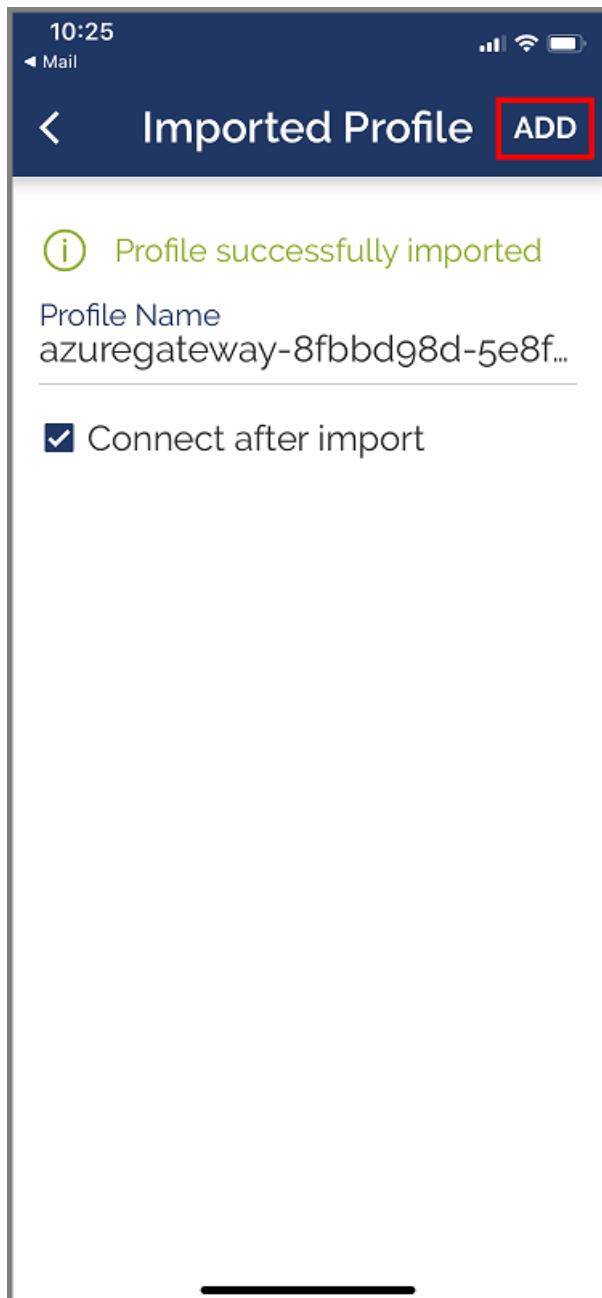
10. Tap on **Copy to OpenVPN**



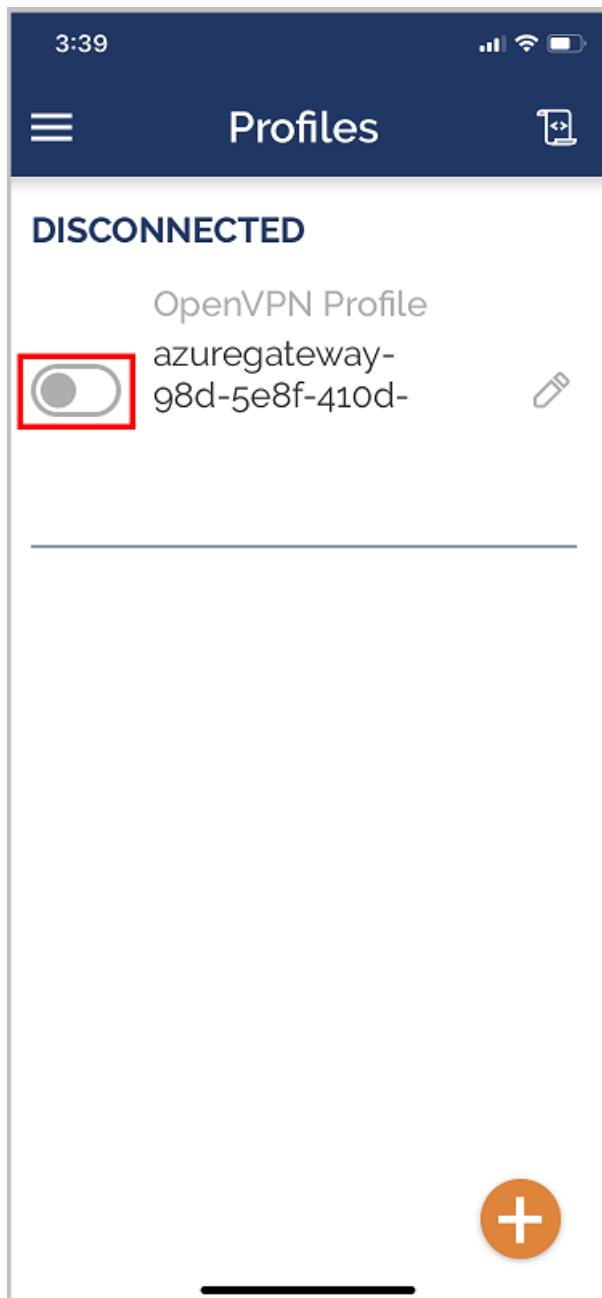
11. Tap on **ADD** in the **Import Profile** page



12. Tap on **ADD** in the **Imported Profile** page



13. Launch the OpenVPN app and slide the switch in the **Profile** page right to connect



Linux clients

1. Open a new Terminal session. You can open a new session by pressing 'Ctrl + Alt + t' at the same time.
2. Enter the following command to install needed components:

```
sudo apt-get install openvpn  
sudo apt-get -y install network-manager-openvpn  
sudo service network-manager restart
```

3. Download the VPN profile for the gateway. This can be done from the Point-to-site configuration tab in the Azure portal.
4. [Export](#) the P2S client certificate you created and uploaded to your P2S configuration on the gateway.
5. Extract the private key and the base64 thumbprint from the .pfx. There are multiple ways to do this. Using OpenSSL on your computer is one way.

```
openssl.exe pkcs12 -in "filename.pfx" -nodes -out "profileinfo.txt"
```

The *profileinfo.txt* file will contain the private key and the thumbprint for the CA, and the Client certificate. Be sure to use the thumbprint of the client certificate.

6. Open *profileinfo.txt* in a text editor. To get the thumbprint of the client (child) certificate, select the text including and between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" for the child certificate and copy it. You can identify the child certificate by looking at the subject=/ line.
7. Open the *vpnconfig.ovpn* file and find the section shown below. Replace everything between the and "cert" and "/cert".

```
# P2S client certificate
# please fill this field with a PEM formatted cert
<cert>
$CLIENTCERTIFICATE
</cert>
```

8. Open the *profileinfo.txt* in a text editor. To get the private key, select the text including and between "----- BEGIN PRIVATE KEY-----" and "-----END PRIVATE KEY-----" and copy it.
9. Open the *vpnconfig.ovpn* file in a text editor and find this section. Paste the private key replacing everything between and "key" and "/key".

```
# P2S client root certificate private key
# please fill this field with a PEM formatted key
<key>
$PRIVATEKEY
</key>
```

10. Do not change any other fields. Use the filled in configuration in client input to connect to the VPN.

11. To connect using the command line, type the following command:

```
sudo openvpn --config <name and path of your VPN profile file>&
```

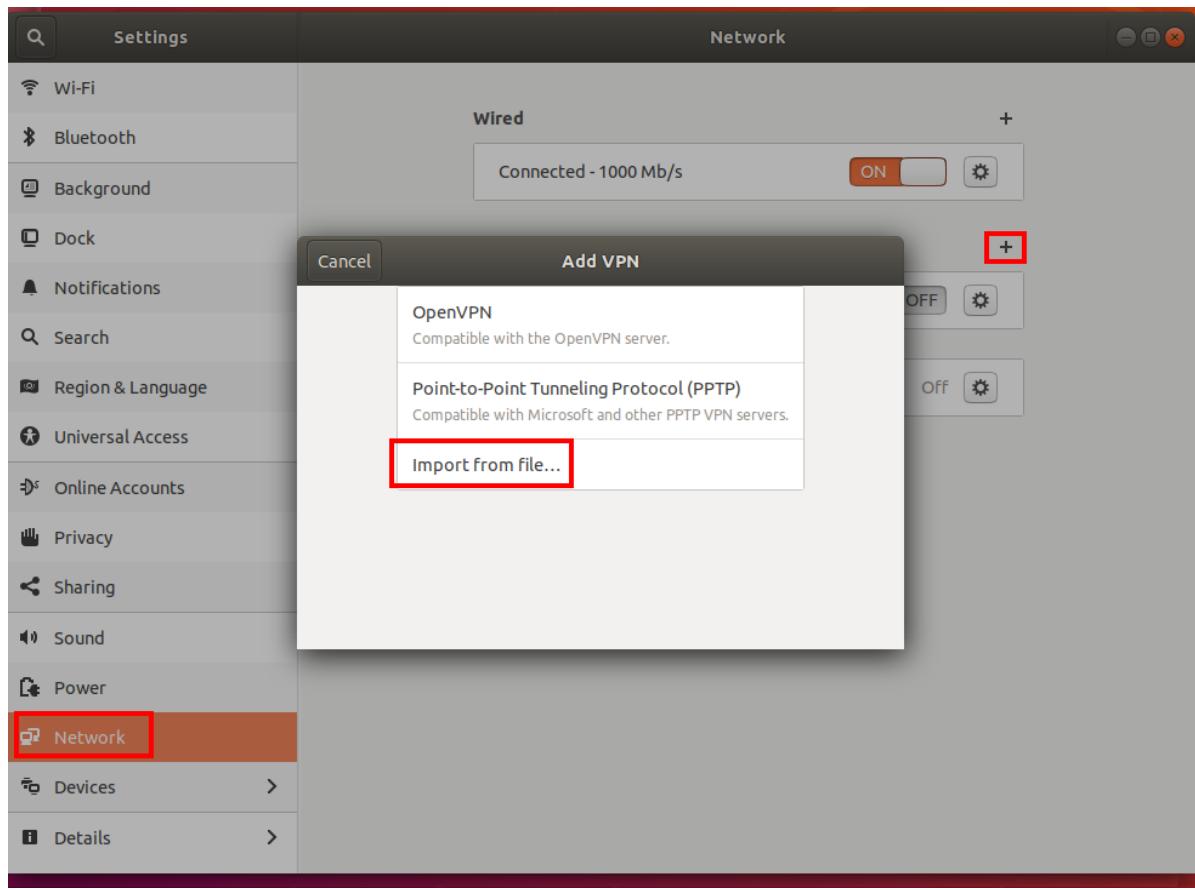
12. To connect using the GUI, go to system settings.

13. Click + to add a new VPN connection.

14. Under **Add VPN**, pick **Import from file...**

15. Browse to the profile file and double-click or pick **Open**.

16. Click **Add** on the **Add VPN** window.



17. You can connect by turning the VPN **ON** on the **Network Settings** page, or under the network icon in the system tray.

Next steps

If you want the VPN clients to be able to access resources in another VNet, then follow the instructions on the [VNet-to-VNet](#) article to set up a vnet-to-vnet connection. Be sure to enable BGP on the gateways and the connections, otherwise traffic will not flow.

"**OpenVPN**" is a trademark of **OpenVPN Inc.**

2 minutes to read

Configure an Always On VPN device tunnel

1/13/2020 • 3 minutes to read • [Edit Online](#)

One of the new features of the Windows 10 Virtual Private Network (VPN) client is the ability to maintain a VPN connection. Always On is a Windows 10 feature that enables the active VPN profile to connect automatically and remain connected based on triggers — namely, user sign-in, network state change, or device screen active.

Azure virtual network gateways can be used with Windows 10 Always On to establish persistent user tunnels as well as device tunnels to Azure. This article will help you configure an Always ON VPN device tunnel.

Always On VPN connections include two types of tunnels:

- **Device tunnel** connects to specified VPN servers before users sign in the device. Pre-login connectivity scenarios and device management purposes use device tunnel.
- **User tunnel** connects only after a user sign in the device. User tunnel allows users to access organization resources through VPN servers.

Both Device tunnel and User tunnel operate independently with their VPN profiles. They can be connected at the same time, and can use different authentication methods and other VPN configuration settings as appropriate.

1. Configure the gateway

Configure the VPN gateway to use IKEv2 and certificate-based authentication using this [point-to-site article](#).

2. Configure the device tunnel

The following requirements must be met in order to successfully establish a device tunnel:

- The device must be a domain joined computer running Windows 10 Enterprise or Education version 1809 or later.
- The tunnel is only configurable for the Windows built-in VPN solution and is established using IKEv2 with computer certificate authentication.
- Only one device tunnel can be configured per device.

1. Install client certificates on the Windows 10 client as shown in this [point-to-site VPN client article](#). The certificate needs to be in the Local Machine store.
2. Use [these instructions](#) to create a VPN Profile and configure device tunnel in the context of the LOCAL SYSTEM account.

Configuration example for device tunnel

After you have configured the virtual network gateway and installed the client certificate in the Local Machine store on the Windows 10 client, use the following examples to configure a client device tunnel.

1. Copy the following text and save it as **devicecert.ps1**.

```

Param(
[string]$xmlFilePath,
[string]$ProfileName
)

$a = Test-Path $xmlFilePath
echo $a

$ProfileXML = Get-Content $xmlFilePath

echo $XML

$ProfileNameEscaped = $ProfileName -replace ' ', '%20'

$Version = 201606090004

$ProfileXML = $ProfileXML -replace '<', '&lt;'
$ProfileXML = $ProfileXML -replace '>', '&gt;'
$ProfileXML = $ProfileXML -replace "'", "&quot;"

$nodeCSPURI = './Vendor/MSFT/VPNv2'
$namespaceName = "root\cimv2\mdm\dmmap"
$className = "MDM_VPNv2_01"

$session = New-CimSession

try
{
    $newInstance = New-Object Microsoft.Management.Infrastructure.CimInstance $className, $namespaceName
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("ParentID", "$nodeCSPURI",
    'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("InstanceID",
    '$ProfileNameEscaped', 'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("ProfileXML", "$ProfileXML",
    'String', 'Property')
    $newInstance.CimInstanceProperties.Add($property)

    $session.CreateInstance($namespaceName, $newInstance)
    $Message = "Created $ProfileName profile."
    Write-Host "$Message"
}
catch [Exception]
{
    $Message = "Unable to create $ProfileName profile: $_"
    Write-Host "$Message"
    exit
}
$Message = "Complete."
Write-Host "$Message"

```

2. Copy the following text and save it as **VPNProfile.xml** in the same folder as **devicecert.ps1**. Edit the following text to match your environment.

- <Servers>azuregateway-1234-56-78dc.cloudapp.net</Servers> <= Can be found in the VpnSettings.xml in the downloaded profile zip file
- <Address>192.168.3.5</Address> <= IP of resource in the vnet or the vnet address space
- <Address>192.168.3.4</Address> <= IP of resource in the vnet or the vnet address space

```

<VPNProfile>
  <NativeProfile>
    <Servers>azuregateway-1234-56-78dc.cloudapp.net</Servers>
    <NativeProtocolType>IKEv2</NativeProtocolType>
    <Authentication>
      <MachineMethod>Certificate</MachineMethod>
    </Authentication>
    <RoutingPolicyType>SplitTunnel</RoutingPolicyType>
      <!-- disable the addition of a class based route for the assigned IP address on the VPN interface -->
    <DisableClassBasedDefaultRoute>true</DisableClassBasedDefaultRoute>
  </NativeProfile>
  <!-- use host routes(/32) to prevent routing conflicts -->
  <Route>
    <Address>192.168.3.5</Address>
    <PrefixSize>32</PrefixSize>
    </Route>
    <Route>
      <Address>192.168.3.4</Address>
      <PrefixSize>32</PrefixSize>
      </Route>
    <!-- need to specify always on = true -->
    <AlwaysOn>true</AlwaysOn>
    <!-- new node to specify that this is a device tunnel -->
    <DeviceTunnel>true</DeviceTunnel>
    <!--new node to register client IP address in DNS to enable manage out -->
    <RegisterDNS>true</RegisterDNS>
  </Route>
</VPNProfile>

```

3. Download **PsExec** from [Sysinternals](#) and extract the files to **C:\PSTools**.

4. From an Admin CMD prompt, launch PowerShell by running:

```

PsExec.exe Powershell for 32-bit Windows
PsExec64.exe Powershell for 64-bit Windows

```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - PsExec64.exe Powershell". The window displays the following text:

```

Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd \PSTools

C:\PSTools>PsExec64.exe Powershell

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

```

5. In PowerShell, switch to the folder where **devicecert.ps1** and **VPNProfile.xml** are located, and run the following command:

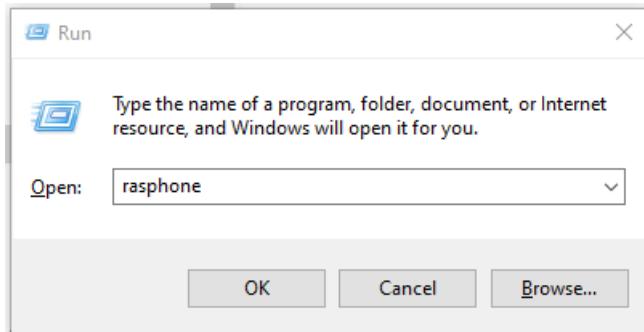
```
.\devicecert.ps1 .\VPNProfile.xml MachineCertTest
```

```
PS E:\Scripts> .\devicecert.ps1 .\VPNProfile.xml MachineCertTest
Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run E:\Scripts\devicecert.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): r
True

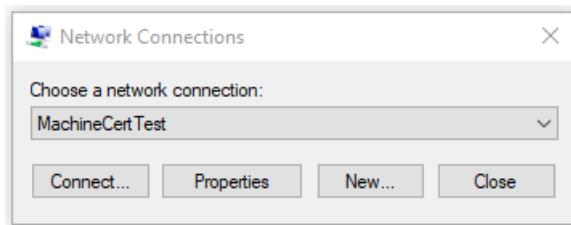
AlwaysOn
ByPassForLocal
DnsSuffix
EdpModeId
InstanceId : MachineCertTest
LockDown
ParentID : ./Vendor/MSFT/VPNv2
ProfileXML
RememberCredentials
TrustedNetworkDetection
PSCoputerName

Created MachineCertTest profile.
Complete.
```

6. Run rasphone.



7. Look for the **MachineCertTest** entry and click **Connect**.



8. If the connection succeeds, reboot the computer. The tunnel will connect automatically.

Cleanup

To remove the profile, run the following command:

```
PS E:\Scripts> Remove-VpnConnection -Name MachineCertTest
Confirm
Deleting VPN connection MachineCertTest. Do you want to continue?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PS E:\Scripts>
```

Next steps

For troubleshooting, see [Azure point-to-site connection problems](#)

Configure an Always On VPN user tunnel

1/22/2020 • 3 minutes to read • [Edit Online](#)

A new feature of the Windows 10 VPN client, Always On, is the ability to maintain a VPN connection. With Always On, the active VPN profile can connect automatically and remain connected based on triggers, such as user sign-in, network state change, or device screen active.

You can use Azure virtual network gateways with Windows 10 Always On to establish persistent user tunnels and device tunnels to Azure. This article helps you configure an Always On VPN user tunnel.

Always On VPN connections include either of two types of tunnels:

- **Device tunnel:** Connects to specified VPN servers before users sign in to the device. Pre-sign-in connectivity scenarios and device management use a device tunnel.
- **User tunnel:** Connects only after users sign in to the device. By using user tunnels, you can access organization resources through VPN servers.

Device tunnels and user tunnels operate independent of their VPN profiles. They can be connected at the same time, and they can use different authentication methods and other VPN configuration settings, as appropriate.

In the following sections, you configure a VPN gateway and a user tunnel.

Step 1: Configure a VPN gateway

You configure the VPN gateway to use IKEv2 and certificate-based authentication by following the instructions in this [point-to-site](#) article.

Step 2: Configure a user tunnel

1. Install client certificates on the Windows 10 client, as shown in this [point-to-site VPN client](#) article. The certificate must be in the current user store.
2. Configure the Always On VPN client through PowerShell, Configuration Manager, or Intune by following the instructions in [Configure Windows 10 client Always On VPN connections](#).

Example configuration for the user tunnel

After you've configured the virtual network gateway and installed the client certificate in the local machine store on the Windows 10 client, configure a client device tunnel by using the following examples:

1. Copy the following text, and save it as *usercert.ps1*:

```

Param(
[string]$xmlFilePath,
[string]$ProfileName
)

$a = Test-Path $xmlFilePath
echo $a

$ProfileXML = Get-Content $xmlFilePath

echo $XML

$ProfileNameEscaped = $ProfileName -replace ' ', '%20'

$Version = 201606090004

$ProfileXML = $ProfileXML -replace '<', '&lt;'
$ProfileXML = $ProfileXML -replace '>', '&gt;'
$ProfileXML = $ProfileXML -replace "'", "&quot;"

$nodeCSPURI = './Vendor/MSFT/VPNv2'
$namespaceName = "root\cimv2\mdm\dmmap"
$className = "MDM_VPNv2_01"

$session = New-CimSession

try
{
    $newInstance = New-Object Microsoft.Management.Infrastructure.CimInstance $className, $namespaceName
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("ParentID", "$nodeCSPURI",
    'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("InstanceID",
    '$ProfileNameEscaped', 'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property = [Microsoft.Management.Infrastructure.CimProperty]::Create("ProfileXML", "$ProfileXML",
    'String', 'Property')
    $newInstance.CimInstanceProperties.Add($property)

    $session.CreateInstance($namespaceName, $newInstance)
    $Message = "Created $ProfileName profile."
    Write-Host "$Message"
}
catch [Exception]
{
    $Message = "Unable to create $ProfileName profile: $_"
    Write-Host "$Message"
    exit
}
$Message = "Complete."
Write-Host "$Message"

```

2. Copy the following text, and save it as *VPNProfile.xml* in the same folder as *usercert.ps1*. Edit the following text to match your environment:

- <Servers>azuregateway-1234-56-78dc.cloudapp.net</Servers> <= Can be found in the VpnSettings.xml in the downloaded profile zip file
- <Address>192.168.3.5</Address> <= IP of resource in the vnet or the vnet address space
- <Address>192.168.3.4</Address> <= IP of resource in the vnet or the vnet address space
- <PrefixSize>32</PrefixSize> <= Subnet mask

```

<VPNProfile>
  <NativeProfile>
    <Servers>azuregateway-b115055e-0882-49bc-a9b9-7de45cba12c0-8e6946892333.vpn.azure.com</Servers>
    <NativeProtocolType>IKEv2</NativeProtocolType>
    <Authentication>
      <UserMethod>Eap</UserMethod>
      <Eap>
        <Configuration>
          <EapHostConfig xmlns="http://www.microsoft.com/provisioning/EapHostConfig"><EapMethod><Type
            xmlns="http://www.microsoft.com/provisioning/EapCommon">13</Type><VendorId
            xmlns="http://www.microsoft.com/provisioning/EapCommon">0</VendorId><VendorType
            xmlns="http://www.microsoft.com/provisioning/EapCommon">0</VendorType><AuthorId
            xmlns="http://www.microsoft.com/provisioning/EapCommon">0</AuthorId></EapMethod><Config
            xmlns="http://www.microsoft.com/provisioning/EapHostConfig"><Eap
            xmlns="http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1"><Type>13</Type><EapType
            xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV1"><CredentialsSource>
              <CertificateStore><SimpleCertSelection>true</SimpleCertSelection></CertificateStore></CredentialsSource>
            <ServerValidation><DisableUserPromptForServerValidation>false</DisableUserPromptForServerValidation>
            <ServerNames></ServerNames></ServerValidation><DifferentUsername>false</DifferentUsername>
            <PerformServerValidation
            xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2">false</PerformServerValidation>
            <AcceptServerName
            xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2">false</AcceptServerName>
          </EapType></Eap></Config></EapHostConfig>
        </Configuration>
      </Eap>
    </Authentication>
    <RoutingPolicyType>SplitTunnel</RoutingPolicyType>
    <!-- disable the addition of a class based route for the assigned IP address on the VPN interface -->
    <DisableClassBasedDefaultRoute>true</DisableClassBasedDefaultRoute>
    </NativeProfile>
    <!-- use host routes(/32) to prevent routing conflicts -->
    <Route>
      <Address>192.168.3.5</Address>
      <PrefixSize>32</PrefixSize>
      </Route>
      <Route>
        <Address>192.168.3.4</Address>
        <PrefixSize>32</PrefixSize>
        </Route>
    <!-- traffic filters for the routes specified above so that only this traffic can go over the device
    tunnel -->
    <TrafficFilter>
      <RemoteAddressRanges>192.168.3.4, 192.168.3.5</RemoteAddressRanges>
    </TrafficFilter>
    <!-- need to specify always on = true -->
    <AlwaysOn>true</AlwaysOn>
    <RememberCredentials>true</RememberCredentials>
    <!--new node to register client IP address in DNS to enable manage out -->
    <RegisterDNS>true</RegisterDNS>
  </VPNProfile>

```

3. Run PowerShell as an administrator.
4. In PowerShell, switch to the folder where *usercert.ps1* and *VPNProfile.xml* are located, and run the following command:

```
C:\> .\usercert.ps1 .\VPNProfile.xml UserTest
```

```
Administrator: Windows PowerShell
PS C:\Users\alzam\OneDrive - Microsoft\Always ON> .\usercert.ps1 .\VPNProfileUser.xml UserTest
True

AlwaysOn      :
ByPassForLocal   :
DnsSuffix      :
EdpModeId      :
InstanceId      : UserTest
LockDown      :
ParentID        : ./Vendor/MSFT/VPNv2
ProfileXML     :
RememberCredentials  :
TrustedNetworkDetection  :
PSCoputerName    :

Created UserTest profile.
Complete.
```

5. Under **VPN Settings**, look for the **UserTest** entry, and then select **Connect**.
6. If the connection succeeds, you've successfully configured an Always On user tunnel.

Clean up your resources

To remove the profile, do the following:

1. Run the following command:

```
C:\> Remove-VpnConnection UserTest
```

2. Disconnect the connection, and clear the **Connect automatically** check box.

```
Administrator: Windows PowerShell
PS C:\Users\alzam\OneDrive - Microsoft\Always ON> Remove-VpnConnection UserTest
Confirm
Deleting VPN connection UserTest. Do you want to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): y
PS C:\Users\alzam\OneDrive - Microsoft\Always ON>
```

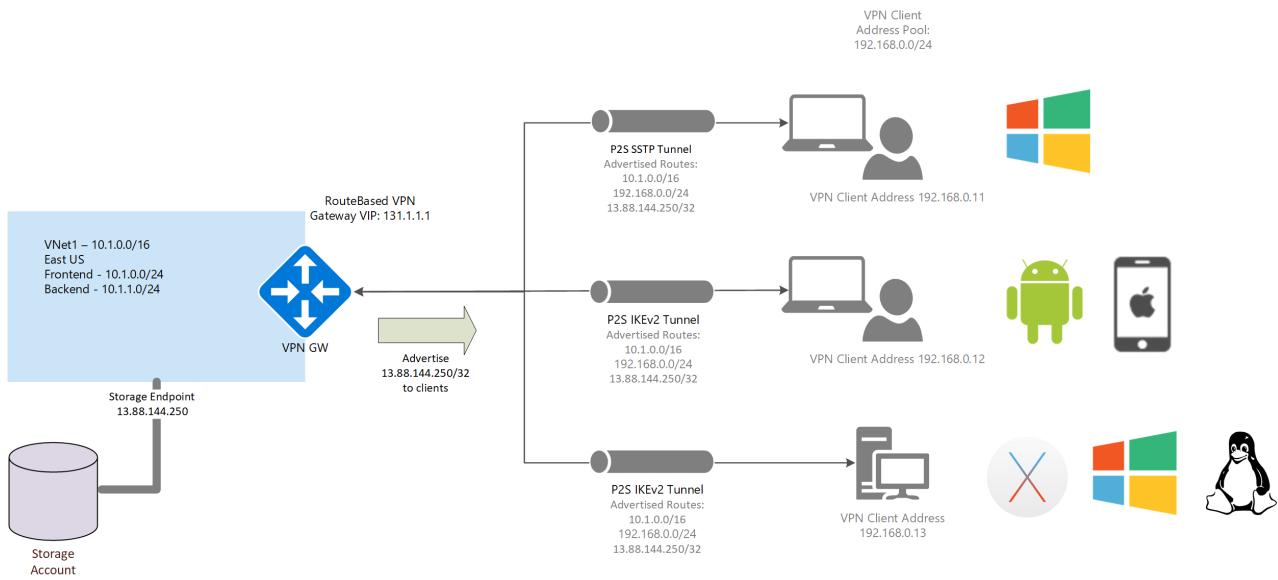
Next steps

To troubleshoot any connection issues that might occur, see [Azure point-to-site connection problems](#).

Advertise custom routes for P2S VPN clients

11/17/2019 • 2 minutes to read • [Edit Online](#)

You may want to advertise custom routes to all of your point-to-site VPN clients. For example, when you have enabled storage endpoints in your VNet and want the remote users to be able to access these storage accounts over the VPN connection. You can advertise the IP address of the storage end-point to all your remote users so that the traffic to the storage account goes over the VPN tunnel, and not the public Internet.



To advertise custom routes

To advertise custom routes, use the `Set-AzVirtualNetworkGateway cmdlet`. The following example shows you how to advertise the IP for the [Contoso storage account tables](#).

1. Ping `contoso.table.core.windows.net` and note the IP address. For example:

```
C:\>ping contoso.table.core.windows.net
Pinging table.by4prdstr05a.store.core.windows.net [13.88.144.250] with 32 bytes of data:
```

2. Run the following PowerShell commands:

```
$gw = Get-AzVirtualNetworkGateway -Name <name of gateway> -ResourceGroupName <name of resource group>
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -CustomRoute 13.88.144.250/32
```

3. To add multiple custom routes, use a coma and spaces to separate the addresses. For example:

```
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -CustomRoute x.x.x.x/xx , y.y.y.y/yy
```

To view custom routes

Use the following example to view custom routes:

```
$gw = Get-AzVirtualNetworkGateway -Name <name of gateway> -ResourceGroupName <name of resource group>
$gw.CustomRoutes | Format-List
```

To delete custom routes

Use the following example to delete custom routes:

```
$gw = Get-AzVirtualNetworkGateway -Name <name of gateway> -ResourceGroupName <name of resource group>
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -CustomRoute @0
```

Next steps

For additional P2S routing information, see [About point-to-site routing](#).

Create a zone-redundant virtual network gateway in Azure Availability Zones

2/11/2020 • 4 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. For information, see [About zone-redundant virtual network gateways](#) and [About Azure Availability Zones](#).

Before you begin

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell, an interactive shell environment hosted in Azure and used through the browser. Azure Cloud Shell comes with the Azure PowerShell cmdlets pre-installed.

To run any code contained in this article on Azure Cloud Shell, open a Cloud Shell session, use the **Copy** button on a code block to copy the code, and paste it into the Cloud Shell session with **Ctrl+Shift+V** on Windows and Linux, or **Cmd+Shift+V** on macOS. Pasted text is not automatically executed, so press **Enter** to run code.

You can launch Azure Cloud Shell using any of the following methods:

Select Try It in the upper-right corner of a code block. This doesn't automatically copy text to Cloud Shell.	
Open shell.azure.com in your browser.	
Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal .	

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you have not installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see [Install the Azure PowerShell module](#).

1. Declare your variables

Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary. If you close your PowerShell/Cloud Shell session at any point during the exercise, just copy and paste the values again to re-declare the variables. 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"
```

2. Create the virtual network

Create a resource group.

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

Create a virtual network.

```
$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubnet1 -AddressPrefix $FEPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubnet1 -AddressPrefix $BEPrefix1
$vnet = New-AzVirtualNetwork -Name $VNet1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNet1Prefix -Subnet $fesub1,$besub1
```

3. 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-AzVirtualNetwork -ResourceGroupName $RG1 -Name VNet1
Add-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27 -VirtualNetwork $getvnet
```

Set the gateway subnet configuration for the virtual network.

```
$getvnet | Set-AzVirtualNetwork
```

4. 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-AzPublicIpAddress -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-AzPublicIpAddress -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-AzPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod Dynamic -Sku Basic
```

5. Create the IP configuration

```
$getvnet = Get-AzVirtualNetwork -ResourceGroupName $RG1 -Name $VNet1  
$subnet = Get-AzVirtualNetworkSubnetConfig -Name $GwSubnet1 -VirtualNetwork $getvnet  
$gwipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GwIPConf1 -Subnet $subnet -PublicIpAddress $pip1
```

6. Create the gateway

Create the virtual network gateway.

For ExpressRoute

```
New-AzVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations $GwIPConf1 -GatewayType ExpressRoute -GatewaySku ErGw1AZ
```

For VPN Gateway

```
New-AzVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations $GwIPConf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1AZ
```

FAQ

What will change when I deploy these new SKUs?

From your perspective, 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 to deploy the new SKUs. However, you will see these new SKUs only in those Azure regions that have Azure Availability Zones.

What regions are available for me to use the new SKUs?

See [Availability Zones](#) for the latest list of available regions.

Can I change/migrate/upgrade 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. However, you should reserve a /27 IP address range for the gateway subnet.

Configure IPsec/IKE policy for S2S VPN or VNet-to-VNet connections

2/12/2020 • 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

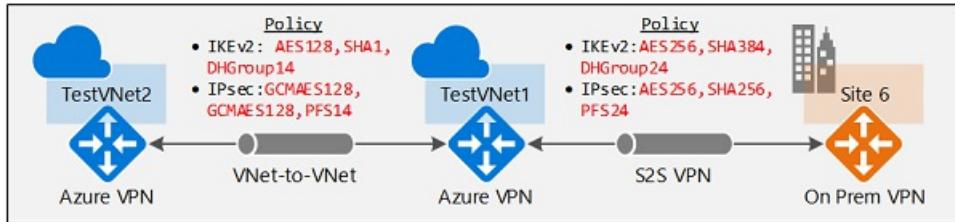
1. Note that IPsec/IKE policy only works on the following gateway SKUs:
 - **VpnGw1**, **VpnGw2**, **VpnGw3** (route-based)
 - **Standard** and **HighPerformance** (route-based)
2. You can only specify **one** policy combination for a given connection.
3. You must specify all algorithms and parameters 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	GCMASE256, 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](#).

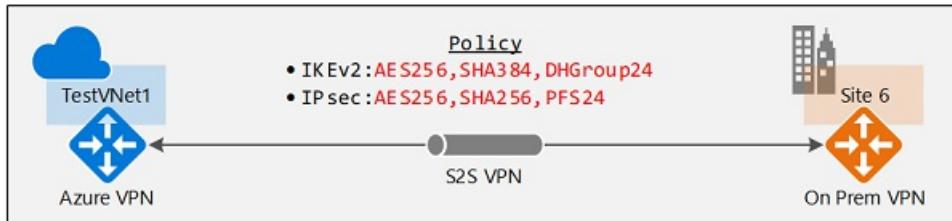
The following table lists the corresponding Diffie-Hellman Groups supported by the custom policy:

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	ECP384	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-AzAccount
Select-AzSubscription -SubscriptionName $Sub1
New-AzResourceGroup -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-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1

$gw1pip1 = New-AzPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$vnet1 = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gw1ipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress
$gw1pip1

New-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -IpConfigurations
$gw1ipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1

New-AzLocalNetworkGateway -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 & 10240000KB

```

$ipsecpolicy6 = New-AzIpsecPolicy -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng6 = Get-AzLocalNetworkGateway -Name $LNGName6 -ResourceGroupName $RG1

New-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1 -VirtualNetworkGateway1
$vnet1gw -LocalNetworkGateway2 $lng6 -Location $Location1 -ConnectionType IPsec -IpsecPolicies $ipsecpolicy6
-SharedKey 'AzureA1b2C3'

```

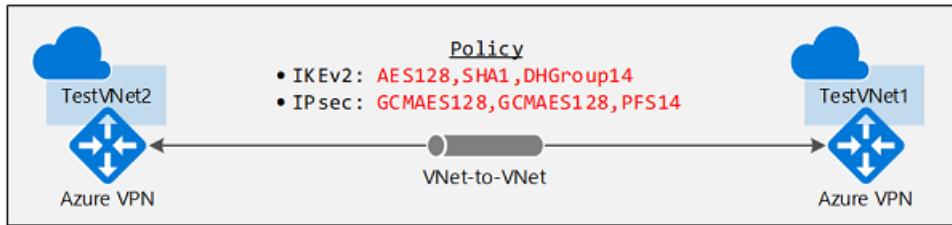
You can optionally add "-UsePolicyBasedTrafficSelectors \$True" to the create connection cmdlet to enable Azure VPN gateway to connect to policy-based VPN devices on premises, as described above.

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-AzResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$besub2 = New-AzVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsb2 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix
$VNetPrefix21,$VNetPrefix22 -Subnet $fesub2,$besub2,$gwsb2

$gw2pip1 = New-AzPublicIpAddress -Name $GW2IPName1 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic
$vnet2 = Get-AzVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gw2ipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GW2IPConf1 -Subnet $subnet2 -PublicIpAddress
$gw2pip1

New-AzVirtualNetworkGateway -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-AzIpsecPolicy -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet2gw = Get-AzVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2

New-AzVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1
$vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -IpsecPolicies
$ipsecpolicy2 -SharedKey 'AzureA1b2C3'

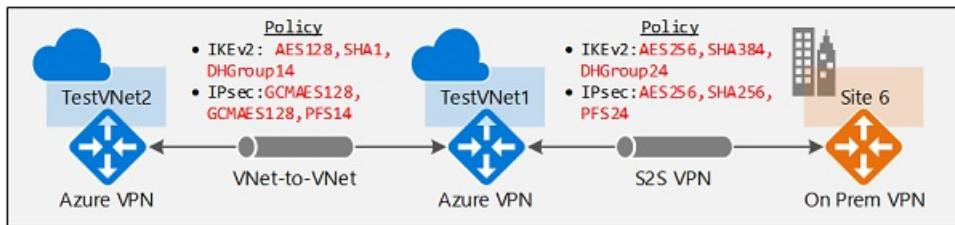
New-AzVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -VirtualNetworkGateway1
$vnet2gw -VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType Vnet2Vnet -IpsecPolicies
$ipsecpolicy2 -SharedKey 'AzureA1b2C3'

```

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-AzVirtualNetworkGatewayConnection -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
SADaDataSizeKilobytes : 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.IpsecPolicies) 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-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

$newpolicy6 = New-AzIpsecPolicy -IkeEncryption AES128 -IkeIntegrity SHA1 -DhGroup DHGroup14 -
IpsecEncryption AES256 -IpsecIntegrity SHA256 -PfsGroup None -SALifeTimeSeconds 14400 -SADataSizeKilobytes
102400000

Set-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection6 -IpsecPolicies
$newpolicy6 -UsePolicyBasedTrafficSelectors $True

```

You can get the connection again to check if the policy is updated.

```

$connection6 = Get-AzVirtualNetworkGatewayConnection -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-AzVirtualNetworkGatewayConnection -Name $Connection16 -ResourceGroupName $RG1

$currentpolicy = $connection6.IpsecPolicies[0]
$connection6.IpsecPolicies.Remove($currentpolicy)

Set-AzVirtualNetworkGatewayConnection -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

2/11/2020 • 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-AzAccount
Select-AzSubscription -SubscriptionName $Sub1
New-AzResourceGroup -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-AzVirtualNetworkSubnetConfig -Name $FEsubName1 -AddressPrefix $FEsubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzVirtualNetwork -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-AzPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$gw1pip2 = New-AzPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic

$vnet1 = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gw1ipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress $gw1pip1
$gw1ipconf2 = New-AzVirtualNetworkGatewayIpConfig -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-AzVirtualNetworkGateway -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-AzPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1
$gw1pip2 = Get-AzPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1
$vnet1gw = Get-AzVirtualNetworkGateway -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:

```

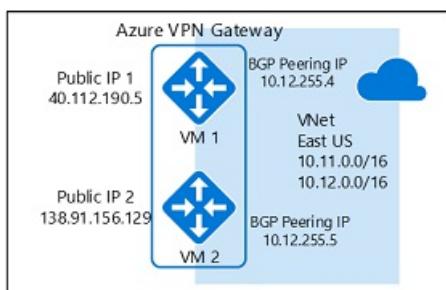
PS D:\> $gw1pip1.IpAddress
40.112.190.5

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"  
$LNGASNS5 = 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-AzResourceGroup -Name $RG5 -Location $Location5  
New-AzLocalNetworkGateway -Name $LNGName51 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress  
$LNGIP51 -AddressPrefix $LNGPrefix51 -Asn $LNGASNS5 -BgpPeeringAddress $BGPPeerIP51
```

Step 2 - Connect the VNet gateway and local network gateway

1. Get the two gateways

```
$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$lwgw1 = Get-AzLocalNetworkGateway -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.

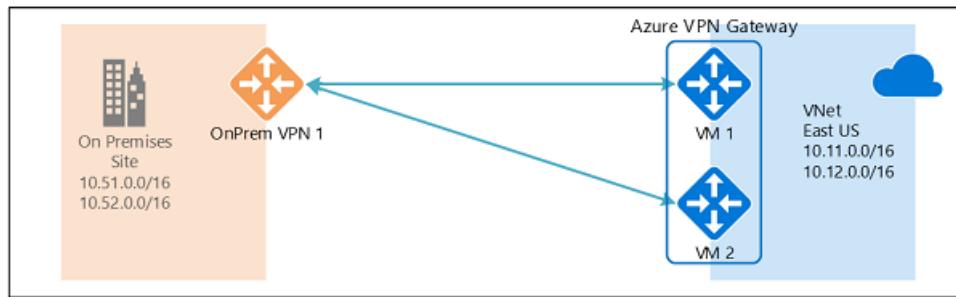
```
New-AzVirtualNetworkGatewayConnection -Name $Connection151 -ResourceGroupName $RG1 -VirtualNetworkGateway1
$vnet1gw -LocalNetworkGateway2 $lNG5gw1 -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -
EnableBGP $True
```

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-AzLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress
$LNGIP52 -AddressPrefix $LNGPrefix52 -Asn $LNGAS5 -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-AzLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5
```

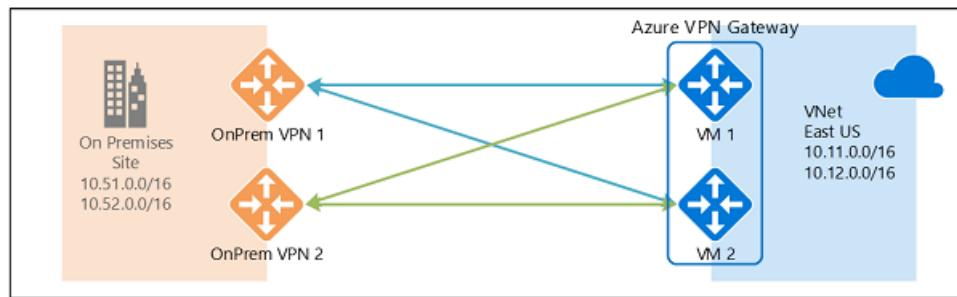
```
New-AzVirtualNetworkGatewayConnection -Name $Connection152 -ResourceGroupName $RG1 -VirtualNetworkGateway1
$vnet1gw -LocalNetworkGateway2 $lng5gw2 -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -
EnableBGP $True
```

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-AzResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$besub2 = New-AzVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsb2 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix
$VNetPrefix21,$VNetPrefix22 -Subnet $fesub2,$besub2,$gwsb2

```

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-AzPublicIpAddress -Name $GW2IPName1 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic
$gw2pip2 = New-AzPublicIpAddress -Name $GW2IPName2 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic

$vnet2 = Get-AzVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gw2IPconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GW2IPconf1 -Subnet $subnet2 -PublicIpAddress $gw2pip1
$gw2IPconf2 = New-AzVirtualNetworkGatewayIpConfig -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-AzVirtualNetworkGateway -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$vnet2gw = Get-AzVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2
```

2. Create both connections

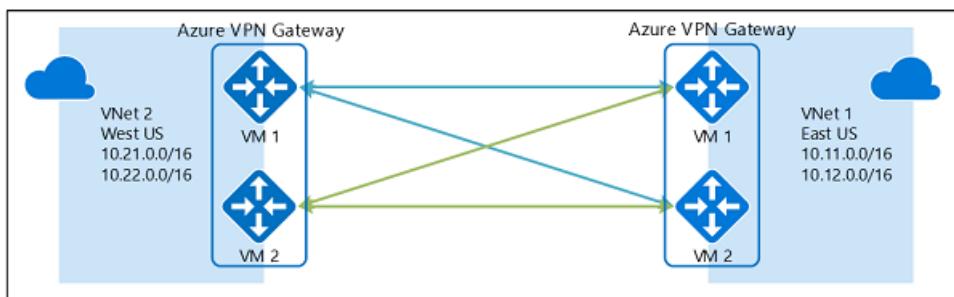
In this step, you will create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1.

```
New-AzVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1  
$vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey  
'AzureA1b2C3' -EnableBgp $True  
  
New-AzVirtualNetworkGatewayConnection -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 = "TestVPNActiveActive01"  
$GWIPName2 = "gwpip2"  
$GWIPconf2 = "gwipconf2"
```

After declaring the variables, you can copy and paste this example to your PowerShell console.

```
$vnet = Get-AzVirtualNetwork -Name $VNetName -ResourceGroupName $RG
$subnet = Get-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gw = Get-AzVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
$location = $gw.Location
```

2. Create the public IP address, then add the second gateway IP configuration

```
$gwpip2 = New-AzPublicIpAddress -Name $GWIPName2 -ResourceGroupName $RG -Location $location -AllocationMethod Dynamic
Add-AzVirtualNetworkGatewayIpConfig -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-AzVirtualNetworkGateway -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-AzVirtualNetworkGateway -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-AzVirtualNetworkGatewayIpConfig -Name $ipconfname -VirtualNetworkGateway $gw
Set-AzVirtualNetworkGateway -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.

How to configure BGP on Azure VPN Gateways using PowerShell

2/11/2020 • 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.

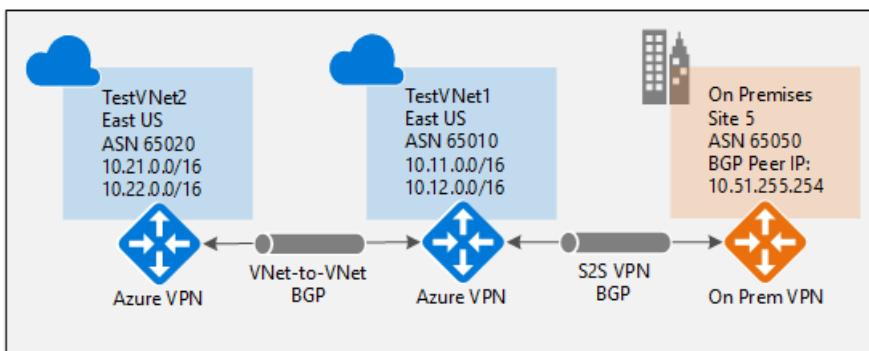
See [Overview of BGP with Azure VPN Gateways](#) for more discussion on benefits of BGP and to understand the technical requirements and considerations of using BGP.

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](#)

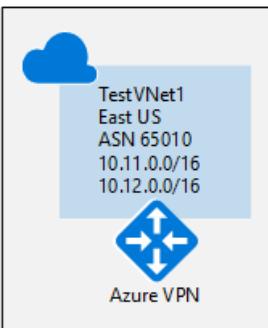
Each part of the instructions forms a basic building block for enabling BGP in your network connectivity. If you complete all three parts, you build the topology as shown in the following diagram:



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-AzAccount
Select-AzSubscription -SubscriptionName $Sub1
New-AzResourceGroup -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.

```

$zesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsu1 = New-AzVirtualNetworkSubnetConfig -Name $GWSuName1 -AddressPrefix $GWSuPrefix1

New-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $zesub1,$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-AzPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic

$vnet1 = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwpipconf1 = New-AzVirtualNetworkGatewayIpConfig -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -IpConfigurations
$gwpipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet1gw.BgpSettingsText

```

The last command shows the corresponding BGP configurations on the Azure VPN Gateway; for example:

```

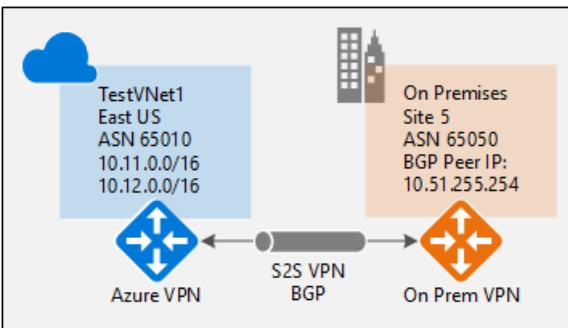
$vnet1gw.BgpSettingsText
{
    "Asn": 65010,
    "BgpPeeringAddress": "10.12.255.30",
    "PeerWeight": 0
}

```

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 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 `BgpPeeringAddress`.

```
New-AzResourceGroup -Name $RG5 -Location $Location5

New-AzLocalNetworkGateway -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -VirtualNetworkGateway1  
$vnet1gw -LocalNetworkGateway2 $lwg5gw -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -  
EnableBGP $True
```

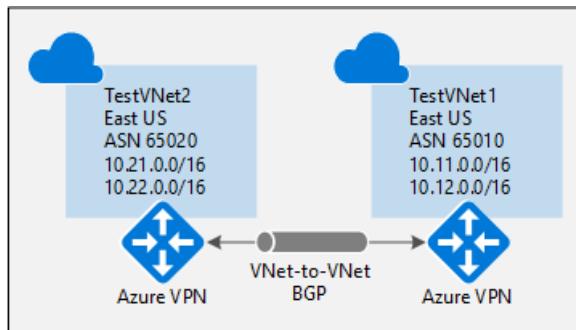
The following example lists the parameters you enter into the BGP configuration section on your on-premises VPN device for this exercise:

```
- 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-AzResourceGroup -Name $RG2 -Location $Location2

$fesub2 = New-AzVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$besub2 = New-AzVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$gwsb2 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzVirtualNetwork -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-AzPublicIpAddress -Name $GWIPName2 -ResourceGroupName $RG2 -Location $Location2 -
AllocationMethod Dynamic

$vnet2      = Get-AzVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2    = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet2
$gwipconf2 = New-AzVirtualNetworkGatewayIpConfig -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-AzVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -IpConfigurations
$gwipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -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-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet2gw = Get-AzVirtualNetworkGateway -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.

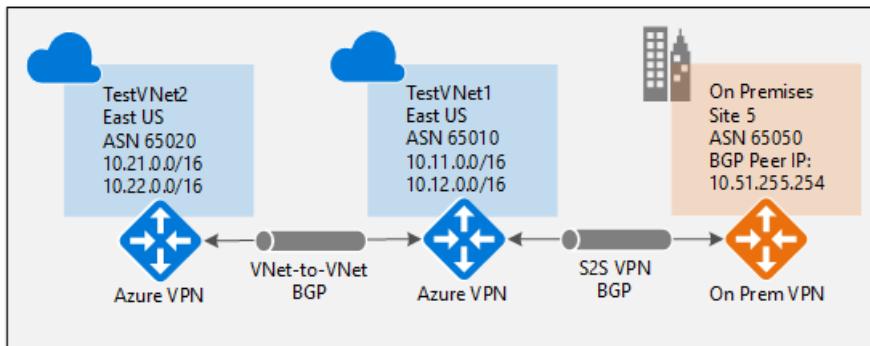
```
New-AzVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1  
$vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey  
'AzureA1b2C3' -EnableBgp $True  
  
New-AzVirtualNetworkGatewayConnection -Name $Connection21 -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

1/10/2020 • 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](#).

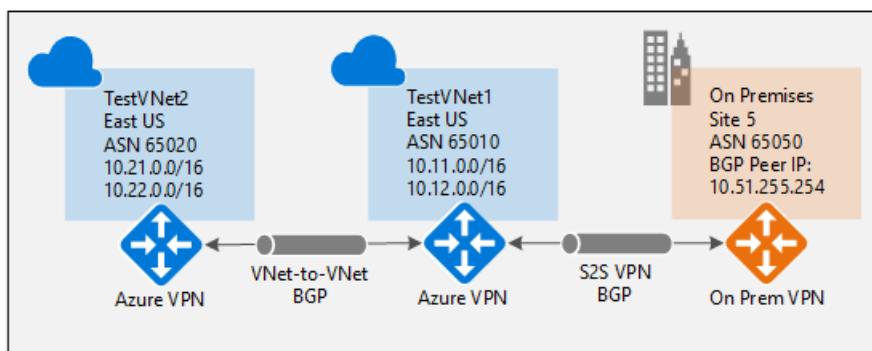
This article helps you with the following tasks:

- [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](#)

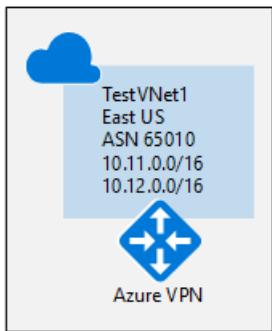
Each of these three sections forms a basic building block for enabling BGP in your network connectivity. If you complete all three sections, you build the topology as shown in the following diagram:



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 the Azure CLI](#) and [Get Started with Azure CLI](#).

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](#).

```
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:

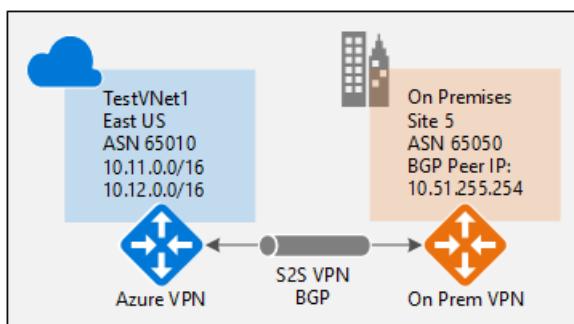
```
az network vnet-gateway list -g TestBGPRG1
```

```
"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 10.51.255.254/32.
- 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
```

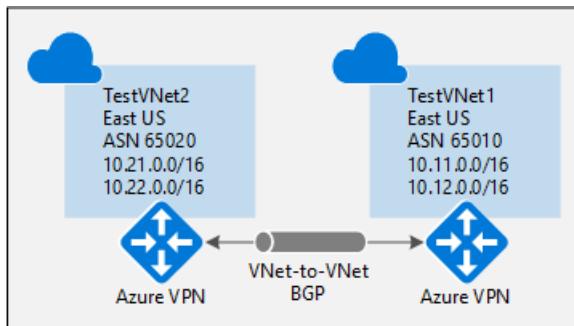
For this exercise, the following example lists the parameters to enter in the BGP configuration section of your on-premises VPN device:

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

2/11/2020 • 6 minutes to read • [Edit Online](#)

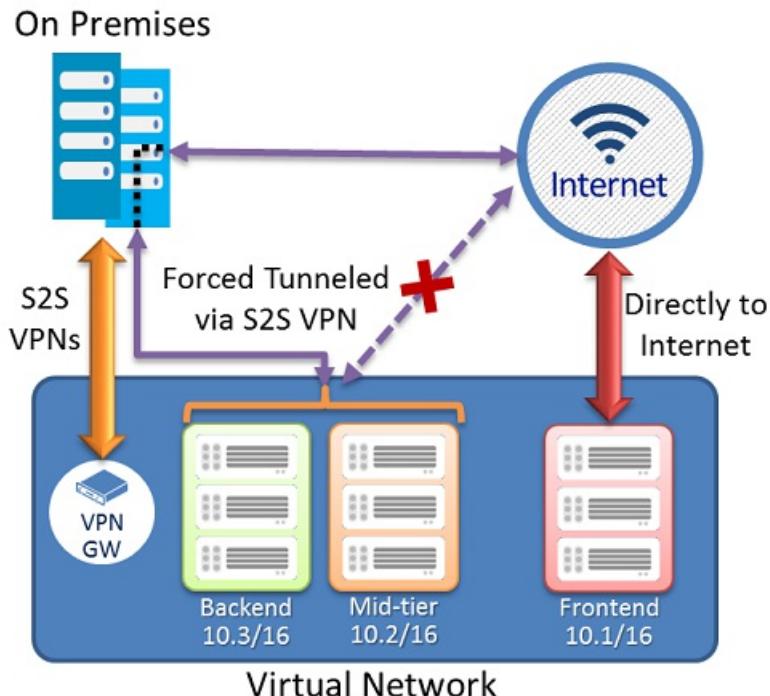
Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies. Without forced tunneling, Internet-bound traffic from your VMs 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

Open your PowerShell console with elevated privileges.

If you are running Azure PowerShell locally, connect to your Azure account. The `Connect-AzAccount` cmdlet prompts you for credentials. After authenticating, it downloads your account settings so that they are available to Azure PowerShell. If you are using Azure Cloud Shell instead, you do not need to run `Connect-AzAccount`. Azure Cloud Shell connects to your Azure account automatically.

```
Connect-AzAccount
```

If you have more than one subscription, get a list of your Azure subscriptions.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Name of subscription"
```

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-AzResourceGroup -Name 'ForcedTunneling' -Location 'North Europe'
```

2. Create a virtual network and specify subnets.

```
$s1 = New-AzVirtualNetworkSubnetConfig -Name "Frontend" -AddressPrefix "10.1.0.0/24"
$s2 = New-AzVirtualNetworkSubnetConfig -Name "Midtier" -AddressPrefix "10.1.1.0/24"
$s3 = New-AzVirtualNetworkSubnetConfig -Name "Backend" -AddressPrefix "10.1.2.0/24"
$s4 = New-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix "10.1.200.0/28"
$vnet = New-AzVirtualNetwork -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-AzLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling" -Location
"North Europe" -GatewayIpAddress "111.111.111.111" -AddressPrefix "192.168.1.0/24"
$lng2 = New-AzLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -GatewayIpAddress "111.111.111.112" -AddressPrefix "192.168.2.0/24"
$lng3 = New-AzLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -GatewayIpAddress "111.111.111.113" -AddressPrefix "192.168.3.0/24"
$lng4 = New-AzLocalNetworkGateway -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-AzRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling" -Location "North Europe"
$rt = Get-AzRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling"
Add-AzRouteConfig -Name "DefaultRoute" -AddressPrefix "0.0.0.0/0" -NextHopType VirtualNetworkGateway -
RouteTable $rt
Set-AzRouteTable -RouteTable $rt
```

5. Associate the route table to the Midtier and Backend subnets.

```
$vnet = Get-AzVirtualNetwork -Name "MultiTier-Vnet" -ResourceGroupName "ForcedTunneling"
Set-AzVirtualNetworkSubnetConfig -Name "MidTier" -VirtualNetwork $vnet -AddressPrefix "10.1.1.0/24" -
RouteTable $rt
Set-AzVirtualNetworkSubnetConfig -Name "Backend" -VirtualNetwork $vnet -AddressPrefix "10.1.2.0/24" -
RouteTable $rt
Set-AzVirtualNetwork -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-AzPublicIpAddress -Name "GatewayIP" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -AllocationMethod Dynamic
$gws subnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$ipconfig = New-AzVirtualNetworkGatewayIpConfig -Name "gwIpConfig" -SubnetId $gws subnet.Id -
PublicIpAddressId $pip.Id
New-AzVirtualNetworkGateway -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-AzLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling"
$VirtualGateway = Get-AzVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling"
Set-AzVirtualNetworkGatewayDefaultSite -GatewayDefaultSite $LocalGateway -VirtualNetworkGateway
$VirtualGateway
```

8. Establish the Site-to-Site VPN connections.

```
$gateway = Get-AzVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling"
$lng1 = Get-AzLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling"
$lng2 = Get-AzLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling"
$lng3 = Get-AzLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling"
$lng4 = Get-AzLocalNetworkGateway -Name "Branch3" -ResourceGroupName "ForcedTunneling"

New-AzVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng1 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzVirtualNetworkGatewayConnection -Name "Connection2" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng2 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzVirtualNetworkGatewayConnection -Name "Connection3" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng3 -ConnectionType
IPsec -SharedKey "preSharedKey"
New-AzVirtualNetworkGatewayConnection -Name "Connection4" -ResourceGroupName "ForcedTunneling" -
Location "North Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng4 -ConnectionType
IPsec -SharedKey "preSharedKey"

Get-AzVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling"
```

Configure forced tunneling using the classic deployment model

2/13/2020 • 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 Resource Manager 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:

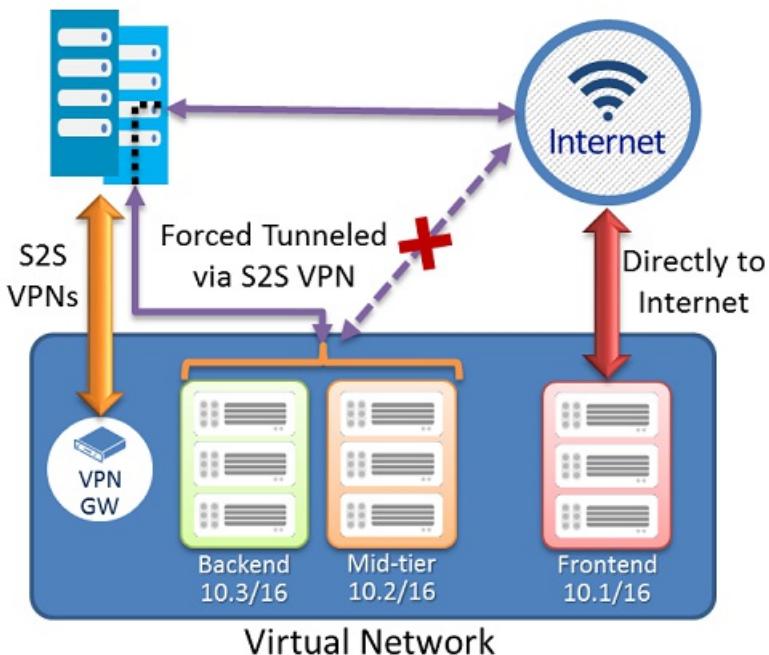
- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network.
 - **On-premises routes:** To the Azure VPN gateway.
 - **Default route:** Directly 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. 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.
- When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

To sign in

1. Open your PowerShell console with elevated rights. To switch to service management, use this command:

```
azure config mode asm
```

2. Connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

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

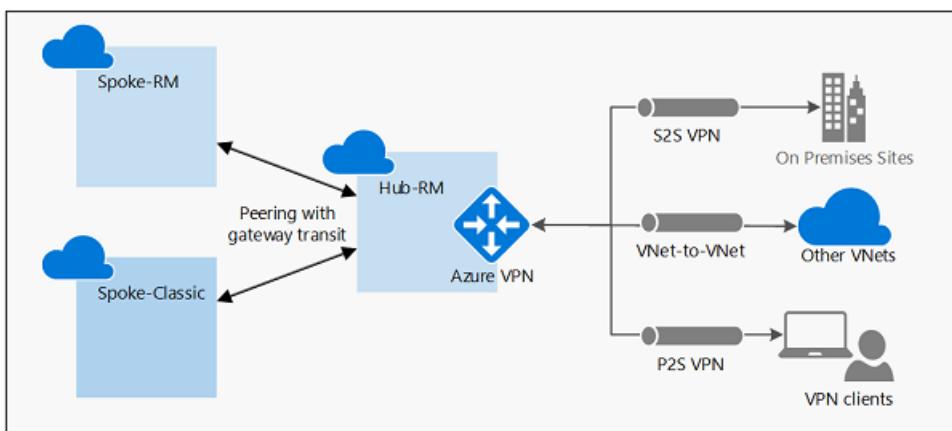
```
Remove-AzureVnetGatewayDefaultSite -VNetName <virtualNetworkName>
```

Configure VPN gateway transit for virtual network peering

2/12/2020 • 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

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:

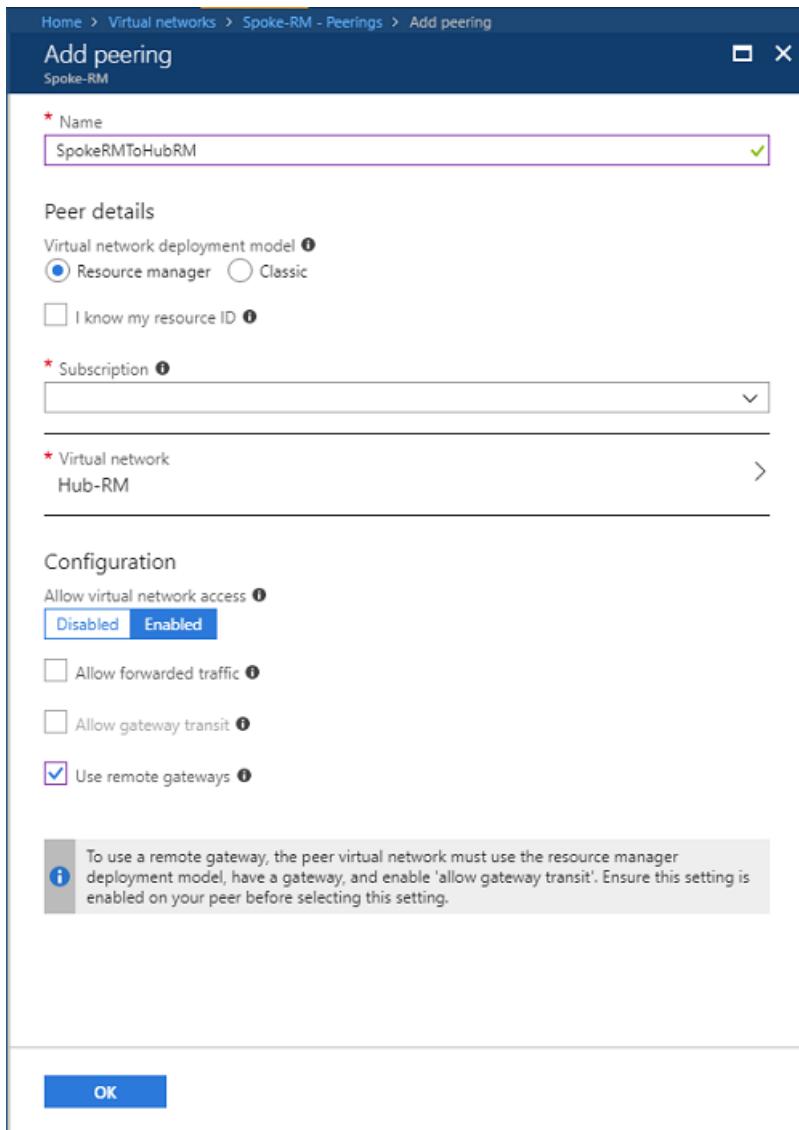
VIRTUAL NETWORK	DEPLOYMENT MODEL	ROLE	PERMISSIONS
Hub-RM	Resource Manager	Network Contributor	Microsoft.Network/virtualNetworks/virtualNetworkPeering/write
	Classic	Classic Network Contributor	N/A
Spoke-Classic	Resource Manager	Network Contributor	Microsoft.Network/virtualNetworks/peer
	Classic	Classic 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-AzVirtualNetwork -Name $SpokeRM -ResourceGroup $SpokeRG
$hubrmvnet   = Get-AzVirtualNetwork -Name $HubRM -ResourceGroup $HubRG

Add-AzVirtualNetworkPeering ` 
    -Name SpokeRMtoHubRM ` 
    -VirtualNetwork $spokermvnet ` 
    -RemoteVirtualNetworkId $hubrmvnet.Id ` 
    -UseRemoteGateways

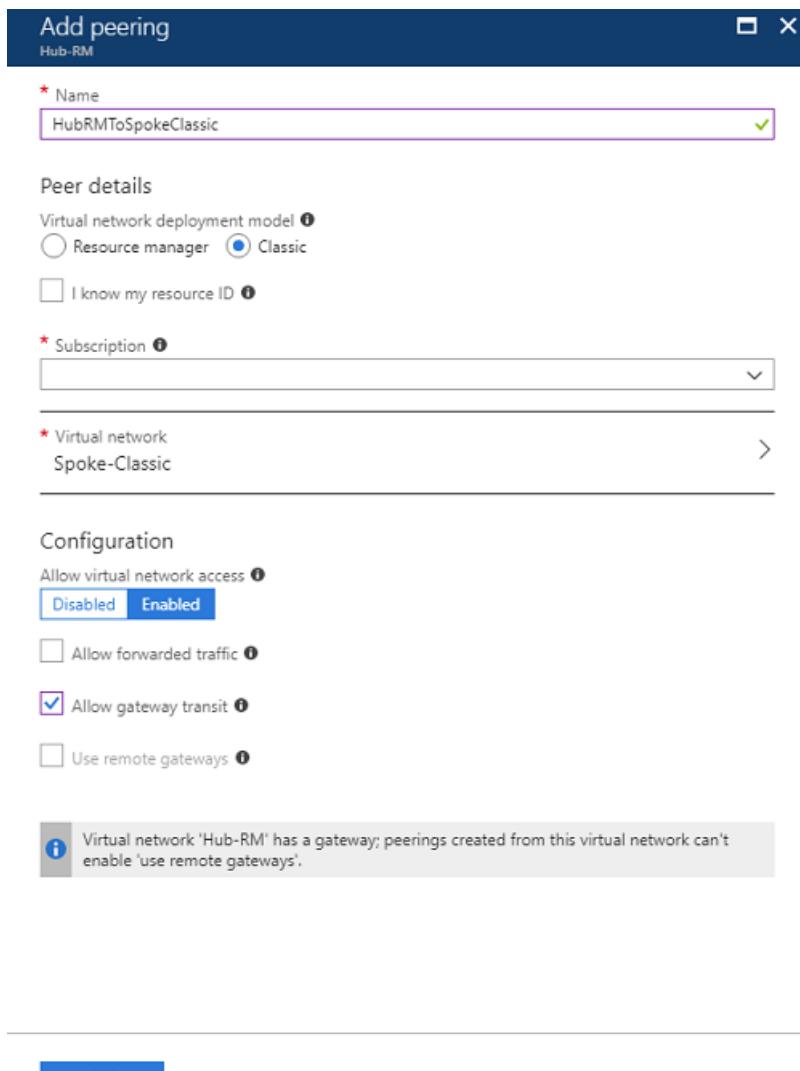
Add-AzVirtualNetworkPeering ` 
    -Name HubRMTospokeRM ` 
    -VirtualNetwork $hubrmvnet ` 
    -RemoteVirtualNetworkId $spokermvnet.Id ` 
    -AllowGatewayTransit

```

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-AzVirtualNetwork -Name $HubRM -ResourceGroup $HubRG

Add-AzVirtualNetworkPeering ` 
-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

1/10/2020 • 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 delete the connection, you may want to download the configuration for your connecting devices in order to get the defined PSK. That way, you don't need to redefine it on the other side.

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

1/10/2020 • 4 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:

1. Set the variable for the LocalNetworkGateway.

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

2. Modify the prefixes.

```
Set-AzLocalNetworkGateway -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.

1. Set the variable for the LocalNetworkGateway.

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

2. Set the gateway with the updated prefixes.

```
Set-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection -Name VNet1toSite1 -ResourceGroupName TestRG1
```

- Set the local network gateway with the modified address prefixes.

Set the variable for the LocalNetworkGateway.

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

Modify the prefixes.

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

- 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-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

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

```
New-AzVirtualNetworkGatewayConnection -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-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection' cmdlet.

```
Remove-AzVirtualNetworkGatewayConnection -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-AzLocalNetworkGateway -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-AzVirtualNetworkGateway' cmdlet.

Set the variables.

```
$local = Get-AzLocalNetworkGateway -Name Site1 -ResourceGroupName TestRG1  
  
$vnetgw = Get-AzVirtualNetworkGateway -Name VNet1GW -ResourceGroupName TestRG1
```

Create the connection.

```
New-AzVirtualNetworkGatewayConnection -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

1/10/2020 • 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 the Azure CLI](#).

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](#).

```
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](#).

Overview of partner VPN device configurations

2/11/2020 • 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-AzAccount
Select-AzSubscription -SubscriptionName $Sub1
New-AzResourceGroup -Name $RG1 -Location $Location1

# Create virtual network

$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNetPrefix11,$VNetPrefix12 -Subnet $fesub1,$besub1,$gwsb1

# Create VPN gateway

$gwip1    = New-AzPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 -Location $Location1 -
AllocationMethod Dynamic
$vnet1    = Get-AzVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1  = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 -Subnet $subnet1 -PublicIpAddress
$gwip1

New-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -InConfigurations
```

```
$gwpipconf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1 -Asn $VNet1ASN

# Create local network gateway

New-AzLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1 -Location $Location1 -GatewayIpAddress $LNGIP5 -AddressPrefix $LNGPrefix51,$LNGPrefix52 -Asn $LNGASNS -BgpPeeringAddress $BGPPeerIP5

# Create the S2S VPN connection

$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1

New-AzVirtualNetworkGatewayConnection -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-AzIpsecPolicy -IkeEncryption AES256 -IkeIntegrity SHA384 -DhGroup DHGroup24 -IpsecEncryption AES256 -IpsecIntegrity SHA1 -PfsGroup PFS24 -SALifeTimeSeconds 7200 -SADataSizeKilobytes 20480000

$vnet1gw = Get-AzVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG1

New-AzVirtualNetworkGatewayConnection -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-AzLocalNetworkGateway -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-AzVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -  
VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw -Location $Location1 -ConnectionType  
IPsec -SharedKey 'AzureA1b2C3' -EnableBGP $True
```

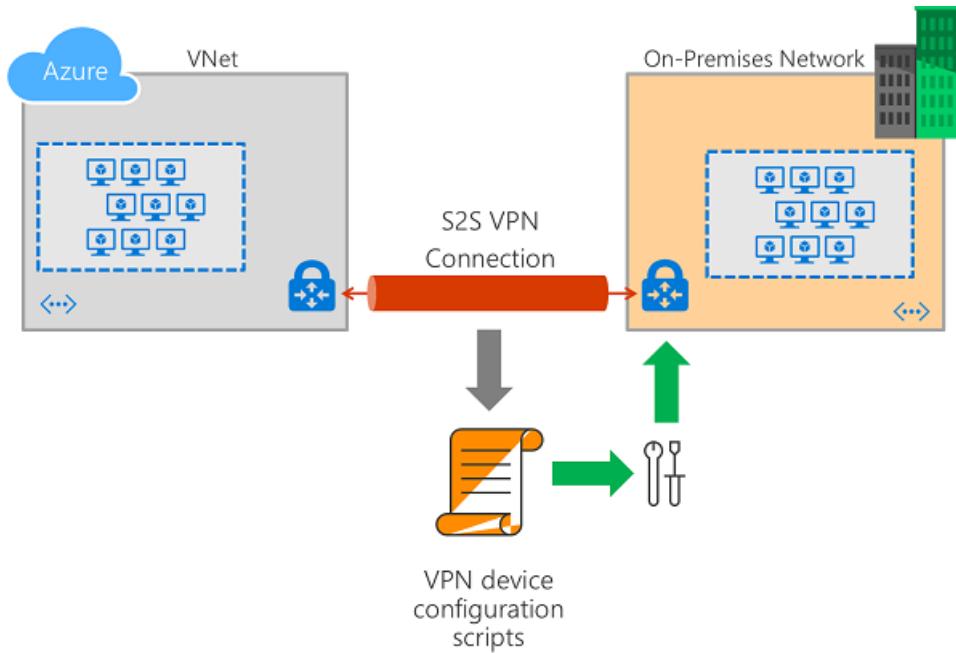
Next steps

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](#).

Download VPN device configuration scripts for S2S VPN connections

2/12/2020 • 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
Juniper	SRX	JunOS 12.x RouteBased BGP
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased VTI
Ubiquiti	EdgeRouter	EdgeOS v1.10x RouteBased BGP

NOTE

(*) Required: NarrowAzureTrafficSelectors (enable UsePolicyBasedTrafficSelectors option) 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. In the top navigation bar, there are links for 'Report a bug' and 'Search resources, services and docs'. Below the navigation bar, the 'Connections' section is displayed. On the left sidebar, under 'FAVORITES', are 'Dashboard', 'Resource groups', 'All resources', 'Virtual machines (classic)', 'Virtual network gateway...', and 'Virtual networks'. Under 'All services', there is a '+ Create a resource' button. The main content area shows a table titled 'Subscriptions: 2 of 44 selected' with one item listed: 'VNet1toSite5' (Status: Connecting, Peer 1: VNet1GW, Peer 2: Site5, Resource Group: TestRG1, Location: East US 2, Subscription: TestUS2). There are buttons for 'Add', 'Edit columns', 'Refresh', and 'Assign Tags'.

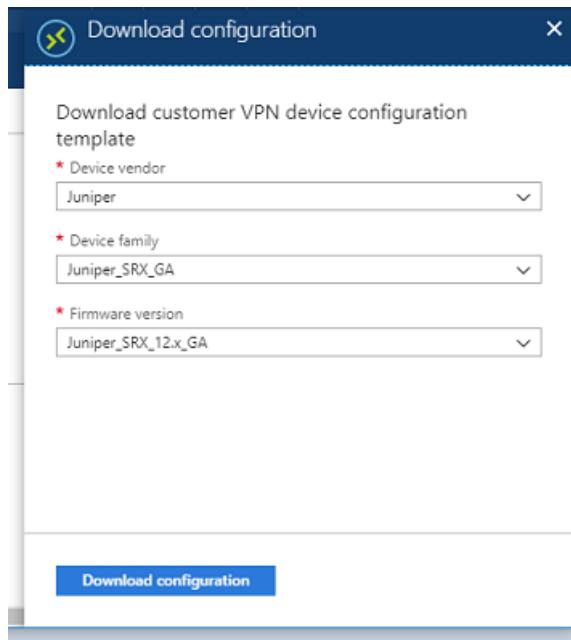
3. Click on the connection you want to configure.

The screenshot shows the 'VNet1toSite5 Connection' overview page. At the top, there is a breadcrumb trail: Home > Connections > VNet1toSite5. Below the title, there is a search bar and buttons for 'Move', 'Download configuration' (which is highlighted with a red box), and 'Delete'. The main content area displays details about the connection, including its resource group (TestRG1), location (East US 2), subscription (TestVNet1), and various network components like VNet1GW and Site5. On the left side, there is a sidebar with links for 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'SETTINGS', 'Shared key', 'Configuration', and 'Properties'.

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. At the top, there is a breadcrumb trail: Home > Connections > VNet1toSite5 > Download configuration. Below the title, there is a search bar and buttons for 'Move', 'Download configuration' (highlighted with a red box), and 'Delete'. The main content area is titled 'Download customer VPN device configuration template' and includes fields for 'Device vendor' (dropdown menu), 'Device family' (dropdown menu), and 'Firmware version' (dropdown menu). At the bottom right, there 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.

```

228 set security ipsec proposal azure-ipsec-proposal-VNettoSite5-40.70.3.155 lifetime-seconds 3600
229
230 ! DEFINING THE IPSEC (PHASE 2) POLICY FOR AZURE
231 set security ipsec policy azure-ipsec-policy-VNettoSite5-40.70.3.155 proposal: Find Replace Find in Files Mark
232
233 ! DEFINING THE IPSEC (PHASE 2) VPN FOR AZURE: Binding to the secure tunnel is
234 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 bind-interface
235
236 ! Note: Please REPLACE the destination-ip below by the IP address of an active
237 ! to for the VPN Monitor functionality (Phase 2 livelihood) of your SRX
238
239 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 vpn-monitor < Find
240 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 ike gateway & Replace Find in Files Mark
241 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 ike proxy-ida < Find what: REPLACE >
242 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 ike proxy-ida < Find Next >
243 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 ike ipsec-pol < Count >
244 set security ipsec vprf azure-ipsec-vprf-VNettoSite5-40.70.3.155 establish-tunnel < Find All in All Opened Documents >
245
246 !
247 ! SETTING THE SECURITY ZONES FOR AZURE
248 !—————
249 ! NOTE: The zones are defined as:
250 ! 1) "Internal" from the ingress (internal facing) on-premises network behavior saw INT
251 ! "External" from the egress (external facing) Azure network, which the external interface is connected to.
252 ! REPLACE these as needed.
253 !
254 ! 2) The on-premises network is tied to a virtual interface on the SRX labeled "gian.1".
255 ! The external (public facing) interface and port on the SRX for Azure is labeled "fx-0/0/0.0".
256 ! REPLACE these as needed.
257 !
258
259
260
261 ! INTERNAL ZONE DEFINITION
262
263 set security zones security-zone Internal address-book address onprexg-networks-VNettoSite5-40.70.3.155 10.51.0.0/16
264 set security zones security-zone Internal address-book address onprexg-networks-VNettoSite5-40.70.3.155 10.52.0.0/16
265
266 ! The VLAN (gian.1) representing the on-premises network. REPLACE the INT as needed.
267
268 set security zones security-zone Internal host-inbound-traffic system-services all
269 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services all
270 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services ping
271 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services ssh
272 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services http
273 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services https
274 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services ssh
275 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services telnet
276 set security zones security-zone Internal interfaces gian.1 host-inbound-traffic system-services all
277
278 ! The physical LAN interface (fx-0/0/1.0) for the on-premises network. REPLACE the INT as needed.
279 set security zones security-zone Internal interfaces fx-0/0/1.0 host-inbound-traffic system-services all
280
281 ! INTERNET ZONE DEFINITION
282

```

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-AzVirtualNetworkGatewaySupportedVpnDevice -Name $GWName -ResourceGroupName $RG

# Download the configuration script for the connection
Get-AzVirtualNetworkGatewayConnectionVpnDeviceConfigScript -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](#).

Sample configuration: Cisco ASA device (IKEv2/no BGP)

1/9/2020 • 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 proposal, 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](#).

Set up alerts on VPN Gateway metrics

7/5/2019 • 2 minutes to read • [Edit Online](#)

This article helps you set up alerts on Azure VPN Gateway metrics. Azure Monitor provides the ability to set up alerts for Azure resources. You can set up alerts for virtual network gateways of the "VPN" type.

METRIC	UNIT	GRANULARITY	DESCRIPTION
AverageBandwidth	Bytes/s	5 minutes	Average combined bandwidth utilization of all site-to-site connections on the gateway.
P2SBandwidth	Bytes/s	1 minute	Average combined bandwidth utilization of all point-to-site connections on the gateway.
P2SConnectionCount	Count	1 minute	Count of point-to-site connections on the gateway.
TunnelAverageBandwidth	Bytes/s	5 minutes	Average bandwidth utilization of tunnels created on the gateway.
TunnelEgressBytes	Bytes	5 minutes	Outgoing traffic on tunnels created on the gateway.
TunnelEgressPackets	Count	5 minutes	Count of outgoing packets on tunnels created on the gateway.
TunnelEgressPacketDropTSMismatch	Count	5 minutes	Count of outgoing packets dropped on tunnels caused by traffic-selector mismatch.
TunnelIngressBytes	Bytes	5 minutes	Incoming traffic on tunnels created on the gateway.
TunnelIngressPackets	Count	5 minutes	Count of incoming packets on tunnels created on the gateway.
TunnelIngressPacketDropTSMismatch	Count	5 minutes	Count of incoming packets dropped on tunnels caused by traffic-selector mismatch.

Set up Azure Monitor alerts based on metrics by using the Azure portal

The following example steps will create an alert on a gateway for:

- **Metric:** TunnelAverageBandwidth

- **Condition:** Bandwidth > 10 bytes/second
- **Window:** 5 minutes
- **Alert action:** Email

1. Go to the virtual network gateway resource and select **Alerts** from the **Monitoring** tab. Then create a new alert rule or edit an existing alert rule.

The screenshot shows the Azure portal interface for managing alerts. On the left, there's a navigation menu with 'Monitoring' selected, which is highlighted with a red box. Within the monitoring section, 'Alerts' is also highlighted with a red box. At the top, there are buttons for 'New alert rule', 'Manage alert rules', 'Manage action groups', and 'View classic alerts'. Below these are dropdown menus for 'Subscription' (set to 'TestRG1'), 'Resource group' (set to 'TestRG1'), and 'Resource' (set to 'VNet1GW'). A time range selector shows 'Past 24 hours'. The main area has a heading 'Pay attention to what matters.' and a message stating 'You have not configured any alert rules.' It includes a callout to 'Configure alert rules and attend to fired alerts to efficiently monitor your Azure resources.' There's also a diagram of a network with a central cloud icon and various connection points. At the bottom, there's a 'New Alert Rule' button and a note about classic alerts.

2. Select your VPN gateway as the resource.

The screenshot shows the 'Create rule' wizard in progress. The first step, 'Create rule', is completed. The second step, 'Select a resource', is active. On the left, there's a 'RESOURCE' section with a 'VNet1GW' item, which is highlighted with a red box. Below it is a 'CONDITION' section with a placeholder 'No condition defined...'. The 'ACTION GROUPS' section shows a single item 'VNet1GW' selected, also highlighted with a red box. The 'ALERT DETAILS' section includes fields for 'Alert rule name' (with a sample value 'Percentage CPU greater than 70') and 'Description'. At the bottom, there's a 'Create alert rule' button and a 'Done' button.

3. Select a metric to configure for the alert.

Dashboard > Resource groups > TestRG1 > VNetGW - Alerts > Create rule

Create rule

RESOURCE VNnetGW

HIERARCHY TestRG1

CONDITION No condition defined, click on 'Add condition' to select a signal and define its logic.

ACTION GROUPS Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more here.

ACTION GROUP NAME ACTION GROUP TYPE

No action group selected

Select existing Create New

ALERT DETAILS

- Alert rule name: Sample - Percentage CPU greater than 70
- Description: Specify alert description here...

Enable rule upon creation: Yes No

Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

All signals (99)

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Gateway P2S Bandwidth	Metric	Platform
Gateway P2S Bandwidth	Metric	Platform
P2S Connection Count	Metric	Platform
Tunnel Bandwidth	Metric	Platform
Tunnel Egress Bytes	Metric	Platform
Tunnel Ingress Bytes	Metric	Platform
Tunnel Egress Packets	Metric	Platform
Tunnel Ingress Packets	Metric	Platform
Tunnel Egress TS Mismatch Packet Drop	Metric	Platform
Tunnel Ingress TS Mismatch Packet Drop	Metric	Platform
All Administrative operations	Activity Log	Administrative
List Supported Vpn Devices (virtualnetworkgateways)	Activity Log	Administrative
Get VirtualNetworkGateway (virtualnetworkgateways)	Activity Log	Administrative
Creates or updates a VirtualNetworkGateway (virtualnetworkgateways)	Activity Log	Administrative
Deletes a virtualNetworkGateway (virtualnetworkgateways)	Activity Log	Administrative
Generate VpnClient package for VirtualNetworkGateway (virtual...	Activity Log	Administrative
Generate VpnProfile package for VirtualNetworkGateway (virtual...	Activity Log	Administrative
Get P2S Client Connection Health for VirtualNetworkGateway...	Activity Log	Administrative
Gets the URL of a pre-generated vpn client profile package (virtua...	Activity Log	Administrative
Set Vpnclient Ipcac parameters for VirtualNetworkGateway P2S cl...	Activity Log	Administrative
Get Vpnclient Ipcac parameters for VirtualNetworkGateway P2S cl...	Activity Log	Administrative
Reset Vpnclient shared key for VirtualNetworkGateway P2S client...	Activity Log	Administrative
Resets a virtualNetworkGateway (virtualnetworkgateways)	Activity Log	Administrative
Gets virtualNetworkGateway advertised routes (virtualnetworkgateways)	Activity Log	Administrative
Gets virtualNetworkGateway bgp peer status (virtualnetworkgateways)	Activity Log	Administrative

4. Configure the signal logic. There are three components to it:

a. **Dimensions:** If the metric has dimensions, you can select specific dimension values so that the alert evaluates only data of that dimension. These are optional.

b. **Condition:** This is the operation to evaluate the metric value.

c. **Time:** Specify the granularity of metric data, and the period of time to evaluate the alert.

Dashboard > Resource groups > TestRG1 > VNetGW - Alerts > Create rule

Create rule

RESOURCE VNnetGW

HIERARCHY TestRG1

CONDITION No condition defined, click on 'Add condition' to select a signal and define its logic.

ACTION GROUPS Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more here.

ACTION GROUP NAME ACTION GROUP TYPE

No action group selected

Select existing Create New

ALERT DETAILS

- Alert rule name: Sample - Percentage CPU greater than 70
- Description: Specify alert description here...

Enable rule upon creation: Yes No

Configure signal logic

Connection Name: Over the last 6 hours

Tunnel Bandwidth (Avg) 0 bytes

This metric supports dimensions. Selecting the dimension values will help you filter to the right time series. If you do not select any value for a dimension, that dimension will be ignored.

DIMENSION NAME	DIMENSION VALUES	SELECT *
Connection Name	0 selected	<input checked="" type="checkbox"/>
Remote IP	0 selected	<input type="checkbox"/>

Checking "Select *" will dynamically select all current and future dimension values for the dimension.

Alert logic

Threshold: Static

Operator: Greater than Aggregation type: Average Threshold value: 10 bytes/second

Condition preview: Whenever the tunnel bandwidth is greater than 10 bytes/second

Evaluated based on: Aggregation granularity (Period): 5 minutes Frequency of evaluation: Every 5 Minutes

Done

5. To view the configured rules, select **Manage alert rules**.

The screenshot shows the Azure portal interface for managing alerts on a Virtual Network Gateway named 'VNet1GW'. The top navigation bar includes a search bar, 'New alert rule', 'Manage alert rules' (which is highlighted with a red box), 'Manage action groups', 'View classic alerts', and 'Refresh'. Below the navigation is a message about subscription settings and a dropdown for selecting a resource group ('TestRG1'). The main content area shows the resource group hierarchy: AlZam > TestRG1 > VNet1GW. On the left sidebar, under the 'Monitoring' section, the 'Alerts' link is also highlighted with a red box. Other monitoring options like 'Metrics' are listed below it. The sidebar also contains sections for 'Settings' (Configuration, Connections, Point-to-site configuration, Properties, Locks, Export template) and 'Support + troubleshooting' (Resource health, Reset, VPN troubleshoot, New support request).

Next steps

To configure alerts on tunnel diagnostic logs, see [Set up alerts on VPN Gateway diagnostic logs](#).

Set up alerts on diagnostic log events from VPN Gateway

1/16/2020 • 2 minutes to read • [Edit Online](#)

This article helps you set up alerts based on diagnostic log events from Azure VPN Gateway using Azure Log Analytics.

The following logs are available in Azure:

NAME	DESCRIPTION
GatewayDiagnosticLog	Contains diagnostic logs for gateway configuration events, primary changes and maintenance events
TunnelDiagnosticLog	Contains tunnel state change events. Tunnel connect/disconnect events have a summarized reason for the state change if applicable
RouteDiagnosticLog	Logs changes to static routes and BGP events that occur on the gateway
IKEDiagnosticLog	Logs IKE control messages and events on the gateway
P2SDiagnosticLog	Logs point-to-site control messages and events on the gateway

Set up alerts

The following example steps will create an alert for a disconnection event that involves a site-to-site VPN tunnel:

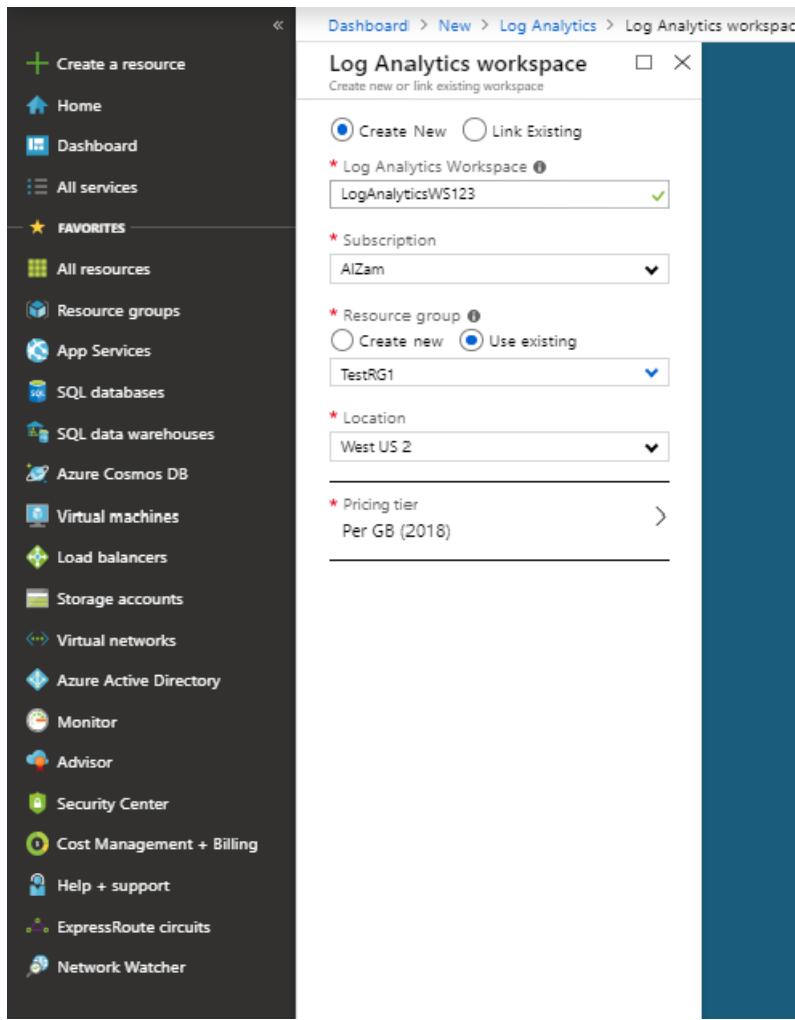
1. In the Azure portal, search for **Log Analytics** under **All services** and select **Log Analytics workspaces**.

The screenshot shows the Azure portal's left sidebar with various service icons and names. A red box highlights the 'All services' link under the 'FAVORITES' section. The main content area shows a search bar with 'log analytics' typed in, and a list of categories like 'Everything', 'General', 'Compute', etc., below it.

2. Select **Create** on the **Log Analytics** page.

The screenshot shows the 'Log Analytics' creation page. The left sidebar is identical to the previous one. The main area features a Microsoft logo and the text 'Log Analytics'. Below this are two buttons: a large blue 'Create' button with a red box around it, and a smaller 'Save for later' button. Further down, there are sections for creating a new workspace and linking an existing one, along with an 'About Log Analytics' summary and useful links.

3. Select **Create New** and fill in the details.



4. Find your VPN gateway on the **Monitor > Diagnostics settings** blade.

The screenshot shows the 'Monitor - Diagnostics settings' blade. The left sidebar highlights the 'Monitor' section. The main area displays a table of resources:

NAME	RESOURCE TYPE	RESOURCE GROUP	DIAGNOSTICS STATUS
VNetGW	Virtual network gateway	TestRG1	Disabled

5. To turn on diagnostics, double-click the gateway and then select **Turn on diagnostics**.

The screenshot shows the Azure portal's 'Monitor - Diagnostics settings' page for a Microsoft resource. The left sidebar includes 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES' (with 'All resources' selected), 'Resource groups', 'App Services', 'SQL databases', 'SQL data warehouses', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Azure Active Directory', 'Monitor', 'Advisor', 'Security Center', 'Cost Management + Billing', 'Help + support', 'ExpressRoute circuits', and 'Network Watcher'. The main area shows 'Subscription: [empty]', 'Resource group: TestRG1', and 'Resource type: Virtual network gateways'. A red box highlights the 'Turn on diagnostics' section, which contains a list of log types: GatewayDiagnosticLog, TunnelDiagnosticLog, RouteDiagnosticLog, IKEDiagnosticLog, P2SDiagnosticLog, and AllMetrics.

6. Fill in the details, and ensure that **Send to Log Analytics** and **TunnelDiagnosticLog** are selected. Choose the Log Analytics Workspace that you created in step 3.

The screenshot shows the 'Diagnostics settings' page for a virtual network gateway. The left sidebar is identical to the previous screenshot. The main area has 'Save' and 'Discard' buttons. A red box highlights the 'Send to Log Analytics' checkbox, which is checked. Another red box highlights the 'TunnelDiagnosticLog' checkbox, which is also checked. Other available log types include 'GatewayDiagnosticLog', 'RouteDiagnosticLog', 'IKEDiagnosticLog', and 'P2SDiagnosticLog'. The 'LOG' section is expanded, and the 'METRIC' section is collapsed.

NOTE

It may take a few hours for the data to show up initially.

7. Go to the overview for the virtual network gateway resource and select **Alerts** from the **Monitoring** tab.

Then create a new alert rule or edit an existing alert rule.

The classic alerts can be accessed from [here](#).

The classic alerts can be accessed from [here](#).

8. Select the Log Analytics workspace and the resource.

Available signal(s): Log, Metric, Activity Log

9. Select **Custom log search** as the signal logic under **Add condition**.

Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

All signals (153)

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Custom log search	Log	Log analytics
All Events	Log(Saved Query)	Log analytics
Count of Events containing the word "start" grouped by Event ID	Log(Saved Query)	Log analytics
Count of Events grouped by Event Log	Log(Saved Query)	Log analytics
Count of Events grouped by Event Source	Log(Saved Query)	Log analytics
Count of Events grouped by Event ID	Log(Saved Query)	Log analytics
All Events with level "Warning"	Log(Saved Query)	Log analytics
Count of Events with level "Warning" grouped by Event ID	Log(Saved Query)	Log analytics
How many connections to Operations Manager Event ID is in	Log(Saved Query)	Log analytics
When did my servers initiate restart?	Log(Saved Query)	Log analytics
Windows Firewall Policy settings have changed	Log(Saved Query)	Log analytics
On which machines and how many times was Windows Firewall	Log(Saved Query)	Log analytics
All IIS Log Entries	Log(Saved Query)	Log analytics
Shows breakdown of response codes	Log(Saved Query)	Log analytics
Find the maximum time taken for each page	Log(Saved Query)	Log analytics
Shows which pages people are getting a 404 for	Log(Saved Query)	Log analytics
Average HTTP Request time by HTTP Method	Log(Saved Query)	Log analytics
Shows servers that are throwing internal server error	Log(Saved Query)	Log analytics

10. Enter the following query in the **Search query** text box. Replace the values in <> and TimeGenerated as appropriate.

```
AzureDiagnostics
| where Category == "TunnelDiagnosticLog"
| where _ResourceId == tolower("<RESOURCEID OF GATEWAY>")
| where TimeGenerated > ago(5m)
| where remoteIP_s == "<REMOTE IP OF TUNNEL>"
| where status_s == "Disconnected"
| project TimeGenerated, OperationName, instance_s, Resource, ResourceGroup, _ResourceId
| sort by TimeGenerated asc
```

Set the threshold value to 0 and select **Done**.

Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

All signals (153)

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Custom log search	Log	Log analytics
All Events	Log(Saved Query)	Log analytics
Count of Events containing the word "start" grouped by Event ID	Log(Saved Query)	Log analytics
Count of Events grouped by Event Log	Log(Saved Query)	Log analytics
Count of Events grouped by Event Source	Log(Saved Query)	Log analytics
Count of Events grouped by Event ID	Log(Saved Query)	Log analytics
All Events with level "Warning"	Log(Saved Query)	Log analytics
Count of Events with level "Warning" grouped by Event ID	Log(Saved Query)	Log analytics
How many connections to Operations Manager Event ID is in	Log(Saved Query)	Log analytics
When did my servers initiate restart?	Log(Saved Query)	Log analytics
Windows Firewall Policy settings have changed	Log(Saved Query)	Log analytics
On which machines and how many times was Windows Firewall	Log(Saved Query)	Log analytics
All IIS Log Entries	Log(Saved Query)	Log analytics
Shows breakdown of response codes	Log(Saved Query)	Log analytics
Find the maximum time taken for each page	Log(Saved Query)	Log analytics
Shows which pages people are getting a 404 for	Log(Saved Query)	Log analytics
Average HTTP Request time by HTTP Method	Log(Saved Query)	Log analytics
Shows servers that are throwing internal server error	Log(Saved Query)	Log analytics

11. On the **Create rule** page, select **Create New** under the **ACTION GROUPS** section. Fill in the details and select **OK**.

Add action group

* Action group name: Email Administrators

* Short name: Email

* Subscription: AlZam

* Resource group: Default-ActivityLogAlerts

Actions

Action Name	Action Type	Status	Details	Actions
EmailAdministrators	Email/SMS/Push/Voice	✓	Edit details	X

Please configure the action by clicking the link.

Unique name for the action... ▾

[Privacy Statement](#)

[Pricing](#)

Info: Have a consistent format in emails, notifications and other endpoints irrespective of monitoring source. You can enable per action by editing details. [Learn more](#)

Email/SMS/Push/Voice

Name: EmailAdministrators

Email: xyz@microsoft.com

Email Azure Resource Manager Role

Owner: ▾

SMS

Country code: 1 Phone number: 1234567890

Info: Carrier charges may apply.

Azure app Push Notifications

Learn about connecting to your Azure resources using the Azure app.

email@example.com

This is the email you use to log into your Azure account.

Voice

Country code: 1 Phone number: 1234567890

Enable the common alert schema. [Learn more](#)

Yes No

OK

12. On the **Create rule** page, fill in the details for **Customize Actions** and make sure that the correct name appears in the **ACTION GROUP NAME** section. Select **Create alert rule** to create the rule.

The screenshot shows the 'Create rule' page in the Azure portal under 'Rules management'. On the left, a sidebar lists various Azure services. The main area is titled 'Create rule' and shows the following configuration:

- RESOURCE:** LogAnalyticsWS123
- HIERARCHY:** TestRG1
- CONDITION:** Whenever the Custom log search is Greater than 0 count
- ACTION GROUPS:** Email Administrators (selected)
- Customize Actions:** Includes 'Email subject' (checked), 'Subject line' (Redmond VPN tunnel is disconnected), and 'Include custom JSON payload for webhook' (unchecked).
- ALERT DETAILS:** Alert rule name: The Azure to Redmond tunnel is disconnected. Description: The tunnel between Azure and Redmond with IP address 104.42.209.46 is disconnected.
- Severity:** Warning(Sev 1)
- Enable rule upon creation:** Yes

A red box highlights the 'Customize Actions' and 'ALERT DETAILS' sections.

Next steps

To configure alerts on tunnel metrics, see [Set up alerts on VPN Gateway metrics](#).

Configure packet captures for VPN gateways

12/18/2019 • 2 minutes to read • [Edit Online](#)

Connectivity and performance-related issues are often times complex and take significant amount of time and effort just to narrow down the cause of the problem. Ability to packet capture greatly helps reduce time in narrowing down the scope of the problem to certain parts of the network, such as whether the issue is on the customer side of the network, the Azure side of the network, or somewhere in between. Once the issue has been narrowed down, it is much more efficient to debug and take remedial action.

There are some commonly available tools for packet capture. However, getting relevant packet captures using these tools is often times cumbersome especially when working with high volume traffic scenarios. Filtering capabilities provided by a VPN gateway packet capture becomes a major differentiator. You may use a VPN gateway packet capture in addition to commonly available packet capture tools.

VPN gateway packet capture filtering capabilities

VPN gateway packet captures can be run on the gateway or on a specific connection depending on customer needs. You can also run packet captures on multiple tunnels at the same time. You can capture single or bi-direction traffic, IKE and ESP traffic, and inner packets along with filtering on a VPN gateway.

Using 5 tuples filter (source subnet, destination subnet, source port, destination port, protocol) and TCP flags (SYN, ACK, FIN, URG, PSH, RST) is helpful when isolating issues on a high volume traffic.

You can use only one option per property while running the packet capture.

Setup packet capture using PowerShell

See the examples below for PowerShell commands to start and stop packet captures. For more information on parameter options (such as how to create filter), see this [PowerShell document](#).

Start packet capture for a VPN gateway

```
Start-AzVirtualnetworkGatewayPacketCapture -ResourceGroupName "YourResourceGroupName" -Name "YourVPNGatewayName"
```

Optional parameter **-FilterData** can be used to apply filter.

Stop packet capture for a VPN gateway

```
Stop-AzVirtualNetworkGatewayPacketCapture -ResourceGroupName "YourResourceGroupName" -Name "YourVPNGatewayName" -SasUrl "YourSASURL"
```

Start packet capture for a VPN gateway connection

```
Start-AzVirtualNetworkGatewayConnectionPacketCapture -ResourceGroupName "YourResourceGroupName" -Name "YourVPNGatewayConnectionName"
```

Optional parameter **-FilterData** can be used to apply filter.

Stop packet capture on a VPN gateway connection

```
Stop-AzVirtualNetworkGatewayConnectionPacketCapture -ResourceGroupName "YourResourceGroupName" -Name "YourVPNGatewayConnectionName" -SasUrl "YourSASURL"
```

Key considerations

- Running packet captures may affect performance. Remember to stop the packet capture when it is not needed.
- Suggested minimum packet capture duration is 600 seconds. Having shorter packet capture duration may not provide complete data due to sync up issues among multiple components on the path.
- Packet capture data files are generated in PCAP format. Use Wireshark or other commonly available applications to open PCAP files.

Next steps

For more information about VPN Gateway, see [About VPN Gateway](#)

Troubleshoot VPN Gateway

1/9/2020 • 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

1/10/2020 • 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

1/10/2020 • 8 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).

NOTE

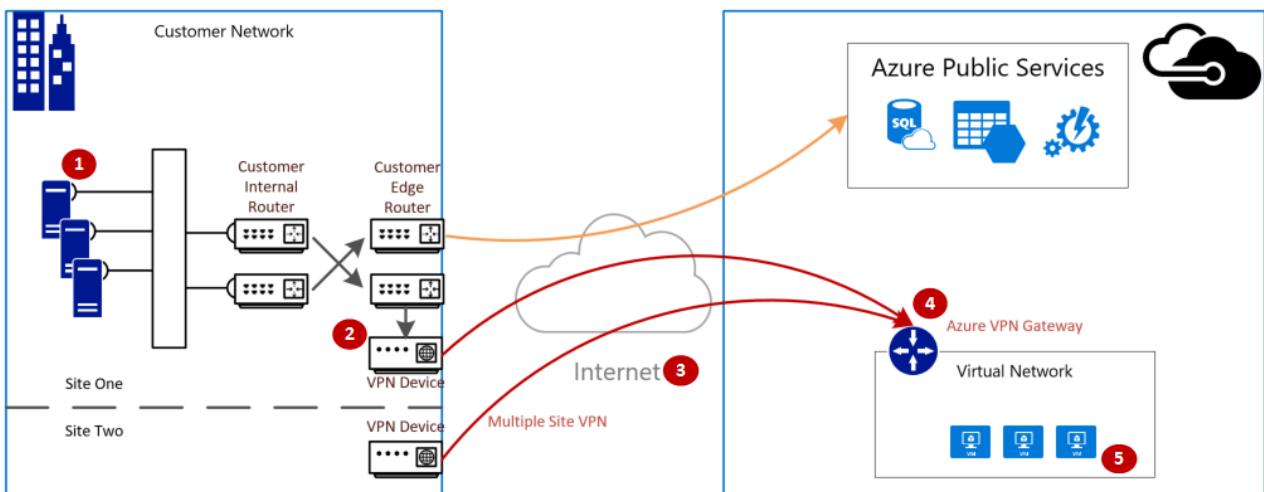
This article is intended to help diagnose and fix common issues. If you're unable to solve the issue by using the following information, [contact support](#).

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 "Gateway SKUs" section of [About 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 by taking the least bandwidth of either the VM, VPN Gateway, or ISP; which is measured in Megabits-per-second (/) divided by eight (8).

If your calculated throughput does not meet your application's baseline throughput requirements, you must 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 the expected Internet bandwidth, you may also contact your ISP.

NOTE

VPN Gateway throughput is an aggregate of all Site-to-Site\VNET-to-VNET, or Point-to-Site connections.

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 3Gbps 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 node acts as the server, and the other node acts as a client. Once this test is completed, we recommend that you reverse the roles of the nodes to test both upload and download throughput on both nodes.

Download iPerf

Download [iPerf](#). For details, see [iPerf documentation](#).

NOTE

The third-party products discussed in this article 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
```

NOTE

Port 5001 is customizable to account for particular firewall restrictions in your environment.

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 directing thirty seconds of traffic on port 5001, to the server. The flag '-P' indicates that we are making 32 simultaneous connections to the server node.

The following screen shows the output from this example:

ID	Interval	Transfer	Bandwidth
[4]	0.00-1.00	sec 15.9 MBbytes	116 Mbits/sec
[4]	1.00-2.00	sec 17.1 MBbytes	128 Mbits/sec
[4]	2.00-3.00	sec 16.4 MBbytes	122 Mbits/sec
[4]	3.00-4.00	sec 17.6 MBbytes	131 Mbits/sec
[4]	4.00-5.00	sec 15.5 MBbytes	114 Mbits/sec
[4]	5.00-6.00	sec 14.8 MBbytes	107 Mbits/sec
[4]	6.00-7.00	sec 17.2 MBbytes	129 Mbits/sec
[4]	7.00-8.00	sec 17.6 MBbytes	132 Mbits/sec
[4]	8.00-9.00	sec 17.5 MBbytes	131 Mbits/sec
[4]	9.00-10.00	sec 17.4 MBbytes	130 Mbits/sec
[4]	0.00-10.00	sec 167 MBbytes	124 Mbits/sec
[4]	0.00-10.00	sec 167 MBbytes	124 Mbits/sec

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 node, and vice-versa.

NOTE

Iperf is not the only tool. [NTTCP](#) is an alternative solution for testing.

Test VMs running Windows

Load Latte.exe onto the VMs

Download the latest version of [Latte.exe](#)

Consider putting Latte.exe in separate folder, such as `c:\tools`

Allow Latte.exe through the Windows firewall

On the receiver, create an Allow rule on the Windows Firewall to allow the Latte.exe traffic to arrive. It's easiest to allow the entire Latte.exe program by name rather than to allow specific TCP ports inbound.

Allow Latte.exe through the Windows Firewall like this

```
netsh advfirewall firewall add rule program=<PATH>\latte.exe name="Latte" protocol=any dir=in action=allow enable=yes profile=ANY
```

For example, if you copied latte.exe to the "c\tools" folder, this would be the command

```
netsh advfirewall firewall add rule program=c:\tools\latte.exe name="Latte" protocol=any dir=in action=allow enable=yes profile=ANY
```

Run latency tests

Start latte.exe on the RECEIVER (run from CMD, not from PowerShell):

```
latte -a <Receiver IP address>:<port> -i <iterations>
```

Around 65k iterations is long enough to return representative results.

Any available port number is fine.

If the VM has an IP address of 10.0.0.4, it would look like this

```
latte -c -a 10.0.0.4:5005 -i 65100
```

Start latte.exe on the SENDER (run from CMD, not from PowerShell)

```
latte -c -a <Receiver IP address>:<port> -i <iterations>
```

The resulting command is the same as on the receiver except with the addition of "-c" to indicate that this is the "client" or sender

```
latte -c -a 10.0.0.4:5005 -i 65100
```

Wait for the results. Depending on how far apart the VMs are, it could take a few minutes to complete. Consider starting with fewer iterations to test for success before running longer tests.

Test VMs running Linux

Use [SockPerf](#) to test VMs.

Install SockPerf on the VMs

On the Linux VMs (both SENDER and RECEIVER), run these commands to prepare SockPerf on your VMs:

CentOS / RHEL - Install GIT and other helpful tools

```
sudo yum install gcc -y -q | sudo yum install git -y -q | sudo yum install gcc-c++ -y  
sudo yum install ncurses-devel -y | sudo yum install -y automake
```

Ubuntu - Install GIT and other helpful tools

```
sudo apt-get install build-essential -y | sudo apt-get install git -y -q | sudo apt-get install -y autotools-dev  
sudo apt-get install -y automake
```

Bash - all

From bash command line (assumes git is installed)

```
git clone https://github.com/mellanox/sockperf | cd sockperf/ | ./autogen.sh | ./configure --prefix=
```

Make is slower, may take several minutes

```
make
```

Make install is fast

```
sudo make install
```

Run SockPerf on the VMs

Sample commands after installation. Server/Receiver - assumes server's IP is 10.0.0.4

```
sudo sockperf sr --tcp -i 10.0.0.4 -p 12345 --full-rtt
```

Client - assumes server's IP is 10.0.0.4

```
sockperf ping-pong -i 10.0.0.4 --tcp -m 1400 -t 101 -p 12345 --full-rtt
```

NOTE

Make sure there are no intermediate hops (e.g. Virtual Appliance) during the throughput testing in between the VM and Gateway. If there are poor results (in terms of overall throughput) coming from the iPERF/NTTCP tests above, please refer to the following article to understand the key factors behind the possible root causes of the problem:

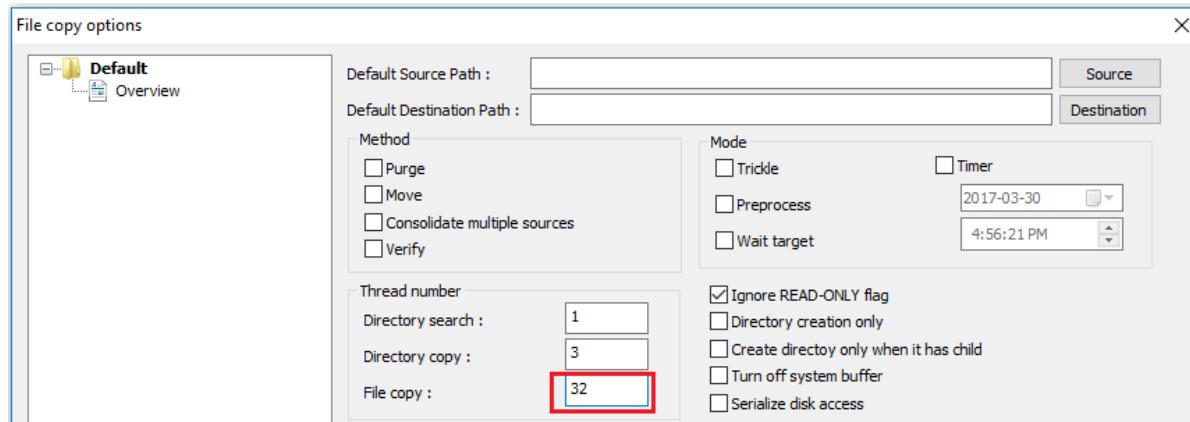
<https://docs.microsoft.com/azure/virtual-network/virtual-network-tcpip-performance-tuning>

In particular, analysis of packet capture traces (Wireshark/Network Monitor) collected in parallel from client and server during those tests will help in the assessments of bad performance. These traces can include packet loss, high latency, MTU size, fragmentation, TCP 0 Window, Out of Order fragments, and so on.

Address slow file copy issues

Even if the overall throughput assessed with the previous steps (iPERF/NTTCP/etc..) was good, you may experience slow file copying when either 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**.



NOTE

Not all application work same, and not all application/process utilizes all the threads. If you run the test, you could see some threads being empty and won't provide accurate throughput results. To check your application file transfer performance, use multi-thread by increasing the # of thread in succession or decrease in order to find the optimal throughput of the application or file transfer.

- Insufficient VM disk read/write speed. For more information, see [Azure Storage Troubleshooting](#).

On-premises device external facing interface

Mentioned the subnets of on-premises ranges that you would like Azure to reach via VPN on Local Network Gateway. Simultaneously, define the VNET address space in Azure to the on-premises device.

- **Route Based Gateway:** The policy or traffic selector for route-based VPNs are configured as any-to-any (or wild cards).
- **Policy Based Gateway:** 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.
- **UsePolicyBasedTrafficSelector** connections: ("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 and from the Azure virtual network prefixes, instead of any-to-any.

Inappropriate configuration may lead to frequent disconnects within the tunnel, packet drops, bad throughput, and latency.

Check latency

You can check latency by using the following tools:

- WinMTR
- TCPTraceroute
- `ping` and `psping` (These tools can provide a good estimate of RTT, but they can't be used in all cases.)

```
C:\WINDOWS\system32\cmd.exe
C:\Users\chadi>tracert 13.72.98.5

Tracing route to 13.72.98.5 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.
```

If you notice a high latency spike at any of the hops before entering MS Network backbone, you may want to proceed with further investigations with your Internet Service Provider.

If a large, unusual latency spike is noticed from hops within "msn.net", please contact MS support for further investigations.

Next steps

For more information or help, check out the following link:

- [Microsoft Support](#)

Troubleshooting: Azure point-to-site connection problems

1/10/2020 • 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
AzureRoot.cer	Local Computer\Trusted Root Certification Authorities

3. Go to C:\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.

The network connection between your computer and the VPN server could not be established because the remote server is not responding

Symptom

When you try and connect to an Azure virtual network gateway using IKEv2 on Windows, you get the following error message:

The network connection between your computer and the VPN server could not be established because the remote server is not responding

Cause

The problem occurs if the version of Windows does not have support for IKE fragmentation

Solution

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

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-Task > 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+gAwIBAgIQFSwLuUrCIdHwI3hzJbdBjANBgkqhkiG9w0BAQsFADAW
MRQwEgYDVQQDDAtQM1NSb290Q2VydDAeFw0xNzA2MTUwMjU4NDZaFw0xODA2MTUw
MzE4NDZaMBYxFDASBgNVBAMMC1AyU1Jvb3RDZXJ0MIIBIjANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEaZ8QUCWxxxxTrxF5yc5uUpL/bzwC5zZ8041tB1NpPa/PI
sa5uwLw/YFb8XG/JCWxUJpUzS/kHUKF1uqkY80U+fAmRmTEMq5wcaMhp3wRfq+1
G90PBNTyqpnHe+154QAnj1DjsHXXNL4AL1N8/TSzYTm7dkiq+EAiYRRMrZ1Ywije
407ChxIp0stB84MtMShyoSm2hg1+3zfwuAGXoJQwWiXh715kMHVTSj9zFechYd7
50LltoRRDyyxsf0qweTFK1gFj13Hn/bq/UJG3AcyQnv1Cv1HwQnXo+hckVBB29wE
sF8QSYk2MMGimPDYYt4ZM5tmYLxxxvGmrGhc+HwXzMeQIDAQABozEwLzAOBgNVHQ8B
AF8EBAMCAgQwHQYDVRO0BBYEFBEl9zZWhQftVLBQNATC/LHLvMb0OMA0GCSqGSIb3
DQEBCwUAQIBAQB7k0ySFUQu72sfj3BdNxrxSyOT4L2rADLhxxxik0U6gHUF6eWz
/0h6y4mNkg3NgLT3j/WclqzHXZruhWAXSF+VbAGkwKA99xGWOCUJ+vKVYL/kDja
gaZrxH1hTYVvMwn4F7DWhteFqhzZ89/w9Mv6p180AimF96qdU8Ez8t860HQAfkU6
2Nw9ZMsGkvLepZZ178yVBDCWMogBMhrRVXG/xQkBajgvL5syLwFB02kWGdC+wylWY
U/Z+EK9UuHnn3Hkq/vxEzRVsYuaxchta0X2UNRzRq+o7061+iyLTpe6fnvW6i1oi
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, re-download and redeploy the Point to Site package on all clients.

Too many VPN clients connected at once

The maximum number of allowable connections is reached. You can see the total number of connected clients in the Azure portal.

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\Roaming\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](#). To make sure that the new routes are being used, the Point-to-Site VPN clients must be downloaded again after virtual network peering has been successfully configured.

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\Roaming\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

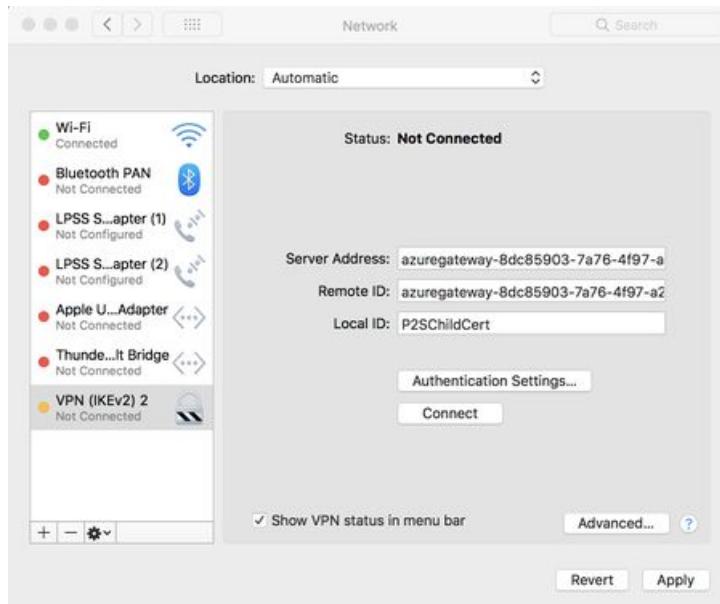
2/17/2020 • 2 minutes to read • [Edit Online](#)

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:

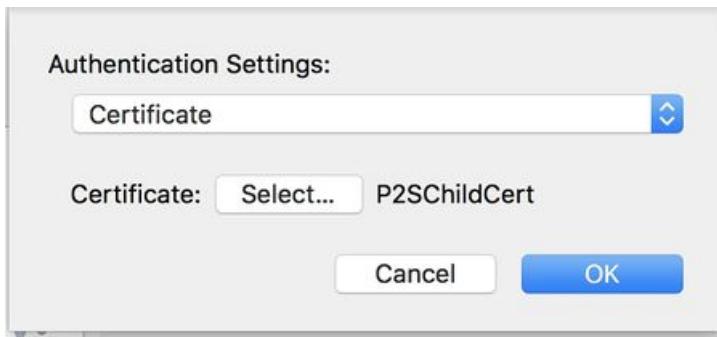
- Server Address
- Remote ID
- Local ID
- Authentication Settings
- OS Version (10.11 or higher)

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.



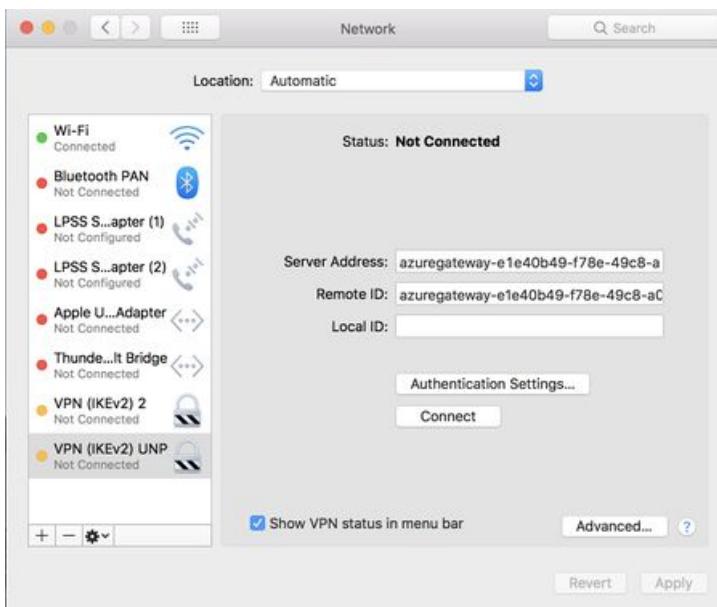
2. Verify that the **Server Address** is the complete FQDN and includes the clouddapp.net.
3. The **Remote ID** should be the same as the Server Address (Gateway FQDN).
4. The **Local ID** should be the same as the **Subject** of the client certificate.
5. Click on **Authentication Settings** to open the Authentication Settings page.



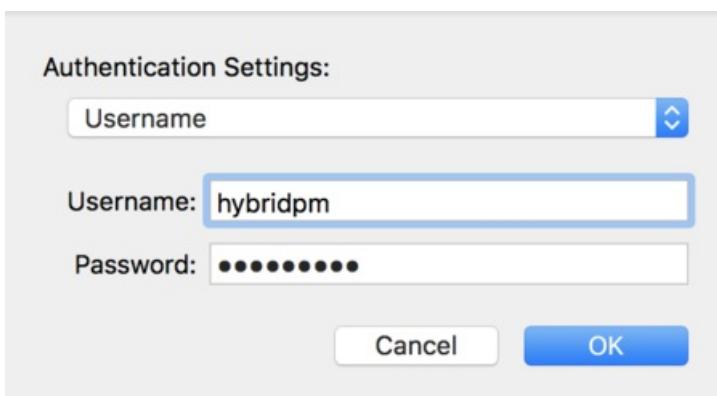
6. Verify that **Certificate** is selected from the dropdown.
7. Click the **Select** button and verify that the correct certificate is selected. Click **OK** to save any changes.

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 clouddapp.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.

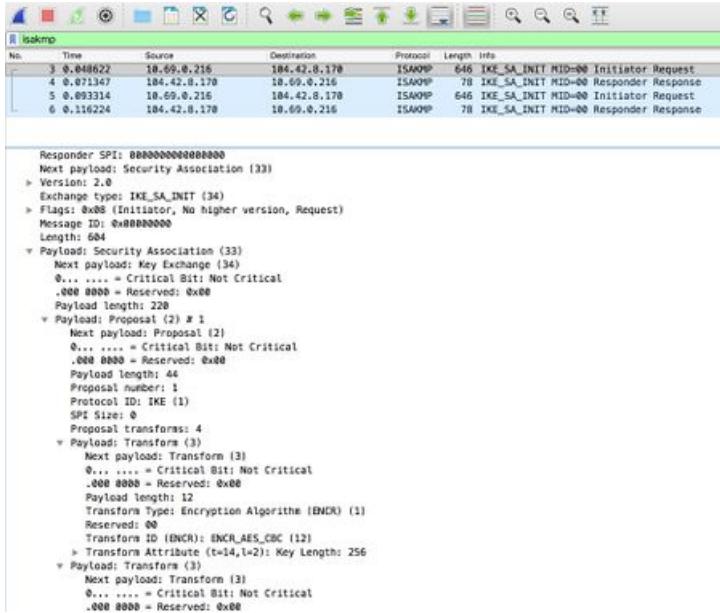


6. Verify that the correct credentials are entered.

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.



3. If there is no server response on the network traces, verify you enabled IKEv2 protocol on the Azure Gateway Configuration page on the Azure portal website.

Next steps

For additional help, see [Microsoft Support](#).

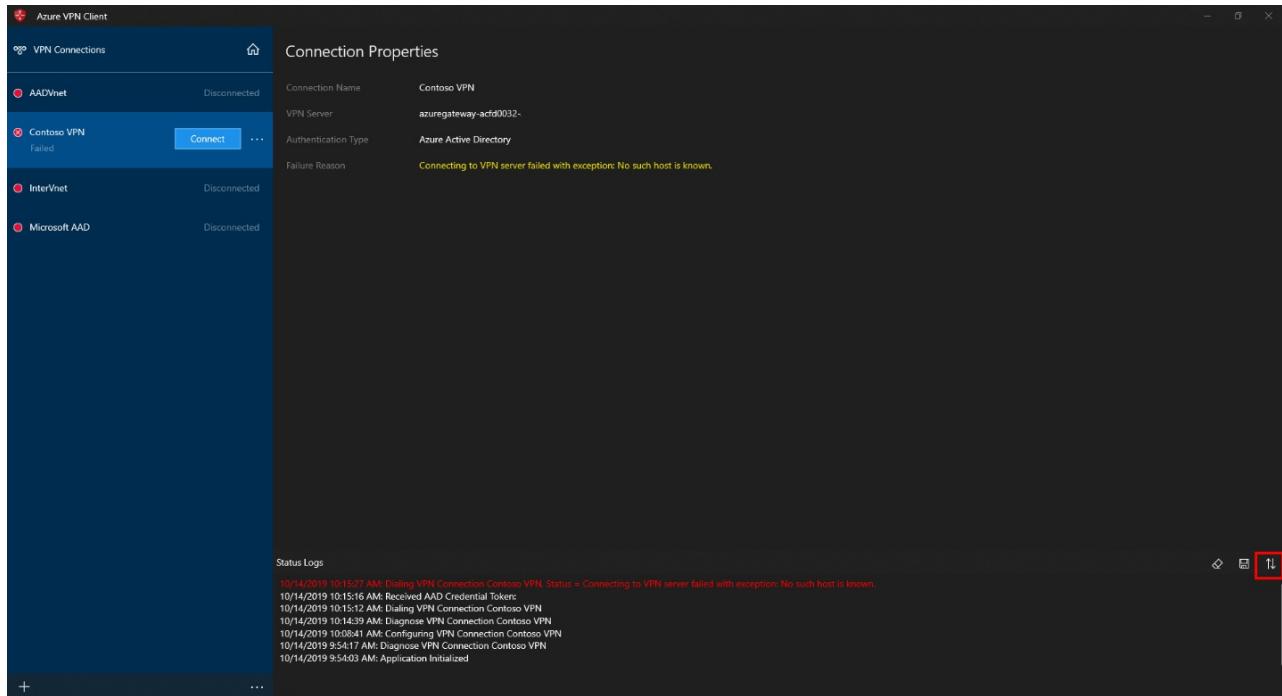
Troubleshoot an Azure AD authentication VPN client

11/17/2019 • 2 minutes to read • [Edit Online](#)

This article helps you troubleshoot a VPN client to connect to a virtual network using Point-to-Site VPN and Azure Active Directory authentication.

View Status Log

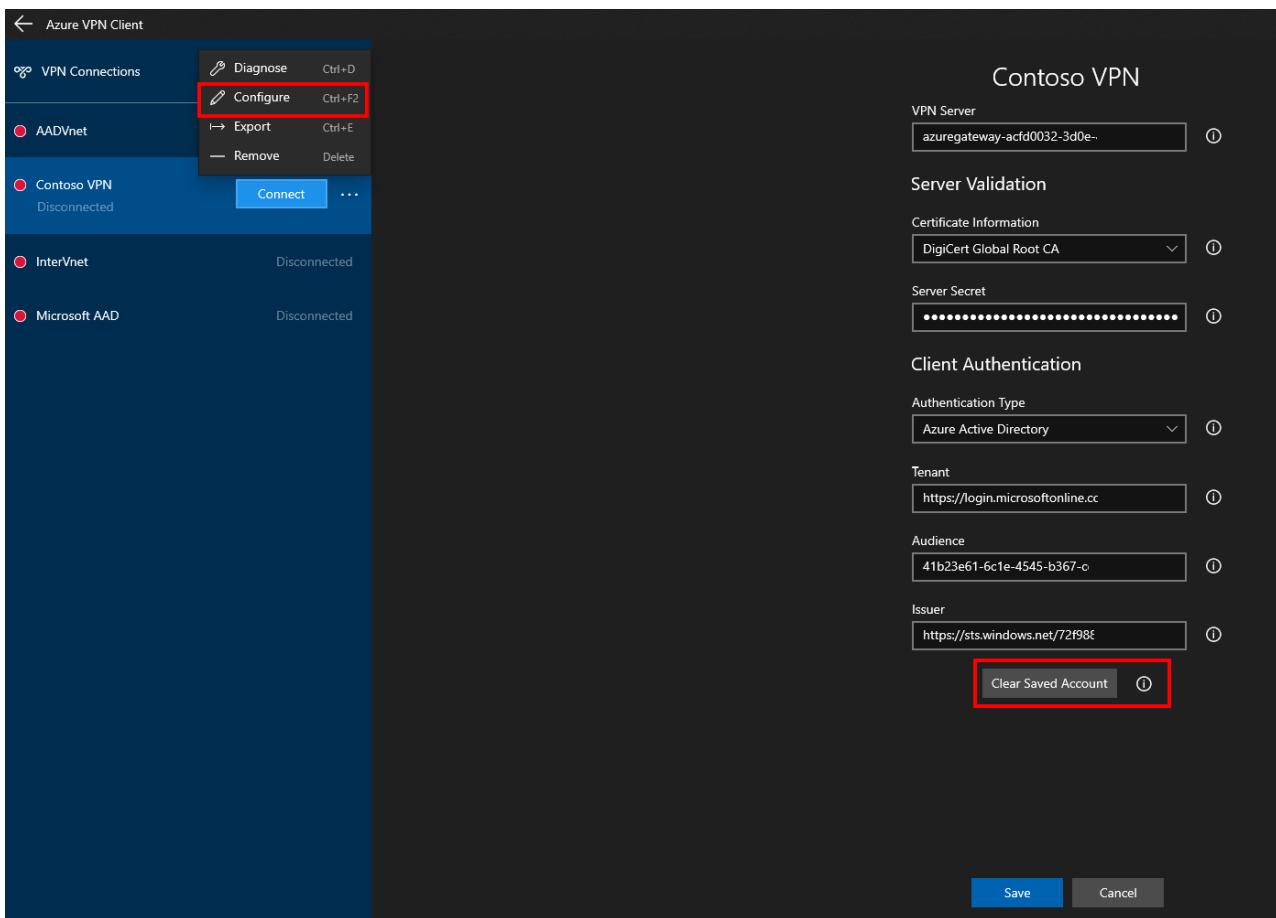
View the status log for error messages.



1. Click the arrows icon at the bottom-right corner of the client window to show the **Status Logs**.
2. Check the logs for errors that may indicate the problem.
3. Error messages are displayed in red.

Clear sign-in information

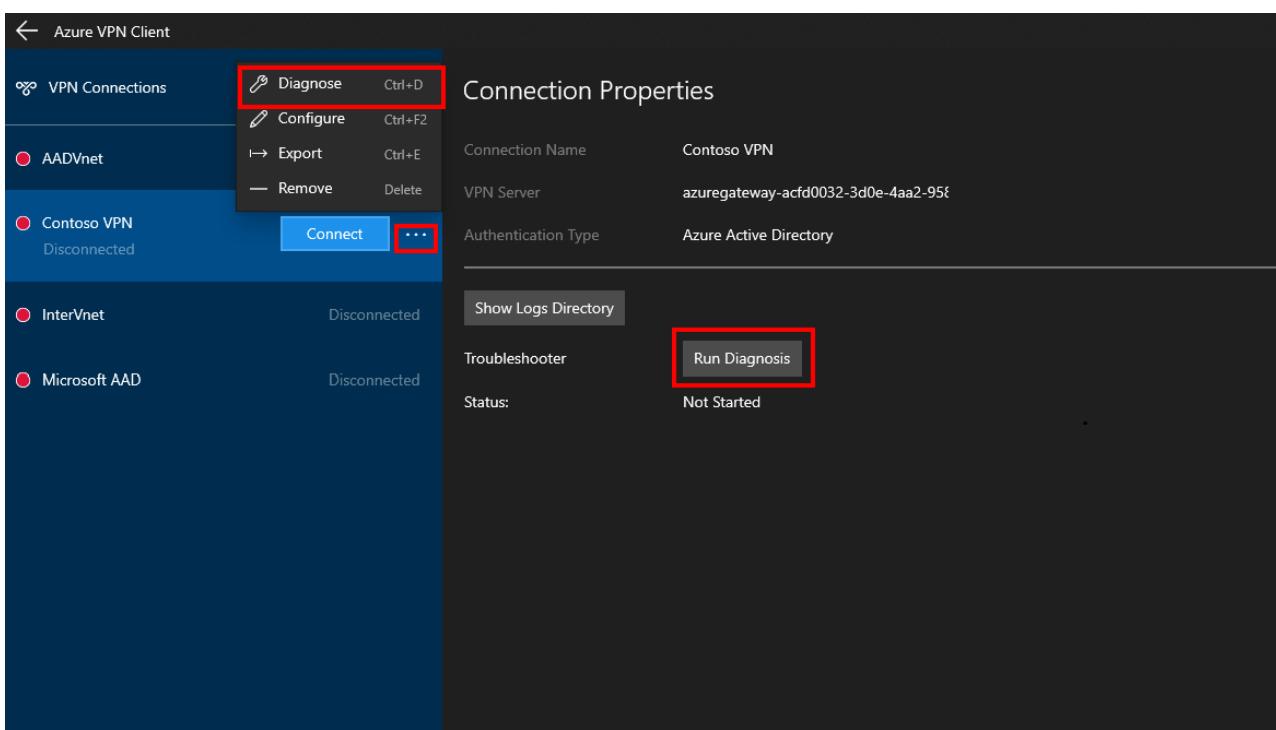
Clear the sign-in information.



1. Select the ... next to the profile that you want to troubleshoot. Select **Configure** -> **Clear Saved Account**.
2. Select **Save**.
3. Try to connect.
4. If the connection still fails, continue to the next section.

Run diagnostics

Run diagnostics on the VPN client.



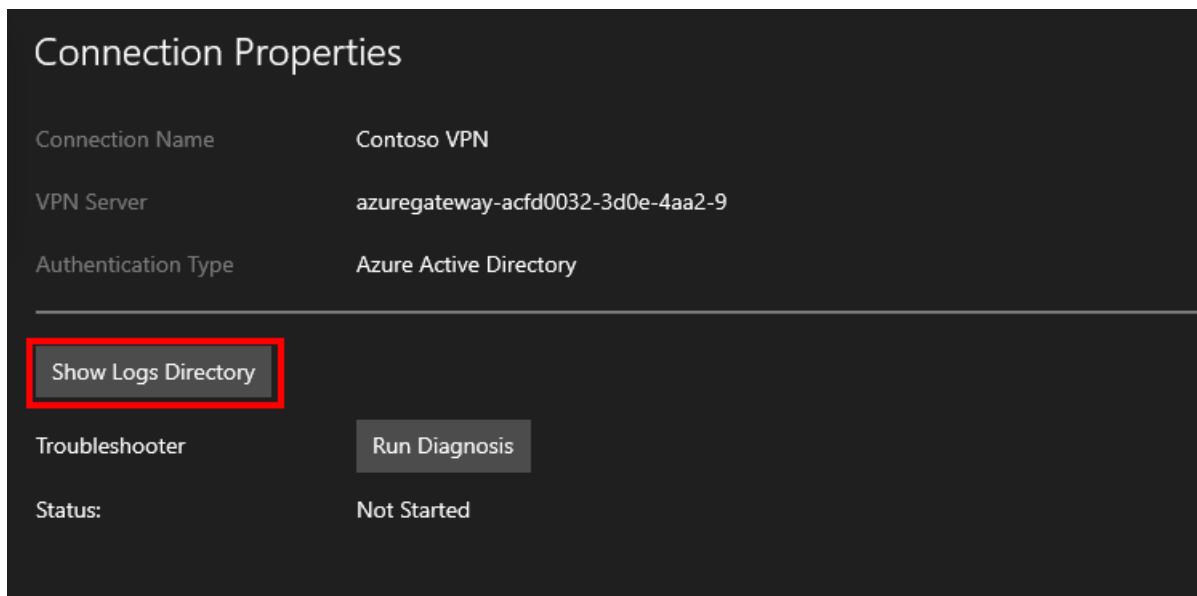
1. Click the ... next to the profile that you want to run diagnostics on. Select **Diagnose** -> **Run Diagnosis**.

2. The client will run a series of tests and display the result of the test
 - Internet Access – Checks to see if the client has Internet connectivity
 - Client Credentials – Check to see if the Azure Active Directory authentication endpoint is reachable
 - Server Resolvable – Contacts the DNS server to resolve the IP address of the configured VPN server
 - Server Reachable – Checks to see if the VPN server is responding or not
3. If any of the tests fail, contact your network administrator to resolve the issue.
4. The next section shows you how to collect the logs, if needed.

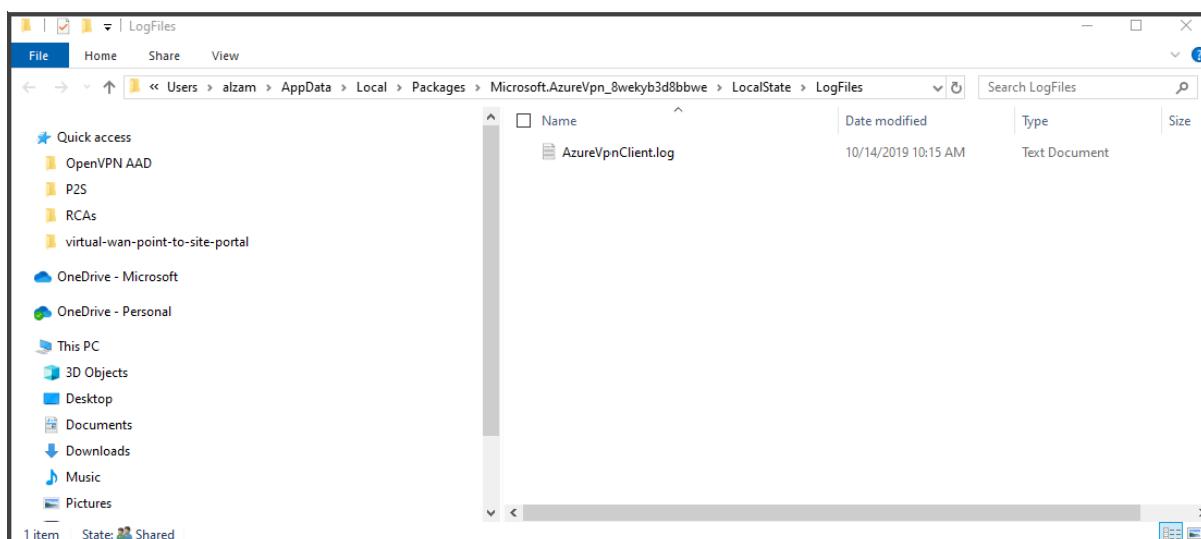
Collect client log files

Collect the VPN client log files. The log files can be sent to support/administrator via a method of your choosing. For example, e-mail.

1. Click the “...” next to the profile that you want to run diagnostics on. Select **Diagnose -> Show Logs Directory**.



2. Windows Explorer opens to the folder that contains the log files.



Next steps

For more information, see [Create an Azure Active Directory tenant for P2S Open VPN connections that use Azure AD authentication](#).

Troubleshooting: An Azure site-to-site VPN connection cannot connect and stops working

1/10/2020 • 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

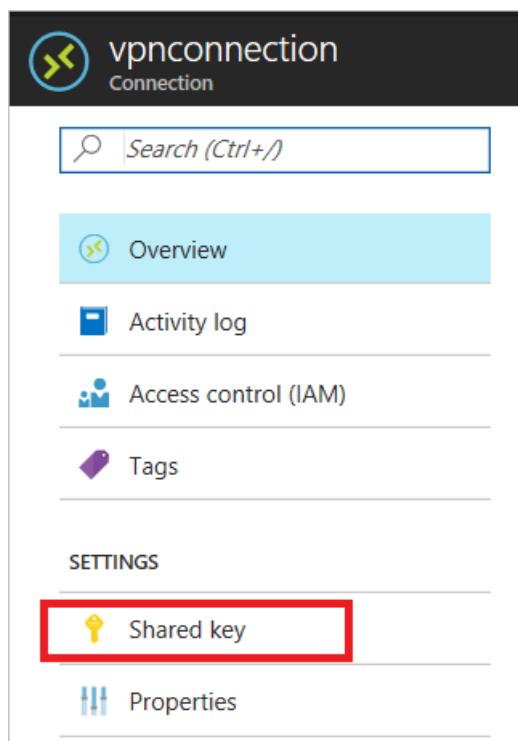
Compare the shared key for the on-premises VPN device to the Azure Virtual Network VPN to make sure that the

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

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

For the Azure Resource Manager deployment model:

```
Get-AzVirtualNetworkGatewayConnectionSharedKey -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"?>
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">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

1/10/2020 • 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.

The screenshot shows the Azure portal interface for a virtual network gateway named 'Gateway01'. On the left, there's a navigation menu with options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', and 'Diagnose and solve problems'. The 'Overview' tab is selected. On the right, under the 'Essentials' section, there are several configuration details. The 'VPN type' field is explicitly highlighted with a red rectangular box. Other visible details include the resource group ('Thomas'), SKU ('VpnGw1'), location ('East Asia'), and subscription information ('<Subscription name>', '<ID>').

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

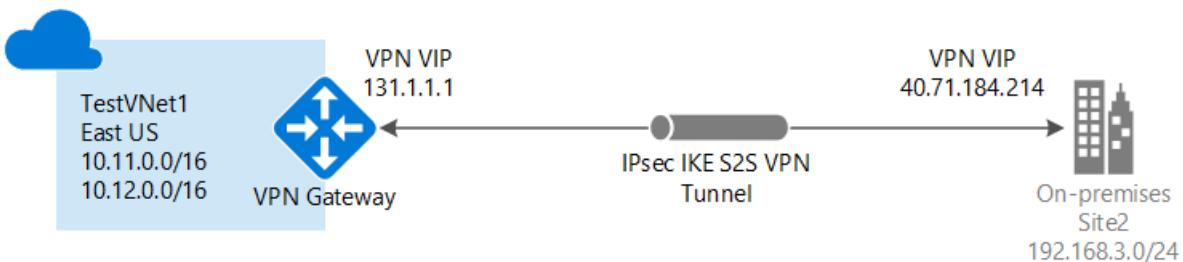
- [Configure a Site-to-Site connection to a virtual network](#)
- [Configure IPsec/IKE policy for Site-to-Site VPN connections](#)

Create a Site-to-Site connection using the Azure portal (classic)

2/13/2020 • 13 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 classic deployment model and do not apply to the current deployment model, Resource Manager. 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.
- 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.
- PowerShell is required in order to specify the shared key and create the VPN gateway connection. When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

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 **+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. click **(change to 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 create the VNet, you can go back and add additional subnets and address spaces.

Create virtual network (cl... □ 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
Content Development ▾

* Resource Group
TestRG1 ▾
[Create new](#)

* Location
East US ▾

Create Automation options

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. 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 network 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 network 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. On the page for your VNet, under **Settings**, click **Diagram**.
2. On the **VPN connections** page, click **You don't have any existing VPN connections. Click here to get started**.
3. For **Connection type**, leave **Site-to-site** selected.
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 must 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.

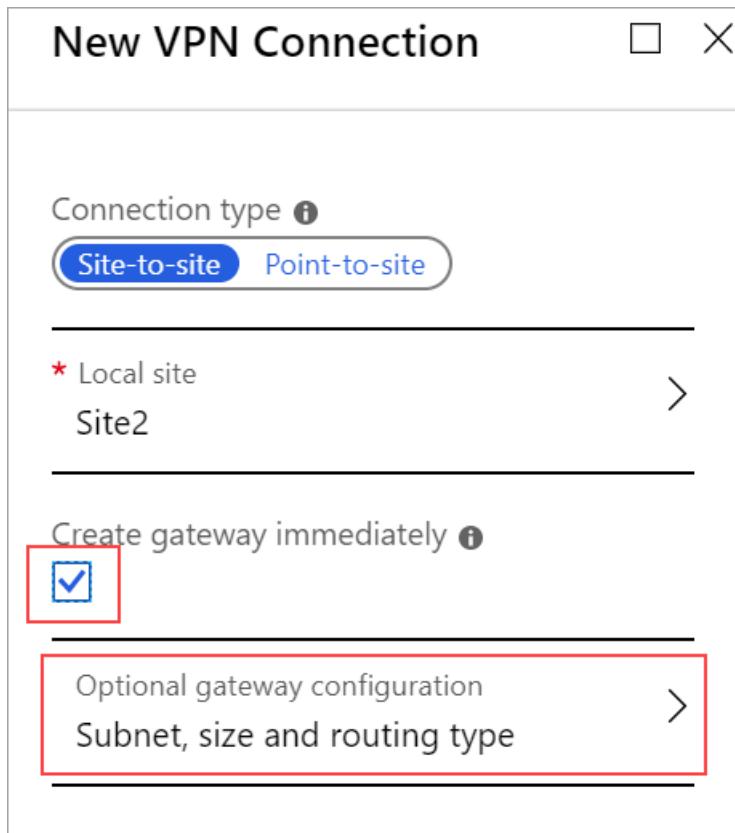
New VPN Connection	X
Connection type ⓘ <input checked="" type="radio"/> Site-to-site <input type="radio"/> Point-to-site	
* Local site Configure required settings >	
Create gateway immediately ⓘ <input type="checkbox"/>	
Local site	
* Name Site2 ✓	
* VPN gateway IP address 4.3.2.1 ✓	
CLIENT ADDRESS SPACE	
192.168.0.0/16 ...	
0.0.0.0/0 ...	
OK	OK

Click **OK** to close the Local site page. **Do not click OK to close the New VPN Connection page.**

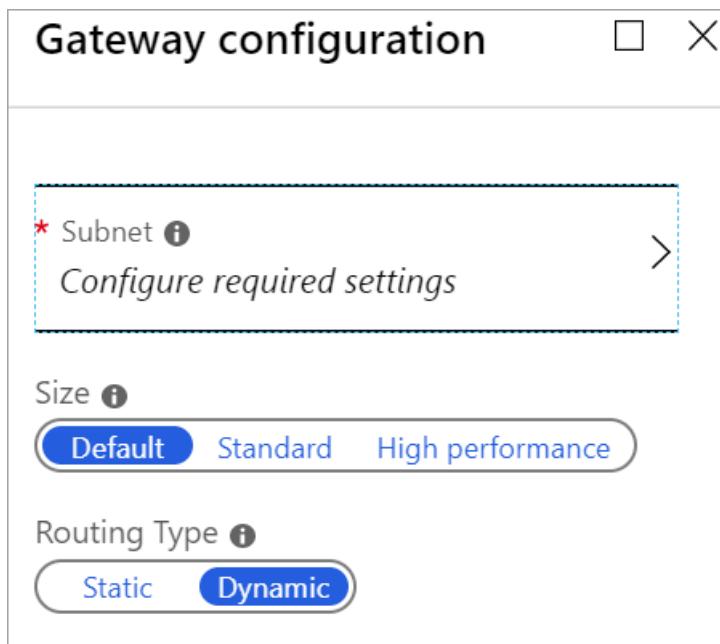
5. Configure the gateway subnet

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. When you are finished configuring these settings, click **OK**.



4. 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.

Add subnet

★ Name	GatewaySubnet
★ Address range (CIDR block) <small>i</small>	10.11.255.0/27 ✓
10.11.255.0-10.11.255.31 (27 addresses)	

6. Specify the SKU and VPN type

1. Select the gateway **Size**. This is the gateway SKU that you use to create your virtual network gateway. Classic VPN gateways use the old (legacy) gateway SKUs. For more information about the legacy gateway SKUs, see [Working with virtual network gateway SKUs \(old SKUs\)](#).

New VPN Connection	Gateway configuration
<p>Connection type <small>i</small></p> <p>Site-to-site Point-to-site</p> <hr/> <p>* Local site</p> <p>Site2 ></p> <hr/> <p>Create gateway immediately <small>i</small></p> <p><input checked="" type="checkbox"/></p> <hr/> <p>Optional gateway configuration</p> <p>Subnet, size and routing type ></p>	<p>★ Subnet <small>i</small></p> <p>10.11.255.0/27 ></p> <hr/> <p>Size <small>i</small></p> <p>Standard Default High performance</p> <hr/> <p>Routing Type <small>i</small></p> <p>Dynamic Static</p>

2. Select the **Routing Type** for your gateway. This is also known as the VPN type. It's important to select the correct 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 Routing 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 deploying 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 [VPN device configuration overview](#).
- 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. See [Before you Begin](#) for information about installing these cmdlets.

Step 1. Connect to your Azure account

You must run these commands locally using the PowerShell service management module.

1. Open your PowerShell console with elevated rights. To switch to service management, use this command:

```
azure config mode asm
```

2. Connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

3. Check the subscriptions for the account.

```
Get-AzureSubscription
```

4. 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 you create a classic VNet in the portal (not using PowerShell), Azure adds the resource group name to the short name. For example, according to Azure, the name of the VNet that you created for this exercise is "Group TestRG1 TestVNet1", not "TestVNet1". PowerShell requires the full name of the virtual network, not the short name that appears in the portal. The long name is not visible in the portal. The following steps help you export the network configuration file to obtain the exact values for the virtual network name.

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 for this exercise to reflect the values in the xml. 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.

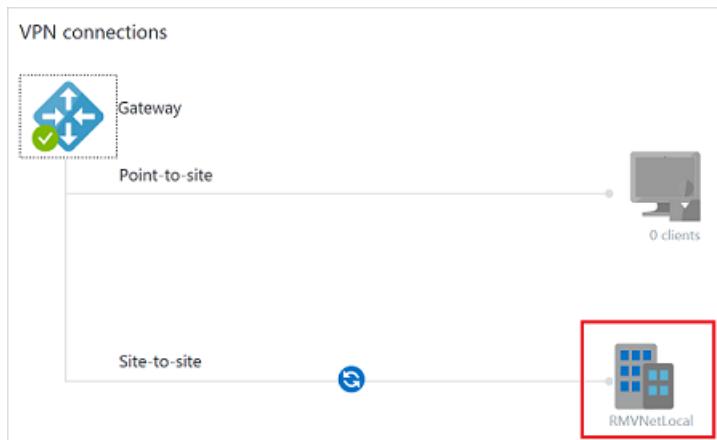
```
Set-AzureVNetGatewayKey -VNetName 'Group TestRG1 TestVNet1' `  
-LocalNetworkSiteName 'D1BFC9CB_Site2' -SharedKey abc123
```

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 by using certificate authentication (classic)

1/10/2020 • 23 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](#).

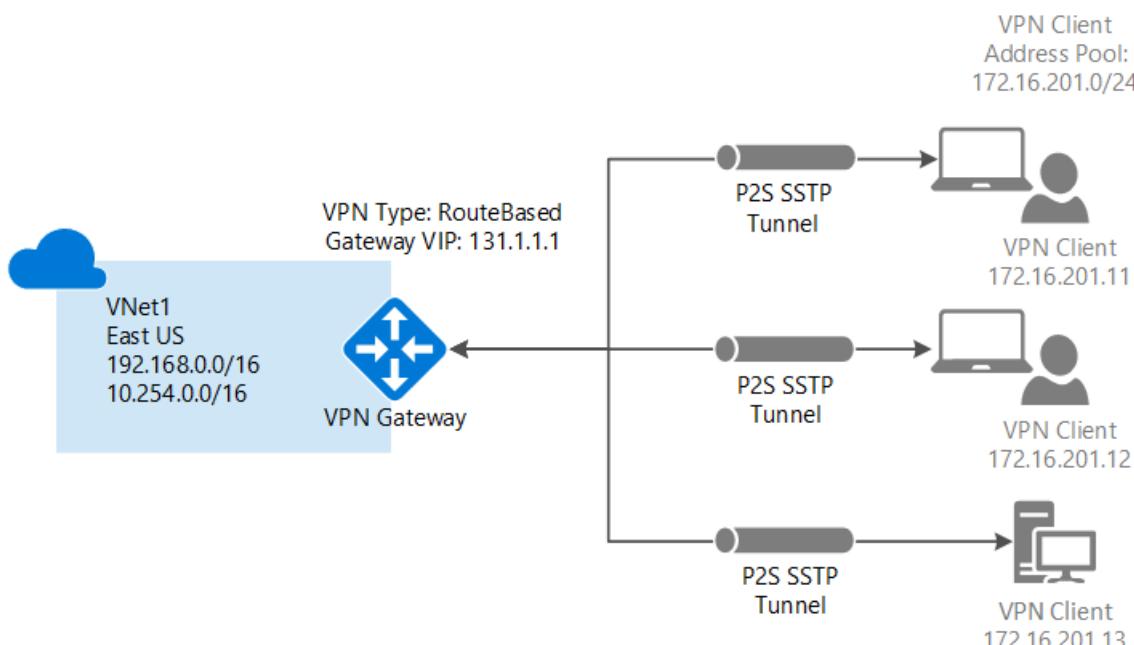
For the Resource Manager version of this article, select it from the drop-down list below, or from the table of contents on the left.

This article shows you how to create a VNet with a Point-to-Site connection. You create this Vnet with the classic deployment model by 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 with a different deployment tool or model by using options that are described in the following articles:

You use a Point-to-Site (P2S) VPN gateway to 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. When you have only a few clients that need to connect to a VNet, a P2S VPN is a useful solution to use instead of a Site-to-Site VPN. 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. To support non-Windows VPN clients, you must create your VNet with the Resource Manager deployment model. The Resource Manager deployment model supports IKEv2 VPN in addition to SSTP. For more information, see [About P2S connections](#).



Prerequisites

Point-to-Site certificate authentication connections require the following prerequisites:

- A Dynamic VPN gateway.
- The public key (.cer file) for a root certificate, which is uploaded to Azure. This key 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's already on the operating system with the necessary information to connect to the VNet.

Point-to-Site connections don't 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 [Point-to-Site FAQ](#).

Example settings

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

- **Create virtual network (classic) settings**

- **Name:** Enter *VNet1*.
 - **Address space:** Enter *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:** Enter *FrontEnd*.
 - **Subnet address range:** Enter *192.168.1.0/24*.
 - **Subscription:** Select a subscription from the list of available subscriptions.
 - **Resource group:** Enter *TestRG*. Select **Create new**, if the resource group doesn't exist.
 - **Location:** Select **East US** from the list.
- **VPN connection settings**
 - **Connection type:** Select **Point-to-site**.
 - **Client Address Space:** Enter *172.16.201.0/24*. VPN clients that connect to the VNet by using this Point-to-Site connection receive an IP address from the specified pool.

- **Gateway configuration subnet settings**

- **Name:** Autofilled with *GatewaySubnet*.
 - **Address range:** Enter *192.168.200.0/24*.

- **Gateway configuration settings:**

- **Size:** Select the gateway SKU that you want to use.
 - **Routing Type:** Select **Dynamic**.

Create a virtual network and a VPN gateway

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](#).

Part 1: Create a virtual network

If you don't already have a virtual network (VNet), 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. On the [Azure portal](#) menu or from the **Home** page, select **Create a resource**. The **New** page opens.
2. In the **Search the marketplace** field, enter *virtual network* and select **Virtual network** from the returned list. The **Virtual network** page opens.
3. From the **Select a deployment model** list, select **Classic**, and then select **Create**. The **Create virtual network** page opens.
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.

The screenshot shows the 'Create virtual network (classic)' dialog box. It includes fields for Name (VNet), Address space (192.168.0.0/16), Subnet name (FrontEnd), Subnet address range (192.168.1.0/24), Subscription (Microsoft Azure), Resource Group (TestRG), and Location (East US). The 'Create' button is highlighted at the bottom left.

5. Select the **Subscription** you want to use from the drop-down list.
6. Select an existing **Resource Group**. Or, create a new resource group by selecting **Create new** and entering a name. If you're creating a new resource group, name the resource group according to your planned configuration values. For more information about resource groups, see [Azure Resource Manager overview](#).
7. Select a **Location** for your VNet. This setting determines the geographical location of the resources that you deploy to this VNet.
8. Select **Create** to create the VNet. From the **Notifications** page, you'll see a **Deployment in progress** message.
9. After your virtual network has been created, the message on the **Notifications** page changes to

Deployment succeeded. Select **Pin to dashboard** if you want to easily find your VNet on the dashboard.

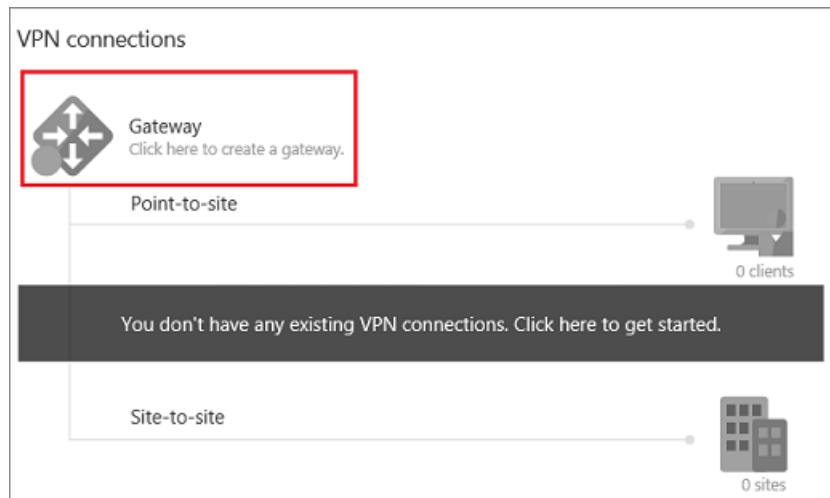
10. Add a DNS server (optional). After you create your virtual network, you can add the IP address of a DNS server for name resolution. The DNS server IP address that you specify should be the address of a DNS server that can resolve the names for the resources in your VNet.

To add a DNS server, select **DNS servers** from your VNet page. Then, enter the IP address of the DNS server that you want to use and select **Save**.

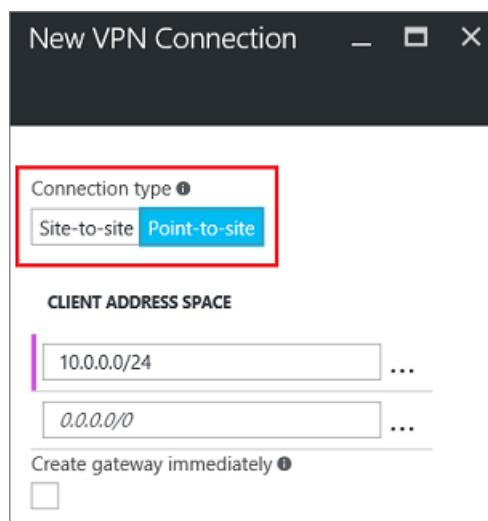
Part 2: Create a gateway subnet and a dynamic routing gateway

In this step, you create a gateway subnet and a dynamic routing gateway. In the Azure portal for the classic deployment model, you create the gateway subnet and the gateway through the same configuration pages. Use the gateway subnet for the gateway services only. Never deploy anything directly to the gateway subnet (such as VMs or other services).

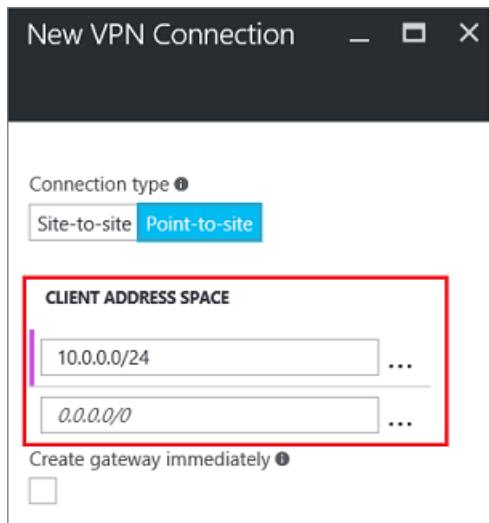
1. In the Azure portal, go to the virtual network for which you want to create a gateway.
2. On the page for your virtual network, select **Overview**, and in the **VPN connections** section, select **Gateway**.



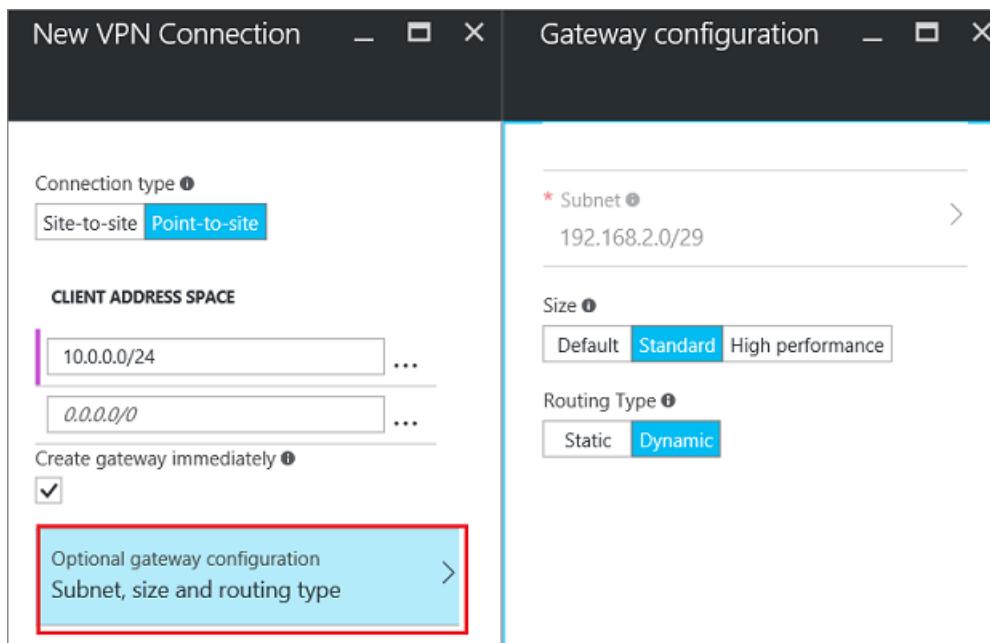
3. On the **New VPN Connection** page, select **Point-to-site**.



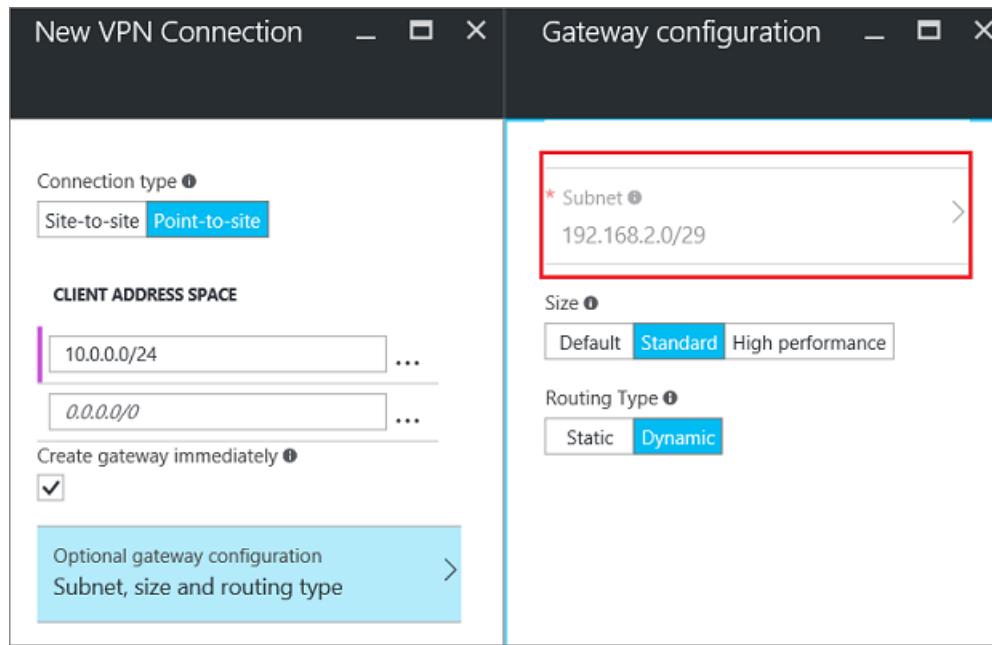
4. For **Client Address Space**, add the IP address range from which the VPN clients receive an IP address when connecting. Use a private IP address range that doesn't overlap with the on-premises location that you connect from, or with the VNet that you connect to. You can overwrite the autofilled range with the private IP address range that you want to use. This example shows the autofilled range.



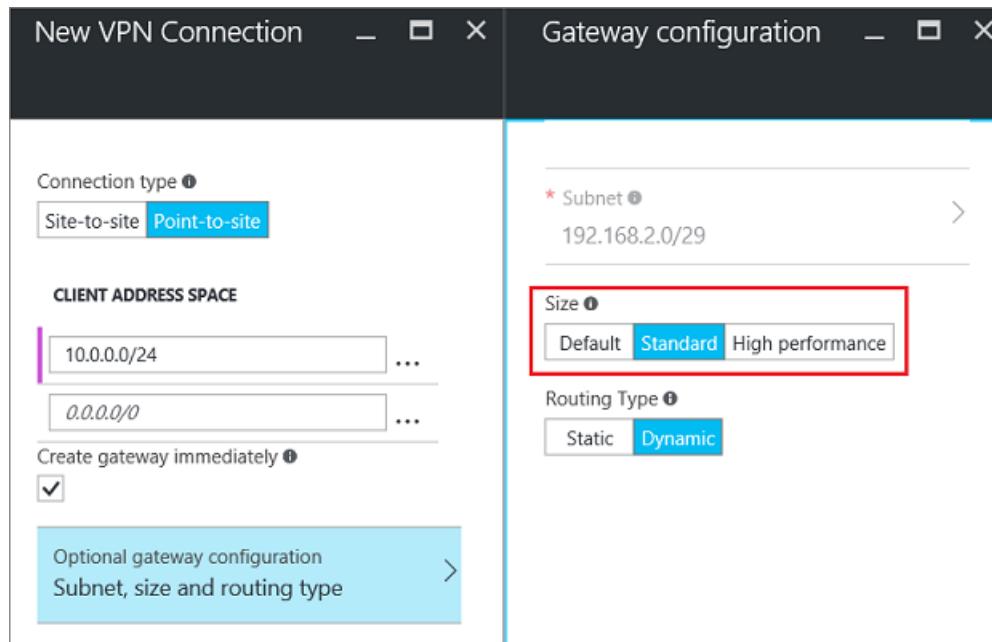
5. Select **Create gateway immediately**, and then select **Optional gateway configuration** to open the **Gateway configuration** page.



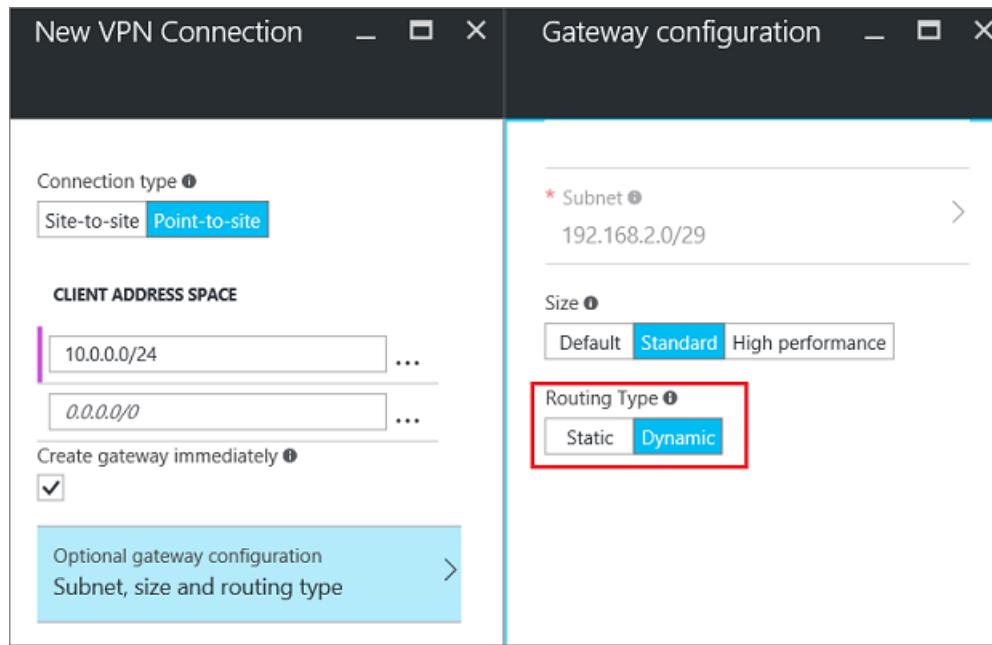
6. From the **Gateway configuration** page, select **Subnet** to add the gateway subnet. It's possible to create a gateway subnet as small as /29. However, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. Doing so will allow for enough addresses 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 not function as expected. Select **OK** to save this setting.



7. Select the gateway **Size**. The size is the gateway SKU for your virtual network gateway. In the Azure portal, the default SKU is **Default**. 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. Select **OK** when you've finished configuring this page.



9. On the **New VPN Connection** page, select **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.

Create certificates

Azure uses certificates 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\Certificates certificate store. The certificate is used to authenticate the client when it connects to the VNet.

If you use self-signed certificates, they must be created by using specific parameters. You can create a self-signed certificate by using the instructions for [PowerShell and Windows 10](#), or [MakeCert](#). It's important to follow the steps in these instructions when you use self-signed root certificates and generate client certificates from the self-signed root certificate. Otherwise, the certificates you create won't be compatible with P2S connections and you'll receive a connection error.

Acquire the public key (.cer) for the root certificate

Use either a root certificate that was generated with an enterprise solution (recommended), or generate a self-signed certificate. After you create the root certificate, export the public certificate data (not the private key) as a Base64 encoded X.509 .cer file. Then, upload the public certificate data to the Azure server.

- **Enterprise certificate:** If you're using an enterprise solution, you can use your existing certificate chain.
Acquire 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, create a self-signed root certificate. Otherwise, the certificates you create won't be compatible with your P2S connections and clients will receive a connection error when they try to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the following articles describe how to generate a compatible self-signed root 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. Although MakeCert is deprecated, you can still use it to generate certificates. Client

certificates that you generate from the root certificate can be installed on any supported P2S client.

- [Linux instructions](#)

Generate a client certificate

Each client computer that you connect to a VNet with a Point-to-Site connection must have a client certificate installed. You generate it from the root certificate and install it on each client computer. If you don't install a valid client certificate, authentication will fail when the client tries to connect to the VNet.

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 use the same client certificate to authenticate and you revoke it, you'll need to generate and install new certificates for every client that uses that certificate.

You can generate client certificates by using the following methods:

- **Enterprise certificate:**

- If you're using an enterprise certificate solution, generate a client certificate with the common name value format *name@yourdomain.com*. Use this format instead of the *domain name\username* format.
- Make sure the client certificate is based on a user certificate template that has *Client Authentication* listed as the first item in the user list. Check the certificate by double-clicking it and viewing **Enhanced Key Usage** in the **Details** tab.

- **Self-signed root certificate:** Follow the steps in one of the following P2S certificate articles so that the client certificates you create will be compatible with your P2S connections. The steps in these articles generate a compatible client certificate:

- [Windows 10 PowerShell instructions](#): These instructions require Windows 10 and PowerShell to generate certificates. The generated certificates can be installed on any supported P2S client.
- [MakeCert instructions](#): Use MakeCert if you don't have access to a Windows 10 computer for generating certificates. Although MakeCert is deprecated, you can still use it to generate certificates. You can install the generated certificates on any supported P2S client.
- [Linux instructions](#)

When you generate a client certificate from a self-signed root certificate, 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, export it as a .pfx file, along with the entire certificate chain. Doing so will create a .pfx file that contains the root certificate information required for the client to authenticate.

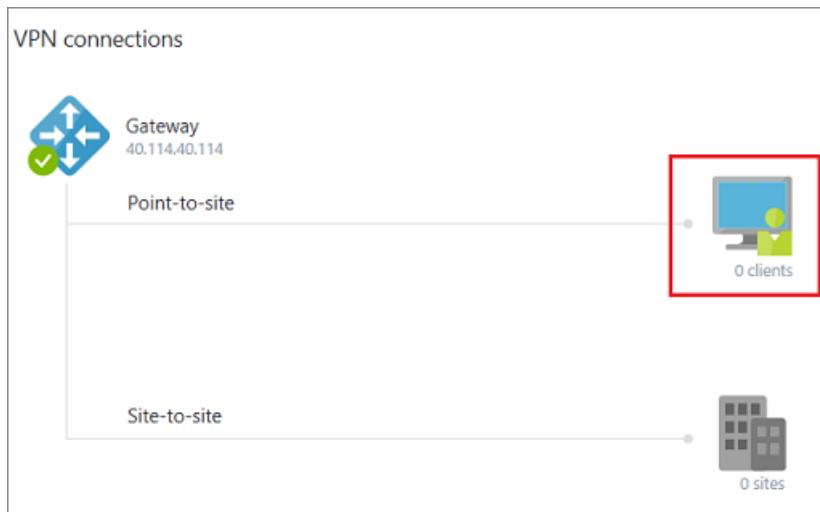
To export the certificate

For steps to export a certificate, see [Generate and export certificates for Point-to-Site using PowerShell](#).

Upload the root certificate .cer file

After the gateway has been created, upload the .cer file (which contains the public key information) for a trusted root certificate to the Azure server. Don't upload the private key for the root certificate. After you upload the certificate, Azure uses it to authenticate clients that have installed a client certificate generated from the trusted root certificate. You can later upload additional trusted root certificate files (up to 20), if needed.

1. On the **VPN connections** section of the page for your VNet, select the clients graphic to open the **Point-to-site VPN connection** page.



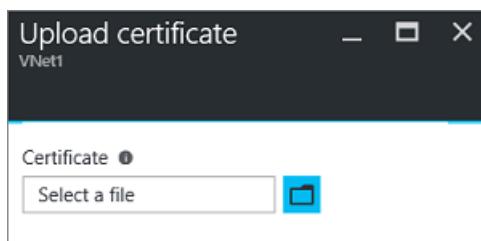
2. On the **Point-to-site VPN connection** page, select **Manage certificate** to open the **Certificates** page.



3. On the **Certificates** page, select **Upload** to open the **Upload certificate** page.



4. Select the folder graphic to browse for the .cer file. Select the file, then select **OK**. The uploaded certificate appears on the **Certificates** page.



Configure the client

To connect to a VNet by 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](#).

Generate and install a VPN client configuration package

1. In the Azure portal, in the **Overview** page for your VNet, in **VPN connections**, select the client graphic to open the **Point-to-site VPN connection** page.
2. From the **Point-to-site VPN connection** page, select the download package that corresponds to the client operating system where it's installed:
 - For 64-bit clients, select **VPN Client (64-bit)**.
 - For 32-bit clients, select **VPN Client (32-bit)**.



- After the package generates, download it and then install it on your client computer. If you see a SmartScreen popup, select **More info**, then select **Run anyway**. You can also save the package to install on other client computers.

Install a client certificate

To create a P2S connection from a different client computer than the one used to generate the client certificates, install a client certificate. When you install a client certificate, you need the password that was created when the client certificate was exported. Typically, you can install the certificate by just double-clicking it. For more information, see [Install an exported client certificate](#).

Connect to your VNet

NOTE

You must have Administrator rights on the client computer from which you are connecting.

- To connect to your VNet, on the client computer, go to **VPN connections** in the Azure portal and locate the VPN connection that you created. The VPN connection has the same name as your virtual network. Select **Connect**. If a pop-up message about the certificate appears, select **Continue** to use elevated privileges.
- On the **Connection** status page, select **Connect** to start the connection. If you see the **Select Certificate** screen, verify that the displayed client certificate is the correct one. If not, select the correct certificate from the drop-down list, and then select **OK**.
- If your connection succeeds, you'll see a **Connected** notification.

Troubleshooting P2S connections

If you have trouble connecting, check the following items:

- If you exported a client certificate with **Certificate Export Wizard**, make sure that you exported it as a .pfx file and selected **Include all certificates in the certification path if possible**. When you export it with this value, the root certificate information is also exported. After you install the certificate on the client computer, the root certificate in the .pfx file is also installed. To verify that the root certificate is installed, open **Manage user certificates** and select **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed, which must be present for authentication to work.
- If you used a certificate that was issued by an Enterprise CA solution and you can't authenticate, verify the authentication order on the client certificate. Check the authentication list order by double-clicking the client certificate, selecting the **Details** tab, and then selecting **Enhanced Key Usage**. Make sure *Client Authentication* is the first item in the list. If it isn't, 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

- Verify that your VPN connection is active. Open an elevated command prompt on your client computer, and run **ipconfig/all**.
- 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

Create a Remote Desktop Connection to connect to a VM that's deployed to your VNet. The best way to verify you can connect to your VM is to connect with its private IP address, rather than its computer name. That way, you're testing to see if you can connect, not whether name resolution is configured properly.

1. Locate the private IP address for your VM. To find the private IP address of a VM, view the properties for the VM in the Azure portal or use PowerShell.
2. Verify that you're connected to your VNet with the Point-to-Site VPN connection.
3. To open Remote Desktop Connection, enter *RDP* or *Remote Desktop Connection* in the search box on the taskbar, then select **Remote Desktop Connection**. You can also open it by using the **mstsc** command in PowerShell.
4. In **Remote Desktop Connection**, enter the private IP address of the VM. If necessary, select **Show Options** to adjust additional settings, then connect.

To troubleshoot an RDP connection to a VM

If you're having trouble connecting to a virtual machine over your VPN connection, there are a few things you can check.

- Verify that your VPN connection is successful.
- Verify that you're connecting to the private IP address for the VM.
- Enter **ipconfig** to check the IPv4 address assigned to the Ethernet adapter on the computer from which you're connecting. An overlapping address space occurs when the IP address is within the address range of the VNet that you're connecting to, or within the address range of your VPNCClientAddressPool. 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 by 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 is generated after you specify the DNS server IP addresses for the VNet. If you update the DNS server IP addresses, generate and install a new VPN client configuration package.

For more troubleshooting information, see [Troubleshoot Remote Desktop connections to a VM](#).

Add or remove trusted root certificates

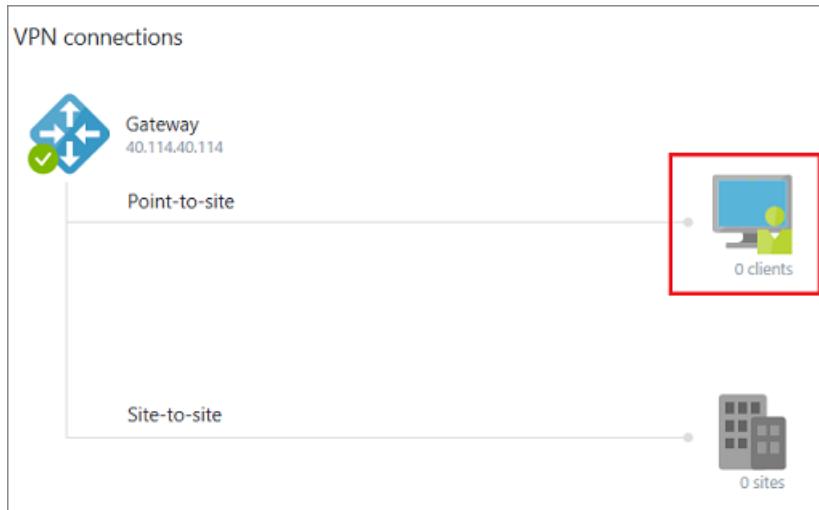
You can add and remove trusted root certificates from Azure. When you remove a root certificate, clients that have a certificate generated from that root can no longer authenticate and connect. For those clients to authenticate and connect again, you must install a new client certificate generated from a root certificate that's trusted by Azure.

To add a trusted root certificate

You can add up to 20 trusted root certificate .cer files to Azure. For instructions, see [Upload the root certificate .cer file](#).

To remove a trusted root certificate

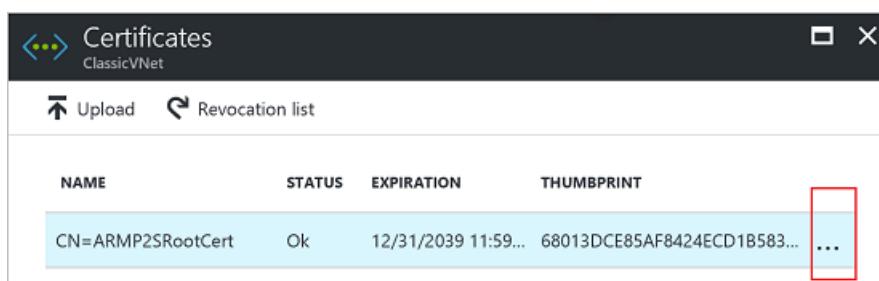
1. On the **VPN connections** section of the page for your VNet, select the clients graphic to open the **Point-to-site VPN connection** page.



2. On the **Point-to-site VPN connection** page, select **Manage certificate** to open the **Certificates** page.



3. On the **Certificates** page, select the ellipsis next to the certificate that you want to remove, then select **Delete**.



Revoke a client certificate

If necessary, you can revoke a client certificate. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. This method 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 its spaces so that it's a continuous string.
3. Go to the classic virtual network. Select **Point-to-site VPN connection**, then select **Manage certificate** to open the **Certificates** page.
4. Select **Revocation list** to open the **Revocation list** page.
5. Select **Add certificate** to open the **Add certificate to revocation list** page.
6. In **Thumbprint**, paste the certificate thumbprint as one continuous line of text, with no spaces. Select **OK** to finish.

After updating has completed, the certificate can no longer be used to connect. Clients that try to connect by using this certificate receive a message saying that the certificate is no longer valid.

Point-to-Site FAQ

This FAQ applies to P2S connections that use 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 that supports SSTP for Point-to-Site?

No. Support is limited only to the listed Windows operating system versions.

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

The amount of VPN client endpoints depends on your gateway sku and protocol.

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
Generation 1	Basic	Max. 10	Max. 128	Not Supported	100 Mbps	Not Supported	No
Generation 1	VpnGw1	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	No
Generation 1	VpnGw2	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	No
Generation 1	VpnGw3	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	No
Generation 1	VpnGw1AZ	Max. 30*	Max. 128	Max. 250	650 Mbps	Supported	Yes
Generation 1	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1 Gbps	Supported	Yes

VPN GATEWAY GENERATION	SKU	S2S/VNET-TO-VNET TUNNELS	P2S SSTP CONNECTIONS	P2S IKEV2/OPEN VPN CONNECTIONS	AGGREGATE THROUGHPUT BENCHMARK	BGP	ZONE-REDUNDANT
Generation 1	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	1.25 Gbps	Supported	Yes
Generation 2	VpnGw2	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	No
Generation 2	VpnGw3	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	No
Generation 2	VpnGw4	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	No
Generation 2	VpnGw5	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	No
Generation 2	VpnGw2AZ	Max. 30*	Max. 128	Max. 500	1.25 Gbps	Supported	Yes
Generation 2	VpnGw3AZ	Max. 30*	Max. 128	Max. 1000	2.5 Gbps	Supported	Yes
Generation 2	VpnGw4AZ	Max. 30*	Max. 128	Max. 5000	5 Gbps	Supported	Yes
Generation 2	VpnGw5AZ	Max. 30*	Max. 128	Max. 10000	10 Gbps	Supported	Yes

(*) Use [Virtual WAN](#) if you need more than 30 S2S VPN tunnels.

- The resizing of VpnGw SKUs is allowed within the same generation, except resizing of the Basic SKU. The Basic SKU is a legacy SKU and has feature limitations. In order to move from Basic to another VpnGw SKU, you must delete the Basic SKU VPN gateway and create a new gateway with the desired Generation and SKU size combination.
- These connection limits are separate. For example, you can have 128 SSTP connections and also 250 IKEv2 connections on a VpnGw1 SKU.
- Pricing information can be found on the [Pricing](#) page.
- SLA (Service Level Agreement) information can be found on the [SLA](#) page.
- On a single tunnel a maximum of 1 Gbps throughput can be achieved. Aggregate Throughput Benchmark in the above table is based on measurements of multiple tunnels aggregated through a single gateway. The Aggregate Throughput Benchmark for a VPN Gateway is S2S + P2S combined. **If you have a lot of P2S connections, it can negatively impact a S2S connection due to throughput limitations.** The Aggregate Throughput Benchmark is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.

To help our customers understand the relative performance of SKUs using different algorithms, we used publicly available iPerf and CTS Traffic tools to measure performances. The table below lists the results of performance tests for Generation 1, VpnGw SKUs. As you can see, the best performance is obtained when we used

GCMAES256 algorithm for both IPsec Encryption and Integrity. We got average performance when using AES256 for IPsec Encryption and SHA256 for Integrity. When we used DES3 for IPsec Encryption and SHA256 for Integrity we got lowest performance.

GENERATION	SKU	ALGORITHMS USED	THROUGHPUT OBSERVED	PACKETS PER SECOND OBSERVED
Generation1	VpnGw1	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000
Generation1	VpnGw1AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	650 Mbps 500 Mbps 120 Mbps	58,000 50,000 50,000
Generation1	VpnGw2AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1 Gbps 500 Mbps 120 Mbps	90,000 80,000 55,000
Generation1	VpnGw3AZ	GCMAES256 AES256 & SHA256 DES3 & SHA256	1.25 Gbps 550 Mbps 120 Mbps	105,000 90,000 60,000

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 up to 20 root certificates.

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

Yes. We use Secure Socket Tunneling Protocol (SSTP) to tunnel through firewalls. This tunnel appears 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 won't reestablish the VPN connection automatically.

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

No. Auto reconnect and DDNS are currently not supported in Point-to-Site VPNs.

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

Yes. Both solutions will work if you have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We don't support Point-to-Site for static routing VPN gateways or gateways that use the **-VpnType PolicyBased** cmdlet.

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

Yes. However, the virtual networks can't 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

- After 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 Linux 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)

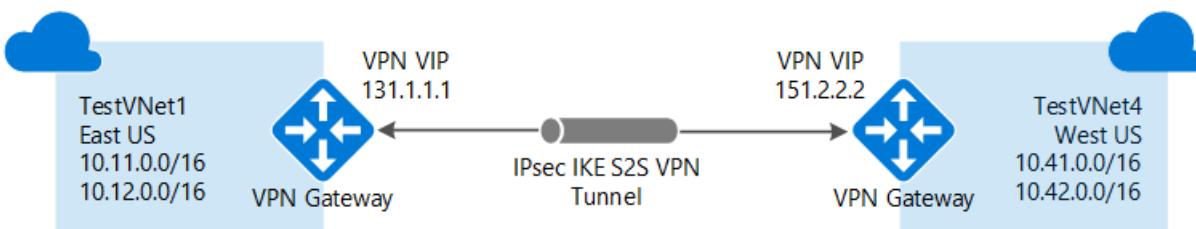
2/13/2020 • 12 minutes to read • [Edit Online](#)

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.

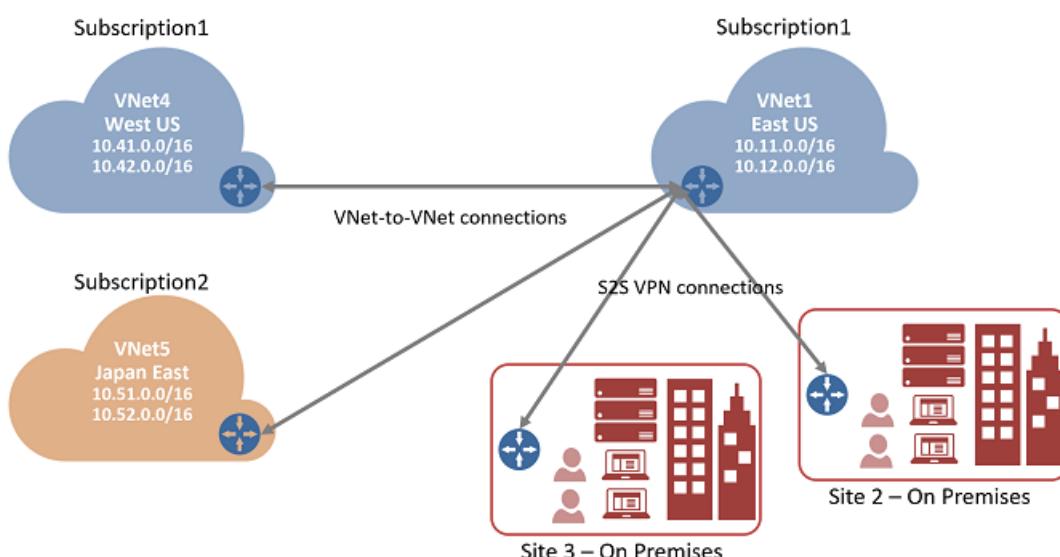
This article helps you create a VPN gateway connection between virtual networks. The virtual networks can be in the same or different regions, and from the same or different subscriptions. The steps in this article apply to the classic deployment model and the Azure portal. You can also create this configuration 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.

Working with 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. When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

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

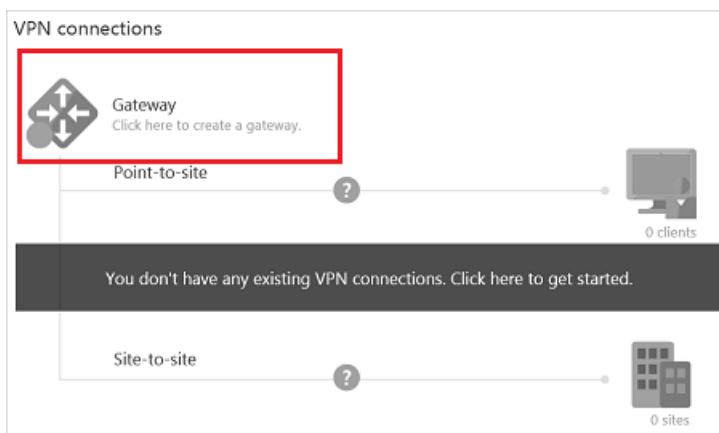
Azure uses the settings specified in each local network site to determine how to route traffic between the VNets. Each VNet must point to the respective local network that you want to route traffic to. You determine the name you want to use to refer to each local network site. It's best to use something 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.

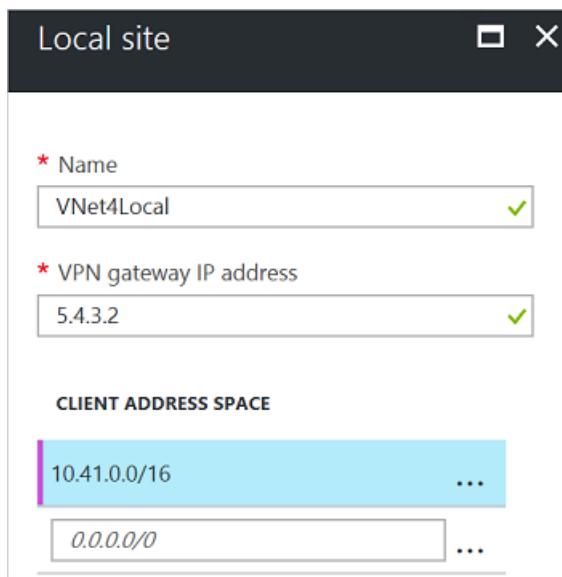
The local site for each VNet is the other VNet. The following example values are used for our configuration:

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.

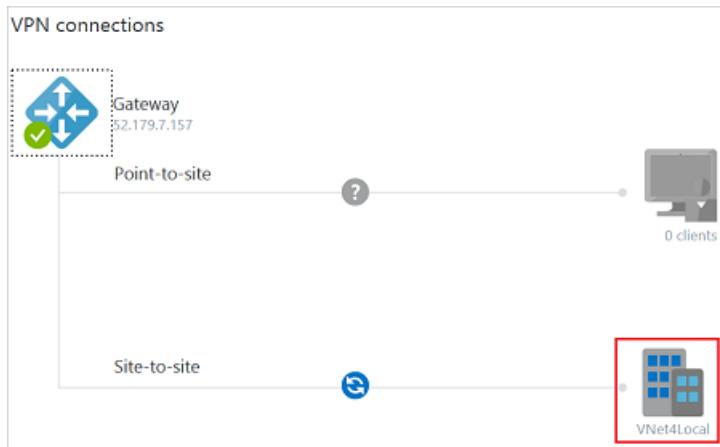
2. Click to open the VNet **Overview** page. On the page, in **VPN connections**, you can view the IP address for your virtual network gateway.



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

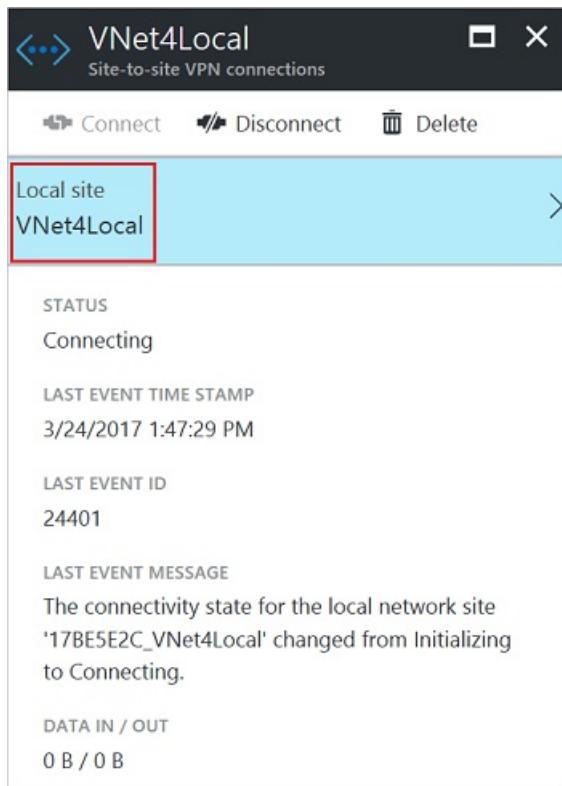
1. Locate your virtual network in the Azure portal.
2. On the VNet **Overview** page, click the local site.



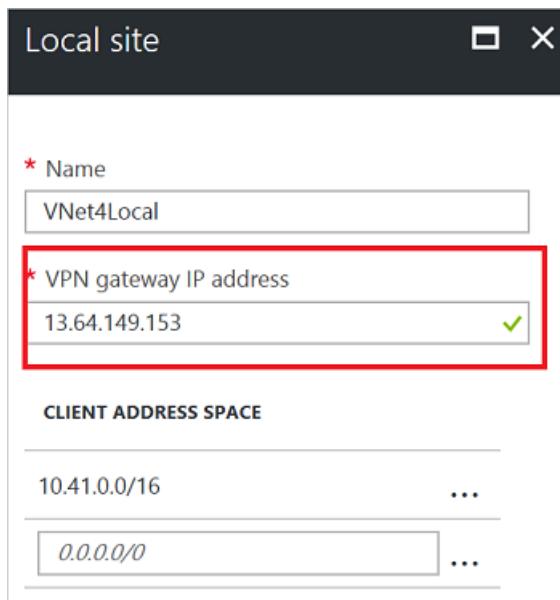
3. On the **Site-to-Site VPN Connections** page, click the name of the local site that you want to modify.

Site-to-site VPN connections			
TestVNet4			
NAME		STATUS	
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 [Working with Azure PowerShell](#).

2. Open your PowerShell console with elevated rights. Use the following examples to help you connect. You must run these commands locally using the PowerShell service management module. To switch to service management, use this command:

```
azure config mode asm
```

3. Connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

4. Check the subscriptions for the account.

```
Get-AzureSubscription
```

5. If you have more than one subscription, select the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionId "Replace_with_your_subscription_ID"
```

6. 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
```

7. Open the file with a text editor and view the names for your VNets and sites. These names will be the names 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

2/13/2020 • 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 Resource Manager 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:

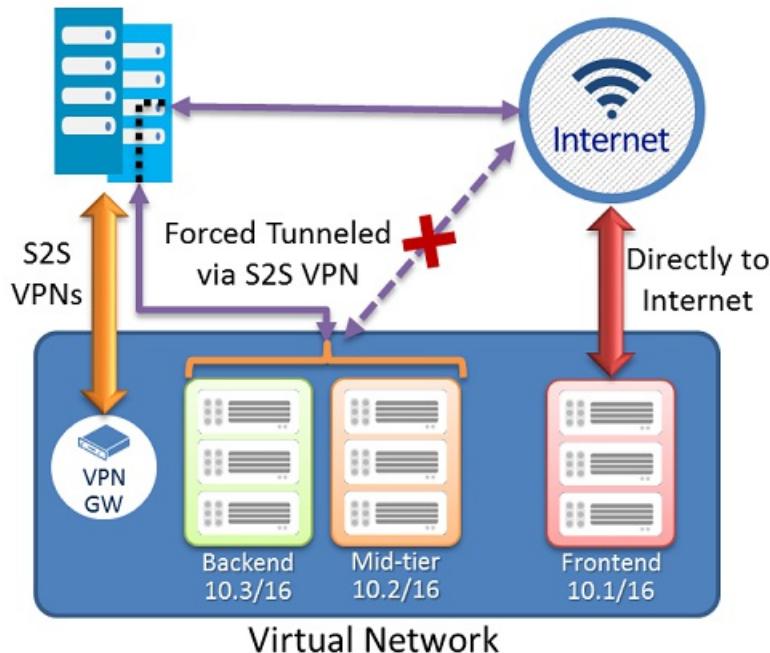
- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network.
 - **On-premises routes:** To the Azure VPN gateway.
 - **Default route:** Directly 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. 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.
- When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

To sign in

1. Open your PowerShell console with elevated rights. To switch to service management, use this command:

```
azure config mode asm
```

2. Connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

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)

2/13/2020 • 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.

When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

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:

1. Open your PowerShell console with elevated rights. To switch to service management, use this command:

```
azure config mode asm
```

2. 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)

2/13/2020 • 7 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.

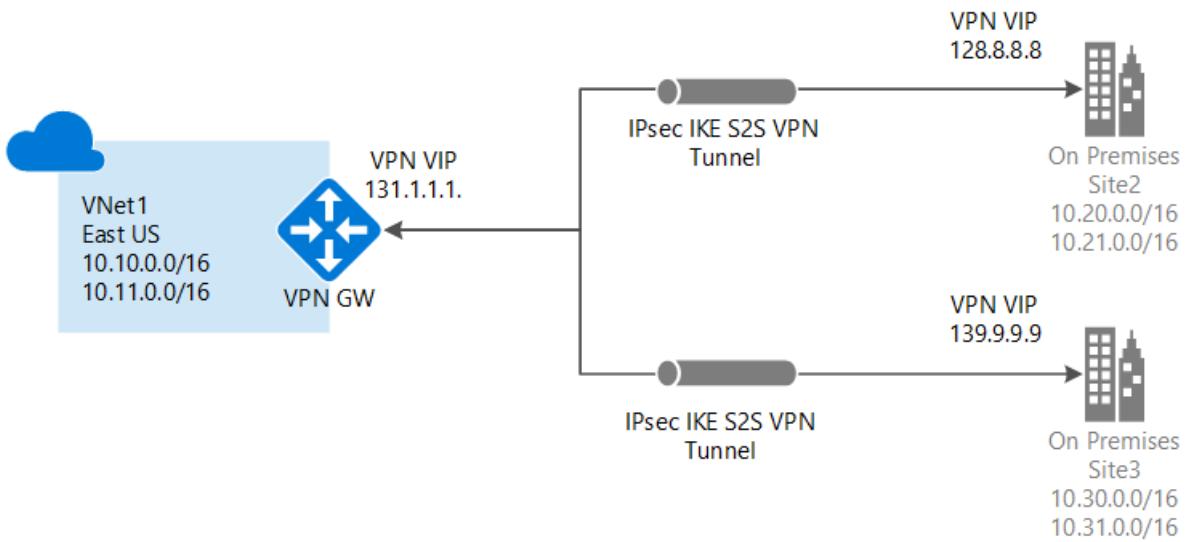
We update this table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from this table.

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

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.
- 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.

Working with Azure PowerShell

When working with the classic deployment model, you can't use Azure Cloud Shell. Instead, you must install the latest version of the Azure Service Management (SM) PowerShell cmdlets locally on your computer. These cmdlets are different from the AzureRM or Az cmdlets. To install the SM cmdlets, see [Install Service Management cmdlets](#). For more information about Azure PowerShell in general, see the [Azure PowerShell documentation](#).

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

Open your PowerShell console with elevated rights. To switch to service management, use this command:

```
azure config mode asm
```

Connect to your account. Use the following example to help you connect:

```
Add-AzureAccount
```

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="https://www.w3.org/2001/XMLSchema"
    xmlns:xsi="https://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

2/6/2020 • 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.

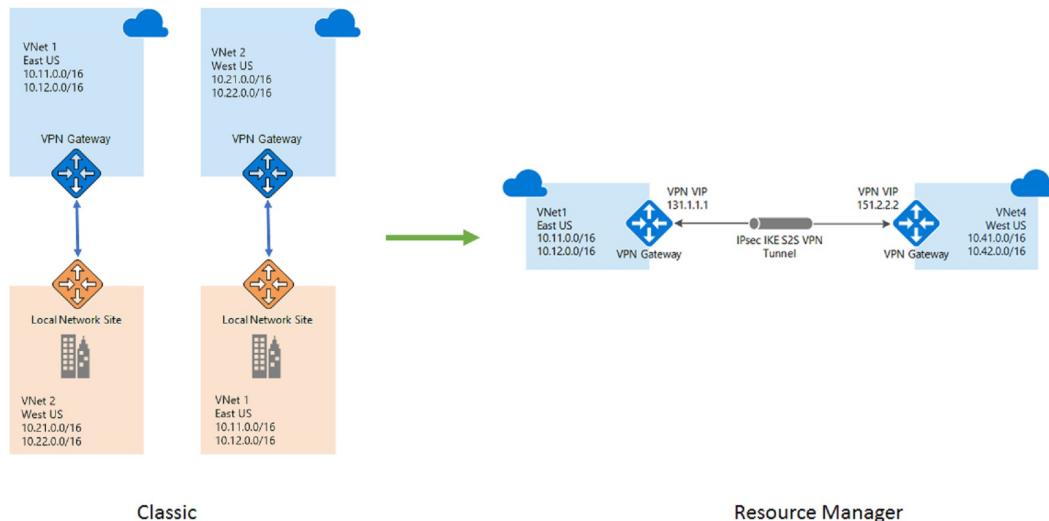
NOTE

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.

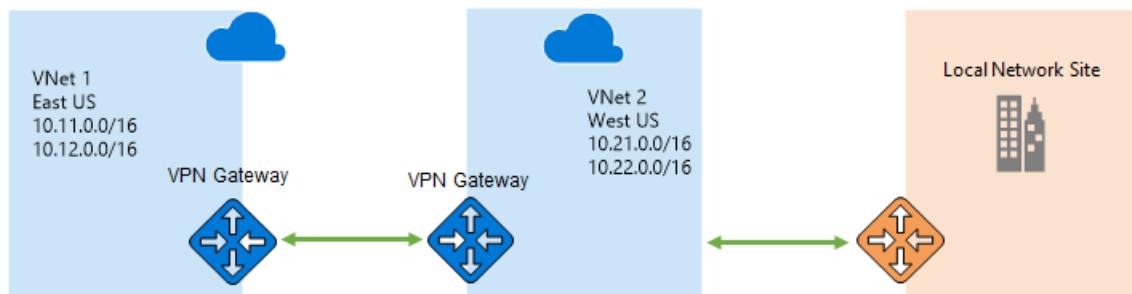
between the two VNets. This model has manageability challenges since any address range changes in one VNet must also be maintained in the corresponding local site representation. In Resource Manager model this workaround is no longer needed. The connection between the two VNets can be directly achieved 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

2/4/2020 • 40 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.

Is there an additional cost for setting up a VPN gateway as active-active?

No.

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 private 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.

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:
 - [Azure portal](#)
 - [Azure PowerShell](#)
 - [Azure PowerShell - classic](#)
3. [Create a new gateway of the type you want 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?

Zone-redundant and zonal gateways (gateway SKUs that have AZ in the name) both rely on a *Standard SKU*

Azure public IP resource. Azure Standard SKU public IP resources must use a static allocation method. Therefore, you will have the public IP address for your VPN gateway as soon as you create the Standard SKU public IP resource you intend to use for it.

For non-zone-redundant and non-zonal gateways (gateway SKUs that do *not* have AZ in the name), you cannot get the VPN gateway IP address before it is created. The IP address changes only if you delete and re-create your VPN gateway.

Can I request a Static Public IP address for my VPN gateway?

As stated above, zone-redundant and zonal gateways (gateway SKUs that have AZ in the name) both rely on a *Standard SKU* Azure public IP resource. Azure Standard SKU public IP resources must use a static allocation method.

For non-zone-redundant and non-zonal gateways (gateway SKUs that do *not* have AZ in the name), only dynamic IP address assignment is supported. However, this doesn't 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 then re-created. The VPN gateway public IP address doesn't change when you resize, reset, or complete other internal maintenance and 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 virtual network 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 virtual network 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 [VPN device configuration overview](#).
- 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?

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or **OpenVPN® Protocol**.

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
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

Azure CLI

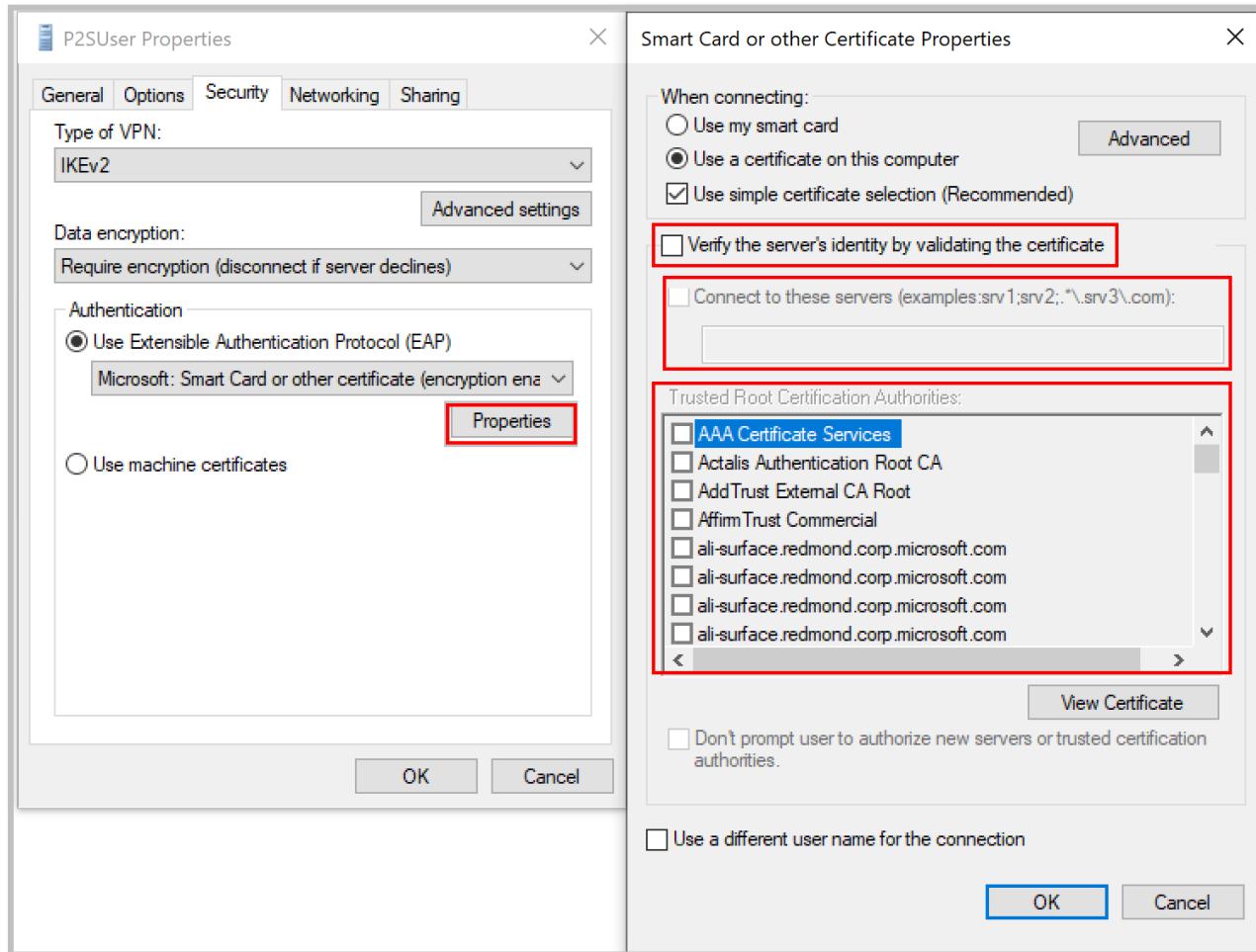
```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

What should I do if I'm getting a certificate mismatch when connecting using certificate authentication?

Uncheck "**Verify the server's identity by validating the certificate**" or **add the server FQDN along with the certificate** when creating a profile manually. You can do this by running **rasphone** from a command prompt and

picking the profile from the drop-down list.

Bypassing server identity validation is not recommended in general, but with Azure certificate authentication, the same certificate is being used for server validation in the VPN tunneling protocol (IKEv2/SSTP) and the EAP protocol. Since the server certificate and FQDN is already validated by the VPN tunneling protocol, it is redundant to validate the same again in EAP.



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

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

Can I use certificates from Azure Key Vault?

No.

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?

It depends on the gateway SKU. For more information on the number of connections supported, see [Gateway SKUs](#).

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 or above
- 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.

NOTE

You will have to set the above registry key if you are running an older version of Windows 10 (10240).

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

Azure supports three 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 outbound TCP port that 443 SSL uses.
- OpenVPN. OpenVPN is a SSL-based solution that can penetrate firewalls since most firewalls open the outbound TCP port that 443 SSL uses.
- IKEv2 VPN. IKEv2 VPN is a standards-based IPsec VPN solution that uses outbound UDP ports 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. However, you can use the OpenVPN client on all platforms to connect over OpenVPN protocol. 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 or [OpenVPN® Protocol](#).

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

OS VERSION	DATE	NUMBER/LINK
Windows 10 Version 1703	January 17, 2018	KB4057144
Windows 10 Version 1709	March 22, 2018	KB4089848

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.

How do I remove the configuration of a P2S connection?

A P2S configuration can be removed using Azure CLI and PowerShell using the following commands:

Azure PowerShell

```
$gw=Get-AzVirtualNetworkGateway -name <gateway-name>
$gw.VPNClientConfiguration = $null
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gw
```

Azure CLI

```
az network vnet-gateway update --name <gateway-name> --resource-group <resource-group name> --remove
"vpnClientConfiguration"
```

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 for SSTP, but depends on the gateway SKU for IKEv2. For more information on the number of connections supported, see [Gateway SKUs](#).

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.

Does RADIUS authentication work with the OpenVPN client?

RADIUS authentication is supported for the OpenVPN protocol only through PowerShell.

VNet-to-VNet and Multi-Site connections

The VNet-to-VNet FAQ applies to VPN gateway connections. For information about VNet peering, see [Virtual network peering](#).

Does Azure charge for traffic between VNets?

VNet-to-VNet traffic within the same region is free for both directions when you use a VPN gateway connection. Cross-region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. For more information, see [VPN Gateway pricing page](#). If you're connecting your VNets by using VNet peering instead of a VPN gateway, see [Virtual network pricing](#).

Does VNet-to-VNet traffic travel across the internet?

No. VNet-to-VNet traffic travels across the Microsoft Azure backbone, not the internet.

Can I establish a VNet-to-VNet connection across Azure Active Directory (AAD) tenants?

Yes, VNet-to-VNet connections that use Azure VPN gateways work across AAD tenants.

Is VNet-to-VNet traffic secure?

Yes, it's protected by IPsec/IKE encryption.

Do I need a VPN device to connect VNets together?

No. Connecting multiple Azure virtual networks together doesn't require a VPN device unless cross-premises connectivity is required.

Do my VNets need to be in the same region?

No. The virtual networks can be in the same or different Azure regions (locations).

If the VNets aren't in the same subscription, do the subscriptions need to be associated with the same Active Directory tenant?

No.

Can I use VNet-to-VNet to connect virtual networks in separate Azure instances?

No. VNet-to-VNet supports connecting virtual networks within the same Azure instance. For example, you can't create a connection between global Azure and Chinese/German/US government Azure instances. Consider using a Site-to-Site VPN connection for these scenarios.

Can I use VNet-to-VNet along with multi-site connections?

Yes. Virtual network connectivity can be used simultaneously with multi-site VPNs.

How many on-premises sites and virtual networks can one virtual network connect to?

See the [Gateway requirements](#) table.

Can I use VNet-to-VNet to connect VMs or cloud services outside of a VNet?

No. VNet-to-VNet supports connecting virtual networks. It doesn't support connecting virtual machines or cloud services that aren't in a virtual network.

Can a cloud service or a load-balancing endpoint span VNets?

No. A cloud service or a load-balancing endpoint can't span across virtual networks, even if they're connected together.

Can I use a PolicyBased VPN type for VNet-to-VNet or Multi-Site connections?

No. VNet-to-VNet and Multi-Site connections require Azure VPN gateways with RouteBased (previously called dynamic routing) VPN types.

Can I connect a VNet with a RouteBased VPN Type to another VNet with a PolicyBased VPN type?

No, both virtual networks MUST use route-based (previously called dynamic routing) VPNs.

Do VPN tunnels share bandwidth?

Yes. All VPN tunnels of the virtual network share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

Are redundant tunnels supported?

Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.

Can I have overlapping address spaces for VNet-to-VNet configurations?

No. You can't have overlapping IP address ranges.

Can there be overlapping address spaces among connected virtual networks and on-premises local sites?

No. You can't have overlapping IP address ranges.

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 printable ASCII 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 all Azure SKUs except the Basic SKU.

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	ECP384	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.

How do I create connections with IKEv1 or IKEv2 protocol type?

IKEv1 connections can be created on all RouteBased VPN type SKUs, except the Basic SKU. You can specify a connection protocol type of IKEv1 or IKEv2 while creating connections. If you do not specify a connection protocol type, IKEv2 is used as default option where applicable. For more information, see the [PowerShell cmdlet](#) documentation. For SKU types and IKEv1/IKEv2 support, see [Connect gateways to policy-based VPN devices](#).

Is transit between between IKEv1 and IKEv2 connections allowed?

Yes. Transit between IKEv1 and IKEv2 connections is supported.

Can I have IKEv1 site-to-site connections on Basic SKUs of RouteBased VPN type?

No. The Basic SKU does not support this.

Can I change the connection protocol type after the connection is created (IKEv1 to IKEv2 and vice versa)?

No. Once the connection is created, IKEv1/IKEv2 protocols cannot be changed. You must delete and recreate a new connection with the desired protocol type.

Where can I find more configuration information for IPsec?

See [Configure IPsec/IKE policy for S2S or VNet-to-VNet connections](#)

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

What Private ASNs can I use?

The useable range of Private ASNs that can be used are:

- 64512-65514, 65521-65534

These ASNs are not reserved by IANA or Azure for use and therefore can be used to assign to your Azure VPN Gateway.

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.**

How many prefixes can I advertise to Azure VPN gateway?

We support up to 4000 prefixes. The BGP session is dropped if the number of prefixes exceeds the limit.

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. For more information, see [About BGP](#).

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 for active-standby VPN gateways, or two IP addresses for active-active VPN gateways. You can get the actual BGP IP address(es) allocated by using PowerShell (Get-AzVirtualNetworkGateway, look for the "bgpPeeringAddress" property), or in the Azure portal (under the "Configure BGP ASN" property on the Gateway Configuration page).

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 or the Vnet address space of the VPN Gateway. 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 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.
- 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](#).

"OpenVPN" is a trademark of OpenVPN Inc.

Azure subscription and service limits, quotas, and constraints

2/25/2020 • 85 minutes to read • [Edit Online](#)

This document lists some of the most common Microsoft Azure limits, which are also sometimes called quotas.

To learn more about Azure pricing, see [Azure pricing overview](#). There, you can estimate your costs by using the [pricing calculator](#). You also can go to the pricing details page for a particular service, for example, [Windows VMs](#). For tips to help manage your costs, see [Prevent unexpected costs with Azure billing and cost management](#).

Managing limits

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's no maximum limit column, the resource doesn't have adjustable limits.

[Free Trial subscriptions](#) aren't 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 your Azure Free Trial subscription to a Pay-As-You-Go subscription](#) and the [Free Trial subscription FAQ](#).

Some limits are managed at a regional level.

Let's use vCPU quotas as an example. To request a quota increase with support for vCPUs, you must decide how many vCPUs you want to use in which regions. You then make a specific request for Azure resource group vCPU quotas for the amounts and regions that you want. If you need to use 30 vCPUs in West Europe to run your application there, you specifically request 30 vCPUs in West Europe. Your vCPU quota isn't increased in any other region--only West Europe has the 30-vCPU quota.

As a result, decide what your Azure resource group quotas must be for your workload in any one region. Then request that amount in each region into which you want to deploy. For help in how to determine your current quotas for specific regions, see [Resolve errors for resource quotas](#).

General limits

For limits on resource names, see [Naming rules and restrictions for Azure resources](#).

For information about Resource Manager API read and write limits, see [Throttling Resource Manager requests](#).

Subscription limits

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Subscriptions per Azure Active Directory tenant	Unlimited.	Unlimited.
Coadministrators per subscription	Unlimited.	Unlimited.
Resource groups per subscription	980	980

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Resource Manager API request size	4,194,304 bytes.	4,194,304 bytes.
Tags per subscription ¹	Unlimited.	Unlimited.
Unique tag calculations per subscription ¹	10,000	10,000
Subscription-level deployments per location	800 ²	800

¹You can apply an unlimited number of tags per subscription. The number of tags per resource or resource group is limited to 50. Resource Manager returns a [list of unique tag name and values](#) in the subscription only when the number of tags is 10,000 or less. You still can find a resource by tag when the number exceeds 10,000.

²If you reach the limit of 800 deployments, delete deployments from the history that are no longer needed. To delete subscription level deployments, use [Remove-AzDeployment](#) or [az deployment delete](#).

Resource group limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resources per resource group	N/A	Resources aren't limited by resource group. Instead, they're limited by resource type in a resource group. See next row.
Resources per resource group, per resource type	800	Some resource types can exceed the 800 limit. See Resources not limited to 800 instances per resource group .
Deployments per resource group in the deployment history	800 ¹	800
Resources per deployment	800	800
Management locks per unique scope	20	20
Number of tags per resource or resource group	50	50
Tag key length	512	512
Tag value length	256	256

¹If you reach the limit of 800 deployments per resource group, delete deployments from the history that are no longer needed. Deleting an entry from the deployment history doesn't affect the deployed resources. For more information, see [Resolve error when deployment count exceeds 800](#).

Template limits

VALUE	DEFAULT LIMIT	MAXIMUM LIMIT
Parameters	256	256

Value	Default Limit	Maximum Limit
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	4 MB	4 MB
Parameter file size	64 KB	64 KB

You can exceed some template limits by using a nested template. For more information, see [Use linked templates when you deploy 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](#).

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 set up all of your domains for federation with on-premises Active Directory, you can add no more than 450 domain names in each directory.
Resources	<ul style="list-style-type: none"> A maximum of 50,000 Azure AD resources can be created in a single directory by users of the Free edition of Azure Active Directory by default. If you have at least one verified domain, the default directory service quota in Azure AD is extended to 300,000 Azure AD resources. A non-admin user can create no more than 250 Azure AD resources. Both active resources and deleted resources that are available to restore count toward this quota. Only deleted Azure AD resources that were deleted fewer than 30 days ago are available to restore. Deleted Azure AD resources that are no longer available to restore count toward this quota at a value of one-quarter for 30 days. If you have developers who are likely to repeatedly exceed this quota in the course of their regular duties, you can create and assign a custom role with permission to create a limitless number of app registrations.

CATEGORY	LIMITS
Schema extensions	<ul style="list-style-type: none"> String-type extensions can have a maximum of 256 characters. Binary-type extensions are limited to 256 bytes. Only 100 extension values, across <i>all</i> types and <i>all</i> applications, can be written to any single Azure AD resource. 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 the 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.
Application Manifest	A maximum of 1200 entries can be added in the Application Manifest.

CATEGORY	LIMITS
Groups	<ul style="list-style-type: none"> A user can create a maximum of 250 groups in an Azure AD organization. An Azure AD organization can have a maximum of 5000 dynamic groups. A maximum of 100 users can be owners of a single group. Any number of Azure AD resources can be members of a single group. A user can be a member of any number of groups. The number of members in a group that you can synchronize from your on-premises Active Directory to Azure Active Directory by using Azure AD Connect is limited to 50,000 members. Nested Groups in Azure AD are not supported within all scenarios <p>At this time the following are the supported scenarios with nested groups.</p> <ul style="list-style-type: none"> One group can be added as a member of another group and you can achieve group nesting. Group membership claims (when an app is configured to receive group membership claims in the token, nested groups the signed-in user is a member of are included) Conditional access (when scoping a conditional access policy to a group) Restricting access to self-serve password reset Restricting which users can do Azure AD Join and device registration <p>The following scenarios DO NOT support nested groups:</p> <ul style="list-style-type: none"> App role assignment (assigning groups to an app is supported, but groups nested within the directly assigned group will not have access), both for access and for provisioning Group-based licensing (assigning a license automatically to all members of a group) Office 365 Groups.
Application Proxy	<ul style="list-style-type: none"> A maximum of 500 transactions per second per App Proxy application A maximum of 750 transactions per second for the Azure AD organization <p>A transaction is defined as a single http request and response for a unique resource. When throttled, clients will receive a 429 response (too many requests).</p>

CATEGORY	LIMITS
Access Panel	<ul style="list-style-type: none"> There's no limit to the number of applications that can be seen in the Access Panel per user. This applies to users assigned licenses for Azure AD Premium or the Enterprise Mobility Suite. A maximum of 10 app tiles can be seen in the Access Panel for each user. This limit applies to users who are assigned licenses for Azure AD Free license plan. Examples of app tiles include Box, Salesforce, or Dropbox. This limit doesn't 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 Azure AD resource can be a member of no more than 30 administrative units.
Admin roles and permissions	<ul style="list-style-type: none"> A group cannot be added as an owner. A group cannot be assigned to a role. Users' ability to read other users' directory information cannot be restricted outside of the Azure AD organization-wide switch to disable all non-admin users' access to all directory information (not recommended). More information on default permissions here. It may take up to 15 minutes or signing out/signing in before admin role membership additions and revocations take effect.

API Management limits

RESOURCE	LIMIT
Maximum number of scale units	10 per region ¹
Cache size	5 GiB per unit ²
Concurrent back-end connections ³ per HTTP authority	2,048 per unit ⁴
Maximum cached response size	2 MiB
Maximum policy document size	256 KiB ⁵
Maximum custom gateway domains per service instance ⁶	20
Maximum number of CA certificates per service instance	10
Maximum number of service instances per subscription ⁷	20
Maximum number of subscriptions per service instance ⁷	500
Maximum number of client certificates per service instance ⁷	50

RESOURCE	LIMIT
Maximum number of APIs per service instance ⁷	50
Maximum number of API operations per service instance ⁷	1,000
Maximum total request duration ⁷	30 seconds
Maximum buffered payload size ⁷	2 MiB
Maximum request URL size ⁸	4096 bytes

¹Scaling limits depend on the pricing tier. To see the pricing tiers and their scaling limits, see [API Management pricing](#).

²Per unit cache size depends on the pricing tier. To see the pricing tiers and their scaling limits, see [API Management pricing](#).

³Connections are pooled and reused unless explicitly closed by the back end.

⁴This limit is per unit of the Basic, Standard, and Premium tiers. The Developer tier is limited to 1,024. This limit doesn't apply to the Consumption tier.

⁵This limit applies to the Basic, Standard, and Premium tiers. In the Consumption tier, policy document size is limited to 4 KiB.

⁶This resource is available in the Premium tier only.

⁷This resource applies to the Consumption tier only.

⁸Applies to the Consumption tier only. Includes an up to 2048 bytes long query string.

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 Azure App Service plan ¹	10	100	Unlimited ²	Unlimited ²	Unlimited ²	Unlimited ²
App Service plan	10 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 (maximum instances)	1 shared	1 shared	3 dedicated ³	10 dedicated ³	30 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 minutes) ⁶	3 minutes	3 minutes	Unlimited, pay at standard rates			

Resource	Free	Shared	Basic	Standard	Premium (V2)	Isolated
CPU time (day) ⁶	60 minutes	240 minutes	Unlimited, pay at standard rates			
Memory (1 hour)	1,024 MB per App Service plan	1,024 MB per app	N/A	N/A	N/A	N/A
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
IP connections	600	600	Depends on instance size ⁸	Depends on instance size ⁸	Depends on instance size ⁸	16,000
Concurrent debugger connections per application	1	1	1	5	5	5
App Service Certificates per subscription ⁹	Not supported	Not supported	10	10	10	10
Custom domains per app	0 (azurewebsites.net subdomain only)	500	500	500	500	500
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
Hybrid connections per plan			5	25	200	200
Integrated load balancer		X	X	X	X	X ¹⁰
Always On			X	X	X	X

Resource	Free	Shared	Basic	Standard	Premium (V2)	Isolated
Scheduled backups				Scheduled backups every 2 hours, a maximum of 12 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)
Autoscale				X	X	X
WebJobs ¹¹	X	X	X	X	X	X
Endpoint monitoring			X	X	X	X
Staging slots				5	20	20
SLA			99.95%	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. For more information, see [App Service pricing](#).

⁴More are allowed upon request.

⁵The storage limit is the total content size across all apps in the same App service plan. The total content size of all apps across all App service plans in a single resource group and region cannot exceed 500GB.

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. For Standard tier and above, there are no theoretical limits to web sockets, but other factors can limit the number of web sockets. For example, maximum concurrent requests allowed (defined by

`maxConcurrentRequestsPerCpu`) are: 7,500 per small VM, 15,000 per medium VM (7,500 x 2 cores), and 75,000 per large VM (18,750 x 4 cores).

⁸The maximum IP connections are per instance and depend on the instance size: 1,920 per B1/S1/P1V2 instance, 3,968 per B2/S2/P2V2 instance, 8,064 per B3/S3/P3V2 instance.

⁹The App Service Certificate quota limit per subscription can be increased via a support request to a maximum limit of 200.

¹⁰App Service Isolated SKUs can be internally load balanced (ILB) with Azure Load Balancer, so there's 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.

¹¹Run 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. There's no predefined limit on the number of WebJobs that can run in an App Service instance. There are practical limits that depend on what the application code is trying to do.

Automation limits

Process automation

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum number of new jobs that can be submitted every 30 seconds per Azure Automation account (nonscheduled jobs)	100	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum number of concurrent running jobs at the same instance of time per Automation account (nonscheduled jobs)	200	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum storage size of job metadata for a 30-day rolling period	10 GB (approximately 4 million jobs)	When this limit is reached, the subsequent requests to create a job fail.
Maximum job stream limit	1MB	A single stream cannot be larger than 1 MB.
Maximum number of modules that can be imported every 30 seconds per Automation account	5	
Maximum size of a module	100 MB	
Job run time, Free tier	500 minutes per subscription per calendar month	
Maximum amount of disk space allowed per sandbox ¹	1 GB	Applies to Azure sandboxes only.
Maximum amount of memory given to a sandbox ¹	400 MB	Applies to Azure sandboxes only.
Maximum number of network sockets allowed per sandbox ¹	1,000	Applies to Azure sandboxes only.
Maximum runtime allowed per runbook ¹	3 hours	Applies to Azure sandboxes only.
Maximum number of Automation accounts in a subscription	No limit	
Maximum number of Hybrid Worker Groups per Automation Account	4,000	
Maximum number of concurrent jobs that can be run on a single Hybrid Runbook Worker	50	
Maximum runbook job parameter size	512 kilobits	
Maximum runbook parameters	50	If you reach the 50-parameter limit, you can pass a JSON or XML string to a parameter and parse it with the runbook.

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum webhook payload size	512 kilobits	
Maximum days that job data is retained	30 days	
Maximum PowerShell workflow state size	5 MB	Applies to PowerShell workflow runbooks when checkpointing workflow.

¹A sandbox is a shared environment that can be used by multiple jobs. Jobs that use the same sandbox are bound by the resource limitations of the sandbox.

Change Tracking and Inventory

The following table shows the tracked item limits per machine for change tracking.

RESOURCE	LIMIT	NOTES
File	500	
Registry	250	
Windows software	250	Doesn't include software updates.
Linux packages	1,250	
Services	250	
Daemon	250	

Update Management

The following table shows the limits for Update Management.

RESOURCE	LIMIT	NOTES
Number of machines per update deployment	1000	

Azure Cache for Redis limits

RESOURCE	LIMIT
Cache size	1.2 TB
Databases	64
Maximum connected clients	40,000
Azure Cache for Redis replicas, for high availability	1
Shards in a premium cache with clustering	10

Azure Cache for Redis limits and sizes are different for each pricing tier. To see the pricing tiers and their associated sizes, see [Azure Cache for Redis pricing](#).

For more information on Azure Cache for Redis configuration limits, see [Default Redis server configuration](#).

Because configuration and management of Azure Cache for Redis instances is done by Microsoft, not all Redis commands are supported in Azure Cache for Redis. For more information, see [Redis commands not supported in Azure Cache for Redis](#).

Azure Cloud Services limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Web or 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
Hosted service certificates per deployment	199	199

¹Each Azure Cloud Service with web or worker roles can have two deployments, one for production and one for staging. This limit refers to the number of distinct roles, that is, configuration. This limit doesn't refer to the number of instances per role, that is, scaling.

Azure Cognitive Search limits

Pricing tiers determine the capacity and limits of your search service. Tiers include:

- **Free** multi-tenant service, shared with other Azure subscribers, is 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**, which includes S1, S2, S3, and 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 can be provisioned at a specific tier. You're 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 an SKU or tier for Azure Cognitive Search](#).

Maximum service limits can be raised upon request. If you need more services within the same subscription, contact Azure Support.

RESOURCE	FREE ¹	BASIC	S1	S2	S3	S3 HD	L1	L2
Maximum services	1	16	16	8	6	6	6	6

Resource	Free	Basic	S1	S2	S3	S3 HD	L1	L2
Maximum scale in search units (SU) ²	N/A	3 SU	36 SU	36 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 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	L1	L2
Service level agreement (SLA) ³	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage per partition	50 MB	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Partitions per service	N/A	1	12	12	12	3	12	12
Partition size	N/A	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Replicas	N/A	3	12	12	12	12	12	12

¹ Basic has one fixed partition. At this tier, additional search units are used for allocating more replicas for increased query workloads.

² S3 HD has a hard limit of three 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 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) SLAs. Three or more replicas are required for query and indexing (read-write) SLAs. The number of partitions isn't 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 Cognitive Search](#).

Azure Cognitive Services limits

The following limits are for the number of Cognitive Services resources per Azure subscription. Each of the Cognitive Services may have additional limitations, for more information see [Azure Cognitive Services](#).

Type	Limit	Example
A mixture of Cognitive Services resources	Maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, 50 Speech Service resources in West US, and 50 Text Analytics resources in East US.
A single type of Cognitive Services resources.	Maximum of 100 resources per region, with a maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, and 100 Computer Vision resources in East US.

Azure Cosmos DB limits

For Azure Cosmos DB limits, see [Limits in Azure Cosmos DB](#).

Azure Data Explorer limits

The following table describes the maximum limits for Azure Data Explorer clusters.

Resource	Limit
Clusters per region per subscription	20
Instances per cluster	1000
Number of databases in a cluster	10,000
Number of attached database configurations in a cluster	70

The following table describes the limits on management operations performed on Azure Data Explorer clusters.

Scope	Operation	Limit
Cluster	read (for example, get a cluster)	500 per 5 minutes
Cluster	write (for example, create a database)	1000 per hour

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](#).

Azure Functions limits

Resource	Consumption Plan	Premium Plan	App Service Plan ¹
Scale out	Event driven	Event driven	Manual/autoscale

RESOURCE	CONSUMPTION PLAN	PREMIUM PLAN	APP SERVICE PLAN
Max instances	200	100	10-20
Default timeout duration (min)	5	30	30 ²
Max timeout duration (min)	10	unbounded ⁸	unbounded ³
Max outbound connections (per instance)	600 active (1200 total)	unbounded	unbounded
Max request size (MB) ⁴	100	100	100
Max query string length ⁴	4096	4096	4096
Max request URL length ⁴	8192	8192	8192
ACU per instance	100	210-840	100-840
Max memory (GB per instance)	1.5	3.5-14	1.75-14
Function apps per plan	100	100	unbounded ⁵
App Service plans	100 per region	100 per resource group	100 per resource group
Storage ⁶	1 GB	250 GB	50-1000 GB
Custom domains per app	500 ⁷	500	500
Custom domain SSL support	unbounded SNI SSL connection included	unbounded SNI SSL and 1 IP SSL connections included	unbounded SNI SSL and 1 IP SSL connections included

¹ For specific limits for the various App Service plan options, see the [App Service plan limits](#).

² By default, the timeout for the Functions 1.x runtime in an App Service plan is unbounded.

³ Requires the App Service plan be set to [Always On](#). Pay at standard [rates](#).

⁴ These limits are [set in the host](#).

⁵ The actual number of function apps that you can host depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.

⁶ The storage limit is the total content size in temporary storage across all apps in the same App Service plan. Consumption plan uses Azure Files for temporary storage.

⁷ When your function app is hosted in a [Consumption plan](#), only the CNAME option is supported. For function apps in a [Premium plan](#) or an [App Service plan](#), you can map a custom domain using either a CNAME or an A record.

⁸ Guaranteed for up to 60 minutes.

Azure Kubernetes Service limits

RESOURCE	DEFAULT LIMIT
Maximum clusters per subscription	100

RESOURCE	DEFAULT LIMIT
Maximum nodes per cluster with Virtual Machine Availability Sets and Basic Load Balancer SKU	100
Maximum nodes per cluster with Virtual Machine Scale Sets and Standard Load Balancer SKU	1000 (100 nodes per node pool)
Maximum pods per node: Basic networking with Kubenet	110
Maximum pods per node: Advanced networking with Azure Container Networking Interface	Azure CLI deployment: 30 ¹ Azure Resource Manager template: 30 ¹ Portal deployment: 30

¹When you deploy an Azure Kubernetes Service (AKS) cluster with the Azure CLI or a Resource Manager template, this value is configurable up to 250 pods per node. You can't configure maximum pods per node after you've already deployed an AKS cluster, or if you deploy a cluster by using the Azure portal.

Azure Machine Learning limits

The latest values for Azure Machine Learning Compute quotas can be found in the [Azure Machine Learning quota page](#)

Azure Maps limits

The following table shows the usage limit for the Azure Maps S0 pricing tier. Usage limit depends on the pricing tier.

RESOURCE	S0 PRICING TIER LIMIT
Maximum request rate per subscription	50 requests per second

The following table shows the data size limit for Azure Maps. The Azure Maps data service is available only at the S1 pricing tier.

RESOURCE	LIMIT
Maximum size of data	50 MB

For more information on the Azure Maps pricing tiers, see [Azure Maps pricing](#).

Azure Monitor limits

Alerts

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Metric alerts (classic)	100 active alert rules per subscription.	Call support.
Metric alerts	1000 active alert rules per subscription in Azure public, Azure China 21Vianet and Azure Government clouds.	Call support.
Activity log alerts	100 active alert rules per subscription.	Same as default.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Log alerts	512	Call support.
Action groups	2,000 action groups per subscription.	Call support.
Autoscale settings	100 per region per subscription.	Same as default.

Action groups

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure app push	10 Azure app actions per action group.	Call support.
Email	1,000 email actions in an action group. No more than 100 emails in an hour. Also see the rate limiting information .	Call support.
ITSM	10 ITSM actions in an action group.	Call support.
Logic app	10 logic app actions in an action group.	Call support.
Runbook	10 runbook actions in an action group.	Call support.
SMS	10 SMS actions in an action group. No more than 1 SMS message every 5 minutes. Also see the rate limiting information .	Call support.
Voice	10 voice actions in an action group. No more than 1 voice call every 5 minutes. Also see the rate limiting information .	Call support.
Webhook	10 webhook actions in an action group. Maximum number of webhook calls is 1500 per minute per subscription. Other limits are available at action-specific information .	Call support.

Log queries and language

LIMIT	DESCRIPTION
Query language	Azure Monitor uses the same Kusto query language as Azure Data Explorer. See Azure Monitor log query language differences for KQL language elements not supported in Azure Monitor.
Azure regions	Log queries can experience excessive overhead when data spans Log Analytics workspaces in multiple Azure regions. See Query limits for details.

LIMIT	DESCRIPTION
Cross resource queries	Maximum number of Application Insights resources and Log Analytics workspaces in a single query limited to 100. Cross-resource query is not supported in View Designer. Cross-resource query in log alerts is supported in the new scheduledQueryRules API. See Cross-resource query limits for details.
Query throttling	A user is limited to 200 queries per 30 seconds on any number of workspaces. This limit applies to programmatic queries or to queries initiated by visualization parts such as Azure dashboards and the Log Analytics workspace summary page.

Log Analytics workspaces

Data collection volume and retention

TIER	LIMIT PER DAY	DATA RETENTION	COMMENT
Current Per GB pricing tier (introduced April 2018)	No limit	30 - 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Free tiers (introduced April 2016)	500 MB	7 days	When your workspace reaches the 500 MB per day limit, data ingestion stops and resumes at the start of the next day. A day is based on UTC. Note that data collected by Azure Security Center is not included in this 500 MB per day limit and will continue to be collected above this limit.
Legacy Standalone Per GB tier (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Per Node (OMS) (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Standard tier	No limit	30 days	Retention can't be adjusted
Legacy Premium tier	No limit	365 days	Retention can't be adjusted

Number of workspaces per subscription.

Pricing tier	Workspace limit	Comments
Free tier	10	This limit can't be increased.
All other tiers	No limit	You're limited by the number of resources within a resource group and the number of resource groups per subscription.

Azure portal

Category	Limits	Comments
Maximum records returned by a log query	10,000	Reduce results using query scope, time range, and filters in the query.

Data Collector API

Category	Limits	Comments
Maximum size for a single post	30 MB	Split larger volumes into multiple posts.
Maximum size for field values	32 KB	Fields longer than 32 KB are truncated.

Search API

Category	Limits	Comments
Maximum records returned in a single query	500,000	
Maximum size of data returned	64,000,000 bytes (~61 MiB)	
Maximum query running time	10 minutes	See Timeouts for details.
Maximum request rate	200 requests per 30 seconds per AAD user or client IP address	See Rate limits for details.

General workspace limits

Category	Limits	Comments
Maximum columns in a table	500	
Maximum characters for column name	500	
Data export	Not currently available	Use Azure Function or Logic App to aggregate and export data.

Data ingestion volume rate

Azure Monitor is a high scale data service that serves thousands of customers sending terabytes of data each month at a growing pace. The default ingestion volume rate limit for data sent from Azure resources using [diagnostic settings](#) is approximately **6 GB/min** per workspace. This is an approximate value since the actual size can vary between data types depending on the log length and its compression ratio. This limit does not apply to

data that is sent from agents or [Data Collector API](#).

If you send data at a higher rate to a single workspace, some data is dropped, and an event is sent to the *Operation* table in your workspace every 6 hours while the threshold continues to be exceeded. If your ingestion volume continues to exceed the rate limit or you are expecting to reach it sometime soon, you can request an increase to your workspace by opening a support request.

To be notified on such an event in your workspace, create a [log alert rule](#) using the following query with alert logic base on number of results grater than zero.

```
Operation  
|where OperationCategory == "Ingestion"  
|where Detail startswith "The rate of data crossed the threshold"
```

NOTE

Depending on how long you've been using Log Analytics, you might have access to legacy pricing tiers. Learn more about [Log Analytics legacy pricing tiers](#).

Application Insights

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 email to AIDataCap@microsoft.com .
Throttling	32,000 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	64,000,000 bytes	
Property and metric name length	150	See type schemas .
Property value string length	8,192	See type schemas .
Trace and exception message length	32,768	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](#).

Azure Policy limits

There's 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	500
Scope	Initiative definitions	100
Tenant	Initiative definitions	1,000
Scope	Policy or initiative assignments	100
Policy definition	Parameters	20
Initiative definition	Policies	100
Initiative definition	Parameters	100
Policy or initiative assignments	Exclusions (notScopes)	400
Policy rule	Nested conditionals	512

Azure SignalR Service limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure SignalR Service units per instance for Free tier	1	1
Azure SignalR Service units per instance for Standard tier	100	100
Azure SignalR Service units per subscription per region for Free tier	5	5
Total Azure SignalR Service unit counts per subscription per region	150	Unlimited
Connections per unit per day for Free tier	20	20
Connections per unit per day for Standard tier	1,000	1,000
Included messages per unit per day for Free tier	20,000	20,000

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Included messages per unit per day for Standard tier	1,000,000	1,000,000

To request an update to your subscription's default limits, open a support ticket.

Backup limits

For a summary of Azure Backup support settings and limitations, see [Azure Backup Support Matrices](#).

Batch limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Batch accounts per region per subscription	1-3	50
Dedicated cores per Batch account	90-900	Contact support
Low-priority cores per Batch account	10-100	Contact support
Active jobs and job schedules per Batch account (completed jobs have no limit)	100-300	1,000 ¹
Pools per Batch account	20-100	500 ¹

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](#).

¹To request an increase beyond this limit, contact Azure Support.

Classic deployment model limits

If you use classic deployment model instead of the Azure Resource Manager deployment model, the following limits apply.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
vCPUs per subscription ¹	20	10,000
Coadministrators per subscription	200	200
Storage accounts per subscription ²	100	100
Cloud services per subscription	20	200
Local networks per subscription	10	500
DNS servers per subscription	9	100

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Reserved IPs per subscription	20	100
Affinity groups per subscription	256	256
Subscription name length (characters)	64	64

¹Extra small instances count as one vCPU toward the vCPU limit despite using a partial CPU core.

²The storage account limit includes both Standard and Premium storage accounts.

Container Instances limits

RESOURCE	DEFAULT LIMIT
Standard sku container groups per region per subscription	100 ¹
Dedicated sku container groups per region per subscription	0 ¹
Number of containers per container group	60
Number of volumes per container group	20
Ports per IP	5
Container instance log size - running instance	4 MB
Container instance log size - stopped instance	16 KB or 1,000 lines
Container creates per hour	300 ¹
Container creates per 5 minutes	100 ¹
Container deletes per hour	300 ¹
Container deletes per 5 minutes	100 ¹

¹To request a limit increase, create an [Azure Support request](#).

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
Maximum image layer size	200 GiB	200 GiB	200 GiB
ReadOps per minute ^{2, 3}	1,000	3,000	10,000
WriteOps per minute ^{2, 4}	100	500	2,000

RESOURCE	BASIC	STANDARD	PREMIUM
Download bandwidth MBps ²	30	60	100
Upload bandwidth MBps ²	10	20	50
Webhooks	2	10	500
Geo-replication	N/A	N/A	Supported
Content trust	N/A	N/A	Supported
Virtual network access	N/A	N/A	Preview
Repository-scoped permissions	N/A	N/A	Preview
• Tokens	N/A	N/A	20,000
• Scope maps	N/A	N/A	20,000
• Repositories per scope map	N/A	N/A	500

¹The 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 [Azure Container Registry pricing](#).

²*ReadOps*, *WriteOps*, and *Bandwidth* are minimum estimates. Azure Container Registry strives to improve performance as usage requires.

³A `docker pull` translates to multiple read operations based on the number of layers in the image, plus the manifest retrieval.

⁴A `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.

Content Delivery Network limits

RESOURCE	DEFAULT LIMIT
Azure Content Delivery Network profiles	25
Content Delivery Network endpoints per profile	25
Custom domains per endpoint	25

A Content Delivery Network subscription can contain one or more Content Delivery Network profiles. A Content Delivery Network profile can contain one or more Content Delivery Network endpoints. You might want to use multiple profiles to organize your Content Delivery Network endpoints by internet domain, web application, or some other criteria.

Data Factory limits

Azure Data Factory is a multitenant service that has the following default limits in place to make sure customer

subscriptions are protected from each other's workloads. To raise the limits up to the maximum for your subscription, contact support.

Version 2

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Data factories in an Azure subscription	50	Contact support .
Total number of entities, such as pipelines, data sets, triggers, linked services, and integration runtimes, within a data factory	5,000	Contact support .
Total CPU cores for Azure-SSIS Integration Runtimes under one subscription	256	Contact support .
Concurrent pipeline runs per data factory that's shared among all pipelines in the factory	10,000	Contact support .
Concurrent External activity runs per subscription per Azure Integration Runtime region External activities are managed on integration runtime but execute on linked services, including Databricks, stored procedure, HDInsights, Web, and others.	3000	Contact support .
Concurrent Pipeline activity runs per subscription per Azure Integration Runtime region Pipeline activities execute on integration runtime, including Lookup, GetMetadata, and Delete.	1000	Contact support .
Concurrent authoring operations per subscription per Azure Integration Runtime region Including test connection, browse folder list and table list, preview data.	200	Contact support .
Concurrent Data Integration Units ¹ consumption per subscription per Azure Integration Runtime region	Region group 1 ² : 6000 Region group 2 ² : 3000 Region group 3 ² : 1500	Contact support .
Maximum activities per pipeline, which includes inner activities for containers	40	40
Maximum number of linked integration runtimes that can be created against a single self-hosted integration runtime	100	Contact support .
Maximum parameters per pipeline	50	50
ForEach items	100,000	100,000

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
ForEach parallelism	20	50
Maximum queued runs per pipeline	100	100
Characters per expression	8,192	8,192
Minimum tumbling window trigger interval	15 min	15 min
Maximum timeout for pipeline activity runs	7 days	7 days
Bytes per object for pipeline objects ³	200 KB	200 KB
Bytes per object for dataset and linked service objects ³	100 KB	2,000 KB
Data Integration Units ¹ per copy activity run	256	Contact support .
Write API calls	1,200/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support .
Read API calls	12,500/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support .
Monitoring queries per minute	1,000	Contact support .
Entity CRUD operations per minute	50	Contact support .
Maximum time of data flow debug session	8 hrs	8 hrs
Concurrent number of data flows per factory	50	Contact support .
Concurrent number of data flow debug sessions per user per factory	3	3
Data Flow Azure IR TTL limit	4 hrs	Contact support .

¹ The data integration unit (DIU) is used in a cloud-to-cloud copy operation, learn more from [Data integration units \(version 2\)](#). For information on billing, see [Azure Data Factory pricing](#).

² [Azure Integration Runtime](#) is [globally available](#) to ensure data compliance, efficiency, and reduced network egress costs.

Region group	Regions
Region group 1	Central US, East US, East US2, North Europe, West Europe, West US, West US 2
Region group 2	Australia East, Australia Southeast, Brazil South, Central India, Japan East, Northcentral US, Southcentral US, Southeast Asia, West Central US
Region group 3	Canada Central, East Asia, France Central, Korea Central, UK South

³ Pipeline, data set, and linked service objects represent a logical grouping of your workload. Limits for these objects don't relate to the amount of data you can move and process with Azure Data Factory. Data Factory is designed to scale to handle petabytes of data.

Version 1

Resource	Default limit	Maximum limit
Pipelines within a data factory	2,500	Contact support .
Data sets within a data factory	5,000	Contact support .
Concurrent slices per data set	10	10
Bytes per object for pipeline objects ¹	200 KB	200 KB
Bytes per object for data set and linked service objects ¹	100 KB	2,000 KB
Azure 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	1,000	MaxInt (32 bit)

¹ Pipeline, data set, and linked service objects represent a logical grouping of your workload. Limits for these objects don't relate to the amount of data you can move and process with Azure Data Factory. 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 previous limit is the Data Factory-enforced core limit for on-demand HDInsight cores. It's different from the core limit that's associated with your Azure subscription.

³ The cloud data movement unit (DMU) for version 1 is used in a cloud-to-cloud copy operation, learn more from [Cloud data movement units \(version 1\)](#). For information on billing, see [Azure Data Factory pricing](#).

Resource	Default lower limit	Minimum limit
Scheduling interval	15 minutes	15 minutes
Interval between retry attempts	1 second	1 second

RESOURCE	DEFAULT LOWER LIMIT	MINIMUM LIMIT
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

Azure Data Lake Analytics makes the complex task of managing distributed infrastructure and complex code easy. It dynamically provisions resources, and you can use it to do analytics on exabytes of data. When the job completes, it winds down resources automatically. You pay only for the processing power that was used. As you increase or decrease the size of data stored or the amount of compute used, you don't have to rewrite code. To raise the default limits for your subscription, contact support.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of concurrent jobs	20	
Maximum number of analytics units (AUs) per account	250	Use any combination of up to a maximum of 250 AUs across 20 jobs. To increase this limit, contact Microsoft Support.
Maximum script size for job submission	3 MB	
Maximum number of Data Lake Analytics accounts per region per subscription	5	To increase this limit, contact Microsoft Support.

Data Lake Store limits

Azure Data Lake Storage Gen1 is an enterprise-wide hyper-scale repository for big data analytic workloads. You can use Data Lake Storage Gen1 to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. There's no limit to the amount of data you can store in a Data Lake Storage Gen1 account.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of Data Lake Storage Gen1 accounts, per subscription, per region	10	To request an increase for this limit, contact support.
Maximum number of access ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.
Maximum number of default ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.

Data Share limits

Azure Data Share enables organizations to simply and securely share data with their customers and partners.

RESOURCE	LIMIT
Maximum number of Data Share resources per Azure subscription	50
Maximum number of sent shares per Data Share resource	100
Maximum number of received shares per Data Share resource	100
Maximum number of invitations per sent share	100
Maximum number of share subscriptions per sent share	100
Maximum number of datasets per share	100
Maximum number of snapshot schedules per share	1

Database Migration Service Limits

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	To request an increase for this limit, contact support.

Event Grid limits

The following limits apply to Azure Event Grid system topics and custom topics, *not* event domains.

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
Publish requests	250 per second
Event size	1 MB (charged in as multiple 64-KB events)

The following limits apply to event domains only.

RESOURCE	LIMIT
Topics per event domain	100,000
Event subscriptions per topic within a domain	500
Domain scope event subscriptions	50

RESOURCE	LIMIT
Publish rate for an event domain (ingress)	5,000 events per second
Publish requests	250 per second
Event Domains per Azure Subscription	100

Event Hubs limits

The following tables provide quotas and limits specific to [Azure Event Hubs](#). For information about Event Hubs pricing, see [Event Hubs pricing](#).

The following limits are common across basic, standard, and dedicated tiers.

LIMIT	SCOPE	NOTES	VALUE
Number of Event Hubs namespaces per subscription	Subscription	-	100
Number of event hubs per namespace	Namespace	Subsequent requests for creation of a new event hub are rejected.	10
Number of partitions per event hub	Entity	-	32
Maximum size of an event hub name	Entity	-	50 characters
Number of non-epoch receivers per consumer group	Entity	-	5
Maximum throughput units	Namespace	Exceeding the throughput unit limit causes your data to be throttled and generates a server busy exception . To request a larger number of throughput units for a Standard tier, file 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 are rejected.	12
Number of calls to the GetRuntimeInformation method	Entity	-	50 per second
Number of virtual network (VNet) and IP Config rules	Entity	-	128

Event Hubs Basic and Standard - quotas and limits

LIMIT	SCOPE	NOTES	BASIC	STANDARD
Maximum size of Event Hubs event	Entity		256 KB	1 MB
Number of consumer groups per event hub	Entity		1	20
Number of AMQP connections per namespace	Namespace	Subsequent requests for additional connections are rejected, and an exception is received by the calling code.	100	5,000
Maximum retention period of event data	Entity		1 day	1-7 days
Apache Kafka enabled namespace	Namespace	Event Hubs namespace streams applications using Kafka protocol	No	Yes
Capture	Entity	When enabled, micro-batches on the same stream	No	Yes

Event Hubs Dedicated - quotas and limits

The Event Hubs Dedicated offering is billed at a fixed monthly price, with a minimum of 4 hours of usage. The Dedicated tier offers all the features of the Standard plan, but with enterprise scale capacity and limits for customers with demanding workloads.

FEATURE	LIMITS
Bandwidth	20 CUs
Namespaces	50 per CU
Event Hubs	1000 per namespace
Ingress events	Included
Message Size	1 MB
Partitions	2000 per CU
Consumer groups	No limit per CU, 1000 per event hub
Brokered connections	100 K included
Message Retention	90 days, 10 TB included per CU
Capture	Included

Identity Manager limits

CATEGORY	LIMIT
User-assigned managed identities	<ul style="list-style-type: none">When you create user-assigned managed identities, only alphanumeric characters (0-9, a-z, and A-Z) and the hyphen (-) are supported. For the assignment to a virtual machine or virtual machine scale set to work properly, the name is limited to 24 characters.If you use 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.

IoT Central limits

IoT Central limits the number of applications you can deploy in a subscription to 10. If you need to increase this limit, contact [Microsoft support](#).

IoT Hub limits

The following table lists the limits associated with the different service tiers S1, S2, S3, and F1. For information about the cost of each *unit* in each tier, see [Azure 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 tier hub 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	100
Maximum free IoT hubs per Azure subscription	1
Maximum number of characters in a device ID	128
Maximum number of device identities returned in a single call	1,000
IoT Hub message maximum retention for device-to-cloud messages	7 days
Maximum size of device-to-cloud message	256 KB

RESOURCE	LIMIT
Maximum size of device-to-cloud batch	AMQP and HTTP: 256 KB for the entire batch MQTT: 256 KB for each message
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 cloud-to-device queue depth per device	50
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	8 KB for tags section, and 32 KB for desired and reported properties sections each
Maximum length of device twin string key	1 KB
Maximum length of device twin string value	4 KB
Maximum depth of object in device twin	10
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, and S3)
Maximum message routing rules	100 (for S1, S2, and S3)
Maximum number of concurrently connected device streams	50 (for S1, S2, S3, and F1 only)
Maximum device stream data transfer	300 MB per day (for S1, S2, S3, and F1 only)

NOTE

If you need more than 100 paid IoT hubs in an Azure subscription, contact Microsoft Support.

NOTE

Currently, the total number of devices plus modules that can be registered to a single IoT hub is capped at 1,000,000. If you want to increase this limit, contact [Microsoft Support](#).

IoT Hub throttles requests when the following quotas are exceeded.

THROTTLE	PER-HUB VALUE
Identity registry operations (create, retrieve, list, update, and delete), individual or bulk import/export	83.33/sec/unit (5,000/min/unit) (for S3). 1.67/sec/unit (100/min/unit) (for S1 and S2).
Device connections	6,000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Device-to-cloud sends	6,000/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 (5,000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S1 and S2).
Cloud-to-device receives	833.33/sec/unit (50,000/min/unit) (for S3), 16.67/sec/unit (1,000/min/unit) (for S1 and S2).
File upload operations	83.33 file upload initiations/sec/unit (5,000/min/unit) (for S3), 1.67 file upload initiations/sec/unit (100/min/unit) (for S1 and S2). 10,000 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	24 MB/sec/unit (for S3), 480 KB/sec/unit (for S2), 160 KB/sec/unit (for S1). Based on 8-KB throttling meter size.
Device twin reads	500/sec/unit (for S3), Maximum of 100/sec or 10/sec/unit (for S2), 100/sec (for S1)
Device twin updates	250/sec/unit (for S3), Maximum of 50/sec or 5/sec/unit (for S2), 50/sec (for S1)
Jobs operations (create, update, list, and delete)	83.33/sec/unit (5,000/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).
Device stream initiation rate	5 new streams/sec (for S1, S2, S3, and F1 only).

IoT Hub Device Provisioning Service limits

The following table lists the limits that apply to Azure IoT Hub Device Provisioning Service resources.

RESOURCE	LIMIT
Maximum device provisioning services per Azure subscription	10
Maximum number of enrollments	1,000,000
Maximum number of registrations	1,000,000
Maximum number of enrollment groups	100
Maximum number of CAs	25
Maximum number of linked IoT hubs	50
Maximum size of message	96 KB

NOTE

To increase the number of enrollments and registrations on your provisioning service, contact [Microsoft Support](#).

NOTE

Increasing the maximum number of CAs is not supported.

The Device Provisioning Service throttles requests when the following quotas are exceeded.

THROTTLE	PER-UNIT VALUE
Operations	200/min/service
Device registrations	200/min/service
Device polling operation	5/10 sec/device

Key Vault limits

Key transactions (maximum 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 2,048-bit	5	1,000	10	2,000
RSA 3,072-bit	5	250	10	500
RSA 4,096-bit	5	125	10	250
ECC P-256	5	1,000	10	2,000
ECC P-384	5	1,000	10	2,000

KEY TYPE	HSM KEY CREATE KEY	HSM KEY ALL OTHER TRANSACTIONS	SOFTWARE KEY CREATE KEY	SOFTWARE KEY ALL OTHER TRANSACTIONS
ECC P-521	5	1,000	10	2,000
ECC SECP256K1	5	1,000	10	2,000

NOTE

In the previous table, we see that for RSA 2,048-bit software keys, 2,000 GET transactions per 10 seconds are allowed. For RSA 2,048-bit HSM-keys, 1,000 GET transactions per 10 seconds are allowed.

The throttling thresholds are weighted, and enforcement is on their sum. For example, as shown in the previous table, when you perform GET operations on RSA HSM-keys, it's eight times more expensive to use 4,096-bit keys compared to 2,048-bit keys. That's because $1,000/125 = 8$.

In a given 10-second interval, an Azure Key Vault client can do *only one* of the following operations before it encounters a 429 throttling HTTP status code:

- 2,000 RSA 2,048-bit software-key GET transactions
- 1,000 RSA 2,048-bit HSM-key GET transactions
- 125 RSA 4,096-bit HSM-key GET transactions
- 124 RSA 4,096-bit HSM-key GET transactions and 8 RSA 2,048-bit HSM-key GET transactions

Secrets, managed storage account keys, and vault transactions:

TRANSACTIONS TYPE	MAXIMUM TRANSACTIONS ALLOWED IN 10 SECONDS, PER VAULT PER REGION ¹
All transactions	2,000

For information on how to handle throttling when these limits are exceeded, see [Azure Key Vault throttling guidance](#).

¹ A subscription-wide limit for all transaction types is five times per key vault limit. For example, HSM-other transactions per subscription are limited to 5,000 transactions in 10 seconds per subscription.

Media Services limits

NOTE

For resources that aren't fixed, open a support ticket to ask for an increase in the quotas. Don't create additional Azure Media Services accounts in an attempt to obtain higher limits.

RESOURCE	DEFAULT LIMIT
Azure Media Services accounts in a single subscription	25 (fixed)
Media reserved units per Media Services account	25 (S1) 10 (S2, S3) ¹
Jobs per Media Services account	50,000 ²
Chained tasks per job	30 (fixed)

RESOURCE	DEFAULT LIMIT
Assets per Media Services 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 Media Services account	5
Programs in stopped state per channel	50
Programs in running state per channel	3
Streaming endpoints that are stopped or running per Media Services 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's a limit on the maximum file size supported for processing in Media Services. ⁷

¹If you change the type, for example, from S2 to S1, the maximum reserved unit limits are reset.

²This number includes queued, finished, active, and canceled jobs. It doesn't include deleted jobs. You can delete old jobs by using **IJob.Delete** or the **DELETE** HTTP request.

As of April 1, 2017, any job record in your account older than 90 days is automatically deleted, along with its associated task records. Automatic deletion occurs even if the total number of records is below the maximum quota. To archive the job and task information, use the code described in [Manage assets with the Media Services .NET SDK](#).

³When you make a request to list job entities, a maximum of 1,000 jobs is returned per request. To keep track of all submitted jobs, use the top or skip queries as described in [OData system query options](#).

⁴Locators aren't 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 [Protect your content with Azure Media Services](#).

⁵The storage accounts must be from the same Azure subscription.

⁶There's a limit of 1,000,000 policies for different Media Services policies. An example is for the Locator policy or ContentKeyAuthorizationPolicy.

NOTE

If you always use the same days and access permissions, use the same policy ID. For information and an example, see [Manage assets with the Media Services .NET SDK](#).

⁷The maximum size supported for a single blob is currently up to 5 TB in Azure Blob Storage. Additional limits apply in Media Services based on the VM sizes that are used by the service. The size limit applies to the files that you upload and also the files that get generated as a result of Media Services processing (encoding or analyzing). If your source file is larger than 260-GB, your Job will likely fail.

The following table shows the limits on the media reserved units S1, S2, and S3. If your source file is larger than the limits defined in the table, your encoding job fails. If you encode 4K resolution sources of long duration, you're required to use S3 media reserved units to achieve the performance needed. If you have 4K content that's larger than the 260-GB limit on the S3 media reserved units, open a support ticket.

MEDIA RESERVED UNIT TYPE	MAXIMUM INPUT SIZE (GB)
S1	26
S2	60
S3	260

Mobile Services limits

TIER	FREE	BASIC	STANDARD
API calls	500,000	1.5 million per unit	15 million per unit
Active devices	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Unlimited units
Push notifications	Azure Notification Hubs Free tier included, up to 1 million pushes	Notification Hubs Basic tier included, up to 10 million pushes	Notification Hubs Standard tier included, up to 10 million pushes
Real-time messaging/ Web Sockets	Limited	350 per mobile service	Unlimited
Offline synchronizations	Limited	Included	Included
Scheduled jobs	Limited	Included	Included
Azure SQL Database (required) Standard rates apply for additional capacity	20 MB included	20 MB included	20 MB included
CPU capacity	60 minutes per day	Unlimited	Unlimited
Outbound data transfer	165 MB per day (daily rollover)	Included	Included

For more information on limits and pricing, see [Azure Mobile Services pricing](#).

Multi-Factor Authentication limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of trusted IP addresses or ranges per subscription	0	50
Remember my devices, number of days	14	60
Maximum number of app passwords	0	No limit
Allow X attempts during MFA call	1	99
Two-way text message timeout seconds	60	600
Default one-time bypass seconds	300	1,800
Lock user account after X consecutive MFA denials	Not set	99
Reset account lockout counter after X minutes	Not set	9,999
Unlock account after X minutes	Not set	9,999

Networking limits

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's no maximum limit column, the resource doesn't have adjustable limits. If you had these limits increased by support in the past and don't see updated limits in the following tables, [open an online customer support request at no charge](#)

RESOURCE	DEFAULT/MAXIMUM LIMIT
Virtual networks	1,000
Subnets per virtual network	3,000
Virtual network peerings per virtual network	500
Virtual network gateways (VPN gateways) per virtual network	1
Virtual network gateways (ExpressRoute gateways) per virtual network	1
DNS servers per virtual network	20
Private IP addresses per virtual network	65,536

RESOURCE	DEFAULT/MAXIMUM LIMIT
Private IP addresses per network interface	256
Private IP addresses per virtual machine	256
Public IP addresses per network interface	256
Public IP addresses per virtual machine	256
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000
Network interface cards	65,536
Network Security Groups	5,000
NSG rules per NSG	1,000
IP addresses and ranges specified for source or destination in a security group	4,000
Application security groups	3,000
Application security groups per IP configuration, per NIC	20
IP configurations per application security group	4,000
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 Azure VPN Gateway	20
Virtual network TAPs	100
Network interface TAP configurations per virtual network TAP	100

Public IP address limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP addresses ¹	10 for Basic.	Contact support.
Static Public IP addresses ¹	10 for Basic.	Contact support.
Standard Public IP addresses ¹	10	Contact support.
Public IP Prefixes	limited by number of Standard Public IPs in a subscription	Contact support.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP prefix length	/28	Contact support.

¹Default limits for Public IP addresses vary by offer category type, such as Free Trial, Pay-As-You-Go, CSP. For example, the default for Enterprise Agreement subscriptions is 1000.

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](#).

Standard Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	1,500
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	600
Backend pool size	1,000 IP configurations, single virtual network
High-availability ports	1 per internal frontend
Outbound rules per Load Balancer	20

Basic Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	250
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	200
Backend pool size	300 IP configurations, single availability set
Availability sets per Load Balancer	150

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	100	100
Local network sites	20	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
DNS servers per virtual network	20	20
Private IP addresses per virtual network	4,096	4,096
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000, up to 1,000,000 for two or more NICs.	500,000, up to 1,000,000 for two or more NICs.
Network Security Groups (NSGs)	200	200
NSG rules per NSG	1,000	1,000
User-defined route tables	200	200
User-defined routes per route table	400	400
Public IP addresses (dynamic)	500	500
Reserved public IP addresses	500	500
Public VIP per deployment	5	Contact support
Private VIP (internal load balancing) per deployment	1	1
Endpoint access control lists (ACLs)	50	50

ExpressRoute limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription, with Azure Resource Manager	10
Maximum number of routes advertised to Azure private peering with ExpressRoute Standard	4,000
Maximum number of routes advertised to Azure private peering with ExpressRoute Premium add-on	10,000
Maximum number of routes advertised from Azure private peering from the VNet address space for an ExpressRoute connection	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Standard	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Premium add-on	200

RESOURCE	DEFAULT/MAXIMUM LIMIT
Maximum number of ExpressRoute circuits linked to the same virtual network in the same peering location	4
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 the Number of virtual networks per ExpressRoute circuit table.

Number of virtual networks per ExpressRoute circuit

CIRCUIT SIZE	NUMBER OF VIRTUAL NETWORK LINKS FOR STANDARD	NUMBER OF VIRTUAL NETWORK LINKS WITH PREMIUM ADD-ON
50 Mbps	10	20
100 Mbps	10	25
200 Mbps	10	25
500 Mbps	10	40
1 Gbps	10	50
2 Gbps	10	60
5 Gbps	10	75
10 Gbps	10	100
40 Gbps*	10	100
100 Gbps*	10	100

*100 Gbps ExpressRoute Direct Only

NOTE

Global Reach connections count against the limit of virtual network connections per ExpressRoute Circuit. For example, a 10 Gbps Premium Circuit would allow for 5 Global Reach connections and 95 connections to the ExpressRoute Gateways or 95 Global Reach connections and 5 connections to the ExpressRoute Gateways or any other combination up to the limit of 100 connections for the circuit.

Virtual WAN limits

RESOURCE	LIMIT
Virtual WAN hubs per region	1
Virtual WAN hubs per virtual wan	Azure regions
VPN (branch) connections per hub	1,000

RESOURCE	LIMIT
VNet connections per hub	500
Point-to-Site users per hub	10,000
Aggregate throughput per Virtual WAN VPN gateway	20 Gbps
Throughput per Virtual WAN VPN connection (2 tunnels)	2 Gbps with 1 Gbps/IPsec tunnel
Aggregate throughput per Virtual WAN ExpressRoute gateway	20 Gbps

Application Gateway limits

The following table applies to v1, v2, Standard, and WAF SKUs unless otherwise stated.

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Azure Application Gateway	1,000 per subscription	
Front-end IP configurations	2	1 public and 1 private
Front-end ports	100 ¹	
Back-end address pools	100 ¹	
Back-end servers per pool	1,200	
HTTP listeners	100 ¹	
HTTP load-balancing rules	100 ¹	
Back-end HTTP settings	100 ¹	
Instances per gateway	V1 SKU - 32 V2 SKU - 125	
SSL certificates	100 ¹	1 per HTTP listener
Maximum SSL certificate size	V1 SKU - 10 KB V2 SKU - 16 KB	
Authentication certificates	100	
Trusted root certificates	100	
Request timeout minimum	1 second	
Request timeout maximum	24 hours	
Number of sites	100 ¹	1 per HTTP listener
URL maps per listener	1	

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Maximum path-based rules per URL map	100	
Redirect configurations	100 ¹	
Concurrent WebSocket connections	Medium gateways 20k Large gateways 50k	
Maximum URL length	32KB	
Maximum header size for HTTP/2	4KB	
Maximum file upload size, Standard	2 GB	
Maximum file upload size WAF	V1 Medium WAF gateways, 100 MB V1 Large WAF gateways, 500 MB V2 WAF, 750 MB	
WAF body size limit, without files	128 KB	
Maximum WAF custom rules	100	
Maximum WAF exclusions	100	

¹ In case of WAF-enabled SKUs, we recommend that you limit the number of resources to 40 for optimal performance.

Network Watcher limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT	NOTE
Azure Network Watcher	1 per region	1 per region	Network Watcher is created to enable access to the service. Only one instance of Network Watcher is required per subscription per region.
Packet capture sessions	10,000 per region	10,000	Number of sessions only, not saved captures.

Private Link limits

The following limits apply to Azure private link:

RESOURCE	LIMIT
Number of private endpoints per virtual network	1000
Number of private endpoints per subscription	64000
Number of private link service per subscription	800
Number of IP Configurations on a private link service	8 (This number is for the NAT IP addresses used per PLS)

RESOURCE	LIMIT
Number of private endpoints on the same private link service	1000

Traffic Manager limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Profiles per subscription	200
Endpoints per profile	200

Azure Bastion limits

RESOURCE	DEFAULT LIMIT
Concurrent RDP connections	25*
Concurrent SSH connections	More than 50**

*May vary due to other on-going RDP sessions or other on-going SSH sessions.

**May vary if there are existing RDP connections or usage from other on-going SSH sessions.

Azure DNS limits

Public DNS zones

RESOURCE	DEFAULT LIMIT
Public DNS Zones per subscription	250 ¹
Record sets per public DNS zone	10,000 ¹
Records per record set in public DNS zone	20
Number of Alias records for a single Azure resource	20
Private DNS zones per subscription	1000
Record sets per private DNS zone	25000
Records per record set for private DNS zones	20
Virtual Network Links per private DNS zone	1000
Virtual Networks Links per private DNS zones with auto-registration enabled	100
Number of private DNS zones a virtual network can get linked to with auto-registration enabled	1
Number of private DNS zones a virtual network can get linked	1000

RESOURCE	DEFAULT LIMIT
Number of DNS queries a virtual machine can send to Azure DNS resolver, per second	500 ²
Maximum number of DNS queries queued (pending response) per virtual machine	200 ²

¹If you need to increase these limits, contact Azure Support.

²These limits are applied to every individual virtual machine and not at the virtual network level. DNS queries exceeding these limits are dropped.

Azure Firewall limits

RESOURCE	DEFAULT LIMIT
Data throughput	30 Gbps ¹
Rules	10,000. All rule types combined.
Maximum DNAT rules	299
Minimum AzureFirewallSubnet size	/26
Port range in network and application rules	0-64,000. Work is in progress to relax this limitation.
Public IP addresses	100 maximum (Currently, SNAT ports are added only for the first five public IP addresses.)
Route table	<p>By default, AzureFirewallSubnet has a 0.0.0.0/0 route with the NextHopType value set to Internet.</p> <p>Azure Firewall must have direct Internet connectivity. If your AzureFirewallSubnet learns a default route to your on-premises network via BGP, you must override that with a 0.0.0.0/0 UDR with the NextHopType value set as Internet to maintain direct Internet connectivity. By default, Azure Firewall doesn't support forced tunneling to an on-premises network.</p> <p>However, if your configuration requires forced tunneling to an on-premises network, Microsoft will support it on a case by case basis. Contact Support so that we can review your case. If accepted, we'll allow your subscription and ensure the required firewall Internet connectivity is maintained.</p>

¹If you need to increase these limits, contact Azure Support.

Azure Front Door Service limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Azure Front Door Service resources per subscription	100
Front-end hosts, which includes custom domains per resource	100

RESOURCE	DEFAULT/MAXIMUM LIMIT
Routing rules per resource	100
Back-end pools per resource	50
Back ends per back-end pool	100
Path patterns to match for a routing rule	25
Custom web application firewall rules per policy	10
Web application firewall policy per subscription	100
Web application firewall match conditions per custom rule	10
Web application firewall IP address ranges per match condition	600
Web application firewall string match values per match condition	10
Web application firewall string match value length	256
Web application firewall POST body parameter name length	256
Web application firewall HTTP header name length	256
Web application firewall cookie name length	256
Web application firewall HTTP request body size inspected	128 KB
Web application firewall custom response body length	2 KB

Timeout values

Client to Front Door

- Front Door has an idle TCP connection timeout of 61 seconds.

Front Door to application back-end

- If the response is a chunked response, a 200 is returned if or when the first chunk is received.
- After the HTTP request is forwarded to the back end, Front Door waits for 30 seconds for the first packet from the back end. Then it returns a 503 error to the client.
- After the first packet is received from the back end, Front Door waits for 30 seconds in an idle timeout. Then it returns a 503 error to the client.
- Front Door to the back-end TCP session timeout is 30 minutes.

Upload and download data limit

	WITH CHUNKED TRANSFER ENCODING (CTE)	WITHOUT HTTP CHUNKING
Download	There's no limit on the download size.	There's no limit on the download size.

	WITH CHUNKED TRANSFER ENCODING (CTE)	WITHOUT HTTP CHUNKING
Upload	There's no limit as long as each CTE upload is less than 2 GB.	The size can't be larger than 2 GB.

Other limits

- Maximum URL size - 8,192 bytes - Specifies maximum length of the raw URL (scheme + hostname + port + path + query string of the URL)
- Maximum Query String size - 4,096 bytes - Specifies the maximum length of the query string, in bytes.

Notification Hubs limits

TIER	FREE	BASIC	STANDARD
Included pushes	1 million	10 million	10 million
Active devices	500	200,000	10 million
Tag quota per installation or registration	60	60	60

For more information on limits and pricing, see [Notification Hubs pricing](#).

Role-based access control limits

RESOURCE	LIMIT
Role assignments for Azure resources per Azure subscription	2,000
Role assignments for Azure resources per management group	500
Custom roles for Azure resources per tenant	5,000
Custom roles for Azure resources per tenant (specialized clouds, such as Azure Government, Azure Germany, and Azure China 21Vianet)	2,000

Service Bus limits

The following table lists quota information specific to Azure Service Bus messaging. For information about pricing and other quotas for Service Bus, see [Service Bus pricing](#).

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum number of Basic or Standard namespaces per Azure subscription	Namespace	Subsequent requests for additional Basic or Standard namespaces are rejected by the Azure 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.	100
Queue or topic size	Entity	Defined upon creation of the queue or 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, and the Standard SKU with partitioning enabled, the maximum queue or 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 don't count toward concurrent TCP connections.	NetMessaging: 1,000. AMQP: 5,000.
Number of concurrent receive requests on a queue, topic, or 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 or 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 for the Basic or Standard tier. The total number of topics and queues in a namespace must be less than or equal to 10,000. For the Premium tier, 1,000 per messaging unit (MU). Maximum limit is 4,000.
Number of partitioned topics or 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, the exception QuotaExceededException is received by the calling code.	Basic and Standard tiers: 100. Partitioned entities aren't supported in the Premium tier. Each partitioned queue or topic counts toward the quota of 1,000 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 ID	Entity	-	128
Maximum size of a message session ID	Entity	-	128
Message size for a queue, topic, or subscription entity	Entity	<p>Incoming messages that exceed these quotas are rejected, and an exception is received by the calling code.</p>	<p>Maximum message size: 256 KB for Standard tier, 1 MB for Premium tier.</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>
Message property size for a queue, topic, or subscription entity	Entity	The exception SerializationException is generated.	Maximum message property size for each property is 32,000. Cumulative size of all properties can't exceed 64,000. This limit applies to the entire header of the BrokeredMessage , which has both user properties and system properties, such as SequenceNumber , Label , and MessageId .
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 per-topic for the Standard tier.
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 or 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: 1,024 (1 K). Maximum length of rule action string: 1,024 (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 per entity type: 12. Rules that are configured on a Service Bus namespace apply to all types: queues, topics.
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.
Number of virtual network and IP filter rules	Namespace		128

Site Recovery limits

The following limits apply to Azure Site Recovery.

Limit Identifier	Default Limit
Number of vaults per subscription	500
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

SQL Database limits

For SQL Database limits, see [SQL Database resource limits for single databases](#), [SQL Database resource limits for elastic pools and pooled databases](#), and [SQL Database resource limits for managed instances](#).

SQL Data Warehouse limits

For SQL Data Warehouse limits, see [SQL Data Warehouse resource limits](#).

Storage limits

The following table describes default limits for Azure general-purpose v1, v2, and Blob storage accounts. The *ingress* limit refers to all data from requests that are sent to a storage account. The *egress* limit refers to all data from responses that are received from a storage account.

RESOURCE	DEFAULT LIMIT
Number of storage accounts per region per subscription, including both standard and premium accounts	250
Maximum storage account capacity	2 PiB for US and Europe, and 500 TiB for all other regions (including the UK) ¹
Maximum number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate ¹ per storage account	20,000 requests per second
Maximum ingress ¹ per storage account (US, Europe regions)	25 Gbps
Maximum ingress ¹ per storage account (regions other than US and Europe)	5 Gbps if RA-GRS/GRS is enabled, 10 Gbps for LRS/ZRS ²
Maximum egress for general-purpose v2 and Blob storage accounts (all regions)	50 Gbps
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS ²
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS ²
Maximum number of virtual network rules per storage account	200
Maximum number of IP address rules per storage account	200

¹Azure Storage standard accounts support higher capacity limits and higher limits for ingress by request. To request an increase in account limits for ingress, contact [Azure Support](#). For more information, see [Announcing larger, higher scale storage accounts](#).

² If your storage account has read-access enabled with geo-redundant storage (RA-GRS) or geo-zone-redundant storage (RA-GZRS), then the egress targets for the secondary location are identical to those of the primary location. [Azure Storage replication](#) options include:

- [Locally redundant storage \(LRS\)](#)
- [Zone-redundant storage \(ZRS\)](#)

- [Geo-redundant storage \(GRS\)](#)
- [Read-access geo-redundant storage \(RA-GRS\)](#)
- [Geo-zone-redundant storage \(GZRS\)](#)
- [Read-access geo-zone-redundant storage \(RA-GZRS\)](#)

NOTE

Microsoft recommends that you use a general-purpose v2 storage account for most scenarios. You can easily upgrade a general-purpose v1 or an Azure Blob storage account to a general-purpose v2 account with no downtime and without the need to copy data. For more information, see [Upgrade to a general-purpose v2 storage account](#).

If the needs of your application exceed the scalability targets of a single storage account, you can build your application to use multiple storage accounts. You can then partition your data objects across those storage accounts. For information on volume pricing, see [Azure Storage pricing](#).

All storage accounts run on a flat network topology and support the scalability and performance targets outlined in this article, regardless of when they were created. For more information on the Azure Storage flat network architecture and on scalability, see [Microsoft Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency](#).

For more information on limits for standard storage accounts, see [Scalability targets for standard storage accounts](#).

Storage resource provider limits

The following limits apply only when you perform management operations by using Azure Resource Manager with Azure Storage.

RESOURCE	DEFAULT LIMIT
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	1200 per hour
Storage account management operations (list)	100 per 5 minutes

Azure Blob storage limits

RESOURCE	TARGET
Maximum size of single blob container	Same as maximum storage account capacity
Maximum number of blocks in a block blob or append blob	50,000 blocks
Maximum size of a block in a block blob	100 MiB
Maximum size of a block blob	50,000 X 100 MiB (approximately 4.75 TiB)
Maximum size of a block in an append blob	4 MiB
Maximum size of an append blob	50,000 x 4 MiB (approximately 195 GiB)
Maximum size of a page blob	8 TiB
Maximum number of stored access policies per blob container	5

RESOURCE	TARGET
Target request rate for a single blob	Up to 500 requests per second
Target throughput for a single page blob	Up to 60 MiB per second
Target throughput for a single block blob	Up to storage account ingress/egress limits ¹

¹ Throughput for a single blob depends on several factors, including, but not limited to: concurrency, request size, performance tier, speed of source for uploads, and destination for downloads. To take advantage of the performance enhancements of [high-throughput block blobs](#), upload larger blobs or blocks. Specifically, call the [Put Blob](#) or [Put Block](#) operation with a blob or block size that is greater than 4 MiB for standard storage accounts. For premium block blob or for Data Lake Storage Gen2 storage accounts, use a block or blob size that is greater than 256 KiB.

Azure Files limits

For more information on Azure Files limits, see [Azure Files scalability and performance targets](#).

RESOURCE	STANDARD FILE SHARES	PREMIUM FILE SHARES
Minimum size of a file share	No minimum; pay as you go	100 GiB; provisioned
Maximum size of a file share	100 TiB*, 5 TiB	100 TiB
Maximum size of a file in a file share	1 TiB	1 TiB
Maximum number of files in a file share	No limit	No limit
Maximum IOPS per share	10,000 IOPS*, 1,000 IOPS	100,000 IOPS
Maximum number of stored access policies per file share	5	5
Target throughput for a single file share	up to 300 MiB/sec*, Up to 60 MiB/sec ,	See premium file share ingress and egress values
Maximum egress for a single file share	See standard file share target throughput	Up to 6,204 MiB/s
Maximum ingress for a single file share	See standard file share target throughput	Up to 4,136 MiB/s
Maximum open handles per file	2,000 open handles	2,000 open handles
Maximum number of share snapshots	200 share snapshots	200 share snapshots
Maximum object (directories and files) name length	2,048 characters	2,048 characters
Maximum pathname component (in the path \A\B\C\D, each letter is a component)	255 characters	255 characters

* Available in most regions, see [Regional availability](#) for the details on available regions.

Azure File Sync limits

RESOURCE	TARGET	HARD LIMIT
Storage Sync Services per region	20 Storage Sync Services	Yes
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	30 server endpoints	Yes
File system objects (directories and files) per sync group	100 million objects	No
Maximum number of file system objects (directories and files) in a directory	5 million objects	Yes
Maximum object (directories and files) security descriptor size	64 KiB	Yes
File size	100 GiB	No
Minimum file size for a file to be tiered	V9: Based on file system cluster size (double file system cluster size). For example, if the file system cluster size is 4kb, the minimum file size will be 8kb. V8 and older: 64 KiB	Yes

NOTE

An Azure File Sync endpoint can scale up to the size of an Azure file share. If the Azure file share size limit is reached, sync will not be able to operate.

Azure Queue storage limits

RESOURCE	TARGET
Maximum size of a single queue	500 TiB
Maximum size of a message in a queue	64 KiB
Maximum number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second, which assumes a 1-KiB message size
Target throughput for a single queue (1-KiB messages)	Up to 2,000 messages per second

Azure Table storage limits

RESOURCE	TARGET
Maximum size of a single table	500 TiB
Maximum size of a table entity	1 MiB
Maximum number of properties in a table entity	255, which includes three system properties: PartitionKey, RowKey, and Timestamp
Maximum total size of property values in an entity	1 MiB
Maximum total size of an individual property in an entity	Varies by property type. For more information, see Property Types in Understanding the Table Service Data Model .
Maximum number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second, which assumes a 1-KiB entity size
Target throughput for a single table partition (1 KiB-entities)	Up to 2,000 entities per second

Virtual machine disk limits

You can attach a number of data disks to an Azure virtual machine. Based on the scalability and performance targets for a VM's data disks, you can determine 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 aren't highly utilized at the same time, the virtual machine can support a larger number of disks.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. There is no limit for the number of Managed Disks, snapshots and images per resource group.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Standard managed disks	50,000	50,000
Standard SSD managed disks	50,000	50,000
Premium managed disks	50,000	50,000
Standard_LRS snapshots	50,000	50,000
Standard_ZRS snapshots	50,000	50,000
Managed image	50,000	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, which is $20,000/300$ IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is $20,000/500$ 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.

For more information, see [Virtual machine sizes](#).

Managed virtual machine disks

Standard HDD managed disks

STAND ARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000							
Throughput per disk	Up to 60 MiB/sec	Up to 300 MiB/sec	Up to 500 MiB/sec	Up to 500 MiB/sec							

Standard SSD managed disks

STA NDAR SSD SIZE S	E1*	E2*	E3*	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 120	Up to 120	Up to 120	Up to 120	Up to 240	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000					
Throughput per disk	Up to 25 MiB/sec	Up to 50 MiB/sec	Up to 60 MiB/sec	Up to 400 MiB/sec	Up to 600 MiB/sec	Up to 750 MiB/sec								

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

Premium SSD managed disks: Per-disk limits

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOP S per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
Throughput per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max burst IOP S per disk **	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500
Max burst throughput per disk **	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec								
Max burst duration**	30 min	30 min	30 min	30 min	30 min	30 min								
Eligible for reservation	No	Yes, up to one year												

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

Premium SSD managed disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum 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	4,095 GB	4,095 GB
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum 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
Maximum bandwidth per account (ingress + egress) ¹	<=50 Gbps

¹Ingress refers to all data from requests that are sent to a storage account. Egress refers to all data from responses that are 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	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

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.
Maximum size of a tiered volume on physical devices	64 TB for StorSimple 8100 and StorSimple 8600	StorSimple 8100 and StorSimple 8600 are physical devices.
Maximum size of a tiered volume on virtual devices in Azure	30 TB for StorSimple 8010 64 TB for StorSimple 8020	StorSimple 8010 and StorSimple 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 StorSimple 8100 24 TB for StorSimple 8600	StorSimple 8100 and StorSimple 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 amount 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're processed sequentially as processing slots become available. New backups of a cloned or a restored tiered volume can't occur until the operation is finished. 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 a restore or clone operation, regardless of the volume size. The volume performance might initially be slower than normal as most of the data and metadata still resides in the cloud. Performance might 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 might 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 can't 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 might initially be slower than normal as most of the data and metadata still resides in the cloud. Performance might 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 might be affected by Internet bandwidth to the cloud. Unlike tiered volumes, if there are 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 might 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/sec with a single 10-gigabit Ethernet network interface	Up to two times with MPIO and two network interfaces.
Maximum client read/write throughput, when served from the HDD tier*	120/250 MB/sec	
Maximum client read/write throughput, when served from the cloud tier*	11/41 MB/sec	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 might be lower and depends on I/O mix and network conditions.

Stream Analytics limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of streaming units per subscription per region	500	To request an increase in streaming units for your subscription beyond 500, contact Microsoft Support .
Maximum number of inputs per job	60	There's a hard limit of 60 inputs per Azure Stream Analytics job.
Maximum number of outputs per job	60	There's a hard limit of 60 outputs per Stream Analytics job.
Maximum number of functions per job	60	There's a hard limit of 60 functions per Stream Analytics job.
Maximum number of streaming units per job	192	There's a hard limit of 192 streaming units per Stream Analytics job.
Maximum number of jobs per region	1,500	Each subscription can have up to 1,500 jobs per geographical region.
Reference data blob MB	300	Reference data blobs can't be larger than 300 MB each.

Virtual Machines limits

Virtual Machines limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual machines per cloud service ¹	50	50
Input endpoints per cloud service ²	150	150

¹Virtual machines created by using the classic deployment model instead of Azure 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. For more information, see [How to set up endpoints to a virtual machine](#).

Virtual Machines limits - Azure Resource Manager

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT
VMs per subscription	25,000 ¹ per region.
VM total cores per subscription	20 ¹ per region. Contact support to increase limit.
Azure Spot VM total cores per subscription	20 ¹ per region. Contact support to increase limit.
VM per series, such as Dv2 and F, cores per subscription	20 ¹ per region. Contact support to increase limit.

RESOURCE	DEFAULT LIMIT
Availability sets per subscription	2,000 per region.
Virtual machines per availability set	200
Certificates per subscription	Unlimited ²

¹Default limits vary by offer category type, such as Free Trial and Pay-As-You-Go, and by series, such as Dv2, F, and G. For example, the default for Enterprise Agreement subscriptions is 350.

²With Azure Resource Manager, certificates are stored in the Azure Key Vault. The number of certificates is unlimited for a subscription. There's a 1-MB limit of certificates per deployment, which consists of either a single VM or an availability set.

NOTE

Virtual machine cores have a regional total limit. They also have a limit for regional per-size series, such as Dv2 and F. These limits 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 can deploy 30 A1 VMs, or 30 D1 VMs, or a combination of the two not to exceed a total of 30 cores. An example of a combination is 10 A1 VMs and 20 D1 VMs.

Shared Image Gallery limits

There are limits, per subscription, for deploying resources using Shared Image Galleries:

- 100 shared image galleries, per subscription, per region
- 1,000 image definitions, per subscription, per region
- 10,000 image versions, per subscription, per region

Virtual machine scale sets limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of VMs in a scale set	1,000	1,000
Maximum number of VMs based on a custom VM image in a scale set	600	600
Maximum number of scale sets in a region	2,000	2,000

See also

- [Understand Azure limits and increases](#)
- [Virtual machine and cloud service sizes for Azure](#)
- [Sizes for Azure Cloud Services](#)
- [Naming rules and restrictions for Azure resources](#)