

Table of Contents

Overview

[About VPN Gateway](#)

[VPN Gateway FAQ](#)

[Subscription and service limits](#)

Get Started

[Planning and design for VPN Gateway](#)

[About VPN Gateway settings](#)

[About VPN devices for Site-to-Site VPN Gateway connections](#)

[About BGP and VPN Gateway](#)

[About highly available connections](#)

How To

Site-to-Site

[Configure a Site-to-Site connection using the Azure portal](#)

[Configure a Site-to-Site connection using PowerShell](#)

[Configure a Site-to-Site connection using the Azure classic portal \(Classic\)](#)

Point-to-Site

[Configure a Point-to-Site connection to a VNet using the Azure portal](#)

[Configure a Point-to-Site connection to a VNet using PowerShell](#)

[Configure a Point-to-Site connection to a VNet using the Azure portal \(Classic\)](#)

[Configure a Point-to-Site connection to a VNet using the classic portal \(Classic\)](#)

VNet-to-VNet

[Configure a VNet-to-VNet connection using the Azure portal](#)

[Configure a VNet-to-VNet connection using PowerShell](#)

[Configure a VNet-to-VNet connection \(Classic\)](#)

VNet-to-VNet connections between the Resource Manager and classic deployment models

[Connect virtual networks from different deployment models in the portal](#)

[Connect virtual networks from different deployment models using PowerShell](#)

Site-to-Site and ExpressRoute coexisting connections

Create Site-to-Site and ExpressRoute coexisting connections

Create Site-to-Site and ExpressRoute coexisting connections (Classic)

Forced tunneling

Configure forced tunneling

Configure forced tunneling (Classic)

Multiple Site-to-Site connections

Add multiple Site-to-Site connections to a VPN gateway

Add multiple Site-to-Site connections to a VPN gateway (Classic)

Configure BGP for Azure VPN Gateways using PowerShell

Configure highly available active-active connections

Modify local network gateway settings using PowerShell

Verify a gateway connection

Reset an Azure VPN Gateway using PowerShell

How to work with self-signed certificates for Point-to-Site connections

Configure a VPN gateway (Classic)

Reference

PowerShell (resource manager)

PowerShell (classic)

REST (resource manager)

REST (classic)

Related

Virtual Machines

Application Gateway

Azure DNS

Traffic Manager

Load Balancer

VPN Gateway

ExpressRoute

Resources

Blog

Forum

Pricing

SLA

Videos

About VPN Gateway

1/17/2017 • 9 min to read • [Edit on GitHub](#)

A virtual network gateway is used to send network traffic between Azure virtual networks and on-premises locations and also between virtual networks within Azure (VNet-to-VNet). When you configure a VPN gateway, you must create and configure a virtual network gateway and a virtual network gateway connection.

In the Resource Manager deployment model, when you create a virtual network gateway resource, you specify several settings. One of the required settings is '-GatewayType'. There are two virtual network gateway types: Vpn and ExpressRoute.

When network traffic is sent on a dedicated private connection, you use the gateway type 'ExpressRoute'. This is also referred to as an ExpressRoute gateway. When network traffic is sent encrypted across a public connection, you use the gateway type 'Vpn'. This is referred to as a VPN gateway. Site-to-Site, Point-to-Site, and VNet-to-VNet connections all use a VPN gateway.

Each virtual network can have only one virtual network gateway per gateway type. For example, you can have one virtual network gateway that uses -GatewayType ExpressRoute, and one that uses -GatewayType Vpn. This article focuses primarily on VPN Gateway. For more information about ExpressRoute, see the [ExpressRoute Technical Overview](#).

Pricing

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

Virtual network gateway compute costs

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

Data transfer costs

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

- If you are sending traffic to your on-premises VPN device, it will be charged with the Internet egress data transfer rate.
- If you are sending traffic between virtual networks in different regions, the pricing is based 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.

Gateway SKUs

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. When you select a higher gateway SKU, more CPUs and network bandwidth are allocated to the gateway, and as a result, the gateway can support higher network throughput to the virtual network.

VPN Gateway can use the following SKUs:

- Basic
- Standard
- HighPerformance

VPN Gateway does not use the UltraPerformance gateway SKU. For information about the UltraPerformance SKU, see the [ExpressRoute](#) documentation.

When selecting a SKU, 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.

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

Estimated aggregate throughput by SKU

The following table shows the gateway types and the estimated aggregate throughput by gateway SKU. This table applies to both 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)	100 Mbps	10	500 Mbps	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.

Configuring a VPN Gateway

When you configure a VPN gateway, the instructions you use 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](#).

A VPN gateway connection relies on multiple resources that are configured with specific settings. Most of the resources can be configured separately, although they must be configured in a certain order in some cases. You can start out creating and configuring resources using one configuration tool, such as the Azure portal. You can then later decide to switch to another tool, such as PowerShell, to configure additional resources, or to 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. For information about individual resources and settings for VPN Gateway, see [About VPN Gateway settings](#).

The following sections contain tables that list:

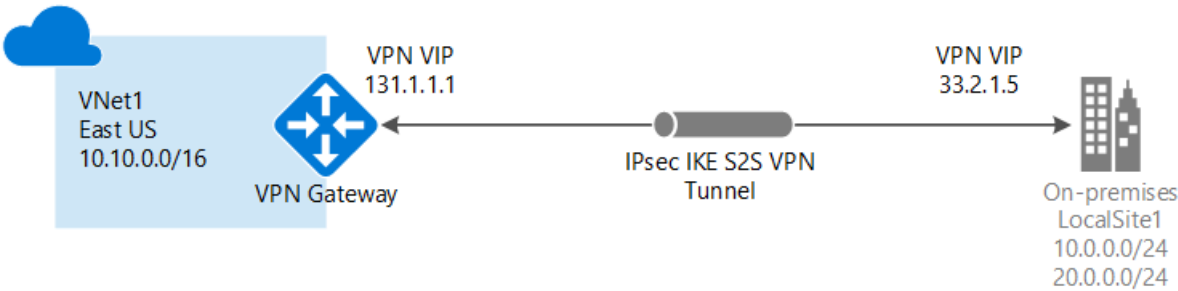
- available deployment model
- available configuration tools
- links that take you directly to an article, if available

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

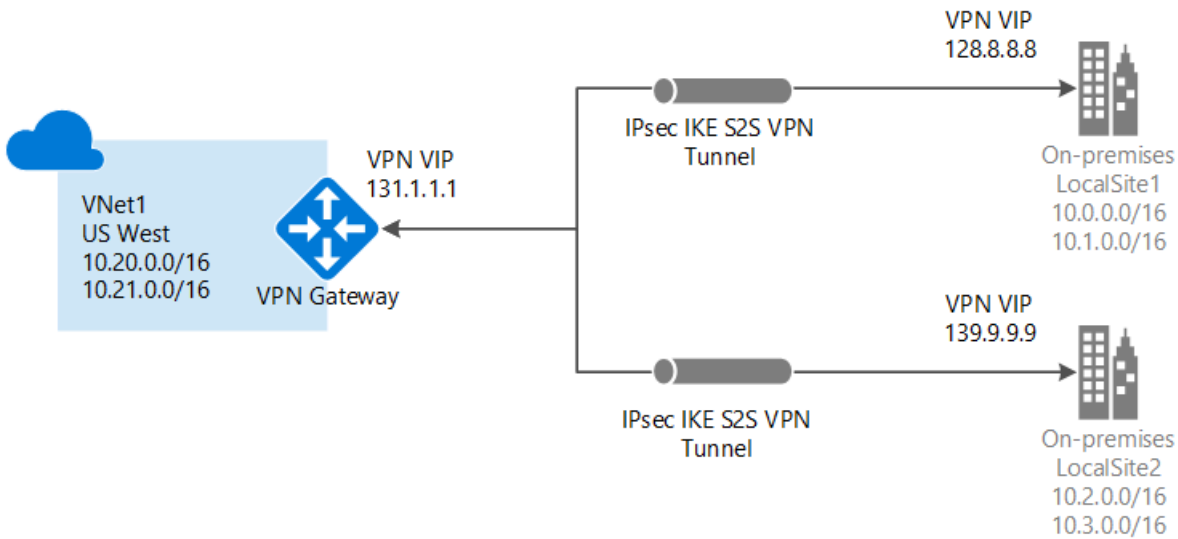
Site-to-Site

A Site-to-Site (S2S) VPN gateway connection is a connection over IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has a public IP address assigned to it and is not located behind a NAT. S2S connections can be used for cross-premises and hybrid configurations.



Multi-Site

You can create and configure a VPN gateway connection between your VNet and multiple on-premises networks. When working with multiple connections, you must use a RouteBased VPN type (dynamic gateway for classic VNets). Because a VNet can only have one VPN gateway, all connections through the gateway share the available bandwidth. This is often called a "multi-site" connection.



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

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

(*) denotes that the classic portal can only support creating one S2S VPN connection.

(**) denotes that an end-to-end scenario is not yet available for the Azure portal.

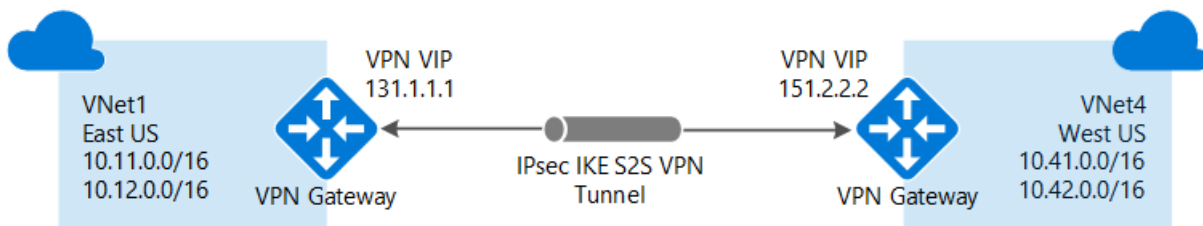
(+) denotes that this article is written for multi-site connections.

VNet-to-VNet

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 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

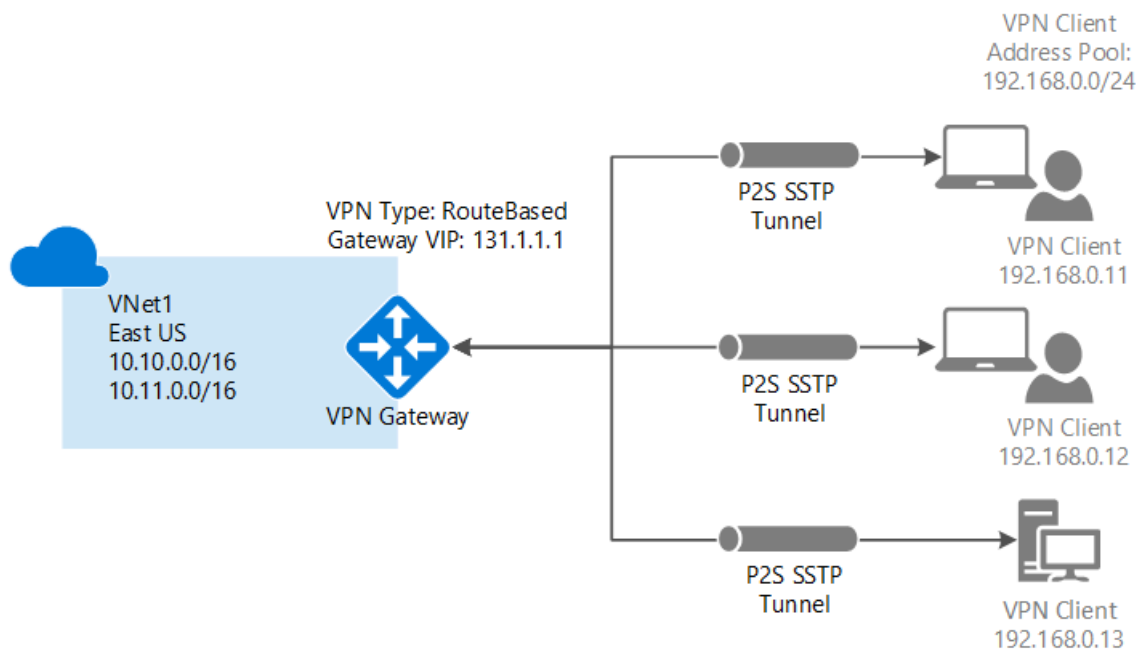
DEPLOYMENT MODEL/METHOD	AZURE PORTAL	CLASSIC PORTAL	POWERSHELL
Classic	Not Supported	Article*	Supported
Resource Manager	Article+	Not Supported	Article
Connections between different deployment models	Article*	Article*	Article

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

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

Point-to-Site

A Point-to-Site (P2S) VPN gateway connection allows you to create a secure connection to your virtual network from an individual client computer. P2S is a VPN connection over SSTP (Secure Socket Tunneling Protocol). P2S connections do not require a VPN device or a public-facing IP address to work. You establish the VPN connection by starting it from the client computer. This solution is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a VNet. P2S connections can be used in conjunction with S2S connections through the same VPN gateway, provided that all of the configuration requirements for both connections are compatible.

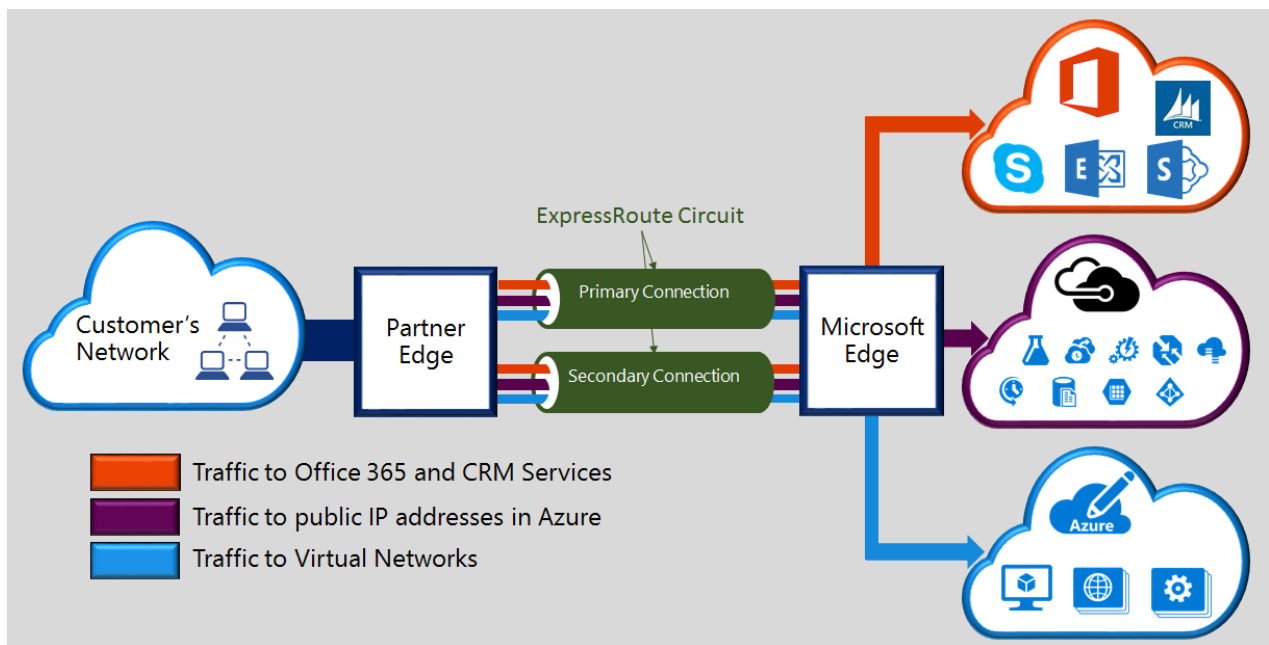


Deployment models and methods for Point-to-Site

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

ExpressRoute

Microsoft Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, 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.

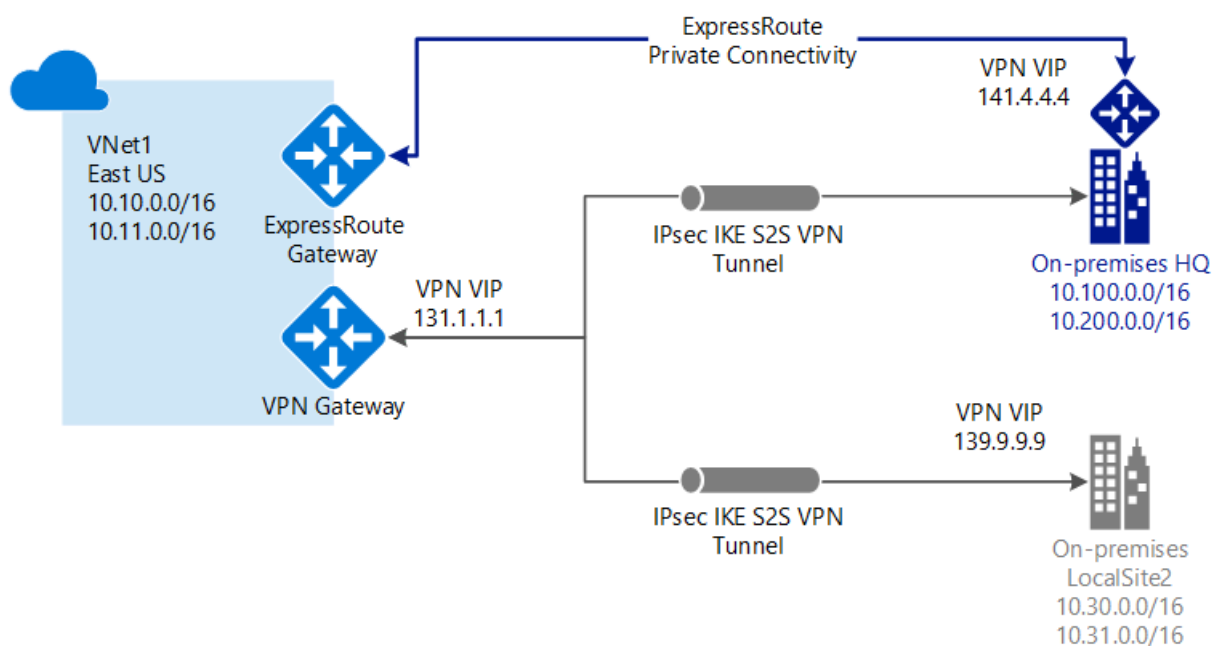


In an ExpressRoute connection, a virtual network gateway is configured with the gateway type 'ExpressRoute', rather than 'Vpn'. For more information about ExpressRoute, see the [ExpressRoute technical overview](#).

Site-to-Site and ExpressRoute coexisting connections

ExpressRoute is a direct, dedicated 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 requires two virtual network gateways for the same virtual network, one using -GatewayType Vpn, and the other using -GatewayType ExpressRoute.



Deployment models and methods for S2S and ExpressRoute

	CLASSIC DEPLOYMENT	RESOURCE MANAGER DEPLOYMENT
Classic Portal	Not Supported	Not Supported

	CLASSIC DEPLOYMENT	RESOURCE MANAGER DEPLOYMENT
Azure Portal	Not Supported	Not Supported
PowerShell	Article	Article

Next steps

Plan your VPN gateway configuration. See [VPN Gateway Planning and Design](#) and [Connecting your on-premises network to Azure](#).

VPN Gateway FAQ

1/17/2017 • 18 min to read • [Edit on GitHub](#)

Connecting to Virtual Networks

Can I connect virtual networks in different Azure regions?

Yes. In fact, there is no region constraint. One virtual network can connect to another virtual network in the same region, or in a different Azure region.

Can I connect virtual networks in different subscriptions?

Yes.

Can I connect to multiple sites from a single virtual network?

You can connect to multiple sites by using Windows PowerShell and the Azure REST APIs. See the [Multi-Site and VNet-to-VNet Connectivity](#) FAQ section.

What are my cross-premises connection options?

The following cross-premises connections are supported:

- [Site-to-Site](#) – VPN connection over IPsec (IKE v1 and IKE v2). This type of connection requires a VPN device or RRAS.
- [Point-to-Site](#) – VPN connection over SSTP (Secure Socket Tunneling Protocol). This connection does not require a VPN device.
- [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.
- [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.
- [ExpressRoute](#) – ExpressRoute is a direct connection to Azure from your WAN, not over the public Internet. See the [ExpressRoute Technical Overview](#) and the [ExpressRoute FAQ](#) for more information.

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

What is the difference between a Site-to-Site connection and Point-to-Site?

Site-to-Site connections let you connect between any of the computers located on your premises to any virtual machine or role instance within your virtual network, depending on how you choose to configure routing. 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 or soft appliance), which must be deployed at the edge of your network. To create this type of connection, you must have the required VPN hardware and an externally facing IPv4 address.

Point-to-Site connections 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, provided that 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.

What is ExpressRoute?

ExpressRoute lets you create private connections between Microsoft datacenters and infrastructure that's on your premises or in a co-location environment. With ExpressRoute, you can establish connections to Microsoft cloud services such as Microsoft Azure and Office 365 at an ExpressRoute partner co-location facility, or directly connect from your existing WAN network (such as an MPLS VPN provided by a network service provider).

ExpressRoute connections offer better security, more reliability, higher bandwidth, and lower latencies than typical connections over the Internet. In some cases, using ExpressRoute connections to transfer data between your on-premises network and Azure can also yield significant cost benefits. If you already have created a cross-premises connection from your on-premises network to Azure, you can migrate to an ExpressRoute connection while keeping your virtual network intact.

See the [ExpressRoute FAQ](#) for more details.

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 [here](#). 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.

What do I do if I have a VPN device that isn't in the known compatible device list?

If you do not see your device listed as a known compatible VPN device and you want to use it for your VPN connection, you'll need to verify that it meets the supported IPsec/IKE configuration options and parameters listed [here](#). Devices meeting the minimum requirements should work well with VPN gateways. Contact your device manufacturer for additional support and configuration instructions.

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 connections

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

The following operating systems are supported:

- Windows 7 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit only)
- Windows 8 (32-bit and 64-bit)
- Windows 8.1 (32-bit and 64-bit)
- Windows Server 2012 (64-bit only)
- Windows Server 2012 R2 (64-bit only)

- Windows 10

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

No. Support is limited only to the Windows operating system versions listed above.

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

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

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

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

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

Yes. We use SSTP (Secure Socket Tunneling Protocol) to tunnel through firewalls. This tunnel will appear as an HTTPs connection.

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

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

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

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

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

Yes. Both these solutions will work if you have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or gateways using the `-VpnType PolicyBased` cmdlet.

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

Yes, it is possible. But the virtual networks cannot have overlapping IP prefixes and the Point-to-Site address spaces must not overlap between the virtual networks.

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

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet.

Gateways

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 get my VPN gateway IP address before I create it?

No. You have to create your gateway first to get the IP address. The IP address changes if you delete and recreate your VPN gateway.

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.

What is the "GatewaySubnet" and why is it needed?

We have a gateway service that we run to enable cross-premises connectivity.

You'll need to create a gateway subnet for your VNet to configure a VPN 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.

The gateway subnet minimum size depends entirely on the configuration that you want to create. Although it is possible to create a gateway subnet as small as /29 for some configurations, we recommend that you create a gateway subnet of /28 or larger (/28, /27, /26 etc.).

Can I deploy Virtual Machines or role instances to my gateway subnet?

No.

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.
- Classic portal: Add each range that you want sent through the gateway for your virtual network on the Networks page under Local Networks.

Can I configure Forced Tunneling?

Yes. See [Configure forced 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 will need to configure user defined routes in your virtual network to ensure traffic is routed properly between your on-premises networks and your virtual network subnets.

Why are certain ports opened on my VPN gateway?

They are required for Azure infrastructure communication. They are protected (locked down) by Azure certificates. Without proper certificates, external entities, including the customers of those gateways, will not be able to cause any effect on those endpoints.

A VPN gateway is fundamentally a multi-homed device with one NIC tapping into the customer private network, and one NIC facing the public network. Azure infrastructure entities cannot tap into customer private networks for compliance reasons, so they need to utilize public endpoints for infrastructure communication. The public endpoints are periodically scanned by Azure security audit.

More information about gateway types, requirements, and throughput

For more information, see [About VPN Gateway Settings](#).

Multi-Site and VNet-to-VNet connectivity

Which type of gateways can support multi-site and VNet-to-VNet connectivity?

Only route-based (dynamic routing) VPNs.

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

No, both virtual networks MUST be using route-based (dynamic routing) VPNs.

Is the VNet-to-VNet traffic secure?

Yes, it is protected by IPsec/IKE encryption.

Does VNet-to-VNet traffic travel over the Azure backbone?

Yes, this traffic traverses the Azure backbone. It does not go over the Internet.

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

Max. 10 combined for the Basic and Standard Dynamic Routing gateways; 30 for the High Performance VPN gateways.

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

No. Overlapping address spaces will cause the network configuration file upload or "Creating Virtual Network" to fail.

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 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 alphanumeric string of length between 1 to 128 characters.

Does Azure charge for traffic between virtual networks?

For traffic between different Azure virtual networks, Azure charges only for traffic traversing from one Azure region to another. The charge rate is listed in the Azure [VPN Gateway Pricing](#) page.

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](#).

BGP

Is BGP supported on all Azure VPN Gateway SKUs?

No, BGP is supported on Azure **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.

Are there ASNs reserved by Azure?

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

- Public ASNs: 8075, 8076, 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.

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

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

What address prefixes will Azure VPN gateways advertise to me?

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

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

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

Yes.

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

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

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

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

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

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

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

Does Azure VPN gateway support BGP transit routing?

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

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

Yes, you can establish more than one S2S VPN tunnel between an Azure VPN gateway and your on-premises

network. Please note that all these tunnels will be counted against the total number of tunnels for your Azure VPN gateways and you must enable BGP on both tunnels.

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

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

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

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

Yes.

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

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

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

Your on-premises BGP peer address **MUST NOT** be the same as the public IP address of your VPN device. Use a different IP address on the VPN device for your BGP Peer IP. It can be an address assigned to the loopback interface on the device. 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 and you have created an endpoint, you can connect to your virtual machine by using the VIP. In that case, you would specify the VIP and the port that you want to connect to. You'll need to configure the port on your virtual machine for the traffic. Typically, you would go to the Azure Classic Portal and save the settings for the RDP connection to your computer. The settings contain the necessary connection information.

If you have a virtual network with cross-premises connectivity configured, you can connect to your virtual machine by using the internal DIP or private IP address. You can also connect to your virtual machine by internal DIP from another virtual machine that's located on the same virtual network. You can't RDP to your virtual machine by using the DIP 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 DIP.

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. If you are troubleshooting, it's important to make sure that you have all the ranges listed in your Local Network that you want to send through the gateway. Verify that the Local Network address ranges do not overlap with any of the address ranges in the virtual network. Also, you want to verify that the DNS server you are using is resolving the name to the proper IP address.

Virtual Network FAQ

You view additional virtual network information in the [Virtual Network FAQ](#).

Azure subscription and service limits, quotas, and constraints

1/17/2017 • 49 min to read • [Edit on GitHub](#)

This document lists some of the most common Microsoft Azure limits, which are also sometimes called quotas. This document doesn't currently cover all Azure services. Over time, the list will be expanded and updated to cover more of the platform.

Please visit [Azure Pricing Overview](#) to learn more about Azure pricing. There, you can estimate your costs using the [Pricing Calculator](#) or by visiting the pricing details page for a service (for example, [Windows VMs](#)).

NOTE

If you want to raise the limit or quota above the **Default Limit**, [open an online customer support request at no charge](#). The limits can't be raised above the **Maximum Limit** value shown in the following tables. If there is no **Maximum Limit** column, then the resource doesn't have adjustable limits.

Free Trial subscriptions are not eligible for limit or quota increases. If you have a Free Trial, you can upgrade to a [Pay-As-You-Go](#) subscription. For more information, see [Upgrade Azure Free Trial to Pay-As-You-Go](#).

Limits and the Azure Resource Manager

It is now possible to combine multiple Azure resources in to a single Azure Resource Group. When using Resource Groups, limits that once were global become managed at a regional level with the Azure Resource Manager. For more information about Azure Resource Groups, see [Azure Resource Manager overview](#).

In the limits below, a new table has been added to reflect any differences in limits when using the Azure Resource Manager. For example, there is a **Subscription Limits** table and a **Subscription Limits - Azure Resource Manager** table. When a limit applies to both scenarios, it is only shown in the first table. Unless otherwise indicated, limits are global across all regions.

NOTE

It is important to emphasize that quotas for resources in Azure Resource Groups are per-region accessible by your subscription, and are not per-subscription, as the service management quotas are. Let's use core quotas as an example. If you need to request a quota increase with support for cores, you need to decide how many cores you want to use in which regions, and then make a specific request for Azure Resource Group core quotas for the amounts and regions that you want. Therefore, if you need to use 30 cores in West Europe to run your application there; you should specifically request 30 cores in West Europe. But you will not have a core quota increase in any other region -- only West Europe will have the 30-core quota.

As a result, you may find it useful to consider deciding what your Azure Resource Group quotas need to be for your workload in any one region, and request that amount in each region into which you are considering deployment. See [troubleshooting deployment issues](#) for more help discovering your current quotas for specific regions.

Service-specific limits

- [Active Directory](#)
- [API Management](#)
- [App Service](#)

- [Application Gateway](#)
- [Application Insights](#)
- [Automation](#)
- [Azure Redis Cache](#)
- [Azure RemoteApp](#)
- [Backup](#)
- [Batch](#)
- [BizTalk Services](#)
- [CDN](#)
- [Cloud Services](#)
- [Data Factory](#)
- [Data Lake Analytics](#)
- [DNS](#)
- [DocumentDB](#)
- [Event Hubs](#)
- [IoT Hub](#)
- [Key Vault](#)
- [Media Services](#)
- [Mobile Engagement](#)
- [Mobile Services](#)
- [Monitoring](#)
- [Multi-Factor Authentication](#)
- [Networking](#)
- [Notification Hub Service](#)
- [Operational Insights](#)
- [Resource Group](#)
- [Scheduler](#)
- [Search](#)
- [Service Bus](#)
- [Site Recovery](#)
- [SQL Database](#)
- [Storage](#)
- [StorSimple System](#)
- [Stream Analytics](#)
- [Subscription](#)
- [Traffic Manager](#)
- [Virtual Machines](#)
- [Virtual Machine Scale Sets](#)

Subscription limits

Subscription limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Cores per subscription ¹	20	10,000
Co-administrators per subscription	200	200

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Storage accounts per subscription ²	200	250
Cloud services per subscription	20	200
Local networks per subscription	10	500
SQL Database servers per subscription	6	150
DNS servers per subscription	9	100
Reserved IPs per subscription	20	100
Hosted service certificates per subscription	400	400
Affinity groups per subscription	256	256
Batch accounts per region per subscription	1	50
Alert rules per subscription	250	250

¹Extra Small instances count as one core towards the core limit despite using a partial core.

²This includes both Standard and Premium storage accounts. If you require more than 200 storage accounts, make a request through [Azure Support](#). The Azure Storage team will review your business case and may approve up to 250 storage accounts.

Subscription limits - Azure Resource Manager

The following limits apply when using the Azure Resource Manager and Azure Resource Groups. Limits that have not changed with the Azure Resource Manager are not listed below. Please refer to the previous table for those limits.

For information about handling limits on Resource Manager requests, see [Throttling Resource Manager requests](#).

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
VMs per subscription	20 ¹ per Region	10,000 per Region
VM total cores per subscription	20 ¹ per Region	10,000 per Region
VM per series (Dv2, F, etc.) cores per subscription	20 ¹ per Region	10,000 per Region
Co-administrators per subscription	Unlimited	Unlimited
Storage accounts per subscription	200	200 ²
Resource Groups per subscription	800	800
Availability Sets per subscription	2000 per Region	2000 per Region

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resource Manager API Reads	15000 per hour	15000 per hour
Resource Manager API Writes	1200 per hour	1200 per hour
Resource Manager API request size	4194304 bytes	4194304 bytes
Cloud services per subscription	Not Applicable ³	Not Applicable ³
Affinity groups per subscription	Not Applicable ³	Not Applicable ³

¹Default limits vary by offer Category Type, such as Free Trial, Pay-As-You-Go, and series, such as Dv2, F, G, etc.

²This includes both Standard and Premium storage accounts. If you require more than 200 storage accounts, make a request through [Azure Support](#). The Azure Storage team will review your business case and may approve up to 250 storage accounts.

³These features are no longer required with Azure Resource Groups and the Azure Resource Manager.

NOTE

It is important to emphasize that virtual machine cores have a regional total limit as well as a regional per size series (Dv2, F, etc.) limit that are separately enforced. For example, consider a subscription with a US East total VM core limit of 30, an A series core limit of 30, and a D series core limit of 30. This subscription would be allowed to deploy 30 A1 VMs, or 30 D1 VMs, or a combination of the two not to exceed a total of 30 cores (e.g. 10 A1 VMs and 20 D1 VMs).

Resource Group limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resources per resource group (per resource type)	800	Varies per resource type
Deployments per resource group	800	800
Resources per deployment	800	800
Management Locks (per unique scope)	20	20
Number of Tags (per resource or resource group)	15	15
Tag key length	512	512
Tag value length	256	256

Virtual Machines limits

Virtual Machine limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual machines per cloud service ¹	50	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Input endpoints per cloud service ²	150	150

¹Virtual machines created in Service Management (instead of Resource Manager) are automatically stored in a cloud service. You can add more virtual machines to that cloud service for load balancing and availability. See [How to Connect Virtual Machines with a Virtual Network or Cloud Service](#).

²Input endpoints allow communications to a virtual machine from outside the virtual machine's cloud service. Virtual machines in the same cloud service or virtual network can automatically communicate with each other. See [How to Set Up Endpoints to a Virtual Machine](#).

Virtual Machines limits - Azure Resource Manager

The following limits apply when using the Azure Resource Manager and Azure Resource Groups. Limits that have not changed with the Azure Resource Manager are not listed below. Please refer to the previous table for those limits.

RESOURCE	DEFAULT LIMIT
Virtual machines per availability set	100
Certificates per subscription	Unlimited ¹

¹With Azure Resource Manager, certificates are stored in the Azure Key Vault. Although the number of certificates is unlimited for a subscription, there is still a 1 MB limit of certificates per deployment (which consists of either a single VM or an availability set).

Virtual Machine Scale Sets limits

RESOURCE	MAXIMUM LIMIT
Maximum number of VMs in a scale set	100
Maximum number of scale sets in a region	200

Networking limits

ExpressRoute Limits

The following limits apply to ExpressRoute resources per subscription.

RESOURCE	DEFAULT LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription for ARM	10
Maximum number of routes for Azure private peering with ExpressRoute standard	4,000
Maximum number of routes for Azure private peering with ExpressRoute premium add-on	10,000
Maximum number of routes for Azure public peering with ExpressRoute standard	200

RESOURCE	DEFAULT LIMIT
Maximum number of routes for Azure public peering with ExpressRoute premium add-on	200
Maximum number of routes for Azure Microsoft peering with ExpressRoute standard	200
Maximum number of routes for Azure Microsoft peering with ExpressRoute premium add-on	200
Number of virtual network links allowed per ExpressRoute circuit	see table below

Number of Virtual Networks per ExpressRoute circuit

CIRCUIT SIZE	NUMBER OF VNET LINKS FOR STANDARD	NUMBER OF VNET LINKS WITH PREMIUM ADD-ON
50 Mbps	10	20
100 Mbps	10	25
200 Mbps	10	25
500 Mbps	10	40
1 Gbps	10	50
2 Gbps	10	60
5 Gbps	10	75
10 Gbps	10	100

Networking limits

The following limits apply only for networking resources managed through the classic deployment model per subscription.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual networks per subscription	50	100
Local network sites per subscription	20	contact support
DNS Servers per virtual network	20	100
Private IP Addresses per virtual network	4096	4096
Concurrent TCP connections for a virtual machine or role instance	500K	500K
Network Security Groups (NSG)	100	200

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
NSG rules per NSG	200	400
User defined route tables	100	200
User defined routes per route table	100	400
Public IP addresses (dynamic)	5	contact support
Reserved public IP addresses	20	contact support
Public VIP per deployment	5	contact support
Private VIP (ILB) per deployment	1	1
Endpoint Access Control Lists (ACLs)	50	50

Networking Limits - Azure Resource Manager

The following limits apply only for networking resources managed through Azure Resource Manager per region per subscription.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual networks per subscription	50	500
Subnets per virtual network	1,000	contact support
DNS Servers per virtual network	9	25
Private IP Addresses per virtual network	4096	4096
Concurrent TCP connections for a virtual machine or role instance	500K	500K
Network Interfaces (NIC)	300	10000
Network Security Groups (NSG)	100	400
NSG rules per NSG	200	500
User defined route tables	100	200
User defined routes per route table	100	400
Public IP addresses (dynamic)	60	contact support
Public IP addresses (Static)	20	contact support
Load balancers (internal and internet facing)	100	contact support
Load balancer rules per load balancer	150	150

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public front end IP per load balancer	5	contact support
Private front end IP per load balancer	30	contact support
VNets peerings per Virtual Network	10	50
Point-to-Site Root Certificates per VPN Gateway	20	20

Contact support in case you need to increase limits from default.

Application Gateway limits

RESOURCE	DEFAULT LIMIT	NOTE
Application Gateway	50 per subscription	
Frontend IP Configurations	2	1 public and 1 private
Frontend Ports	20	
Backend Address Pools	20	
Backend Servers per pool	100	
HTTP Listeners	20	
HTTP load balancing rules	200	# of HTTP Listeners * n, n=10 Default
Backend HTTP settings	20	1 per Backend Address Pool
Instances per gateway	10	
SSL certificates	20	1 per HTTP Listeners
Request timeout min	1 second	
Request timeout max	24hrs	
Number of sites	20	1 per HTTP Listeners
URL Maps per listener	1	

Traffic Manager limits

RESOURCE	DEFAULT LIMIT
Profiles per subscription	100 ¹
Endpoints per profile	200

¹Contact support in case you need to increase these limits.

DNS limits

RESOURCE	DEFAULT LIMIT
Zones per subscription	100 ¹
Record sets per zone	5000 ¹
Records per record set	20

¹ Contact Azure Support in case you need to increase these limits.

Storage limits

For additional details on storage account limits, see [Azure Storage Scalability and Performance Targets](#).

Storage Service limits

RESOURCE	DEFAULT LIMIT
Number of storage accounts per subscription	200 ¹
TB per storage account	500 TB
Max number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	Only limit is the 500 TB storage account capacity
Max size of a single blob container, table, or queue	500 TB
Max number of blocks in a block blob or append blob	50,000
Max size of a block in a block blob	100 MB
Max size of a block blob	50,000 X 100 MB (approx. 4.75 TB)
Max size of a block in an append blob	4 MB
Max size of an append blob	50,000 X 4 MB (approx. 195 GB)
Max size of a page blob	1 TB
Max size of a table entity	1 MB
Max number of properties in a table entity	252
Max size of a message in a queue	64 KB
Max size of a file share	5 TB
Max size of a file in a file share	1 TB
Max number of files in a file share	Only limit is the 5 TB total capacity of the file share
Max 8 KB IOPS per share	1000

RESOURCE	DEFAULT LIMIT
Max number of files in a file share	Only limit is the 5 TB total capacity of the file share
Max number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	Only limit is the 500 TB storage account capacity
Max number of stored access policies per container, file share, table, or queue	5
Total Request Rate (assuming 1 KB object size) per storage account	Up to 20,000 IOPS, entities per second, or messages per second
Target throughput for single blob	Up to 60 MB per second, or up to 500 requests per second
Target throughput for single queue (1 KB messages)	Up to 2000 messages per second
Target throughput for single table partition (1 KB entities)	Up to 2000 entities per second
Target throughput for single file share	Up to 60 MB per second
Max ingress ² per storage account (US Regions)	10 Gbps if GRS/ZRS ³ enabled, 20 Gbps for LRS
Max egress ² per storage account (US Regions)	20 Gbps if RA-GRS/GRS/ZRS ³ enabled, 30 Gbps for LRS
Max ingress ² per storage account (European and Asian Regions)	5 Gbps if GRS/ZRS ³ enabled, 10 Gbps for LRS
Max egress ² per storage account (European and Asian Regions)	10 Gbps if RA-GRS/GRS/ZRS ³ enabled, 15 Gbps for LRS

¹This includes both Standard and Premium storage accounts. If you require more than 200 storage accounts, make a request through [Azure Support](#). The Azure Storage team will review your business case and may approve up to 250 storage accounts.

²*Ingress* refers to all data (requests) being sent to a storage account. *Egress* refers to all data (responses) being received from a storage account.

³Azure Storage replication options include:

- **RA-GRS:** Read-access geo-redundant storage. If RA-GRS is enabled, egress targets for the secondary location are identical to those for the primary location.
- **GRS:** Geo-redundant storage.
- **ZRS:** Zone-redundant storage. Available only for block blobs.
- **LRS:** Locally redundant storage.

Virtual Machine disk limits

An Azure virtual machine supports attaching a number of data disks. For optimal performance, you will want to limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all disks are not being highly utilized at the same time, the storage account can support a larger number disks.

- **For standard storage accounts:** A standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a standard storage account should not exceed this limit.

You can roughly calculate the number of highly utilized disks supported by a single standard storage account based on the request rate limit. For example, for a Basic Tier VM, the maximum number of highly utilized disks is about 66 (20,000/300 IOPS per disk), and for a Standard Tier VM, it is about 40 (20,000/500 IOPS per disk), as shown in the table below.

- **For premium storage accounts:** A premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

See [Virtual machine sizes](#) for additional details.

Standard storage accounts

Virtual machine disks: per disk limits

VM TIER	BASIC TIER VM	STANDARD TIER VM
Disk size	1023 GB	1023 GB
Max 8 KB IOPS per persistent disk	300	500
Max number of disks performing max IOPS	66	40

Premium storage accounts

Virtual machine disks: per account limits

RESOURCE	DEFAULT LIMIT
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB
Max bandwidth per account (ingress + egress ¹)	<=50 Gbps

¹*Ingress* refers to all data (requests) being sent to a storage account. *Egress* refers to all data (responses) being received from a storage account.

Virtual machine disks: per disk limits

PREMIUM STORAGE DISK TYPE	P10	P20	P30
Disk size	128 GiB	512 GiB	1024 GiB (1 TB)
Max IOPS per disk	500	2300	5000
Max throughput per disk	100 MB per second	150 MB per second	200 MB per second
Max number of disks per storage account	280	70	35

Virtual machine disks: per VM limits

RESOURCE	DEFAULT LIMIT
Max IOPS Per VM	80,000 IOPS with GS5 VM ¹

RESOURCE	DEFAULT LIMIT
Max throughput per VM	2,000 MB/s with GS5 VM ¹

¹Refer to [VM Size](#) for limits on other VM sizes.

Storage Resource Provider limits

The following limits apply when using the Azure Resource Manager and Azure Resource Groups only.

RESOURCE	DEFAULT LIMIT
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	200 per hour
Storage account management operations (list)	100 per 5 minutes

Cloud Services limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Web/worker roles per deployment ¹	25	25
Instance Input Endpoints per deployment	25	25
Input Endpoints per deployment	25	25
Internal Endpoints per deployment	25	25

¹Each Cloud Service with Web/Worker roles can have two deployments, one for production and one for staging. Also note that this limit refers to the number of distinct roles (configuration) and not the number of instances per role (scaling).

App Service limits

The following App Service limits include limits for Web Apps, Mobile Apps, API Apps, and Logic Apps.

RESOURCE	FREE	SHARED (PREVIEW)	BASIC	STANDARD	PREMIUM (PREVIEW)
Web, mobile, or API apps per App Service plan ¹	10	100	Unlimited ²	Unlimited ²	Unlimited ²
Logic apps per App Service plan ¹	10	10	10	20 per core	20 per core
App Service plan	1 per region	10 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

RESOURCE	FREE	SHARED (PREVIEW)	BASIC	STANDARD	PREMIUM (PREVIEW)
Scale-Out (max instances)	1 shared	1 shared	3 dedicated ³	10 dedicated ³	20 dedicated (50 in ASE) ^{3,4}
Storage ⁵	1 GB ⁵	1 GB ⁵	10 GB ⁵	50 GB ⁵	500 GB ^{4,5}
CPU time (5 min) ⁶	3 minutes	3 minutes	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates
CPU time (day) ⁶	60 minutes	240 minutes	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates
Memory (1 hour)	1024 MB per App Service plan	1024 MB per app	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
Application architecture	32-bit	32-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit
Web Sockets per instance ⁷	5	35	350	Unlimited	Unlimited
Concurrent debugger connections per application	1	1	1	5	5
azurewebsites.net subdomain with FTP/S and SSL	X	X	X	X	X
Custom domain support		X	X	X	X
Custom domain SSL support			Unlimited	Unlimited, 5 SNI SSL and 1 IP SSL connections included	Unlimited, 5 SNI SSL and 1 IP SSL connections included
Integrated Load Balancer		X	X	X	X
Always On			X	X	X
Scheduled Backups				Once per day	Once every 5 minutes ⁸
Auto Scale			X	X	X
WebJobs ⁹	X	X	X	X	X

RESOURCE	FREE	SHARED (PREVIEW)	BASIC	STANDARD	PREMIUM (PREVIEW)
Azure Scheduler support		X	X	X	X
Endpoint monitoring			X	X	X
Staging Slots				5	20
Custom domains per app		500	500	500	500
SLA			99.9%	99.95% ¹⁰	99.95% ¹⁰

¹Apps and storage quotas are per App Service plan unless noted otherwise.

²The actual number of apps that you can host on these machines depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.

³Dedicated instances can be of different sizes. See [App Service Pricing](#) for more details.

⁴Premium tier allows up to 50 computes instances (subject to availability) and 500 GB of disk space when using App Service Environments, and 20 compute instances and 250 GB storage otherwise.

⁵The storage limit is the total content size across all apps in the same App Service plan. More storage options are available in [App Service Environment](#)

⁶These resources are constrained by physical resources on the dedicated instances (the instance size and the number of instances).

⁷If you scale an app in the Basic tier to two instances, you have 350 concurrent connections for each of the two instances.

⁸Premium tier allows backup intervals down up to every 5 minutes when using App Service Environments, and 50 times per day otherwise.

⁹Run custom executables and/or scripts on demand, on a schedule, or continuously as a background task within your App Service instance. Always On is required for continuous WebJobs execution. Azure Scheduler Free or Standard is required for scheduled WebJobs. There is no predefined limit on the number of WebJobs that can run in an App Service instance, but there are practical limits that depend on what the application code is trying to do.

¹⁰SLA of 99.95% provided for deployments that use multiple instances with Azure Traffic Manager configured for failover.

Scheduler limits

The following table describes each of the major quotas, limits, defaults, and throttles in Azure Scheduler.

RESOURCE	LIMIT DESCRIPTION
Job size	Maximum job size is 16K. If a PUT or a PATCH results in a job larger than these limits, a 400 Bad Request status code is returned.
Request URL size	Maximum size of the request URL is 2048 chars.
Aggregate header size	Maximum aggregate header size is 4096 chars.
Header count	Maximum header count is 50 headers.
Body size	Maximum body size is 8192 chars.

RESOURCE	LIMIT DESCRIPTION
Recurrence span	Maximum recurrence span is 18 months.
Time to start time	Maximum "time to start time" is 18 months.
Job history	Maximum response body stored in job history is 2048 bytes.
Frequency	The default max frequency quota is 1 hour in a free job collection and 1 minute in a standard job collection. The max frequency is configurable on a job collection to be lower than the maximum. All jobs in the job collection are limited the value set on the job collection. If you attempt to create a job with a higher frequency than the maximum frequency on the job collection then request will fail with a 409 Conflict status code.
Jobs	The default max jobs quota is 5 jobs in a free job collection and 50 jobs in a standard job collection. The maximum number of jobs is configurable on a job collection. All jobs in the job collection are limited the value set on the job collection. If you attempt to create more jobs than the maximum jobs quota, then the request fails with a 409 Conflict status code.
Job collections	Maximum number of job collection per subscription is 200,000.
Job history retention	Job history is retained for up to 2 months or up to the last 1000 executions.
Completed and faulted job retention	Completed and faulted jobs are retained for 60 days.
Timeout	There's a static (not configurable) request timeout of 60 seconds for HTTP actions. For longer running operations, follow HTTP asynchronous protocols; for example, return a 202 immediately but continue working in the background.

Batch limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Cores per Batch account	20	N/A ¹
Jobs and job schedules ² per Batch account	20	10,000
Pools per Batch account	20	5000

¹ The number of cores per Batch account can be increased, but the maximum number is unspecified. Contact customer support to discuss increase options.

² Includes run-once active jobs and active job schedules. Completed jobs and job schedules are not limited.

BizTalk Services limits

The following table shows the limits for Azure Biztalk Services.

RESOURCE	FREE (PREVIEW)	DEVELOPER	BASIC	STANDARD	PREMIUM
Scale out	N/A	N/A	Yes, in increments of 1 Basic Unit	Yes, in increments of 1 Standard Unit	Yes, in increments of 1 Premium Unit
Scale Limit	N/A	N/A	Up to 8 units	Up to 8 units	Up to 8 units
EAI Bridges per Unit	N/A	25	25	125	500
EDI Agreements per Unit	N/A	10	50	250	1000
Hybrid Connections per Unit	5	5	10	50	100
Hybrid Connection Data Transfer (GBs) per Unit	5	5	50	250	500
Number of connections using BizTalk Adapter Service per Unit	N/A	1	2	5	25
Archiving	N/A	Available	N/A	N/A	Available
High Availability	N/A	N/A	Available	Available	Available

DocumentDB limits

DocumentDB is a global scale database in which throughput and storage can be scaled to handle whatever your application requires. If you have any questions about the scale DocumentDB provides, please send email to askdocdb@microsoft.com.

Mobile Engagement limits

RESOURCE	MAXIMUM LIMIT
App Collection Users	5 per App Collection
Average Data points	200 per Active User/Day
Average App-Info set	50 per Active User/Day
Average Messages pushed	20 per Active User/Day
Segments	100 per app
Criteria per segment	10
Active Push Campaigns	50 per app

RESOURCE	MAXIMUM LIMIT
Total Push Campaigns (includes Active & Completed)	1000 per app

Search limits

Pricing tiers determine the capacity and limits of your search service. Tiers include:

- *Free* multi-tenant service, shared with other Azure subscribers, intended for evaluation and small development projects.
- *Basic* provides dedicated computing resources for production workloads at a smaller scale, with up to three replicas for highly available query workloads.
- *Standard (S1, S2, S3, S3 High Density)* is for larger production workloads. Multiple levels exist within the standard tier so that you can choose a resource configuration that best matches your workload profile.

Limits per subscription

You can create multiple services within a subscription, each one provisioned at a specific tier, limited only by the number of services allowed at each tier. For example, you could create up to 12 services at the Basic tier and another 12 services at the S1 tier within the same subscription. For more information about tiers, see [Choose a SKU or tier for Azure Search](#).

Maximum service limits can be raised upon request. Contact Azure Support if you need more services within the same subscription.

RESOURCE	FREE	BASIC	S1	S2	S3	S3 HD ¹
Maximum services	1	12	12	6	6	6
Maximum scale in SU ²	N/A ³	3 SU ⁴	36 SU	36 SU	36 SU	36 SU

¹ S3 HD does not support [indexers](#) at this time.

² Search units (SU) are billing units, allocated as either a *replica* or a *partition*. You need both resources for storage, indexing, and query operations. To learn more about how search units are computed, plus a chart of valid combinations that stay under the maximum limits, see [Scale resource levels for query and index workloads](#).

³ Free is based on shared resources used by multiple subscribers. At this tier, there are no dedicated resources for an individual subscriber. For this reason, maximum scale is marked as not applicable.

⁴ Basic has one fixed partition. At this tier, additional SUs are used for allocating more replicas for increased query workloads.

Limits per search service

Storage is constrained by disk space or by a hard limit on the *maximum number* of indexes or documents, whichever comes first.

RESOURCE	FREE	BASIC	S1	S2	S3	S3 HD
Service Level Agreement (SLA)	No ¹	Yes	Yes	Yes	Yes	Yes

RESOURCE	FREE	BASIC	S1	S2	S3	S3 HD
Storage per partition	50 MB	2 GB	25 GB	100 GB	200 GB	200 GB
Partitions per service	N/A	1	12	12	12	3 ²
Partition size	N/A	2 GB	25 GB	100 GB	200 GB	200 GB
Replicas	N/A	3	12	12	12	12
Maximum indexes	3	5	50	200	200	1000 per partition or 3000 per service
Maximum documents	10,000	1 million	15 million per partition or 180 million per service	60 million per partition or 720 million per service	120 million per partition or 1.4 billion per service	1 million per index or 200 million per partition
Estimated queries per second (QPS)	N/A	~3 per replica	~15 per replica	~60 per replica	~60 per replica	>60 per replica

¹ Free and Preview SKUs do not come with service level agreements (SLAs). SLAs are enforced once a SKU becomes generally available.

² S3 HD has a hard limit of 3 partitions, which is lower than the partition limit for S3. The lower partition limit is imposed because the index count for S3 HD is substantially higher. Given that service limits exist for both computing resources (storage and processing) and content (indexes and documents), the content limit is reached first.

To learn more about limits on a more granular level, such as document size, queries per second, keys, requests, and responses, see [Service limits in Azure Search](#).

Media Services limits

NOTE

For resources that are not fixed, you may ask for the quotas to be raised, by opening a support ticket. Do **not** create additional Azure Media Services accounts in an attempt to obtain higher limits.

RESOURCE	DEFAULT LIMIT
Azure Media Services (AMS) accounts in a single subscription	25 (fixed)
Assets per AMS account	1,000,000
Chained tasks per job	30 (fixed)
Assets per task	50
Assets per job	100

RESOURCE	DEFAULT LIMIT
Jobs per AMS account	50,000 ²
Unique locators associated with an asset at one time	5 ⁴
Live channels per AMS account	5
Programs in stopped state per channel	50
Programs in running state per channel	3
Streaming endpoints in running state per AMS account	2
Streaming units per streaming endpoint	10
Media Reserved Units (RUs) per AMS account	25 (S1, S2) 10 (S3) ¹
Storage accounts	1,000 ⁵ (fixed)
Policies	

¹ S3 RUs are not available in India West.

² This number includes queued, finished, active, and canceled jobs. It does not include deleted jobs. You can delete the old jobs using **IJob.Delete** or the **DELETE** HTTP request.

³ When making a request to list Job entities, a maximum of 1,000 will be returned per request. If you need to keep track of all submitted Jobs, you can use top/skip as described in [OData system query options](#).

⁴ Locators are not designed for managing per-user access control. To give different access rights to individual users, use Digital Rights Management (DRM) solutions. For more information, see [this](#) section.

⁵ The storage accounts must be from the same Azure subscription.

⁶ There is a limit of 1,000,000 policies for different AMS policies (for example, for Locator policy or ContentKeyAuthorizationPolicy).

NOTE

You should use the same policy ID if you are always using the same days / access permissions / etc.

CDN limits

RESOURCE	SOFT LIMIT
CDN profiles	8
CDN endpoints per profile	10
Custom domains per endpoint	10

Request an update to your subscription's soft limits by opening a support ticket.

Mobile Services limits

TIER:	FREE	BASIC	STANDARD
API Calls	500 K	1.5 M / unit	15 M / unit
Active Devices	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Unlimited units
Push Notifications	Notification Hubs Free Tier included, up to 1 M pushes	Notification Hubs Basic Tier included, up to 10 M pushes	Notification Hubs Standard Tier included, up to 10 M pushes
Real time messaging/ Web Sockets	Limited	350 / mobile service	Unlimited
Offline synchronizations	Limited	Included	Included
Scheduled jobs	Limited	Included	Included
SQL Database (required) Standard rates apply for additional capacity	20 MB included	20 MB included	20 MB included
CPU capacity	60 minutes / day	Unlimited	Unlimited
Outbound data transfer	165 MB per day (daily Rollover)	Included	Included

For additional details on these limits and for information on pricing, see [Mobile Services Pricing](#).

Monitoring limits

RESOURCE	LIMIT
Autoscale Settings	100 per region per subscription

Notification Hub Service limits

TIER:	FREE	BASIC	STANDARD
Included Pushes	1 Million	10 Million	10 Million
Active Devices	500	Unlimited	Unlimited
Tag quota per installation/registration	60	60	60

For additional details on these limits and for information on pricing, see [Notification Hubs Pricing](#).

Event Hubs limits

The following table lists quotas and limits specific to Azure Event Hubs. For information about Event Hubs pricing, see [Event Hubs Pricing](#).

LIMIT	SCOPE	TYPE	BEHAVIOR WHEN EXCEEDED	VALUE
Number of Event Hubs per namespace	Namespace	Static	Subsequent requests for creation of a new namespace will be rejected.	10
Number of partitions per Event Hub	Entity	Static	-	32
Number of consumer groups per Event Hub	Entity	Static	-	20
Number of AMQP connections per namespace	Namespace	Static	Subsequent requests for additional connections will be rejected and an exception will be received by the calling code.	5,000
Maximum size of Event Hubs event	System-wide	Static	-	256KB
Maximum size of an Event Hub name	Entity	Static	-	50 characters
Number of non-epoch receivers per consumer group	Entity	Static	-	5
Maximum retention period of event data	Entity	Static	-	1-7 days
Maximum throughput units	Namespace	Static	Exceeding the throughput unit limit will cause your data to be throttled and generate a ServerBusyException . You can request a larger number of throughput units for a Standard tier by filing a support ticket. Additional throughput units are available in blocks of twenty on a committed purchase basis.	20

Service Bus limits

The following table lists quota information specific to Service Bus messaging. For information about pricing and other quotas for Service Bus, see the [Service Bus Pricing](#) overview.

QUOTA NAME	SCOPE	TYPE	BEHAVIOR WHEN EXCEEDED	VALUE
Maximum number of basic / standard namespaces per Azure subscription	Namespace	Static	Subsequent requests for additional basic / standard namespaces will be rejected by the portal.	100
Maximum number of premium namespaces per Azure subscription	Namespace	Static	Subsequent requests for additional premium namespaces will be rejected by the portal.	10
Queue/topic size	Entity	Defined upon creation of the queue/topic.	Incoming messages will be rejected and an exception will be received by the calling code.	1, 2, 3, 4 or 5 GB. If partitioning is enabled, the maximum queue/topic size is 80 GB.
Number of concurrent connections on a namespace	Namespace	Static	Subsequent requests for additional connections will be rejected and an exception will be received by the calling code. REST operations do not count towards concurrent TCP connections.	NetMessaging: 1,000 AMQP: 5,000
Number of concurrent connections on a queue/topic/subscription entity	Entity	Static	Subsequent requests for additional connections will be rejected and an exception will be received by the calling code. REST operations do not count towards concurrent TCP connections.	Capped by the limit of concurrent connections per namespace.
Number of concurrent receive requests on a queue/topic/subscription entity	Entity	Static	Subsequent receive requests will be rejected and an exception will be received by the calling code. This quota applies to the combined number of concurrent receive operations across all subscriptions on a topic.	5,000

QUOTA NAME	SCOPE	TYPE	BEHAVIOR WHEN EXCEEDED	VALUE
Number of topics/queues per service namespace	System-wide	Static	Subsequent requests for creation of a new topic or queue on the service namespace will be rejected. As a result, if configured through the Azure portal , an error message will be generated. If called from the management API, an exception will be received by the calling code.	10,000 The total number of topics plus queues in a service namespace must be less than or equal to 10,000. This is not applicable to Premium as all entities are partitioned.
Number of partitioned topics/queues per service namespace	System-wide	Static	Subsequent requests for creation of a new partitioned topic or queue on the service namespace will be rejected. As a result, if configured through the Azure portal , an error message will be generated. If called from the management API, a QuotaExceededException exception will be received by the calling code.	Basic and Standard Tiers - 100 Premium - 1,000 Each partitioned queue or topic counts towards the quota of 10,000 entities per namespace.
Maximum size of any messaging entity path: queue or topic	Entity	Static	-	260 characters
Maximum size of any messaging entity name: namespace, subscription, or subscription rule	Entity	Static	-	50 characters

QUOTA NAME	SCOPE	TYPE	BEHAVIOR WHEN EXCEEDED	VALUE
Message size for a queue/topic/subscription entity	System-wide	Static	Incoming messages that exceed these quotas will be rejected and an exception will be received by the calling code.	<p>Maximum message size: 256KB (Standard tier) / 1MB (Premium tier).</p> <p>Note Due to system overhead, this limit is usually slightly less.</p> <p>Maximum header size: 64KB</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/subscription entity	System-wide	Static	A SerializationException exception is generated.	<p>Maximum message property size for each property is 32K. Cumulative size of all properties cannot exceed 64K. This applies to the entire header of the BrokeredMessage, which has both user properties as well as system properties (such as SequenceNumber, Label, MessageId, and so on).</p>
Number of subscriptions per topic	System-wide	Static	Subsequent requests for creating additional subscriptions for the topic will be rejected. As a result, if configured through the portal, an error message will be shown. If called from the management API an exception will be received by the calling code.	2,000

QUOTA NAME	SCOPE	TYPE	BEHAVIOR WHEN EXCEEDED	VALUE
Number of SQL filters per topic	System-wide	Static	Subsequent requests for creation of additional filters on the topic will be rejected and an exception will be received by the calling code.	2,000
Number of correlation filters per topic	System-wide	Static	Subsequent requests for creation of additional filters on the topic will be rejected and an exception will be received by the calling code.	100,000
Size of SQL filters/actions	System-wide	Static	Subsequent requests for creation of additional filters will be rejected and an exception will be received by the calling code.	Maximum length of filter condition string: 1024 (1K). Maximum length of rule action string: 1024 (1K). Maximum number of expressions per rule action: 32.
Number of SharedAccessAuthorizationRule rules per namespace, queue, or topic	Entity, namespace	Static	Subsequent requests for creation of additional rules will be rejected and an exception will be received by the calling code.	Maximum number of rules: 12. Rules that are configured on a Service Bus namespace apply to all queues and topics in that namespace.

IoT Hub limits

The following table lists the limits associated with the different service tiers (S1, S2, S3, F1). For information about the cost of each *unit* in each tier, see [IoT Hub Pricing](#).

RESOURCE	S1 STANDARD	S2 STANDARD	S3 STANDARD	F1 FREE
Messages/day	400,000	6,000,000	300,000,000	8,000
Maximum units	200	200	200	1

NOTE

If you anticipate using more than 200 units with an S1 or S2 or S3 tier hub, please contact Microsoft support.

The following table lists the limits that apply to IoT Hub resources:

RESOURCE	LIMIT
Maximum paid IoT hubs per Azure subscription	10
Maximum free IoT hubs per Azure subscription	1
Maximum number of device identities returned in a single call	1000
IoT Hub message maximum retention for device-to-cloud messages	7 days
Maximum size of device-to-cloud message	256 KB
Maximum size of device-to-cloud batch	256 KB
Maximum messages in device-to-cloud batch	500
Maximum size of cloud-to-device message	64 KB
Maximum TTL for cloud-to-device messages	2 days
Maximum delivery count for cloud-to-device messages	100
Maximum delivery count for feedback messages in response to a cloud-to-device message	100
Maximum TTL for feedback messages in response to a cloud-to-device message	2 days

NOTE

If you need more than 10 paid IoT hubs in an Azure subscription, please contact Microsoft support.

The IoT Hub service throttles requests when the following quotas are exceeded:

THROTTLE	PER-HUB VALUE
Identity registry operations (create, retrieve, list, update, delete), individual or bulk import/export	5000/min/unit (for S3) 100/min/unit (for S1 and S2).
Device connections	6000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Device-to-cloud sends	6000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Cloud-to-device sends	5000/min/unit (for S3), 100/min/unit (for S1 and S2).
Cloud-to-device receives	50000/min/unit (for S3), 1000/min/unit (for S1 and S2).

THROTTLE	PER-HUB VALUE
File upload operations	5000 file upload notifications/min/unit (for S3), 100 file upload notifications/min/unit (for S1 and S2). 10000 SAS URIs can be out for an Azure Storage account at one time. 10 SAS URIs/device can be out at one time.

Data Factory limits

Data factory is a multi-tenant service that has the following default limits in place to make sure customer subscriptions are protected from each other's workloads. Many of the limits can be easily raised for your subscription up to the maximum limit by contacting support.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
data factories in an Azure subscription	50	Contact support
pipelines within a data factory	2500	Contact support
datasets within a data factory	5000	Contact support
concurrent slices per dataset	10	10
bytes per object for pipeline objects ¹	200 KB	200 KB
bytes per object for dataset and linked service objects ¹	100 KB	2000 KB
HDInsight on-demand cluster cores within a subscription ²	60	Contact support
Cloud data movement unit ³	8	Contact support
Retry count for pipeline activity runs	1000	MaxInt (32 bit)

¹ Pipeline, dataset, and linked service objects represent a logical grouping of your workload. Limits for these objects do not relate to amount of data you can move and process with the Azure Data Factory service. Data factory is designed to scale to handle petabytes of data.

² On-demand HDInsight cores are allocated out of the subscription that contains the data factory. As a result, the above limit is the Data Factory enforced core limit for on-demand HDInsight cores and is different from the core limit associated with your Azure subscription.

³ Cloud data movement unit (DMU) is being used in a cloud-to-cloud copy operation. It is a measure that represents the power (a combination of CPU, memory, and network resource allocation) of a single unit in Data Factory. You can achieve higher copy throughput by leveraging more DMUs for some scenarios. Refer to [Cloud data movement units](#) section on details.

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

Data Lake Analytics makes the complex task of managing distributed infrastructure and complex code easy. It dynamically provisions resources and lets you do analytics on exabytes of data. When the job completes, it winds down resources automatically, and you pay only for the processing power used. As you increase or decrease the size of data stored or the amount of compute used, you don't have to rewrite code. Many of the default limits can be easily raised for your subscription by contacting support.

RESOURCE	DEFAULT LIMIT	COMMENTS
max concurrent jobs	3	
Max parallelism per account	60	Use any combination of up to a maximum of 60 units of parallelism across three jobs.

Stream Analytics limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of Streaming Units per subscription per region	50	A request to increase streaming units for your subscription beyond 50 can be made by contacting Microsoft Support .
Maximum throughput of a Streaming Unit	1MB/s*	Maximum throughput per SU depends on the scenario. Actual throughput may be lower and depends upon query complexity and partitioning. Further details can be found in the Scale Azure Stream Analytics jobs to increase throughput article.
Maximum number of inputs per job	60	There is a hard limit of 60 inputs per Stream Analytics job.
Maximum number of outputs per job	60	There is a hard limit of 60 outputs per Stream Analytics job.
Maximum number of functions per job	60	There is a hard limit of 60 functions per Stream Analytics job.
Maximum number of jobs per region	1500	Each subscription may have up to 1500 jobs per geographical region.

Active Directory limits

Here are the usage constraints and other service limits for the Azure Active Directory service.

CATEGORY	LIMITS
Directories	<p>A single user can only be associated with a maximum of 20 Azure Active Directory directories.</p> <p>Examples of possible combinations:</p> <ul style="list-style-type: none"> • A single user creates 20 directories. • A single user is added to 20 directories as a member. • A single user creates 10 directories and later is added by others to 10 different directories.
Objects	<ul style="list-style-type: none"> • A maximum of 500,000 objects can be used in a single directory by users of the Free edition of Azure Active Directory. • A non-admin user can create no more than 250 objects.
Schema extensions	<ul style="list-style-type: none"> • String type extensions can have maximum of 256 characters. • Binary type extensions are limited to 256 bytes. • 100 extension values (across ALL types and ALL applications) can be written to any single object. • Only "User", "Group", "TenantDetail", "Device", "Application" and "ServicePrincipal" entities can be extended with "String" type or "Binary" type single-valued attributes. • Schema extensions are available only in Graph API-version 1.21-preview. The application must be granted write access to register an extension.
Applications	<p>A maximum of 10 users can be owners of a single application.</p>
Groups	<ul style="list-style-type: none"> • A maximum of 10 users can be owners of a single group. • Any number of objects can be members of a single group in Azure Active Directory. • The number of members in a group you can synchronize from your on-premises Active Directory to Azure Active Directory is limited to 15K members, using Azure Active Directory Directory Synchronization (DirSync). • The number of members in a group you can synchronize from your on-premises Active Directory to Azure Active Directory using Azure AD Connect is limited to 50K members.
Access Panel	<ul style="list-style-type: none"> • There is no limit to the number of applications that can be seen in the Access Panel per end user, for users assigned licenses for Azure AD Premium or the Enterprise Mobility Suite. • A maximum of 10 app tiles (examples: Box, Salesforce, or Dropbox) can be seen in the Access Panel for each end user for users assigned licenses for Free or Azure AD Basic editions of Azure Active Directory. This limit does not apply to Administrator accounts.

CATEGORY	LIMITS
Reports	A maximum of 1,000 rows can be viewed or downloaded in any report. Any additional data is truncated.

Azure RemoteApp limits

RESOURCE	DEFAULT LIMIT
Collections per user	1
Published apps per collection	100
Trial collection duration	30 days
Trial collections	2 per subscription
Users per trial collection	10
Trial template images	25
Paid collections	3
Paid template images	25
Users - basic tier*	400 (default)/ 800 (maximum)
Users - standard tier*	250 (default)/ 500 (maximum)
Users- premium tier	100 default.
Users - premium plus tier	50 default.
Concurrent connections across all collections in a subscription	5000
User data storage (UPD) per user per collection	50 GB
Idle timeout	4 hours
Disconnected timeout	4 hours

*User limits in basic and standard tiers cannot be increased beyond the maximum limit listed above.

The number of users is determined by the number of VMs used for your collection:

- Basic = 16 users per VM
- Standard = 10 users per VM
- Premium = 4 users per VM
- Premium plus = 2 users per 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 (24*7).
Maximum size of a tiered volume on physical devices	64 TB for 8100 and 8600	8100 and 8600 are physical devices.
Maximum size of a tiered volume on virtual devices in Azure	30 TB for 8010 64 TB for 8020	8010 and 8020 are virtual devices in Azure that use Standard Storage and Premium Storage respectively.
Maximum size of a locally pinned volume on physical devices	9 TB for 8100 24 TB for 8600	8100 and 8600 are physical devices.
Maximum number of iSCSI connections	512	
Maximum number of iSCSI connections from initiators	512	
Maximum number of access control records per device	64	
Maximum number of volumes per backup policy	24	
Maximum number of backups retained per backup policy	64	
Maximum number of schedules per backup policy	10	
Maximum number of snapshots of any type that can be retained per volume	256	This includes local snapshots and cloud snapshots.
Maximum number of snapshots that can be present in any device	10,000	
Maximum number of volumes that can be processed in parallel for backup, restore, or clone	16	<ul style="list-style-type: none"> • If there are more than 16 volumes, they will be processed sequentially as processing slots become available. • New backups of a cloned or a restored tiered volume cannot occur until the operation is finished. However, for a local volume, backups are allowed after the volume is online.

LIMIT IDENTIFIER	LIMIT	COMMENTS
Restore and clone recover time for tiered volumes	< 2 minutes	<ul style="list-style-type: none"> • The volume is made available within 2 minutes of restore or clone operation, regardless of the volume size. • The volume performance may initially be slower than normal as most of the data and metadata still resides in the cloud. Performance may increase as data flows from the cloud to the StorSimple device. • The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate may be affected by Internet bandwidth to the cloud. • The restore or clone operation is complete when all the metadata is on the device. • Backup operations cannot be performed until the restore or clone operation is fully complete.

LIMIT IDENTIFIER	LIMIT	COMMENTS
Restore recover time for locally pinned volumes	< 2 minutes	<ul style="list-style-type: none"> The volume is made available within 2 minutes of the restore operation, regardless of the volume size. The volume performance may initially be slower than normal as most of the data and metadata still resides in the cloud. Performance may increase as data flows from the cloud to the StorSimple device. The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate may be affected by Internet bandwidth to the cloud. Unlike tiered volumes, in the case of locally pinned volumes, the volume data is also downloaded locally on the device. The restore operation is complete when all the volume data has been brought to the device. The restore operations may be long and the total time to complete the restore will depend on the size of the provisioned local volume, your Internet bandwidth and the existing data on the device. Backup operations on the locally pinned volume are allowed while the restore operation is in progress.
Thin-restore availability	Last failover	
Maximum client read/write throughput (when served from the SSD tier)*	920/720 MB/s with a single 10GbE network interface	Up to 2x with MPIO and two network interfaces.
Maximum client read/write throughput (when served from the HDD tier)*	120/250 MB/s	
Maximum client read/write throughput (when served from the cloud tier)*	11/41 MB/s	Read throughput depends on clients generating and maintaining sufficient I/O queue depth.

* Maximum throughput per I/O type was measured with 100 percent read and 100 percent write scenarios. Actual throughput may be lower and depends on I/O mix and network conditions.

Operational Insights limits

The following limits apply to Operational Insights subscriptions.

	FREE	STANDARD	PREMIUM
Daily data transfer limit	500 MB ¹	None	None
Data retention period	7 days	1 month	12 months
Data storage limit	500 MB * 7 days = 3.5 GB	unlimited	unlimited

¹When customers reach their 500MB daily data transfer limit, data analysis stops and resumes at the start of the next day. A day is based on UTC.

Backup limits

The following limits apply to Azure Backup.

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of servers/machines that can be registered against each vault	50 for Windows Server/Client/SCDPM 200 for IaaS VMs
Size of a data source for data stored in Azure vault storage	54400 GB max ¹
Number of backup vaults that can be created in each Azure subscription	25(Backup vaults) 25 Recovery Services vault per region
Number of times backup can be scheduled per day	3 per day for Windows Server/Client 2 per day for SCDPM Once a day for IaaS VMs
Data disks attached to an Azure virtual machine for backup	16

- ¹The 54400 GB limit does not apply to IaaS VM backup.

Site Recovery limits

The following limits apply to Azure Site Recovery:

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of vaults per subscription	25
Number of servers per Azure vault	250
Number of protection groups per Azure vault	No limit
Number of recovery plans per Azure vault	No limit
Number of servers per protection group	No limit
Number of servers per recovery plan	50

Application Insights limits

There are some limits on the number of metrics and events per application (that is, per instrumentation key).

Limits depend on the [pricing plan](#) that you choose.

RESOURCE	DEFAULT LIMIT	NOTE
Total data per day	500 GB	You can reduce by setting a cap. If you need more, mail AIDataCap@microsoft.com
Free data per month (Basic price plan)	1 GB	Additional data charged per GB
Throttling	16 k events/second	Measured over a minute.
Data retention	90 days	for Search , Analytics and Metrics explorer
Availability multi-step test detailed results retention	90 days	Detailed results of each step
Property and Metric ² name count	200	
Property and metric name length	150	
Property value string length	8192	
Distinct values for properties ^{3,4}	100	> 100 => can't use property as filter in Metrics Explorer
Trace and Exception message length	10000	
Availability tests count per app	10	

1. All these numbers are per instrumentation key.
2. Metric names are defined both in `TrackMetric` and in the measurement parameter of other `Track*()` calls. Metric names are global per instrumentation key.
3. Properties can be used for filtering and group-by only while they have less than 100 unique values for each property. After the number of unique values exceeds 100, you can still search the property, but no longer use it for filters or group-by.
4. Standard properties such as Request Name and Page URL are limited to 1000 unique values per week. After 1000 unique values, additional values are marked as "Other values." The original values can still be used for full text search and filtering.

About pricing and quotas in Application Insights

API Management limits

RESOURCE	LIMIT
API Calls (per unit of scale)	32 million per day ¹
Data transfer (per unit of scale)	161 GB per day ¹
Cache	5 GB ¹
Units of scale	Unlimited ¹

RESOURCE	LIMIT
Azure Active Directory Integration	Unlimited User Accounts ¹

¹API Management limits are different for each pricing tier. To see the pricing tiers and their associated limits and scaling options, see [API Management Pricing](#).

Azure Redis Cache limits

RESOURCE	LIMIT
Cache size	530 GB (contact us for more)
Databases	64
Max connected clients	40,000
Redis Cache replicas (for high availability)	1
Shards in a premium cache with clustering	10

Azure Redis Cache limits and sizes are different for each pricing tier. To see the pricing tiers and their associated sizes, see [Azure Redis Cache Pricing](#).

For more information on Azure Redis Cache configuration limits, see [Default Redis server configuration](#).

Because configuration and management of Azure Redis Cache instances is done by Microsoft, not all Redis commands are supported in Azure Redis Cache. For more information, see [Redis commands not supported in Azure Redis Cache]((redis-cache/cache-configure.md#redis-commands-not-supported-in-azure-redis-cache)).

Key Vault limits

TRANSACTIONS TYPE	MAX TRANSACTIONS ALLOWED IN 10 SECONDS, PER VAULT PER REGION ¹
HSM- CREATE KEY	5
HSM- other transactions	1000
Soft-key CREATE KEY	10
Soft-key other transactions	1500
All secrets, vault related transactions	2000

¹ There is a subscription-wide limit for all transaction types, that is 5x per key vault limit. For example, HSM- other transactions per subscription are limited to 5000 transactions in 10 seconds per subscription.

Multi-Factor Authentication

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Max number of Trusted IP addresses/ranges per subscription ¹	0	12

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Remember my devices - number of days	14	60
Max number of app passwords?	0	No Limit
Allow X attempts during MFA call	1	99
Two-way Text message Timeout Seconds	60	600
Default one-time bypass seconds	300	1800
Lock user account after X consecutive MFA denials	Not Set	99
Reset account lockout counter after X minutes	Not Set	9999
Unlock account after X minutes	Not Set	9999

¹This is expected to increase in the future.

Automation limits

RESOURCE	MAXIMUM LIMIT
Max number of new jobs that can be submitted every 30 seconds per Automation Account (non Scheduled jobs)	100
Max number of concurrent running jobs at the same instance of time per Automation Account (non Scheduled jobs)	200
Max number of modules that can be imported every 30 seconds per Automation Account	5
Max size of a Module	100 MB
Job Run Time - Free tier	500 minutes per subscription per calendar month
Max amount of memory given to a job	400 MB
Max number of network sockets allowed per job	1000

SQL Database limits

For SQL Database limits, see [SQL Database Resource Limits](#).

See also

[Understanding Azure Limits and Increases](#)

[Virtual Machine and Cloud Service Sizes for Azure](#)

[Sizes for Cloud Services](#)

Planning and design for VPN Gateway

1/17/2017 • 10 min to read • [Edit on GitHub](#)

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

Planning

Cross-premises connectivity options

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

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

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	Typically < 100 Mbps aggregate	Typically < 100 Mbps aggregate	50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps
Protocols Supported	Secure Sockets Tunneling Protocol (SSTP)	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	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

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

Gateway requirements by VPN type and SKU

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. When you select a higher gateway SKU, more CPUs and network bandwidth are allocated to the gateway, and as a result, the gateway can support higher network throughput to the virtual network.

VPN Gateway can use the following SKUs:

- Basic
- Standard
- HighPerformance

VPN Gateway does not use the UltraPerformance gateway SKU. For information about the UltraPerformance SKU, see the [ExpressRoute](#) documentation.

When selecting a SKU, 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.

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

Aggregate throughput by SKU and VPN type

The following table shows the gateway types and the estimated aggregate throughput by gateway SKU. This table applies to both 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)	100 Mbps	10	500 Mbps	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.

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

Workflow

The following list outlines the common workflow for cloud connectivity:

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

Design

Connection topologies

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

Design basics

The following sections discuss the VPN gateway basics. Also, consider [networking services limitations](#).

About subnets

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

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

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

About local network gateways

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

About gateway types

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

The gateway types are:

- Vpn
- ExpressRoute

About connection types

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

- IPsec
- Vnet2Vnet
- ExpressRoute
- VPNClient

About VPN types

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

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

Gateway configurations require a RouteBased VPN.

- **RouteBased:** RouteBased VPNs were previously called dynamic routing gateways in the classic deployment model. RouteBased VPNs use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels. The policy (or traffic selector) for RouteBased VPNs are configured as any-to-any (or wild cards). The value for a RouteBased VPN type is *RouteBased*.

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

VPN type - Resource Manager deployment model

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

VPN type - classic deployment model

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

VPN devices for Site-to-Site connections

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

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

You need to have experience configuring your VPN device, or have someone that can configure the device for you. For more information about VPN devices, see [About VPN devices](#). The VPN devices article contains information about validated devices, requirements for devices that have not been validated, and links to device configuration documents if available.

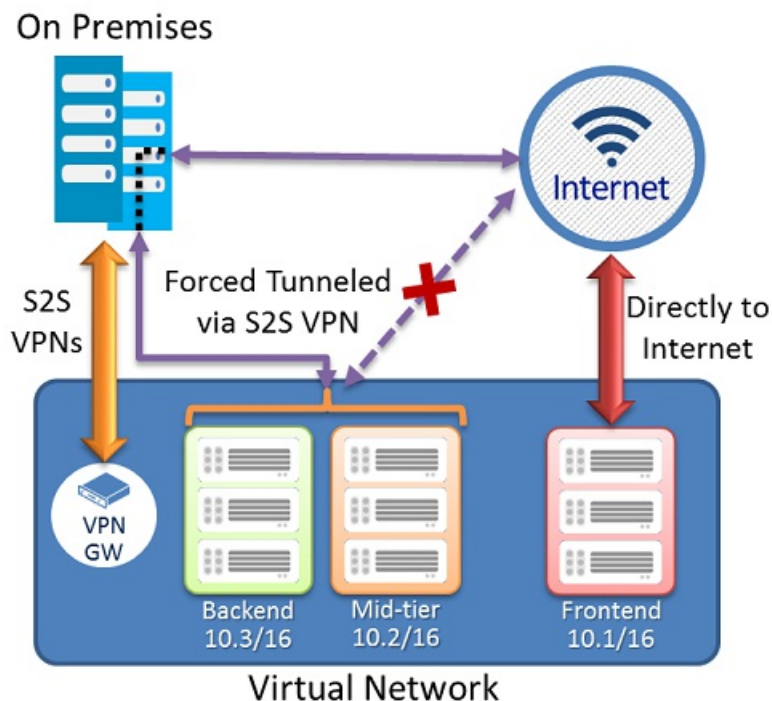
Consider forced tunnel routing

For most configurations, you can configure forced tunneling. Forced tunneling lets you redirect or "force" all

Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies.

Without forced tunneling, Internet-bound traffic from your VMs in Azure will always traverse from Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic. Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches.

Forced tunneling diagram



A forced tunneling connection can be configured in both deployment models and by using different tools. See the following table for more information. We update this table as new articles, new deployment models, and additional tools become available for this configuration. When an article is available, we link directly to it from the table.

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

Next steps

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

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

About VPN Gateway settings

1/17/2017 • 9 min to read • [Edit on GitHub](#)

A VPN gateway connection solution relies on the configuration of multiple resources in order to send network traffic between virtual networks and on-premises locations. Each resource contains configurable settings. The combination of the resources and settings determines the connection outcome.

The sections in this article discuss the resources and settings that relate to a VPN gateway in the **Resource Manager** deployment model. You may find it helpful to view the available configurations by using connection topology diagrams. You can find the descriptions and topology diagrams for each connection solution in the [About VPN Gateway](#) article.

Gateway types

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

The available values for `-GatewayType` are:

- Vpn
- ExpressRoute

A VPN gateway requires the `-GatewayType Vpn`.

Example:

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

Gateway SKUs

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. When you select a higher gateway SKU, more CPUs and network bandwidth are allocated to the gateway, and as a result, the gateway can support higher network throughput to the virtual network.

VPN Gateway can use the following SKUs:

- Basic
- Standard
- HighPerformance

VPN Gateway does not use the UltraPerformance gateway SKU. For information about the UltraPerformance SKU, see the [ExpressRoute](#) documentation.

When selecting a SKU, 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.

Configuring the gateway SKU

Specifying the gateway SKU in the 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.

For example, if you select the gateway type 'VPN' and the VPN type 'Policy-based', you see only the 'Basic' SKU because that is the only SKU available for PolicyBased VPNs. If you select 'Route-based', you can select from Basic, Standard, and HighPerformance SKUs.

Specifying the gateway SKU using PowerShell

The following PowerShell example specifies the `-GatewaySku` as *Standard*.

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

Changing a gateway SKU

If you want to upgrade your gateway SKU to a more powerful SKU (from Basic/Standard to HighPerformance), you can use the `Resize-AzureRmVirtualNetworkGateway` PowerShell cmdlet. You can also downgrade the gateway SKU size using this cmdlet.

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

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

Estimated aggregate throughput by gateway SKU and type

The following table shows the gateway types and the estimated aggregate throughput by gateway SKU. This table applies to both 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)	100 Mbps	10	500 Mbps	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.

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

VPN types

When you create the virtual network gateway for a VPN gateway configuration, you must specify a VPN type. The VPN type that you choose depends on the connection topology that you want to create. For example, a P2S connection requires a RouteBased VPN type. A VPN type can also depend on the hardware that you will be using. S2S configurations require a VPN device. Some VPN devices only support a certain VPN type.

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

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

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

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


```
New-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg `
-Location 'West US' -IpConfigurations $gwipconfig `
-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

Gateway subnet

To configure a virtual network gateway, you first need to create a gateway subnet for your VNet. The gateway subnet must be named *GatewaySubnet* to work properly. This name lets Azure know that this subnet should be used for the gateway.

The minimum size of your gateway subnet depends entirely on the configuration that you want to create. Although it is possible to create a gateway subnet as small as /29, we recommend that you create a gateway subnet of /28 or larger (/28, /27, /26, etc.).

Creating a larger gateway size prevents you from running up against gateway size limitations. For example, you may have created a virtual network gateway with a gateway subnet size /29 for a S2S connection. You now want to configure a S2S/ExpressRoute coexist configuration. That configuration requires a gateway subnet minimum size /28. To create your configuration, you would have to modify the gateway subnet to accommodate the minimum requirement for the connection, which is /28.

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

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

IMPORTANT

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

Local network gateways

When creating a VPN gateway configuration, the local network gateway often represents your on-premises location. In the classic deployment model, the local network gateway was referred to as a Local Site.

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

The following PowerShell example creates a new local network gateway:

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

Sometimes you need to modify the local network gateway settings. For example, when you add or modify the address range, or if the IP address of the VPN device changes. For a classic VNet, you can change these settings in the classic portal on the Local Networks page. For Resource Manager, see [Modify local network gateway settings using PowerShell](#).

REST APIs and PowerShell cmdlets

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

CLASSIC	RESOURCE MANAGER
PowerShell	PowerShell
REST API	REST API

Next steps

See [About VPN Gateway](#) for more information about available connection configurations.

About VPN devices for Site-to-Site VPN Gateway connections

1/17/2017 • 6 min to read • [Edit on GitHub](#)

A VPN device is required to configure a Site-to-Site (S2S) VPN connection. Site-to-Site connections can be used to create a hybrid solution, or whenever you want a secure connection between your on-premises network and your virtual network. This article discusses compatible VPN devices and configuration parameters.

NOTE

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

If your device doesn't appear in the [Validated VPN devices](#) table, see the [Non-validated VPN devices](#) section of this article. It's possible that your device may still work with Azure. For VPN device support, please contact your device manufacturer.

Items to note when viewing the tables:

- There has been a terminology change for static and dynamic routing. You'll likely run into both terms. There is no functionality change, only the names are changing.
 - Static Routing = PolicyBased
 - Dynamic Routing = RouteBased
- Specifications for High Performance 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 Azure High Performance VPN gateway.

Validated VPN devices

We have validated a set of standard VPN devices in partnership with device vendors. All the devices in the device families contained in the following list should work with Azure VPN gateways. See [About VPN Gateway](#) to verify the type of gateway that you need to create for the solution you want to configure.

To help configure your VPN device, refer to the links that correspond to appropriate device family. For VPN device support, please contact your device manufacturer.

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED	ROUTEBASED
Allied Telesis	AR Series VPN Routers	2.9.2	Coming soon	Not compatible
Barracuda Networks, Inc.	Barracuda NextGen Firewall F-series	PolicyBased: 5.4.3 RouteBased: 6.2.0	Configuration instructions	Configuration instructions
Barracuda Networks, Inc.	Barracuda NextGen Firewall X-series	Barracuda Firewall 6.5	Barracuda Firewall	Not compatible
Brocade	Vyatta 5400 vRouter	Virtual Router 6.6R3 GA	Configuration instructions	Not compatible

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED	ROUTEBASED
Check Point	Security Gateway	R75.40 R75.40VS	Configuration instructions	Configuration instructions
Cisco	ASA	8.3	Cisco samples	Not compatible
Cisco	ASR	PolicyBased: IOS 15.1 RouteBased: IOS 15.2	Cisco samples	Cisco samples
Cisco	ISR	PolicyBased: IOS 15.0 RouteBased*: IOS 15.1	Cisco samples	Cisco samples*
Citrix	NetScaler MPX, SDX, VPX	10.1 and above	Integration instructions	Not compatible
Dell SonicWALL	TZ Series, NSA Series SuperMassive Series E-Class NSA Series	SonicOS 5.8.x SonicOS 5.9.x SonicOS 6.x	Configuration guide for SonicOS 6.2 Configuration guide for SonicOS 5.9	Configuration guide for SonicOS 6.2 Configuration guide for SonicOS 5.9
F5	BIG-IP series	12.0	Configuration instructions	Configuration instructions
Fortinet	FortiGate	FortiOS 5.4.x	Configuration instructions	Configuration instructions
Internet Initiative Japan (IIJ)	SEIL Series	SEIL/X 4.60 SEIL/B1 4.60 SEIL/x86 3.20	Configuration instructions	Not compatible
Juniper	SRX	PolicyBased: JunOS 10.2 Routebased: JunOS 11.4	Juniper samples	Juniper samples
Juniper	J-Series	PolicyBased: JunOS 10.4r9 RouteBased: JunOS 11.4	Juniper samples	Juniper samples
Juniper	ISG	ScreenOS 6.3	Juniper samples	Juniper samples
Juniper	SSG	ScreenOS 6.2	Juniper samples	Juniper samples
Microsoft	Routing and Remote Access Service	Windows Server 2012	Not compatible	Microsoft samples
Open Systems AG	Mission Control Security Gateway	N/A	Installation guide	Installation guide
Openswan	Openswan	2.6.32	(Coming soon)	Not compatible

VENDOR	DEVICE FAMILY	MINIMUM OS VERSION	POLICYBASED	ROUTEBASED
Palo Alto Networks	All devices running PAN-OS	PAN-OS PolicyBased: 6.1.5 or later RouteBased: 7.0.5 or later	Configuration instructions	Configuration instructions
WatchGuard	All	Fireware XTM PolicyBased: v11.11.x RouteBased: v11.12.x	Configuration instructions	Configuration instructions

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

Non-validated VPN devices

If you don't see your device listed in the Validated VPN devices table, it still may work with a Site-to-Site connection. Verify that your VPN device meets the minimum requirements outlined in the Gateway Requirements section of the [About VPN Gateway](#) article. Devices meeting the minimum requirements should also work well with VPN gateways. 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
<code><RP_OnPremisesNetwork></code>	Your chosen name for this object. Example: myOnPremisesNetwork
<code><RP_AzureNetwork></code>	Your chosen name for this object. Example: myAzureNetwork
<code><RP_AccessList></code>	Your chosen name for this object. Example: myAzureAccessList
<code><RP_IPSecTransformSet></code>	Your chosen name for this object. Example: myIPSecTransformSet
<code><RP_IPSecCryptoMap></code>	Your chosen name for this object. Example: myIPSecCryptoMap
<code><SP_AzureNetworkIpRange></code>	Specify range. Example: 192.168.0.0
<code><SP_AzureNetworkSubnetMask></code>	Specify subnet mask. Example: 255.255.0.0
<code><SP_OnPremisesNetworkIpRange></code>	Specify on-premises range. Example: 10.2.1.0

SAMPLE TEXT	CHANGE TO
<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 Parameters

NOTE

Although the values listed in the following table are supported by the Azure VPN Gateway, currently there is no way for you to specify or select a specific combination from the Azure VPN Gateway. You must specify any constraints from the on-premises VPN device. In addition, you must clamp MSS at 1350.

IKE Phase 1 setup

PROPERTY	POLICYBASED	ROUTEBASED AND STANDARD OR HIGH PERFORMANCE VPN GATEWAY
IKE Version	IKEv1	IKEv2
Diffie-Hellman Group	Group 2 (1024 bit)	Group 2 (1024 bit)
Authentication Method	Pre-Shared Key	Pre-Shared Key
Encryption Algorithms	AES256 AES128 3DES	AES256 3DES
Hashing Algorithm	SHA1(SHA128)	SHA1(SHA128), SHA2(SHA256)
Phase 1 Security Association (SA) Lifetime (Time)	28,800 seconds	10,800 seconds

IKE Phase 2 setup

PROPERTY	POLICYBASED	ROUTEBASED AND STANDARD OR HIGH PERFORMANCE VPN GATEWAY
IKE Version	IKEv1	IKEv2
Hashing Algorithm	SHA1(SHA128)	SHA1(SHA128)
Phase 2 Security Association (SA) Lifetime (Time)	3,600 seconds	3,600 seconds
Phase 2 Security Association (SA) Lifetime (Throughput)	102,400,000 KB	-
IPsec SA Encryption & Authentication Offers (in the order of preference)	1. ESP-AES256 2. ESP-AES128 3. ESP-3DES 4. N/A	See <i>RouteBased Gateway IPsec Security Association (SA) Offers</i> (below)

PROPERTY	POLICYBASED	ROUTEBASED AND STANDARD OR HIGH PERFORMANCE VPN GATEWAY
Perfect Forward Secrecy (PFS)	No	No (*)
Dead Peer Detection	Not supported	Supported

(*) Azure Gateway as IKE responder can accept PFS DH Group 1, 2, 5, 14, 24.

RouteBased Gateway IPsec Security Association (SA) Offers

The following table lists IPsec SA Encryption and Authentication Offers. Offers are listed the order of preference that the offer is presented or accepted.

IPSEC SA ENCRYPTION AND AUTHENTICATION OFFERS	AZURE GATEWAY AS INITIATOR	AZURE GATEWAY AS RESPONDER
1	ESP AES_256 SHA	ESP AES_128 SHA
2	ESP AES_128 SHA	ESP 3_DES MD5
3	ESP 3_DES MD5	ESP 3_DES SHA
4	ESP 3_DES SHA	AH SHA1 with ESP AES_128 with null HMAC
5	AH SHA1 with ESP AES_256 with null HMAC	AH SHA1 with ESP 3_DES with null HMAC
6	AH SHA1 with ESP AES_128 with null HMAC	AH MD5 with ESP 3_DES with null HMAC, no lifetimes proposed
7	AH SHA1 with ESP 3_DES with null HMAC	AH SHA1 with ESP 3_DES SHA1, no lifetimes
8	AH MD5 with ESP 3_DES with null HMAC, no lifetimes proposed	AH MD5 with ESP 3_DES MD5, no lifetimes
9	AH SHA1 with ESP 3_DES SHA1, no lifetimes	ESP DES MD5
10	AH MD5 with ESP 3_DES MD5, no lifetimes	ESP DES SHA1, no lifetimes
11	ESP DES MD5	AH SHA1 with ESP DES null HMAC, no lifetimes proposed
12	ESP DES SHA1, no lifetimes	AH MD5 with ESP DES null HMAC, no lifetimes proposed
13	AH SHA1 with ESP DES null HMAC, no lifetimes proposed	AH SHA1 with ESP DES SHA1, no lifetimes
14	AH MD5 with ESP DES null HMAC, no lifetimes proposed	AH MD5 with ESP DES MD5, no lifetimes

IPSEC SA ENCRYPTION AND AUTHENTICATION OFFERS	AZURE GATEWAY AS INITIATOR	AZURE GATEWAY AS RESPONDER
15	AH SHA1 with ESP DES SHA1, no lifetimes	ESP SHA, no lifetimes
16	AH MD5 with ESP DES MD5, no lifetimes	ESP MD5, no lifetimes
17	-	AH SHA, no lifetimes
18	-	AH MD5, no lifetimes

- You can specify IPsec ESP NULL encryption with RouteBased and High Performance 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.

Overview of BGP with Azure VPN Gateways

1/17/2017 • 7 min to read • [Edit on GitHub](#)

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

BGP is the standard routing protocol commonly used in the Internet to exchange routing and reachability information between two or more networks. When used in the context of Azure Virtual Networks, BGP enables the Azure VPN Gateways and your on-premises VPN devices, called BGP peers or neighbors, to exchange "routes" that will inform both gateways on the availability and reachability for those prefixes to go through the gateways or routers involved. BGP can also enable transit routing among multiple networks by propagating routes a BGP gateway learns from one BGP peer to all other BGP peers.

Why use BGP?

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

There are several advantages and new capabilities with BGP:

Support automatic and flexible prefix updates

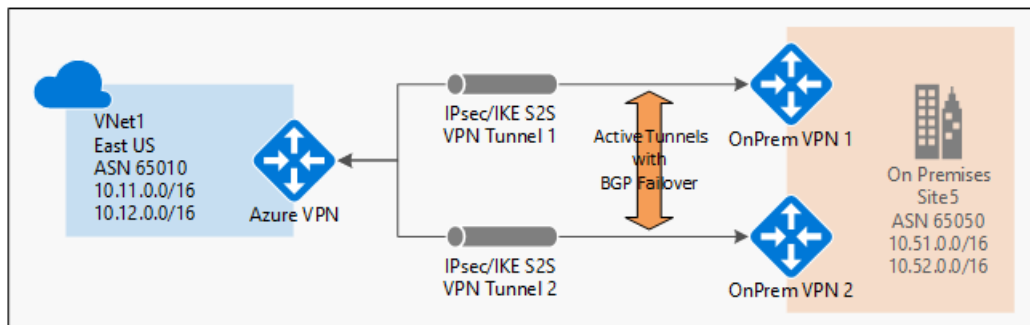
With BGP, you only need to declare a minimum prefix to a specific BGP peer over the IPsec S2S VPN tunnel. It can be as small as a host prefix (/32) of the BGP peer IP address of your on-premises VPN device. You can control which on-premises network prefixes you want to advertise to Azure to allow your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

You can also advertise a larger prefixes that may include some of your VNet address prefixes, such as a large private IP address space (e.g., 10.0.0.0/8). Please 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 will automatically shift to the remaining tunnels.

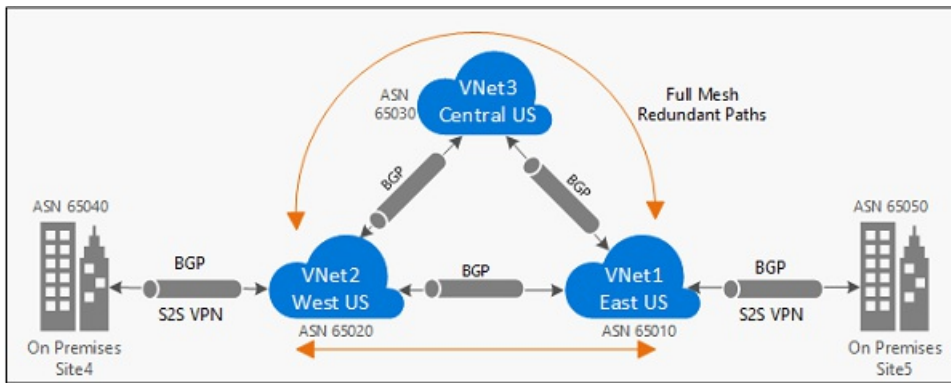
The following diagram shows a simple example of this highly available setup:



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 **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.

Are there ASNs reserved by Azure?

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

- Public ASNs: 8075, 8076, 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.

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 for not for your cross-premises connectivity. You can override this default by assigning a different ASN when creating the VPN gateway, or change the ASN after the gateway is created. You will need to assign your on-premises ASNs to the corresponding Azure Local Network Gateways.

What address prefixes will Azure VPN gateways advertise to me?

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

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

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

Yes.

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

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

Network.

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

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

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

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

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

Does Azure VPN gateway support BGP transit routing?

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

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

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

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

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

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

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

Yes.

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

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

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

Your on-premises BGP peer address **MUST NOT** be the same as the public IP address of your VPN device. Use a different IP address on the VPN device for your BGP Peer IP. It can be an address assigned to the loopback interface on the device. 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.

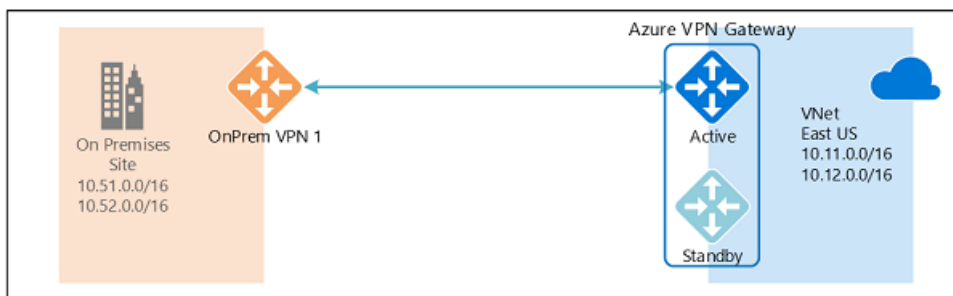
Highly Available Cross-Premises and VNet-to-VNet Connectivity

1/17/2017 • 5 min to read • [Edit on GitHub](#)

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.



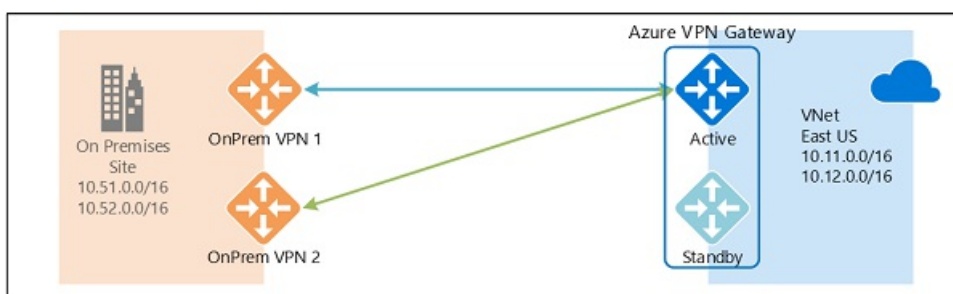
Highly Available Cross-Premises Connectivity

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

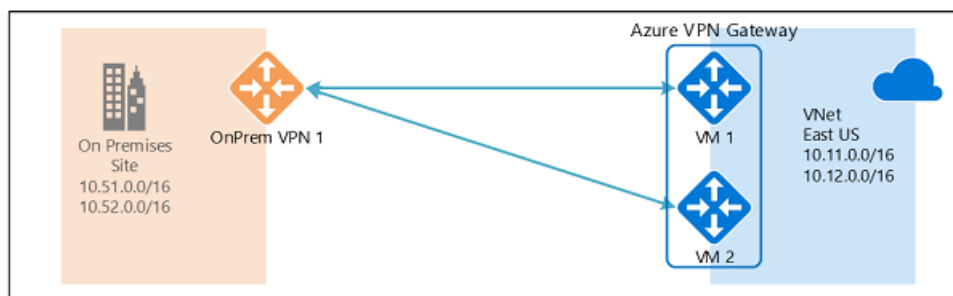
VPN devices from the same on-premises network to Azure, you need to create one local network gateway for each VPN device, and one connection from your Azure VPN gateway to the local network gateway.

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

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

Active-active Azure VPN gateway

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



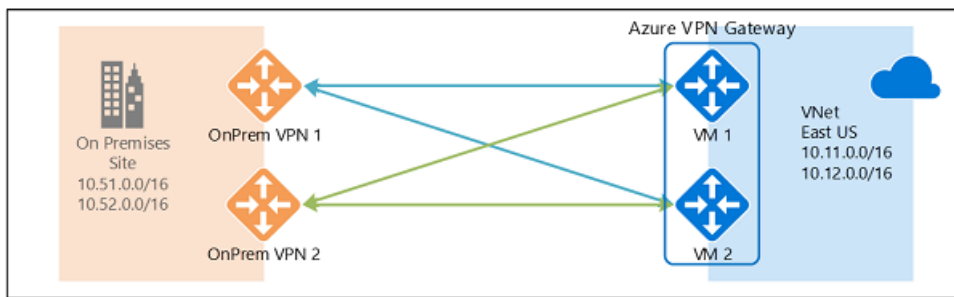
In this configuration, each Azure gateway instance will have a unique public IP address, and each will establish an IPsec/IKE S2S VPN tunnel to your on-premises VPN device specified in your local network gateway and connection. Note that both VPN tunnels are actually part of the same connection. You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

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

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

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

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



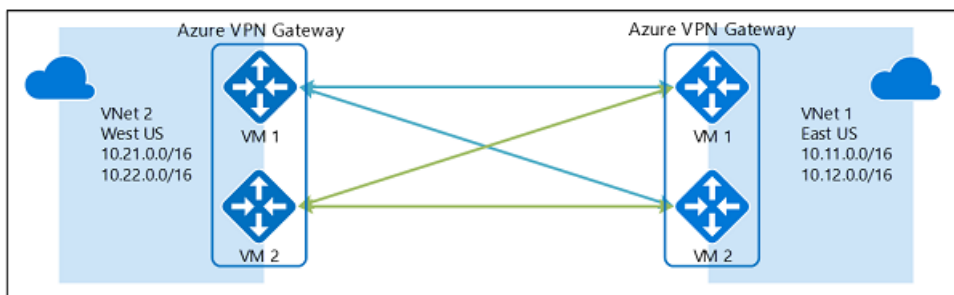
Here you create and setup the Azure VPN gateway in an active-active configuration, and create two local network gateways and two connections for your two on-premises VPN devices as described above. The result is a full mesh connectivity of 4 IPsec tunnels between your Azure virtual network and your on-premises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

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

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

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

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



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

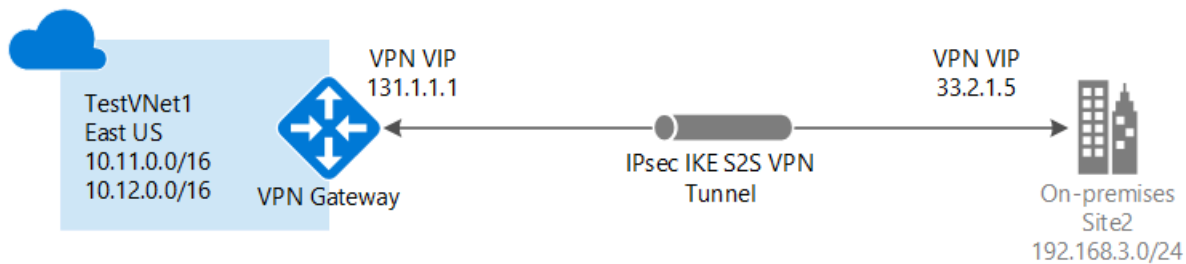
Next steps

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

Create a VNet with a Site-to-Site connection using the Azure portal

1/17/2017 • 14 min to read • [Edit on GitHub](#)

This article walks you through creating a virtual network and a Site-to-Site VPN gateway connection to your on-premises network using the Azure Resource Manager deployment model and the Azure portal. Site-to-Site connections can be used for cross-premises and hybrid configurations.



Deployment models and methods for Site-to-Site connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for Site-to-Site configurations. When an article with configuration steps is available, we link directly to it from this table.

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

(*) denotes that the classic portal can only support creating one S2S VPN connection.

(**) denotes that an end-to-end scenario is not yet available for the Azure portal.

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

Additional configurations

If you want to connect VNets together, but are not creating a connection to an on-premises location, see [Configure a VNet-to-VNet connection](#). If you want to add a Site-to-Site connection to a VNet that already has a connection, see [Add a S2S connection to a VNet with an existing VPN gateway connection](#).

Before you begin

Verify that you have the following items before beginning your configuration:

- A 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.
- An externally facing public IP address for your VPN device. This IP address cannot be located behind a NAT.
- 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](#).

Sample configuration values for this exercise

When using these steps as an exercise, you can use the sample configuration values:

- **VNet Name:** TestVNet1
- **Address Space:** 10.11.0.0/16 and 10.12.0.0/16
- **Subnets:**
 - FrontEnd: 10.11.0.0/24
 - BackEnd: 10.12.0.0/24
 - GatewaySubnet: 10.12.255.0/27
- **Resource Group:** TestRG1
- **Location:** East US
- **DNS Server:** 8.8.8.8
- **Gateway Name:** VNet1GW
- **Public IP:** VNet1GWIP
- **VPN Type:** Route-based
- **Connection Type:** Site-to-site (IPsec)
- **Gateway Type:** VPN
- **Local Network Gateway Name:** Site2
- **Connection Name:** VNet1toSite2

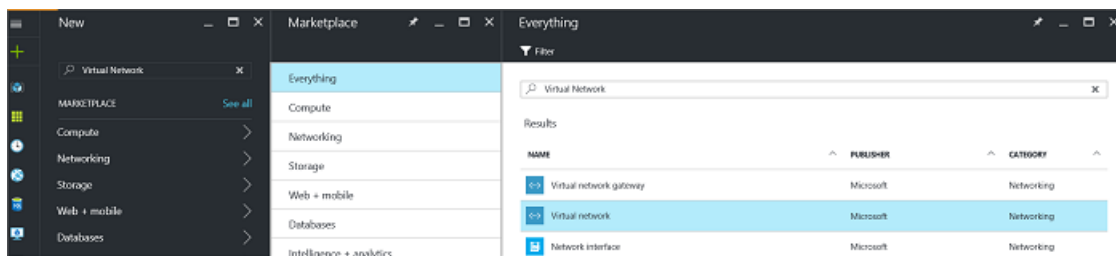
1. Create a virtual network

If you already have a VNet, verify that the settings are compatible with your VPN gateway design. Pay particular attention to any subnets that may overlap with other networks. If you have overlapping subnets, your connection won't work properly. If your VNet is configured with the correct settings, you can begin the steps in the [Specify a DNS server](#) section.

To create a virtual network

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

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



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

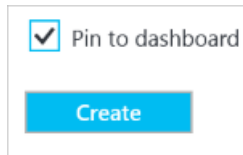
4. On the **Create virtual network** blade, configure the VNet settings. When you fill in the fields, the red exclamation mark will become a green check mark when the characters entered in the field are valid.

5. The **Create virtual network** blade looks similar to the following example. There may be values that are auto-filled. If so, replace the values with your own.

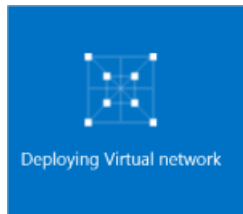
6. **Name**: Enter the name for your Virtual Network.
7. **Address space**: Enter the address space. If you have multiple address spaces to add, add your first address

space. You can add additional address spaces later, after creating the VNet.

8. **Subnet name:** Add the subnet name and subnet address range. You can add additional subnets later, after creating the VNet.
9. **Subscription:** Verify that the Subscription listed is the correct one. You can change subscriptions by using the drop-down.
10. **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](#).
11. **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
12. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



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

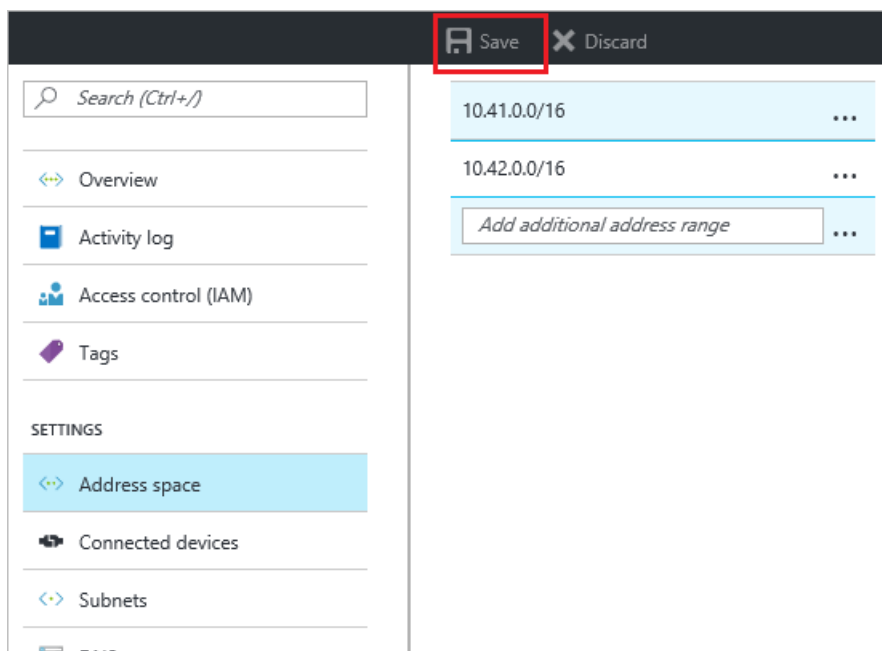


2. Add additional address space and subnets

You can add additional address space and subnets to your VNet once it has been created.

To add address space

1. To add additional address space, under the **Settings** section for your virtual network blade, click **Address space** to open the Address space blade.
2. Add the additional address space, and then click **Save** at the top of the blade.



To create subnets

1. To create subnets, in the **Settings** section of your virtual network blade, click **Subnets** to open the **Subnets** blade.
2. In the Subnets blade, click **+Subnet** to open the **Add subnet** blade. Name your new subnet and specify the address range.

NAME	ADDRESS RANGE	AVAILABLE ADDR...	SECURITY GROUP
FrontEnd	10.41.0.0/24	251	-

Name: BackEnd ✓

Address range (CIDR block): 10.42.0.0/24 ✓
10.42.0.0 - 10.42.0.255 (256 addresses)

3. Click **OK** at the bottom of the blade to save your changes.



3. Specify a DNS server

To specify a DNS server

This setting allows you to specify the DNS server that you want to use for name resolution for this virtual network. It does not create a DNS server.

1. On the **Settings** page for your virtual network, navigate to **DNS Servers** and click to open the DNS servers blade.
2. On the **DNS Servers** page, under **DNS servers**, select **Custom**.
3. In the **DNS Server** field, in the **Add DNS server** box, enter the IP address of the DNS server that you want to use for name resolution.
4. When you are done adding DNS servers, click **Save** at the top of the blade to save your configuration.

Save Discard

Search (Ctrl+/,)

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Address space

Connected devices

Subnets

DNS servers

DNS servers ⓘ

☐ Default (Azure-provided)

☒ Custom

DNS SERVER

Add DNS server ...

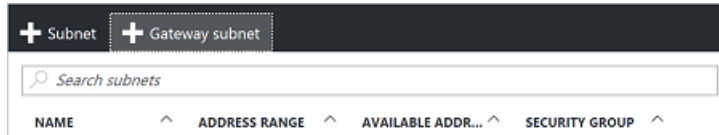
4. Create a gateway subnet

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

If you are creating this configuration as an exercise, refer to these [values](#) when creating your gateway subnet.

To create a gateway subnet

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



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



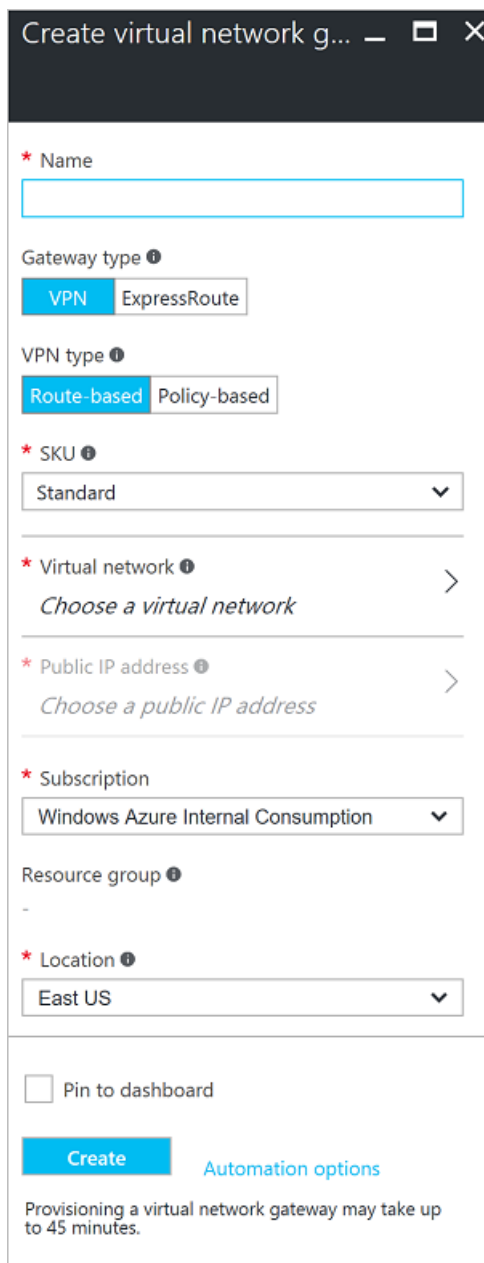
5. Click **OK** at the bottom of the blade to create the subnet.

5. Create a virtual network gateway

If you are creating this configuration as an exercise, you can refer to the [sample configuration values](#).

To create a virtual network gateway

1. In the portal, on the left side, click **+** and type "Virtual Network Gateway" in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** blade, click **Create** at the bottom of the blade. This opens the **Create virtual network gateway** blade.
2. On the **Create virtual network gateway** blade, fill in the values for your virtual network gateway.



Create virtual network gateway

* Name

Gateway type ⓘ

VPN ExpressRoute

VPN type ⓘ

Route-based Policy-based

* SKU ⓘ

Standard

* Virtual network ⓘ

Choose a virtual network

* Public IP address ⓘ

Choose a public IP address

* Subscription

Windows Azure Internal Consumption

Resource group ⓘ

-

* Location ⓘ

East US

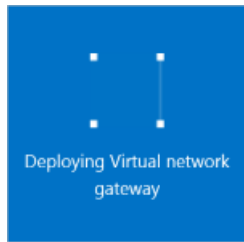
☐ Pin to dashboard

Create Automation options

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

3. **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
4. **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
5. **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
6. **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select.
7. **Location:** Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, the virtual network will not appear in the 'Choose a virtual network' dropdown.
8. Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the **Choose a virtual network** blade. Select the VNet. If you don't see your VNet, make sure the **Location** field is pointing to the region in which your virtual network is located.
9. Choose a public IP address. Click **Public IP address** to open the **Choose public IP address** blade. Click **+Create New** to open the **Create public IP address blade**. Input a name for your public IP address. This blade creates a public IP address object to which a public IP address will be dynamically assigned. Click **OK** to save your changes to this blade.
10. **Subscription:** Verify that the correct subscription is selected.
11. **Resource group:** This setting is determined by the Virtual Network that you select.

12. Don't adjust the **Location** after you've specified the previous settings.
13. Verify the settings. You can select **Pin to dashboard** at the bottom of the blade if you want your gateway to appear on the dashboard.
14. Click **Create** to begin creating the gateway. The settings will be validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.



15. 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 will appear as a connected device. You can click the connected device (your virtual network gateway) to view more information.

6. Create a local network gateway

The 'local network gateway' refers to your on-premises location. Give the local network gateway a name by which Azure can refer to it.

If you are creating this configuration as an exercise, you can refer to the [sample configuration values](#).

To create a local network gateway

1. In the portal, from **All resources**, click **+Add**. In the **Everything** blade search box, type **Local network gateway**, then click to search. This will return a list. Click **Local network gateway** to open the blade, then click **Create** to open the **Create local network gateway** blade.

Create local network gateway

* Name
LocalNetworkName ✓

* IP address ⓘ
33.2.1.5 ✓

Address space ⓘ
192.168.3.0/24 ...
Add additional address range ...

* Subscription
Windows Azure Internal Consumption ▼

* Resource group ⓘ
☐ Create new ☒ Use existing
TestRG1 ▼

* Location
East US ▼

☐ Pin to dashboard

Create Automation options

2. On the **Create local network gateway blade**, specify a **Name** for your local network gateway object.
3. Specify a valid public **IP address** for the VPN device or virtual network gateway to which you want to connect. If this local network represents an on-premises location, this is the public IP address of the VPN device that you want to connect to. It cannot be behind NAT and has to be reachable by Azure. If this local network represents another VNet, you will specify the public IP address that was assigned to the virtual network gateway for that VNet.
4. **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.
5. For **Subscription**, verify that the correct subscription is showing.
6. For **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.
7. For **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.
8. Click **Create** to create the local network gateway.

7. Configure your VPN device

To configure your VPN device, you'll need the public IP address of the virtual network gateway for configuring your on-premises VPN device. Work with your device manufacturer for specific configuration information and configure your device. Refer to the [VPN Devices](#) for more information about VPN devices that work well with

Azure.

To find the public IP address of your virtual network gateway using PowerShell, use the following sample:

```
Get-AzureRmPublicIpAddress -Name GW1PublicIP -ResourceGroupName TestRG
```

You can also view the public IP address for your virtual network gateway by using the Azure portal. Navigate to **Virtual network gateways**, then click the name of your gateway.

8. Create a Site-to-Site VPN connection

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.

Before beginning this section, verify that your virtual network gateway and local network gateways have finished creating. If you are creating this configuration as an exercise, refer to these [values](#) when creating your connection.

To create the VPN connection

1. Locate your virtual network gateway and click **All settings** to open the **Settings** blade.
2. On the **Settings** blade, click **Connections**, and then click **Add** at the top of the blade to open the **Add connection** blade.



Add connection
VNetGW

* Name
Vnet1s2s ✓

Connection type
Site-to-site (IPsec) ▼

* Virtual network gateway
VNetGW

* Local network gateway
VNet1LocalNet >

* Shared key (PSK)
87654321 ✓

Subscription
Windows Azure Internal Consumption ▼

Resource group
TestRG1

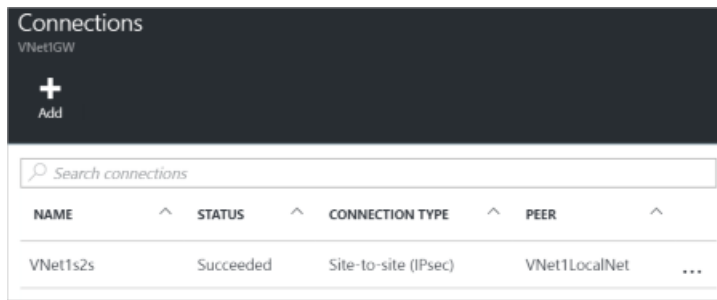
New

Location
East US ▼

3. On the **Add connection** blade, **Name** your connection.
4. For **Connection type**, select **Site-to-site(IPSec)**.
5. For **Virtual network gateway**, the value is fixed because you are connecting from this gateway.
6. For **Local network gateway**, click **Choose a local network gateway** and select the local network gateway that you want to use.
7. For **Shared Key**, the value here must match the value that you are using for your local VPN device. If your VPN device on your local network doesn't provide a shared key, you can make one up and input it here and on your

local device. The important thing is that they both match.

- The remaining values for **Subscription**, **Resource Group**, and **Location** are fixed.
- Click **OK** to create your connection. You'll see *Creating Connection* flash on the screen.
- When the connection is complete, you'll see it appear in the **Connections** blade for your Gateway.



NAME	STATUS	CONNECTION TYPE	PEER
VNet1s2s	Succeeded	Site-to-site (IPsec)	VNet1LocalNet

9. Verify the VPN connection

You can verify your VPN connection either in the portal, or by using PowerShell.

To verify your connection by using PowerShell

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```



- 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.

```
Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1",
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite",
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}
```

To verify your connection by using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	 2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	 RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

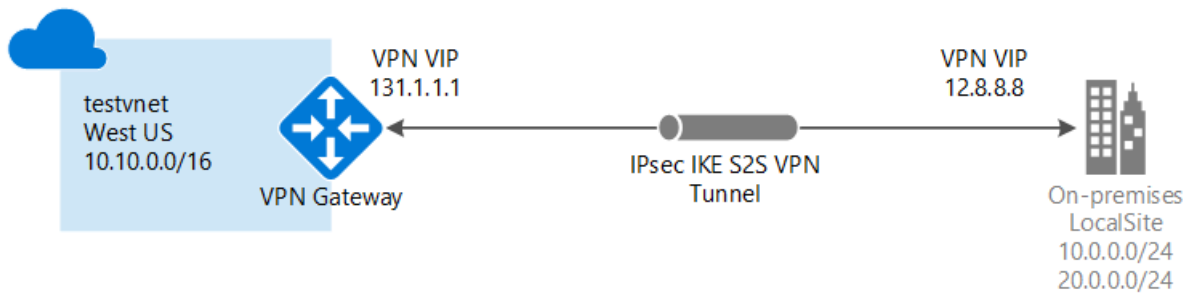
Next steps

- 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](#).

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

1/17/2017 • 14 min to read • [Edit on GitHub](#)

This article walks you through creating a virtual network and a Site-to-Site VPN gateway connection to your on-premises network using the Azure Resource Manager deployment model. Site-to-Site connections can be used for cross-premises and hybrid configurations.



Deployment models and methods for Site-to-Site connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for Site-to-Site configurations. When an article with configuration steps is available, we link directly to it from this table.

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

(*) denotes that the classic portal can only support creating one S2S VPN connection.

(**) denotes that an end-to-end scenario is not yet available for the Azure portal.

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

Additional configurations

If you want to connect VNets together, but are not creating a connection to an on-premises location, see [Configure a VNet-to-VNet connection](#). If you want to add a Site-to-Site connection to a VNet that already has a connection, see [Add a S2S connection to a VNet with an existing VPN gateway connection](#).

Before you begin

Verify that you have the following items before beginning configuration.

- A 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.
- An externally facing public IP address for your VPN device. This IP address cannot be located behind a NAT.
- 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](#).
- 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.

1. Connect to your subscription

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:

```
Login-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

2. Create a virtual network and a gateway subnet

The examples use a gateway subnet of /28. 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.

If you already have a virtual network with a gateway subnet that is /29 or larger, you can jump ahead to [Add your local network gateway](#).

IMPORTANT

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

To create a virtual network and a gateway subnet

Use the following sample to create a virtual network and a gateway subnet. Substitute the values for your own.

First, create a resource group:

```
New-AzureRmResourceGroup -Name testrg -Location 'West US'
```

Next, create your virtual network. Verify that the address spaces you specify don't overlap any of the address

spaces that you have on your on-premises network.

The following sample creates a virtual network named *testvnet* and two subnets, one called *GatewaySubnet* and the other called *Subnet1*. It's important to create one subnet named specifically *GatewaySubnet*. If you name it something else, your connection configuration will fail.

Set the variables.

```
$subnet1 = New-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.0.0.0/28
$subnet2 = New-AzureRmVirtualNetworkSubnetConfig -Name 'Subnet1' -AddressPrefix '10.0.1.0/28'
```

Create the VNet.

```
New-AzureRmVirtualNetwork -Name testvnet -ResourceGroupName testrg `
-Location 'West US' -AddressPrefix 10.0.0.0/16 -Subnet $subnet1, $subnet2
```

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

This step is required only if you need to add a gateway subnet to a VNet that you previously created.

You can create your gateway subnet by using the following sample. Be sure to name the gateway subnet 'GatewaySubnet'. If you name it something else, you create a subnet, but Azure won't treat it as a gateway subnet.

Set the variables.

```
$vnet = Get-AzureRmVirtualNetwork -ResourceGroupName testrg -Name testvnet
```

Create the gateway subnet.

```
Add-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.0.3.0/28 -VirtualNetwork $vnet
```

Set the configuration.

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

3. Add your local network gateway

In a virtual network, the local network gateway typically refers to your on-premises location. You give the site a name by which Azure can refer to it, and also specify the address space prefix for the local network gateway.

Azure uses the IP address prefix you specify to identify which traffic to send to your on-premises location. This means that you have to specify each address prefix that you want to be associated with your local network gateway. You can easily update these prefixes if your on-premises network changes.

When using the PowerShell examples, note the following:

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

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

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

To add a local network gateway with multiple address prefixes:

```
New-AzureRmLocalNetworkGateway -Name LocalSite -ResourceGroupName testrg `
-Location 'West US' -GatewayIpAddress '23.99.221.164' -AddressPrefix @( '10.0.0.0/24','20.0.0.0/24')
```

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.

4. Request a public IP address for the VPN gateway

Next, request a public IP address to be allocated to your Azure VNet VPN gateway. This is not the same IP address that is assigned to your VPN device; rather it's assigned to the Azure VPN gateway itself. You can't specify the IP address that you want to use. It is dynamically allocated to your gateway. You use this IP address when configuring your on-premises VPN device to connect to the gateway.

The Azure VPN gateway for the Resource Manager deployment model currently only supports public IP addresses by using the Dynamic Allocation method. However, this does not mean the IP address will change. The only time the Azure VPN gateway IP address changes is when the gateway is deleted and re-created. The gateway public IP address won't change across resizing, resetting, or other internal maintenance/upgrades of your Azure VPN gateway.

Use the following PowerShell sample:

```
$gwpip= New-AzureRmPublicIpAddress -Name gwpip -ResourceGroupName testrg -Location 'West US' -AllocationMethod Dynamic
```

5. 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.

```
$vnet = Get-AzureRmVirtualNetwork -Name testvnet -ResourceGroupName testrg
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
$gwipconfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name gwipconfig1 -SubnetId $subnet.Id -PublicIpAddressId $gwpip.Id
```

6. Create the virtual network gateway

In this step, you create the virtual network gateway. Creating a gateway can take a long time to complete. Often 45 minutes or more.

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](#).
- The *-GatewaySku* can be *Basic*, *Standard*, or *HighPerformance*.

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

7. Configure your VPN device

At this point, you need the public IP address of the virtual network gateway for configuring your on-premises VPN device. Work with your device manufacturer for specific configuration information. You can refer to the [VPN Devices](#) for more information.

To find the public IP address of your virtual network gateway, use the following sample:

```
Get-AzureRmPublicIpAddress -Name gwpip -ResourceGroupName testrg
```

8. 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-AzureRmVirtualNetworkGateway -Name vnetgw1 -ResourceGroupName testrg  
$local = Get-AzureRmLocalNetworkGateway -Name LocalSite -ResourceGroupName testrg
```

Create the connection.

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

After a short while, the connection will be established.

To verify a VPN connection

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

To verify your connection by using PowerShell

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.


```



Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1"
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite"
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}

```

To verify your connection by using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	 2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	 RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

To modify IP address prefixes for a local network gateway

If you need to change the prefixes for your local network gateway, use the following instructions. Two sets of instructions are provided. The instructions you choose depend on whether you have already created your gateway connection.

How to add or remove prefixes - no gateway connection

- **To add** additional address prefixes to a local network gateway that you created, but that doesn't yet have a gateway connection, use the example below. Be sure to change the values to your own.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @('10.0.0.0/24','20.0.0.0/24','30.0.0.0/24')
```

- **To remove** an address prefix from a local network gateway that doesn't have a VPN connection, use the example below. Leave out the prefixes that you no longer need. In this example, we no longer need prefix 20.0.0.0/24 (from the previous example), so we will update the local network gateway and exclude that prefix.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @('10.0.0.0/24','30.0.0.0/24')
```

How to add or remove prefixes - existing gateway connection

If you have created your gateway connection and want to add or remove the IP address prefixes contained in your local network gateway, you'll need to do the following steps in order. This will result in some downtime for your VPN connection. When updating your prefixes, you'll first remove the connection, modify the prefixes, and then create a new connection. In the examples below, be sure to change the values to your own.

IMPORTANT

Don't delete the VPN gateway. If you do so, you'll have to go back through the steps to recreate it, as well as reconfigure your on-premises router with the new settings.

1. Remove the connection.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName -ResourceGroupName MyRGName
```

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

Set the variable for the LocalNetworkGateway.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName
```

Modify the prefixes.

```
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @('10.0.0.0/24','20.0.0.0/24','30.0.0.0/24')
```

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

Set the variable for the VirtualNetworkGateway.

```
$gateway1 = Get-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName MyRGName
```

Create the connection. Note that this sample uses the variable \$local that you set in the preceding step.

```
New-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName `
-ResourceGroupName MyRGName -Location 'West 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 gateway IP address, use the `New-AzureRmVirtualNetworkGatewayConnection` cmdlet. As long as you keep the name of the local network gateway exactly the same as the existing name, the settings will overwrite. At this time, the "Set" cmdlet does not support modifying the gateway IP address.

How to modify the gateway IP address - no gateway connection

To update the gateway IP address for your local network gateway that doesn't yet have a connection, use the example below. You can also update the address prefixes at the same time. The settings you specify will overwrite the existing settings. Be sure to use the existing name of your local network gateway. If you don't, you'll be creating a new local network gateway, not overwriting the existing one.

Use the following example, replacing the values for your own.

```
New-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName `
-Location "West US" -AddressPrefix @( '10.0.0.0/24','20.0.0.0/24','30.0.0.0/24' ) `
-GatewayIpAddress "5.4.3.2" -ResourceGroupName MyRGName
```

How to modify the gateway IP address - existing gateway connection

If a gateway connection already exists, you'll first need to remove the connection. Then, you can modify the gateway IP address and recreate a new connection. This will result in some downtime for your VPN connection.

IMPORTANT

Don't delete the VPN gateway. If you do so, you'll have to go back through the steps to recreate it, as well as reconfigure your on-premises router with the IP address that will be assigned to the newly created gateway.

1. Remove the connection. You can find the name of your connection by using the

```
Get-AzureRmVirtualNetworkGatewayConnection
```

 cmdlet.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName `
-ResourceGroupName MyRGName
```

2. Modify the GatewayIpAddress value. You can also modify your address prefixes at this time, if necessary. Note that this will overwrite the existing local network gateway settings. Use the existing name of your local network gateway when modifying so that the settings will overwrite. If you don't, you'll be creating a new local network gateway, not modifying the existing one.

```
New-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName `
-Location "West US" -AddressPrefix @( '10.0.0.0/24','20.0.0.0/24','30.0.0.0/24' ) `
-GatewayIpAddress "104.40.81.124" -ResourceGroupName MyRGName
```

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

```
Get-AzureRmVirtualNetworkGateway
```

 cmdlet.

Set the variables:

```
$local = Get-AzureRMLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
$vnnetgw = Get-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName MyRGName
```

Create the connection:

```
New-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName -ResourceGroupName MyRGName `
-Location "West US" `
-VirtualNetworkGateway1 $vnnetgw `
-LocalNetworkGateway2 $local `
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

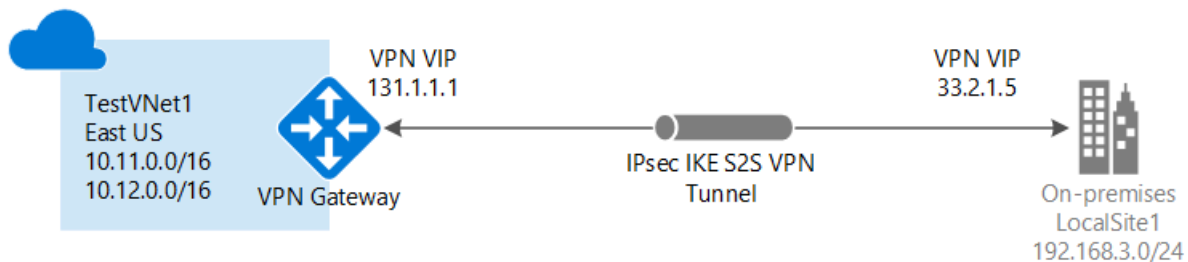
Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).

Create a VNet with a Site-to-Site connection using the Azure classic portal

1/17/2017 • 6 min to read • [Edit on GitHub](#)

This article walks you through creating a virtual network and a site-to-site VPN gateway connection to your on-premises network using the classic deployment model and the classic portal. Site-to-Site connections can be used for cross-premises and hybrid configurations.



Deployment models and methods for Site-to-Site connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for Site-to-Site configurations. When an article with configuration steps is available, we link directly to it from this table.

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

(*) denotes that the classic portal can only support creating one S2S VPN connection.

(**) denotes that an end-to-end scenario is not yet available for the Azure portal.

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

Additional configurations

If you want to connect VNets together, see [Configure a VNet-to-VNet connection for the classic deployment model](#). If you want to add a Site-to-Site connection to a VNet that already has a connection, see [Add a S2S connection to a VNet with an existing VPN gateway connection](#).

Before you begin

Verify that you have the following items before beginning configuration.

- A 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.
- An externally facing public IP address for your VPN device. This IP address cannot be located behind a NAT.
- 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](#).

Create your virtual network

1. Log in to the [Azure classic portal](#).
2. In the lower left corner of the screen, click **New**. In the navigation pane, click **Network Services**, and then click **Virtual Network**. Click **Custom Create** to begin the configuration wizard.
3. To create your VNet, enter your configuration settings on the following pages:

Virtual network details page

Enter the following information:

- **Name:** Name your virtual network. For example, *EastUSVNet*. You'll use this virtual network name when you deploy your VMs and PaaS instances, so you may not want to make the name too complicated.
- **Location:** The location is directly related to the physical location (region) where you want your resources (VMs) to reside. For example, if you want the VMs that you deploy to this virtual network to be physically located in *East US*, select that location. You can't change the region associated with your virtual network after you create it.

DNS servers and VPN connectivity page

Enter the following information, and then click the next arrow on the lower right.

- **DNS Servers:** Enter the DNS server name and IP address, or select a previously registered DNS server from the shortcut menu. 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.
- **Configure Site-To-Site VPN:** Select the checkbox for **Configure a site-to-site VPN**.
- **Local Network:** A local network represents your physical on-premises location. You can select a local network that you've previously created, or you can create a new local network. However, if you select to use a local network that you previously created, go to the **Local Networks** configuration page and verify that the VPN Device IP address (public facing IPv4 address) for the VPN device is accurate.

Site-to-site connectivity page

If you're creating a new local network, you'll see the **Site-To-Site Connectivity** page. If you want to use a local network that you previously created, this page will not appear in the wizard and you can move on to the next section.

Enter the following information, and then click the next arrow.

- **Name:** The name you want to call your local (on-premises) network site.
- **VPN Device IP Address:** The public facing IPv4 address of your on-premises VPN device that you use to connect to Azure. The VPN device cannot be located behind a NAT.
- **Address Space:** Include Starting IP and CIDR (Address Count). You specify the address range(s) that you want

to be sent through the virtual network gateway to your local on-premises location. If a destination IP address falls within the ranges that you specify here, it is routed through the virtual network gateway.

- **Add address space:** If you have multiple address ranges that you want to be sent through the virtual network gateway, specify each additional address range. You can add or remove ranges later on the **Local Network** page.

Virtual network address spaces page

Specify the address range that you want to use for your virtual network. These are the dynamic IP addresses (DIPS) that will be assigned to the VMs and other role instances that you deploy to this virtual network.

It's especially important to select a range that does not overlap with any of the ranges that are used for your on-premises network. You need to coordinate with your network administrator. Your network administrator may need to carve out a range of IP addresses from your on-premises network address space for you to use for your virtual network.

Enter the following information, and then click the checkmark on the lower right to configure your network.

- **Address Space:** Include Starting IP and Address Count. Verify that the address spaces you specify don't overlap any of the address spaces that you have on your on-premises network.
- **Add subnet:** Include Starting IP and Address Count. Additional subnets are not required, but you may want to create a separate subnet for VMs that will have static DIPS. Or you might want to have your VMs in a subnet that is separate from your other role instances.
- **Add gateway subnet:** Click to add the gateway subnet. The gateway subnet is used only for the virtual network gateway and is required for this configuration.

Click the checkmark on the bottom of the page and your virtual network will begin to create. When it completes, you will see **Created** listed under **Status** on the **Networks** page in the Azure Classic Portal. After the VNet has been created, you can then configure your virtual network gateway.

IMPORTANT

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

Configure your virtual network gateway

Configure the virtual network gateway to create a secure site-to-site connection. See [Configure a virtual network gateway in the Azure classic portal](#).

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).

Configure a Point-to-Site connection to a VNet using the Azure Portal

1/17/2017 • 16 min to read • [Edit on GitHub](#)

A Point-to-Site (P2S) configuration lets you create a secure connection from an individual client computer to a virtual network. A P2S connection is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a virtual network.

Point-to-Site connections do not require a VPN device or a public-facing IP address to work. A VPN connection is established by starting the connection from the client computer. For more information about Point-to-Site connections, see the [VPN Gateway FAQ](#) and [Planning and Design](#).

This article walks you through creating a VNet with a Point-to-Site connection in the Resource Manager deployment model using the Azure portal.

Deployment models and methods for P2S connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

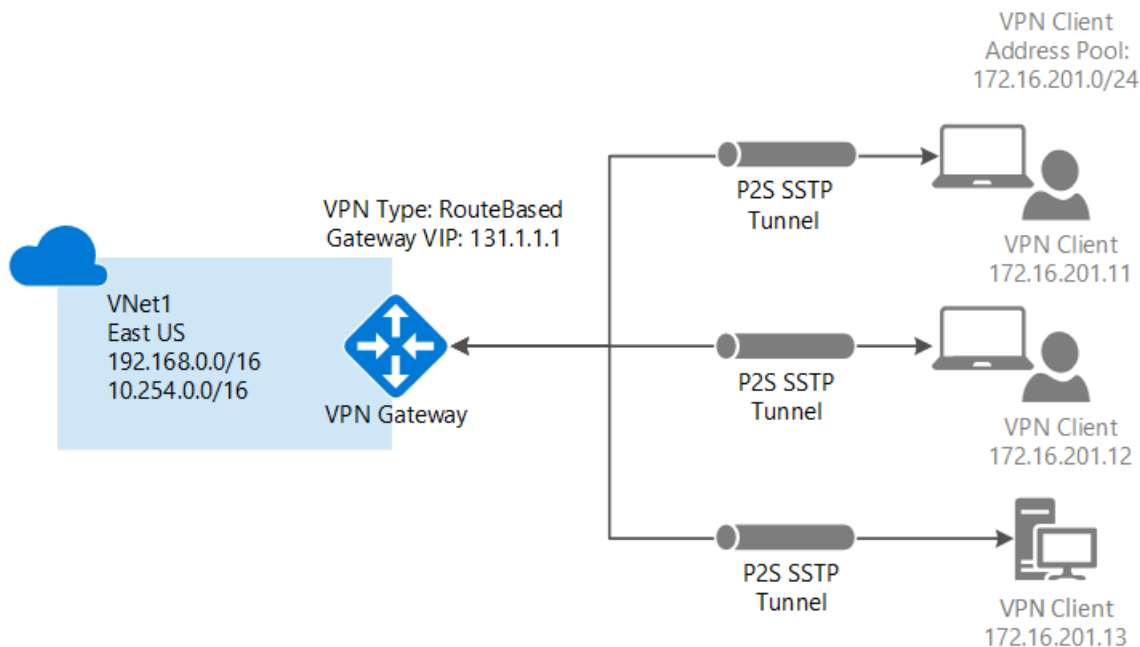
For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the two deployment models and available deployment methods for P2S configurations. When an article with configuration steps is available, we link directly to it from this table.

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

Basic workflow



Example values

- **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:** 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.

Before beginning

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

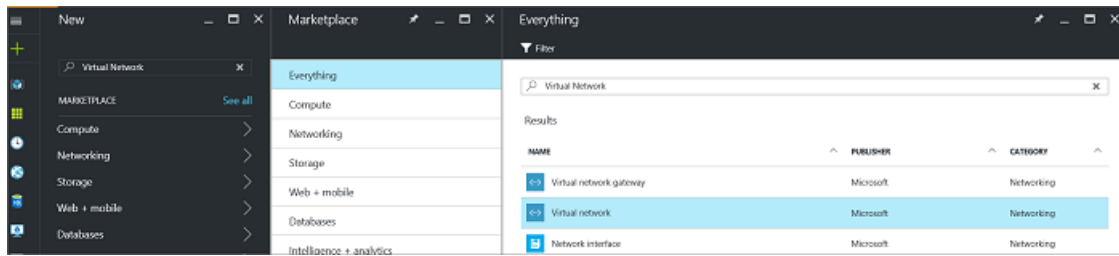
Part 1 - Create a virtual network

If you are creating this configuration as an exercise, you can refer to the [example values](#).

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](#).

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

returned list and click to open the **Virtual Network** blade.



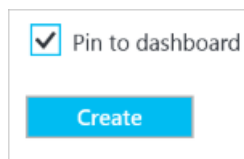
3. Near the bottom of the Virtual Network blade, from the **Select a deployment model** list, select **Resource Manager**, and then click **Create**.

4. On the **Create virtual network** blade, configure the VNet settings. When you fill in the fields, the red exclamation mark will become a green check mark when the characters entered in the field are valid.

5. The **Create virtual network** blade looks similar to the following example. There may be values that are auto-filled. If so, replace the values with your own.

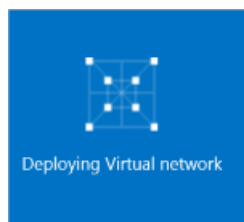
6. **Name:** Enter the name for your Virtual Network.

7. **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.
8. **Subnet name:** Add the subnet name and subnet address range. You can add additional subnets later, after creating the VNet.
9. **Subscription:** Verify that the Subscription listed is the correct one. You can change subscriptions by using the drop-down.
10. **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](#).
11. **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
12. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



A screenshot of a form section. It features a checkbox labeled 'Pin to dashboard' which is checked. Below the checkbox is a blue button with the text 'Create' in white.

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

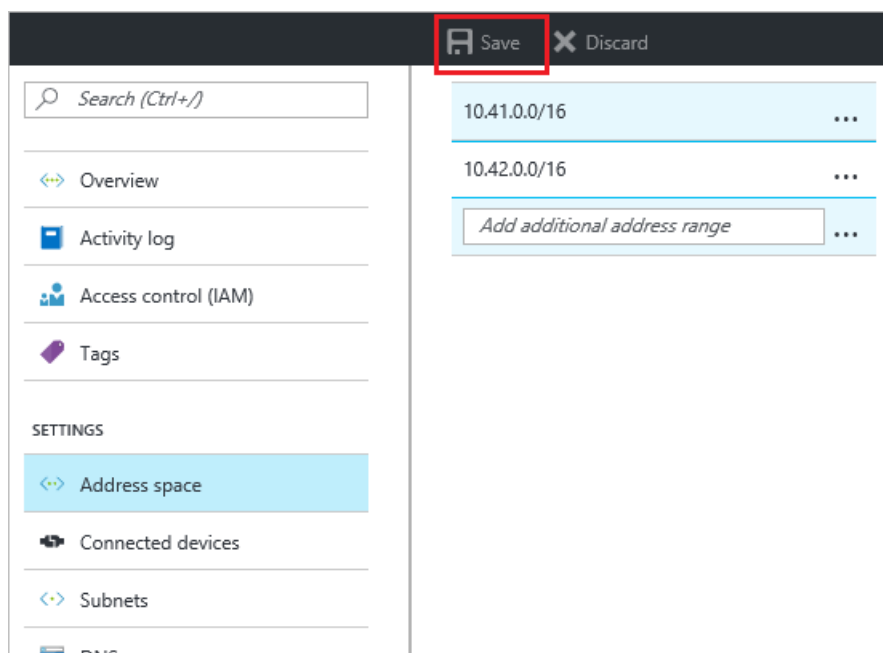


2. Add additional address space and subnets

You can add additional address space and subnets to your VNet once it has been created.

To add address space

1. To add additional address space, under the **Settings** section for your virtual network blade, click **Address space** to open the Address space blade.
2. Add the additional address space, and then click **Save** at the top of the blade.



To create subnets

1. To create subnets, in the **Settings** section of your virtual network blade, click **Subnets** to open the **Subnets** blade.
2. In the Subnets blade, click **+Subnet** to open the **Add subnet** blade. Name your new subnet and specify the address range.

NAME	ADDRESS RANGE	AVAILABLE ADDR...	SECURITY GROUP
FrontEnd	10.41.0.0/24	251	-

* Name: BackEnd ✓

* Address range (CIDR block) ⓘ: 10.42.0.0/24 ✓
10.42.0.0 - 10.42.0.255 (256 addresses)

3. Click **OK** at the bottom of the blade to save your changes.



3. Create a gateway subnet

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

The screenshots in this section are provided as a reference example. Be sure to use the GatewaySubnet address range that corresponds with the required values for your configuration.

To create a gateway subnet

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

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

Add subnet
RMVNet1

* Name: GatewaySubnet

* Address range (CIDR block) ⓘ: 192.168.0.0/24
192.168.0.0 - 192.168.0.255 (256 addresses)

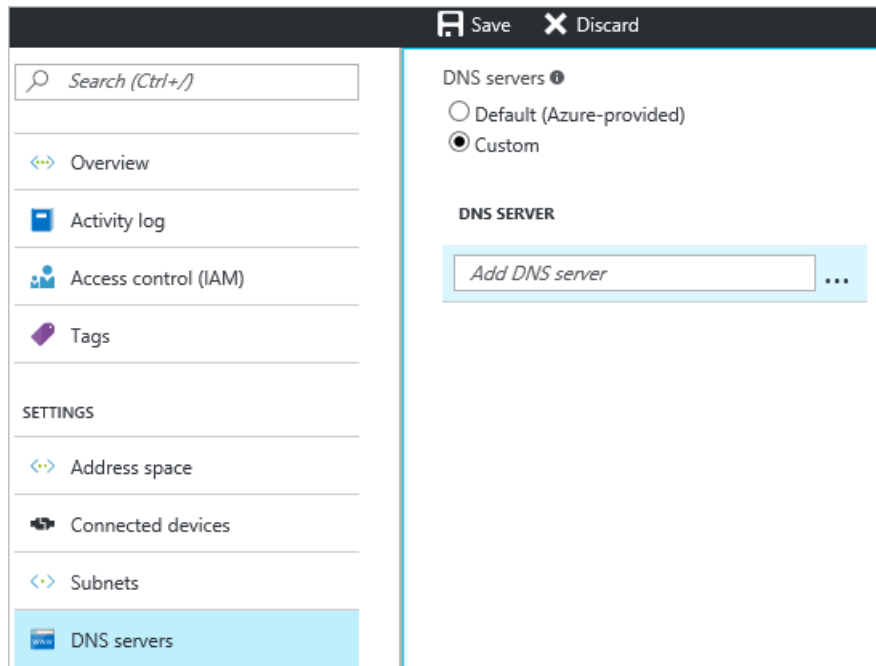
5. Click **OK** at the bottom of the blade to create the subnet.

4. Specify a DNS server (optional)

This setting allows you to specify the DNS server that you want to use for name resolution for this virtual network. It does not create a DNS server.

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

2. On the **DNS Servers** page, under **DNS servers**, select **Custom**.
3. In the **DNS Server** field, in the **Add DNS server** box, enter the IP address of the DNS server that you want to use for name resolution.
4. When you are done adding DNS servers, click **Save** at the top of the blade to save your configuration.



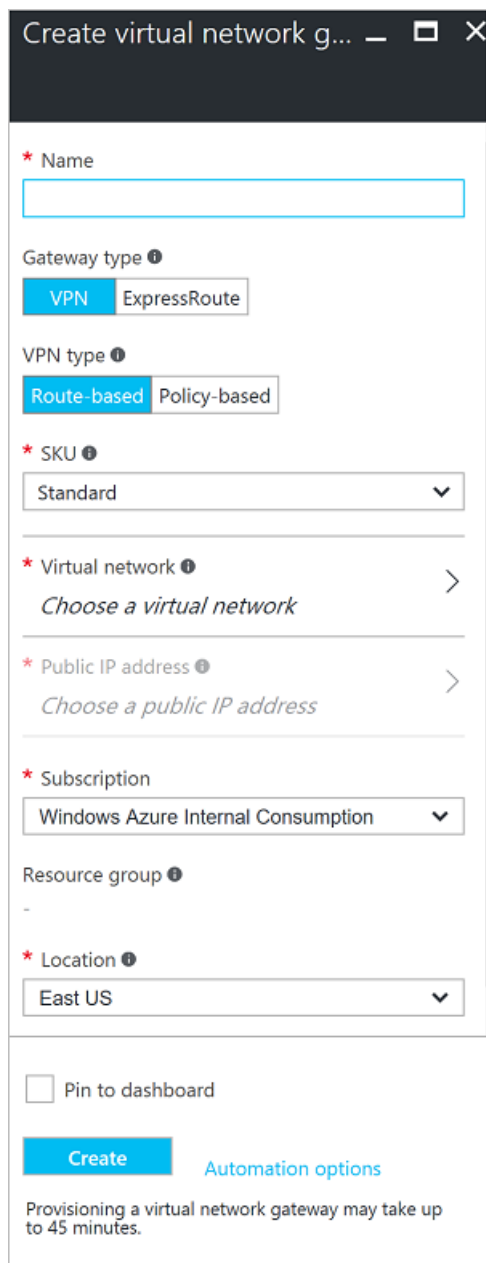
Part 2 - Create a virtual network gateway

Point-to-site connections require the following settings:

- Gateway type: VPN
- VPN type: Route-based

To create a virtual network gateway

1. In the portal, on the left side, click + and type "Virtual Network Gateway" in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** blade, click **Create** at the bottom of the blade. This opens the **Create virtual network gateway** blade.
2. On the **Create virtual network gateway** blade, fill in the values for your virtual network gateway.



Create virtual network gateway

* Name

Gateway type ⓘ

VPN ExpressRoute

VPN type ⓘ

Route-based Policy-based

* SKU ⓘ

Standard

* Virtual network ⓘ

Choose a virtual network

* Public IP address ⓘ

Choose a public IP address

* Subscription

Windows Azure Internal Consumption

Resource group ⓘ

-

* Location ⓘ

East US

☐ Pin to dashboard

Create Automation options

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

3. **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
4. **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
5. **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
6. **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select.
7. **Location:** Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, the virtual network will not appear in the 'Choose a virtual network' dropdown.
8. Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the **Choose a virtual network** blade. Select the VNet. If you don't see your VNet, make sure the **Location** field is pointing to the region in which your virtual network is located.
9. Choose a public IP address. Click **Public IP address** to open the **Choose public IP address** blade. Click **+Create New** to open the **Create public IP address blade**. Input a name for your public IP address. This blade creates a public IP address object to which a public IP address will be dynamically assigned. Click **OK** to save your changes to this blade.
10. **Subscription:** Verify that the correct subscription is selected.
11. **Resource group:** This setting is determined by the Virtual Network that you select.

12. Don't adjust the **Location** after you've specified the previous settings.
13. Verify the settings. You can select **Pin to dashboard** at the bottom of the blade if you want your gateway to appear on the dashboard.
14. Click **Create** to begin creating the gateway. The settings will be validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.



15. 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 will appear as a connected device. You can click the connected device (your virtual network gateway) to view more information.

Part 3 - Generate certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You export public certificate data (not the private key) as a Base-64 encoded X.509 .cer file from either a root certificate generated by an enterprise certificate solution, or a self-signed root certificate. You then import the public certificate data from the root certificate to Azure. Additionally, you need to generate a client certificate from the root certificate for clients. Each client that wants to connect to the virtual network using a P2S connection must have a client certificate installed that was generated from the root certificate.

1. Obtain the .cer file for the root certificate

If you are using an enterprise solution, you can use your existing certificate chain. If you aren't using an enterprise CA solution, you can create a self-signed root cert. One method for creating a self-signed cert is makecert.

- If you are using an enterprise certificate system, obtain the .cer file for the root certificate that you want to use.
 - If you are not using an enterprise certificate solution, you need to generate a self-signed root certificate. For Windows 10 steps, you can refer to [Working with self-signed root certificates for Point-to-Site configurations](#).
1. To obtain a .cer file from a certificate, open **certmgr.msc** and locate the root certificate. Right-click the self-signed root certificate, click **all tasks**, and then click **export**. This opens the **Certificate Export Wizard**.
 2. In the Wizard, click **Next**, select **No, do not export the private key**, and then click **Next**.
 3. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**. Then, click **Next**.
 4. 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**.
 5. Click **Finish** to export the certificate.

2. Generate a client certificate

You can either generate a unique certificate for each client that will connect, or you can use the same certificate on multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate if needed. Otherwise, if everyone is using the same client certificate and you find that you need to revoke the certificate for one client, you will need to generate and install new certificates for all of the clients that use that certificate to authenticate.

- If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
- If you are using a self-signed certificate, see [Working with self-signed root certificates for Point-to-Site configurations](#) to generate a client certificate.

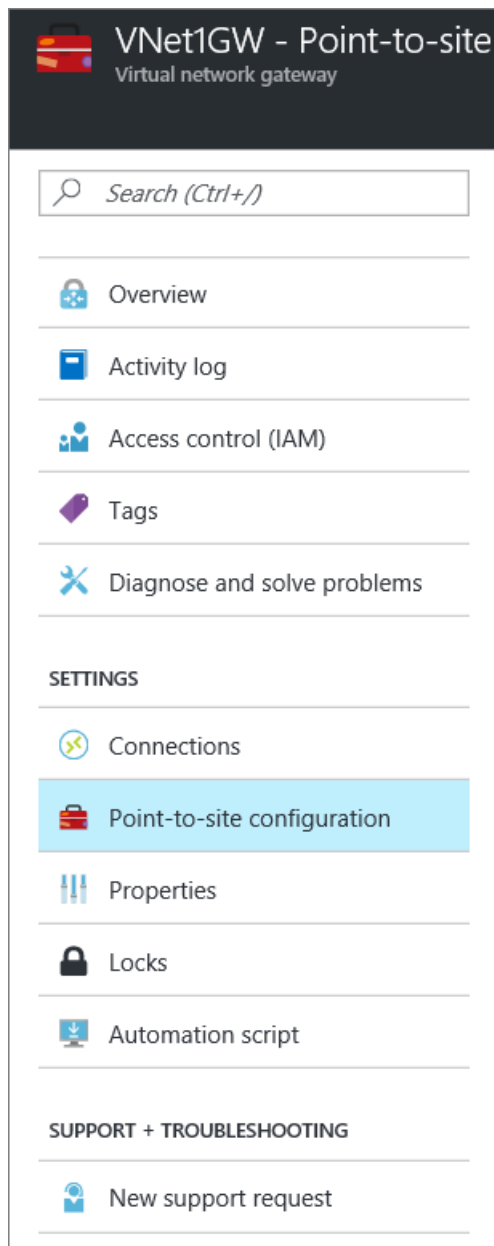
3. Export the client certificate

A client certificate is required for authentication. After generating the client certificate, export it. The client certificate you export will be installed later on each client computer.

1. To export a client certificate, you can use *certmgr.msc*. Right-click the client certificate that you want to export, click **all tasks**, and then click **export**.
2. Export the client certificate with the private key. This is a *.pfx* file. Make sure to record or remember the password (key) that you set for this certificate.

Part 4 - Add the client address pool

1. Once the virtual network gateway has been created, navigate to the **Settings** section of the virtual network gateway blade. In the **Settings** section, click **Point-to-site configuration** to open the **Configuration** blade.



2. **Address pool** is the pool of IP addresses from which clients that connect will receive an IP address. Add the address pool, and then click **Save**.

Save

Discard

Connection health

Connections	0
Ingress (bytes)	0
Egress (bytes)	0

Address pool

172.16.201.0/24

Part 5 - Upload the root certificate .cer file

After the gateway has been created, you can upload the .cer file for a trusted root certificate to Azure. You can upload files for up to 20 root certificates. You do not upload the private key for the root certificate to Azure. Once the .cer file is uploaded, Azure uses it to authenticate clients that connect to the virtual network.

1. Navigate to the **Point-to-site configuration** blade. You will add the .cer files in the **Root certificate** section of this blade.

Save

Discard

Connection health

Connections	0
Ingress (bytes)	0
Egress (bytes)	0

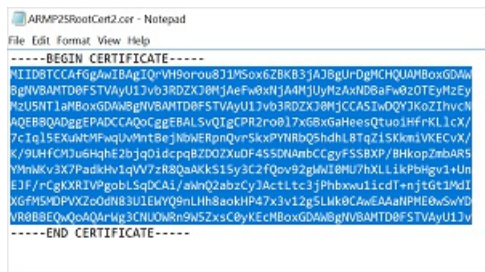
Address pool

172.16.201.0/24

Root certificates

NAME	PUBLIC CERTIFICATE DATA
	...

2. Make sure that you exported the root certificate as a Base-64 encoded X.509 (.cer) file. You need to export it in this format so that you can open the certificate with text editor.
3. Open the certificate with a text editor, such as Notepad. Copy only the following section:



4. Paste the certificate data into the **Public Certificate Data** section of the portal. Put the name of the certificate in the **Name** space, and then click **Save**. You can add up to 20 trusted root certificates.

Root certificates	
NAME	PUBLIC CERTIFICATE DATA
rootcert	MIIDAjCCAe6gAwIBAgIQ8QG2PbAHMrJLRSILZS/iKjAJBç ...
<input type="text"/>	<input type="text"/> ...
Revoked certificates	
NAME	THUMBPRINT
<input type="text"/>	<input type="text"/> ...

Part 6 - Download and install the VPN client configuration package

Clients connecting to Azure using P2S must have both a client certificate, and a VPN client configuration package installed. VPN client configuration packages are available for Windows clients.

The VPN client package contains information to configure the VPN client software that is built into Windows. The configuration is specific to the VPN that you want to connect to. The package does not install additional software. See the [VPN Gateway FAQ](#) for more information.

1. On the **Point-to-site configuration** blade, click **Download VPN client** to open the **Download VPN client** blade.



2. Select the correct package for your client, then click **Download**. For 64-bit clients, select **AMD64**. For 32-bit clients, select **x86**.
3. Install the package on the client computer. If you get a SmartScreen popup, click **More info**, then **Run anyway** in order to install the package.
4. On the client computer, navigate to **Network Settings** and click **VPN**. You will see the connection listed. It will show the name of the virtual network that it will connect to and looks similar to this example:



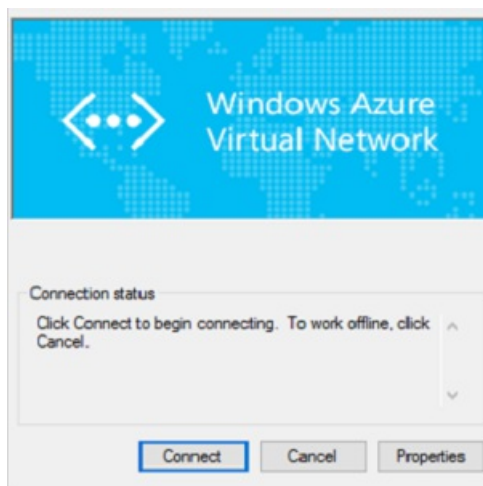
Part 7 - Install the client certificate

Each client computer must have a client certificate in order to authenticate. When installing the client certificate, you will need the password that was created when the client certificate was exported.

1. Copy the .pfx file to the client computer.
2. Double-click the .pfx file to install it. Do not modify the installation location.

Part 8 - Connect to Azure

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection should now be established.



Part 9 - 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 should be something similar to this:

```
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 add or remove trusted root certificates

You can remove trusted root certificate from Azure. When you remove a trusted certificate, the client certificates that were generated from the root certificate will no longer be able to connect to Azure via Point-to-Site. If you want clients to connect, they need to install a new client certificate that is generated from a certificate that is trusted in Azure.

You can manage the list of revoked client certificates on the **Point-to-site configuration** blade. This is the blade that you used to [upload a trusted root certificate](#).

To manage the list of revoked client certificates

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. If you remove a root certificate .cer from Azure, it revokes the access for all client certificates generated/signed by the revoked root certificate. If you want to revoke a particular client certificate, not the root, you can do so. That way the other certificates that were generated from the root certificate will still be valid.

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.

You can manage the list of revoked client certificates on the **Point-to-site configuration** blade. This is the blade that you used to [upload a trusted root certificate](#).

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).

Configure a Point-to-Site connection to a VNet using PowerShell

1/17/2017 • 14 min to read • [Edit on GitHub](#)

A Point-to-Site (P2S) configuration lets you create a secure connection from an individual client computer to a virtual network. A P2S connection is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a virtual network.

Point-to-Site connections do not require a VPN device or a public-facing IP address to work. A VPN connection is established by starting the connection from the client computer. For more information about Point-to-Site connections, see the [VPN Gateway FAQ](#) and [Planning and Design](#).

This article walks you through creating a VNet with a Point-to-Site connection in the Resource Manager deployment model using PowerShell.

Deployment models and methods for P2S connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

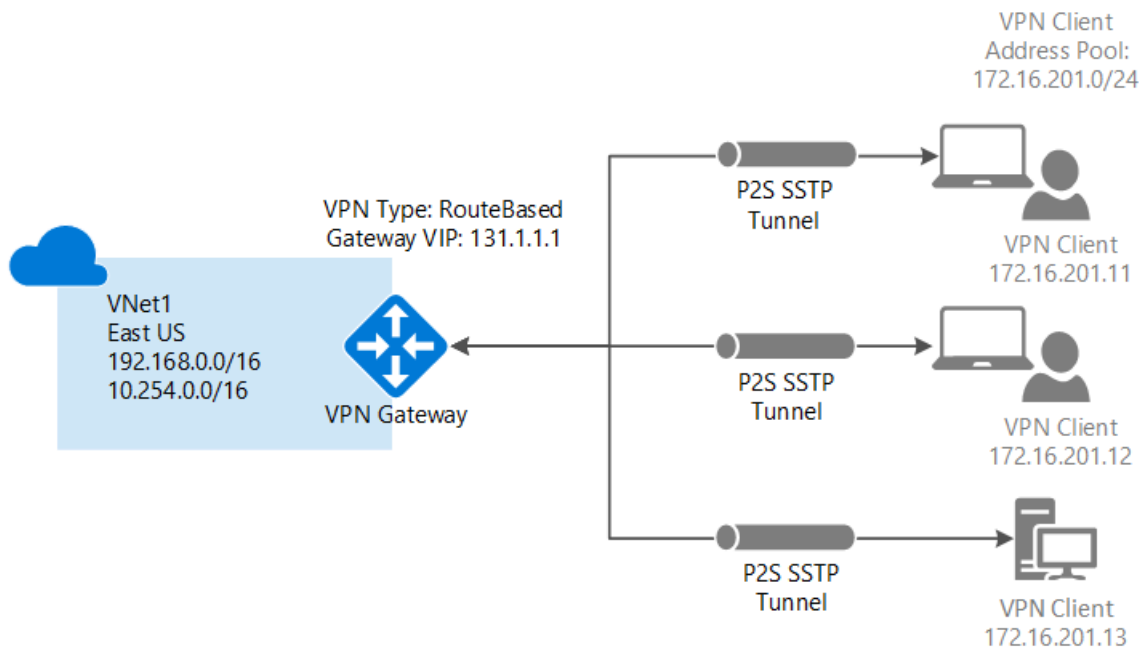
For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the two deployment models and available deployment methods for P2S configurations. When an article with configuration steps is available, we link directly to it from this table.

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

Basic workflow



In this scenario, you will create a virtual network with a Point-to-Site connection. The instructions will also help you generate certificates, which are required for this configuration. A P2S connection is composed of the following items: A VNet with a VPN gateway, a root certificate .cer file (public key), a client certificate, and the VPN configuration on the client.

We use the following values for this configuration. We set the variables 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.

Example values

- **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 will work 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.
 - **Subnet 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.
- **GW Name: Vnet1GW**
- **Public IP name: VNet1GWPIP**
- **VpnType: RouteBased**

Before beginning

- Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- Install the latest version of the Azure Resource Manager PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets. When working with PowerShell for this configuration, make sure that you are running as administrator.

Part 1 - Log in and set variables

In this section, you log in and declare the values used for this configuration. The declared values are used in the sample scripts. Change the values to reflect your own environment. Or, you can use the declared values and go through the steps as an exercise.

1. In the PowerShell console, log in to your Azure account. This 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.

```
Login-AzureRmAccount
```

2. Get a list of your Azure subscriptions.

```
Get-AzureRmSubscription
```

3. Specify the subscription that you want to use.

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

4. Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary.

```
$VNetName = "VNet1"
$FESubName = "FrontEnd"
$BESubName = "Backend"
$GWSubName = "GatewaySubnet"
$VNetPrefix1 = "192.168.0.0/16"
$VNetPrefix2 = "10.254.0.0/16"
$FESubPrefix = "192.168.1.0/24"
$BESubPrefix = "10.254.1.0/24"
$GWSubPrefix = "192.168.200.0/26"
$VPNClientAddressPool = "172.16.201.0/24"
$RG = "TestRG"
$Location = "East US"
$DNS = "8.8.8.8"
$GWName = "VNet1GW"
$GWIPName = "VNet1GWPIP"
$GWIPconfName = "gwipconf"
```

Part 2 - Configure a VNet

1. Create a resource group.

```
New-AzureRmResourceGroup -Name $RG -Location $Location
```

2. Create the subnet configurations for the virtual network, naming them *FrontEnd*, *BackEnd*, and *GatewaySubnet*. These prefixes must be part of the VNet address space that you declared.

```
$fesub = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName -AddressPrefix $FESubPrefix
$besub = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName -AddressPrefix $BESubPrefix
$gwsb = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName -AddressPrefix $GWSubPrefix
```

3. Create the virtual network. The DNS server specified should be a DNS server that can resolve the names for the resources you are connecting to. For this example, we used a public IP address. Be sure to use your own values.

```
New-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG -Location $Location -AddressPrefix $VNetPrefix1,$VNetPrefix2 -Subnet $fesub, $besub, $gwsb -DnsServer $DNS
```

4. Specify the variables for the virtual network you created.

```
$vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
```

5. Request a dynamically assigned public IP address. This IP address is necessary for the gateway to work properly. You later connect the gateway to the gateway IP configuration.

```
$pip = New-AzureRmPublicIpAddress -Name $GWIPName -ResourceGroupName $RG -Location $Location -AllocationMethod Dynamic
$ipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress $pip
```

Part 3 - Certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You export public certificate data (not the private key) as a Base-64 encoded X.509 .cer file from either a root certificate generated by an enterprise certificate solution, or a self-signed root certificate. You then import the public certificate data from the root certificate to Azure. Additionally, you need to generate a client certificate from the root certificate for clients. Each client that wants to connect to the virtual network using a P2S connection must have a client certificate installed that was generated from the root certificate.

1. Obtain the .cer file for the root certificate

You will need to get the public certificate data for the root certificate that you want to use.

- If you are using an enterprise certificate system, obtain the .cer file for the root certificate that you want to use.
- If you are not using an enterprise certificate solution, you need to generate a self-signed root certificate. For Windows 10 steps, you can refer to [Working with self-signed root certificates for Point-to-Site configurations](#).

1. To obtain a .cer file from a certificate, open **certmgr.msc** and locate the root certificate. Right-click the self-signed root certificate, click **all tasks**, and then click **export**. This opens the **Certificate Export Wizard**.
2. In the Wizard, click **Next**, select **No, do not export the private key**, and then click **Next**.
3. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**. Then, click **Next**.
4. 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**.
5. Click **Finish** to export the certificate.

2. Generate the client certificate

Next, generate the client certificates. You can either generate a unique certificate for each client that will connect, or you can use the same certificate on multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate if needed. Otherwise, if everyone is using the same client certificate and you find that you need to revoke the certificate for one client, you will need to generate and install new certificates for all of the clients that use the certificate to authenticate. The client certificates are installed on each client computer later in this exercise.

- If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the NetBIOS 'DOMAIN\username' format.
- If you are using a self-signed certificate, see [Working with self-signed root certificates for Point-to-Site configurations](#) to generate a client certificate.

3. Export the client certificate

A client certificate is required for authentication. After generating the client certificate, export it. The client certificate you export will be installed later on each client computer.

1. To export a client certificate, you can use *certmgr.msc*. Right-click the client certificate that you want to export, click **all tasks**, and then click **export**.
2. Export the client certificate with the private key. This is a *.pfx* file. Make sure to record or remember the password (key) that you set for this certificate.

4. Upload the root certificate .cer file

Declare the variable for your certificate name, replacing the value with your own:

```
$P2SRootCertName = "Mycertificatename.cer"
```

Add the public certificate data for the root certificate to Azure. You can upload files for up to 20 root certificates. You do not upload the private key for the root certificate to Azure. Once the .cer file is uploaded, Azure uses it to authenticate clients that connect to the virtual network.

Replace the file path with your own, and then run the cmdlets.

```
$filePathForCert = "C:\cert\Mycertificatename.cer"
$cert = new-object System.Security.Cryptography.X509Certificates.X509Certificate2($filePathForCert)
$certBase64 = [system.convert]::ToBase64String($cert.RawData)
$sp2srootcert = New-AzureRmVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $certBase64
```

Part 4 - 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**. This can take up to 45 minutes to complete.

```
New-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG `
-Location $Location -IpConfigurations $ipconf -GatewayType Vpn `
-VpnType RouteBased -EnableBgp $false -GatewaySku Standard `
-VpnClientAddressPool $VPNClientAddressPool -VpnClientRootCertificates $sp2srootcert
```

Part 5 - Download the VPN client configuration package

Clients connecting to Azure using P2S must have both a client certificate and a VPN client configuration package installed. VPN client configuration packages are available for Windows clients. The VPN client package contains information to configure the VPN client software that is built into Windows and is specific to the VPN that you want to connect to. The package does not install additional software. See the [VPN Gateway FAQ](#) for more information.

1. After the gateway has been created, you can download the client configuration package. This example downloads the package for 64-bit clients. If you want to download the 32-bit client, replace 'Amd64' with 'x86'. You can also download the VPN client by using the Azure portal.

```
Get-AzureRmVpnClientPackage -ResourceGroupName $RG `
-VirtualNetworkGatewayName $GWName -ProcessorArchitecture Amd64
```

2. The PowerShell cmdlet returns a URL link. This is an example of what the returned URL looks like:

```
"https://mdsbrketwprodslprod.blob.core.windows.net/cmakexe/4a431aa7-b5c2-45d9-97a0-859940069d3f/amd64/4a431aa7-b5c2-45d9-97a0-859940069d3f.exe?sv=2014-02-14&sr=b&sig=jSNCNQ9aUKkCiEokdo%2BqvFjAfyhSXGnRG0vYAv4efg0%3D&st=2016-01-08T07%3A10%3A08Z&se=2016-01-08T08%3A10%3A08Z&sp=r&fileExtension=.exe"
```

3. Copy and paste the link that is returned to a web browser to download the package. Then install the package on the client computer. If you get a SmartScreen popup, click **More info**, then **Run anyway** in order to install the package.
4. On the client computer, navigate to **Network Settings** and click **VPN**. You will see the connection listed. It will show the name of the virtual network that it will connect to and looks similar to this example:



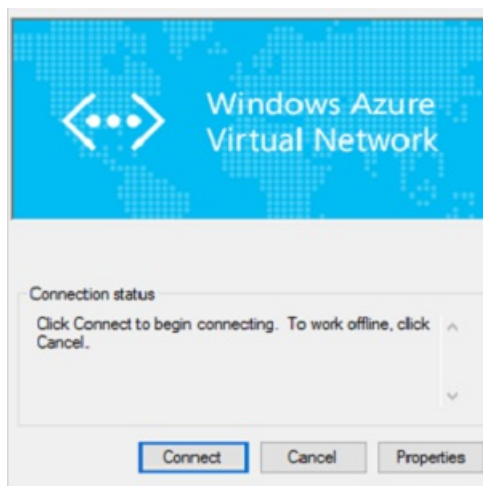
Part 6 - Install the client certificate

Each client computer must have a client certificate in order to authenticate. When installing the client certificate, you will need the password that was created when the client certificate was exported.

1. Copy the .pfx file to the client computer.
2. Double-click the .pfx file to install it. Do not modify the installation location.

Part 7 - Connect to Azure

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection should now be established.



Part 8 - 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 should be something similar to this:

```
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 add or remove a trusted root certificate

Certificates are used to authenticate VPN clients for Point-to-Site VPNs. The following steps walk you through adding and removing root certificates. When you add a Base64-encoded X.509 (.cer) file to Azure, you are telling Azure to trust the root certificate that the file represents.

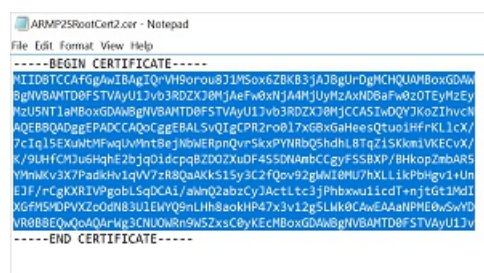
You can add or remove trusted root certificates by using PowerShell, or in the Azure portal. If you want to do this using the Azure portal, go to your **virtual network gateway > settings > Point-to-site configuration > Root certificates**. The following steps walk you through these tasks using PowerShell.

Add a trusted root certificate

You can add up to 20 trusted root certificate .cer files to Azure. Follow the steps below to add a root certificate.

1. Create and prepare the new root certificate that you will add to Azure. Export the public key as a Base-64 encoded X.509 (.CER) and open it with a text editor. Then copy only the section shown below.

Copy the values, as shown in the following example:



```
-----BEGIN CERTIFICATE-----
MIIDBTCCAqGgAwIBAgIQvH9orou8J1MSow6Z8KB3JA3BgUrDgMCHQUAMBoxGDAw
BgNVBAMTD0FSTVAYU1Jv3RDZXJ0MjAeFw0xNjA0MjUyMzA0ND8aFw0zOTYyMzE5
MzUSNTIaMBBoxGDAwBgNVBAMTD0FSTVAYU1Jv3RDZXJ0MjCCASIAwDQVJKozIhvcN
AQEBBQADggEPADCCAQoCggEBALSVQIgcPR2ro017xGBxGaHeesQtwo1HFRKL1cX/
7c1q15EXuWtFwqUvMntBgjN0wERpnQvr5kxPYNRb05hdhL8TqZ15Kkm1VKECvX/
K/9UHFCHU5uHqH2bJqD1cpgBZD0ZXuDF4SSDNaMbCgyfSSBXP/BHkopZmbAR5
YmNkV3X7Padkhv1qVW7zR8QaAKS15y3C2fQpv92gWlI0MU7hXL1kPbHgv1+Un
EJF/rCgKXRIvPgobLsqDCA1/awmQ2abzCy3Actl1c3jPhbxwu11cdT+njTgt1MdI
XGFH5VDPVXZoodN83U1EYVQ9mLHh8aokHP47x3v12g5Lk8CAwEAANPHEBwSwYD
VR8BBEQwQoA0ArHg3CNUOWRn9W5ZxsC8yKEChBoxGDAwBgNVBAMTD0FSTVAYU1Jv
-----END CERTIFICATE-----
```

2. 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/ZCCAeugAwIBAgIQKazzfJmKp9JRiX+tkTfSzAJBgUrDgMCHQUAMBoxGDAwBgNVBAMTDU1UDJTUm9vdENlcuQwHhcNM
TUxMjE5MDI1MTIxWhcNMzcxMjMjMjE0TU5WJAYMRYwFAyDQVQQDEw1NeVAyU1Jv3RDZXJ0MIIIBjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIIBGgKCAQEAyYjXoWy8xE/GF1OSIvUaA0bxBjZ1PJfcXkMWsHPzvhWc2esOKrVQtgFgDz4ggAnOUFEkFaszjiHdnXv3m
jzE2SpmA VIZPf2/yPWqkoHwkmnp6BpOvNVOpKxaGPOuK8+dq11xcL0eCkt69g4by0FGRFkBcSIgVTViS9wjuuS7LPo5+OXgyFkAY3pSDi
MzQckRGfGw5WGMHRDAiruDQF1ciLNoJAQCsDdLnI3pDYsvRW73HZEhmOqRRnJQe6VekvBYKLvnKaxUTKhFTYwuyvHBB96nMF
dRUKCZiWRIy8Hc8+sQEsAML2EitAjQv4+fqgYiFdSWqnQCPf/7IZbotgQIDAQABo0wSzbJBgNVHQEEQjBAGBAkuVrVvFsCJA dK5
pb/eoCNoRowGDEWMBQGA1UEAxMNTXIQMINSb290Q2VydIIQKazzfJmKp9JRiX+tkTfSzAJBgUrDgMCHQUAA4IBAQA223veAZEI
ar9N12ubNH2+HwZASnZDVNqspkPKD97TXfKHIPLcS43TaYkTz38eVrw16E0yDk4jAuPaKnPuPYFRj9w540SvY6PdOUwDoEqplcA Vp+b4
VYwxPL6oyEQ8wnOYuoAK1lhh20lCbo8h9mMy9ofU+RP6HJ7ITqupLfXdID/XevI8tW6Dm+C/wCeV3EmII09KUoblD/e24zlo3YzOtbyXwT
lh34T0f0/zQvUuBqZMcIPfM1cDvqcqIEFLWvWkoAnxbzckye2uk1gHO52d8A VL3mGiX8wBjKjc/pMdxrEvvCzJkdtBmqxtM6XjDJA LuVh
16qFlqgTWClcb7ju"
```

3. Add the new root certificate. You can only add one certificate at a time.

```
Add-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -VirtualNetworkGatewayName "VNet1GW" -ResourceGroupName "TestRG" -PublicCertData $MyP2SCertPubKeyBase64_2
```

4. You can verify that the new certificate was added correctly by using the following cmdlet.

```
Get-AzureRmVpnClientRootCertificate -ResourceGroupName "TestRG" `
-VirtualNetworkGatewayName "VNet1GW"
```

To remove a trusted root certificate

You can remove trusted root certificate from Azure. When you remove a trusted certificate, the client certificates that were generated from the root certificate will no longer be able to connect to Azure via Point-to-Site. If you want clients to connect, they need to install a new client certificate that is generated from a certificate that is trusted in Azure.

1. To remove a trusted root certificate, modify the following sample:

Declare the variables.

```
$P2SRootCertName2 = "ARMP2SRootCert2.cer"
$MyP2SCertPubKeyBase64_2 =
"MIIC/zCCAeugAwIBAgIQKazzFjMkp9JRiX+tkTfSzAJBgUrDgMCHQUAMBgxFjAUBgNVBAMTDU15UDJUm9vdENlcuQwHhcNM
TUxMjE5MDI1MTIxWheNMzcxMjMxMjM1OTU5WjAUMRYwFA YDVoQDEw1NeVAyU1Jvb3RDZXJ0MIIIBjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAYjIXoWy8xE/GF1OSivUaA0bxBjZ1PJfcXkMWsHPzvhWc2esOKrVQtgFgDz4ggAnOUFEkFaszjiHdnXv3m
jzE2SpmA VIZPf2/yPWqkoHwkmnp6BpOvNVOpKsaGPOuK8+dq1LxcL0eCkt69g4ly0FGRFkBCSlgVTViS9wjuuS7LPo5+OXgyFkAY3pSDi
MzQCKRGfNgw5WGMHRDAiruDQF1ciLNojA QCsDdLnI3pDYs vRW73HZEhmOqRRnJQe6VekvBYKLvnKaxUTKhFIYwuyMHBB96nMF
dRUKCZLiWRly8Hc8+sQEsaML2EItAjQv4+fqgYiFdSWqnQCPf/7IZbotgQIDAQABo00wSzBJBgNVHQEEQjBAGBAkuVrWvFsCJAdK5
pb/eoCNoRowGDEWMBQGA1UEAxMNTXIQMINSb290Q2VydIIQKazzFjMkp9JRiX+tkTfSzAJBgUrDgMCHQUAA4IBAQA223veAZEI
ar9N12ubNH2+HwZASnZDVNqspkPKD97TXfKHIPILcS43TaYkTz38eVrwI6E0yDk4jAuPaKnPuPYFRj9w540SvY6PdOUwDoEqpIcA Vp+b4
VYwxPL6oyEQ8wnOYuoAK1h1h20lCbo8h9mMy9ofU+RP6HJ7ITqulFxdID/XevI8tW6Dm+C/wCeV3EmIO9KUobID/e24zlo3YzOtbyXwT
Ih34T0fO/zQvUuBqZMcIPfM1cDvqcqIEFLWvWKOAnxbzkye2uk1gHO52d8A VL3mGiX8wBJkjc/pMdxrEvvcZjklTbmqxTM6XjDJA LuVh
16qFlqgTWC1cb7ju"
```

2. Remove the certificate.

```
Remove-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $P2SRootCertName2 -VirtualNetworkGatewayName
$GWName -ResourceGroupName $RG -PublicCertData $MyP2SCertPubKeyBase64_2
```

3. Use the following cmdlet to verify that the certificate was removed successfully.

```
Get-AzureRmVpnClientRootCertificate -ResourceGroupName "TestRG" `
-VirtualNetworkGatewayName "VNet1GW"
```

To manage the list of revoked client certificates

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. If you remove a root certificate .cer from Azure, it revokes the access for all client certificates generated/signed by the revoked root certificate. If you want to revoke a particular client certificate, not the root, you can do so. That way the other certificates that were generated from the root certificate will still be valid.

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. Get the thumbprint of the client certificate to revoke.

```
$RevokedClientCert1 = "ClientCert1"
$RevokedThumbprint1 = "?ef2af033d0686820f5a3c74804d167b88b69982f"
```

2. Add the thumbprint to the list of revoked thumbprint.

```
Add-AzureRmVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 `
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -Thumbprint $RevokedThumbprint1
```

3. Verify that the thumbprint was added to the certificate revocation list. You must add one thumbprint at a time.

```
Get-AzureRmVpnClientRevokedCertificate -VirtualNetworkGatewayName $GWName -ResourceGroupName $RG
```

Reinstate a client certificate

You can reinstate a client certificate by removing the thumbprint from the list of revoked client certificates.

1. Remove the thumbprint from the list of revoked client certificate thumbprint.

```
Remove-AzureRmVpnClientRevokedCertificate -VpnClientRevokedCertificateName $RevokedClientCert1 `
-VirtualNetworkGatewayName $GWName -ResourceGroupName $RG -Thumbprint $RevokedThumbprint1
```

2. Check if the thumbprint is removed from the revoked list.

```
Get-AzureRmVpnClientRevokedCertificate -VirtualNetworkGatewayName $GWName -ResourceGroupName $RG
```

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).

Configure a Point-to-Site connection to a VNet using the Azure portal

1/17/2017 • 12 min to read • [Edit on GitHub](#)

This article walks you through creating a VNet with a Point-to-Site connection in the classic deployment model using the Azure portal. A Point-to-Site (P2S) configuration lets you create a secure connection from an individual client computer to a virtual network. A P2S connection is useful when you want to connect to your VNet from a remote location, such as from home or a conference. Or, when you only have a few clients that need to connect to a virtual network.

Point-to-Site connections do not require a VPN device or a public-facing IP address to work. A VPN connection is established by starting the connection from the client computer. For more information about Point-to-Site connections, see the [VPN Gateway FAQ](#) and [About VPN Gateway](#).

Deployment models and methods for P2S connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

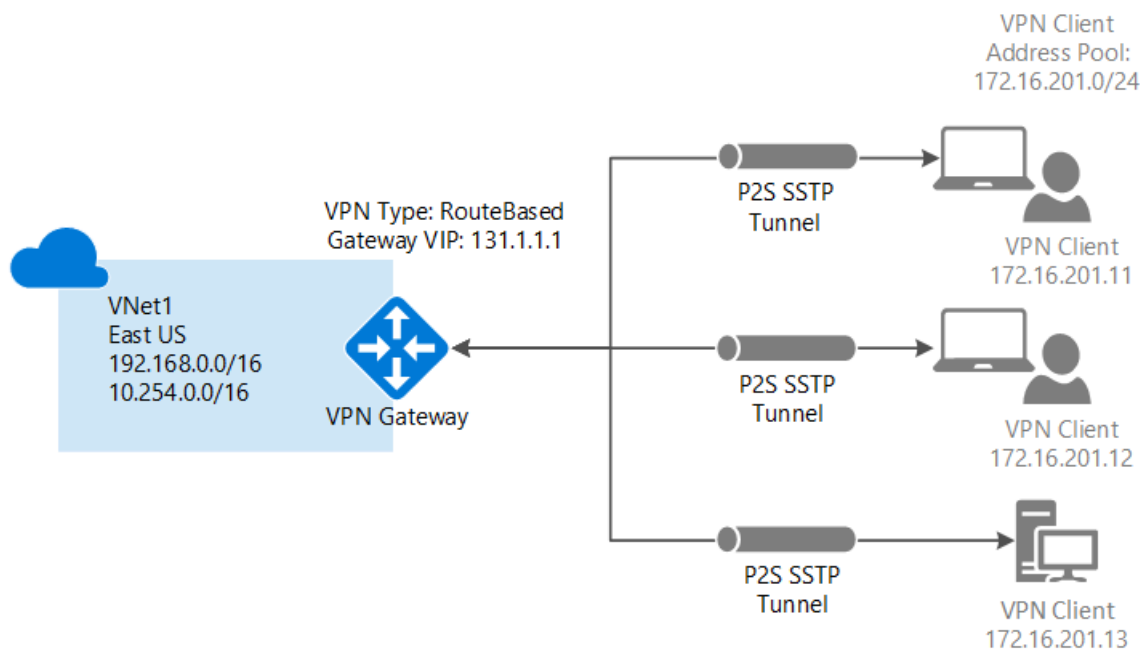
For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the two deployment models and available deployment methods for P2S configurations. When an article with configuration steps is available, we link directly to it from this table.

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

Basic workflow



The following sections walk you through the steps to create a secure Point-to-Site connection to a virtual network.

1. Create a virtual network and VPN gateway
2. Generate certificates
3. Upload the .cer file
4. Generate the VPN client configuration package
5. Configure the client computer
6. Connect to Azure

Example settings

You can use the following example settings:

- **Name:** VNet1
- **Address space:** 192.168.0.0/16
- **Subnet name:** FrontEnd
- **Subnet address range:** 192.168.1.0/24
- **Subscription:** If you have more than one subscription, verify that you are using the correct one.
- **Resource Group:** TestRG
- **Location:** East US
- **Connection type:** Point-to-site
- **Client Address Space:** 172.16.201.0/24. VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the specified pool.
- **GatewaySubnet:** 192.168.200.0/24. The Gateway subnet must use the name "GatewaySubnet".
- **Size:** Select the gateway SKU that you want to use.
- **Routing Type:** Dynamic

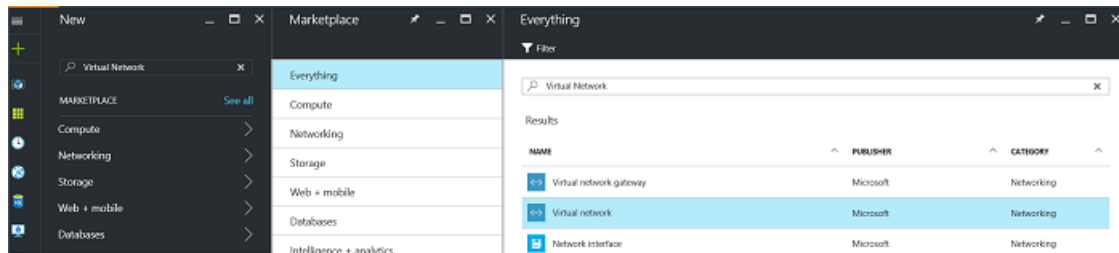
Section 1 - Create a virtual network and a VPN gateway

Part 1: Create a virtual network

If you don't already have a virtual network, create one. Screenshots are provided as examples. Be sure to replace the values with your own. To create a VNet by using the Azure portal, use the following steps:

1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.

- Click **New**. In the **Search the marketplace** field, type "Virtual Network". Locate **Virtual Network** from the returned list and click to open the **Virtual Network** blade.



- Near the bottom of the Virtual Network blade, from the **Select a deployment model** list, select **Classic**, and then click **Create**.

A screenshot of a dialog box titled 'Select a deployment model'. It contains a dropdown menu with 'Classic' selected. Below the dropdown is a blue 'Create' button.

- On the **Create virtual network** blade, configure the VNet settings. In this blade, you'll 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.

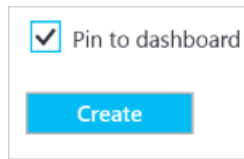
A screenshot of the 'Create virtual network' blade in the Azure portal. The form contains the following fields:

- Name**: VNet (with a green checkmark)
- Address space**: 192.168.0.0/16 (with a green checkmark). Below it, the address range is shown: 192.168.0.0 - 192.168.255.255 (65536 addresses).
- Subnet name**: FrontEnd (with a green checkmark)
- Subnet address range**: 192.168.1.0/24 (with a green checkmark). Below it, the address range is shown: 192.168.1.0 - 192.168.1.255 (256 addresses).
- Subscription**: Windows Azure Internal Consumption (dropdown menu)
- Resource Group**: TestRG (dropdown menu). Above it, there are radio buttons for 'Create new' and 'Use existing', with 'Use existing' selected.
- Location**: East US (dropdown menu)

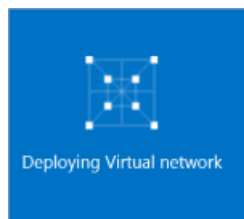
- Verify that the **Subscription** is the correct one. You can change subscriptions by using the drop-down.
- Click **Resource group** and either select an existing resource group, or create a new one by typing a name for your new resource group. If you are creating a new group, name the resource group according to your

planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](#).

- Next, select the **Location** settings for your VNet. The location will determine where the resources that you deploy to this VNet will reside.
- Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



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



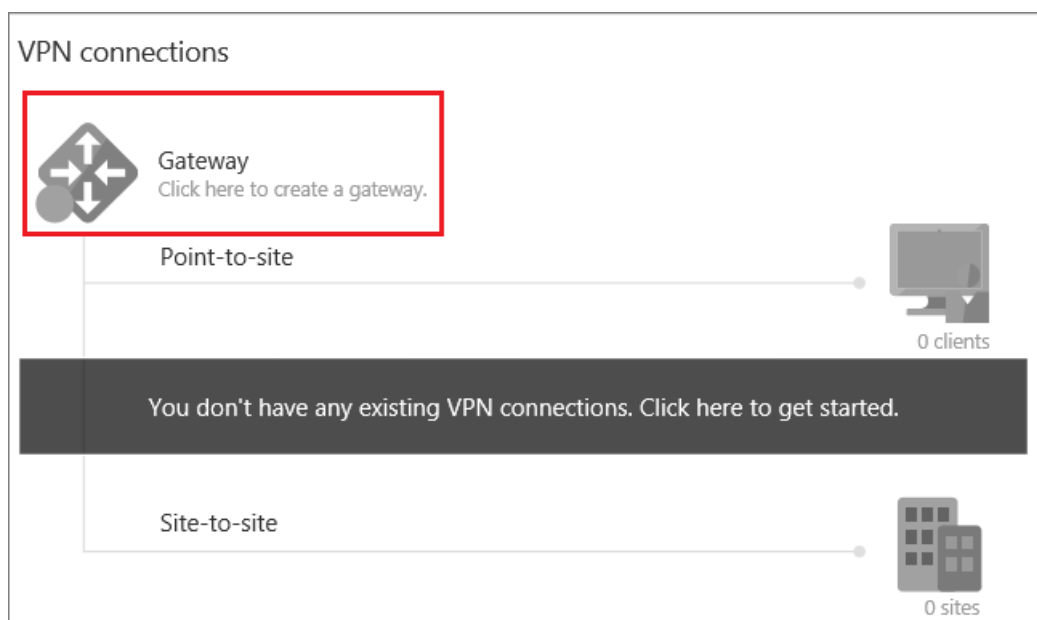
- After you create your virtual network, you can add the IP address of a DNS server in order to handle name resolution. Open the settings for your virtual network, click DNS servers, and add the IP address of the DNS server that you want to use. This setting does not create a new DNS server. Be sure to add a DNS server that your resources can communicate with.

Once your virtual network has been created, you will see **Created** listed under **Status** on the networks page in the Azure classic portal.

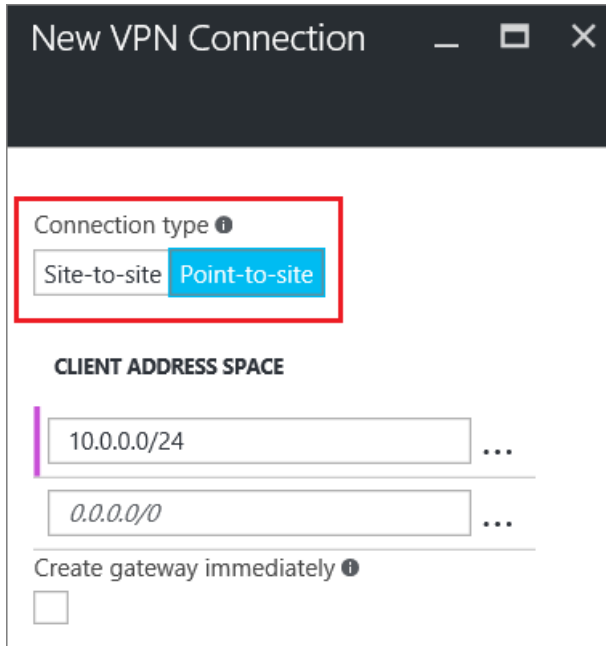
Part 2: Create gateway subnet and a dynamic routing gateway

In this step, you will create a gateway subnet and a Dynamic routing gateway. In the Azure portal for the classic deployment model, creating the gateway subnet and the gateway can be done through the same configuration blades.

- In the portal, navigate to the virtual network for which you want to create a gateway.
- On the blade for your virtual network, on the **Overview** blade, in the VPN connections section, click **Gateway**.

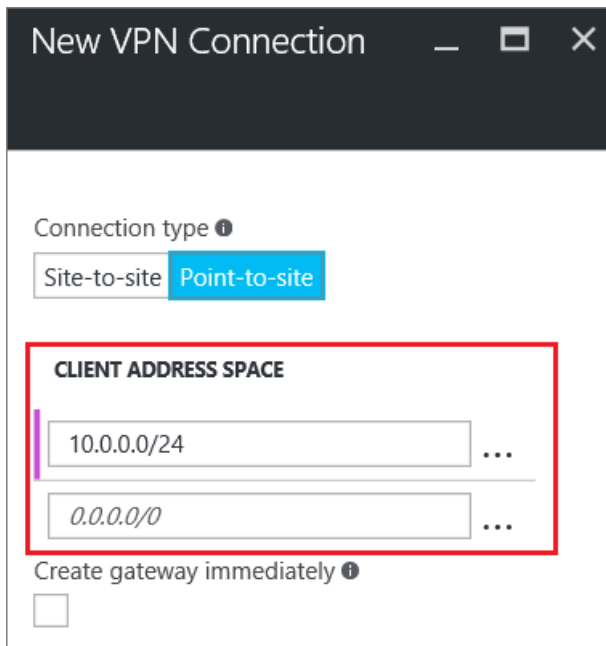


- On the **New VPN Connection** blade, select **Point-to-site**.



The screenshot shows the 'New VPN Connection' window. The 'Connection type' section has two buttons: 'Site-to-site' and 'Point-to-site'. The 'Point-to-site' button is highlighted in blue. Below this, the 'CLIENT ADDRESS SPACE' section contains two input fields. The first field contains '10.0.0.0/24' and the second field contains '0.0.0.0/0'. Both fields have a three-dot menu icon to their right. At the bottom, there is a checkbox labeled 'Create gateway immediately' which is currently unchecked.

- For **Client Address Space**, add the IP address range. This is the range from which the VPN clients will receive an IP address when connecting. Delete the auto-filled range and add your own.



This screenshot is similar to the previous one, but with a red rectangular box highlighting the 'CLIENT ADDRESS SPACE' section. This section includes the two input fields for IP address ranges: '10.0.0.0/24' and '0.0.0.0/0'. The 'Point-to-site' button remains selected. The 'Create gateway immediately' checkbox is still unchecked.

- Select the **Create gateway immediately** checkbox.

New VPN Connection

Connection type ⓘ
 Site-to-site Point-to-site

CLIENT ADDRESS SPACE

10.0.0.0/24 ...

0.0.0.0/0 ...

Create gateway immediately ⓘ
☒

Optional gateway configuration Subnet, size and routing type >

6. Click **Optional gateway configuration** to open the **Gateway configuration** blade.

New VPN Connection

Connection type ⓘ
 Site-to-site Point-to-site

CLIENT ADDRESS SPACE

10.0.0.0/24 ...

0.0.0.0/0 ...

Create gateway immediately ⓘ
☒

Optional gateway configuration Subnet, size and routing type >

Gateway configuration

* Subnet ⓘ
 192.168.2.0/29 >

Size ⓘ
 Default Standard High performance

Routing Type ⓘ
 Static Dynamic

7. Click **Subnet Configure required settings** to add the **gateway subnet**. While it is possible to create a gateway subnet as small as /29, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. This will allow for enough addresses to accommodate possible additional configurations that you may want in the future.

IMPORTANT

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

New VPN Connection

Connection type ⓘ

Site-to-site Point-to-site

CLIENT ADDRESS SPACE

10.0.0.0/24 ...

0.0.0.0/0 ...

Create gateway immediately ⓘ

☒

Optional gateway configuration Subnet, size and routing type >

Gateway configuration

* Subnet ⓘ

192.168.2.0/29 >

Size ⓘ

Default Standard High performance

Routing Type ⓘ

Static Dynamic

8. Select the gateway **Size**. This is the gateway SKU that you will use to create your virtual network gateway. In the portal, the Default SKU is **Basic**. For more information about gateway SKUs, see [About VPN Gateway Settings](#).

New VPN Connection

Connection type ⓘ

Site-to-site Point-to-site

CLIENT ADDRESS SPACE

10.0.0.0/24 ...

0.0.0.0/0 ...

Create gateway immediately ⓘ

☒

Optional gateway configuration Subnet, size and routing type >

Gateway configuration

* Subnet ⓘ

192.168.2.0/29 >

Size ⓘ

Default Standard High performance

Routing Type ⓘ

Static Dynamic

9. Select the **Routing Type** for your gateway. P2S configurations require a **Dynamic** routing type. Click **OK** when you have finished configuring this blade.

- On the **New VPN Connection** blade, click **OK** at the bottom of the blade to begin creating your virtual network gateway. This can take up to 45 minutes to complete.

Section 2 - Generate certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You export public certificate data (not the private key) as a Base-64 encoded X.509 .cer file from either a root certificate generated by an enterprise certificate solution, or a self-signed root certificate. You then import the public certificate data from the root certificate to Azure. Additionally, you need to generate a client certificate from the root certificate for clients. Each client that wants to connect to the virtual network using a P2S connection must have a client certificate installed that was generated from the root certificate.

Part 1: Obtain the .cer file for the root certificate

If you are using an enterprise solution, you can use your existing certificate chain. If you aren't using an enterprise CA solution, you can create a self-signed root cert. One method for creating a self-signed cert is makecert.

- If you are using an enterprise certificate system, obtain the .cer file for the root certificate that you want to use.
 - If you are not using an enterprise certificate solution, you need to generate a self-signed root certificate. For Windows 10 steps, you can refer to [Working with self-signed root certificates for Point-to-Site configurations](#).
- To obtain a .cer file from a certificate, open **certmgr.msc** and locate the root certificate. Right-click the self-signed root certificate, click **all tasks**, and then click **export**. This opens the **Certificate Export Wizard**.
 - In the Wizard, click **Next**, select **No, do not export the private key**, and then click **Next**.
 - On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**. Then, click **Next**.
 - 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**.
 - Click **Finish** to export the certificate.

Part 2: Generate a client certificate

You can either generate a unique certificate for each client that will connect, or you can use the same certificate on multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate if needed. Otherwise, if everyone is using the same client certificate and you find that you need to revoke the certificate for one client, you will need to generate and install new certificates for all of the clients that

use that certificate to authenticate.

- If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
- If you are using a self-signed certificate, see [Working with self-signed root certificates for Point-to-Site configurations](#) to generate a client certificate.

Part 3: Export the client certificate

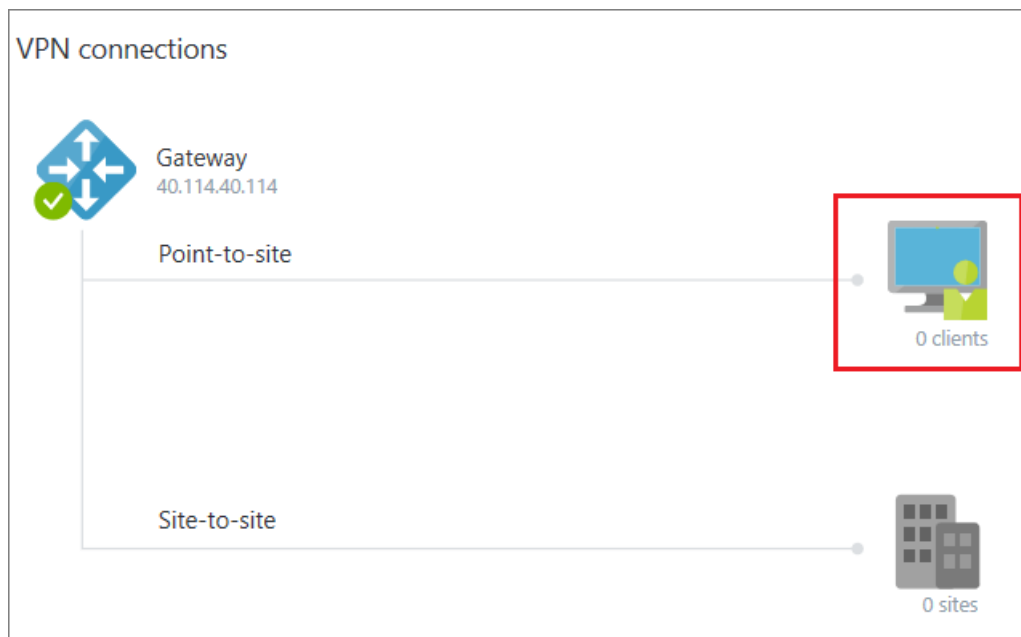
Install a client certificate on each computer that you want to connect to the virtual network. A client certificate is required for authentication. You can automate installing the client certificate, or you can install it manually. The following steps walk you through exporting and installing the client certificate manually.

1. To export a client certificate, you can use *certmgr.msc*. Right-click the client certificate that you want to export, click **all tasks**, and then click **export**.
2. Export the client certificate with the private key. This is a *.pfx* file. Make sure to record or remember the password (key) that you set for this certificate.

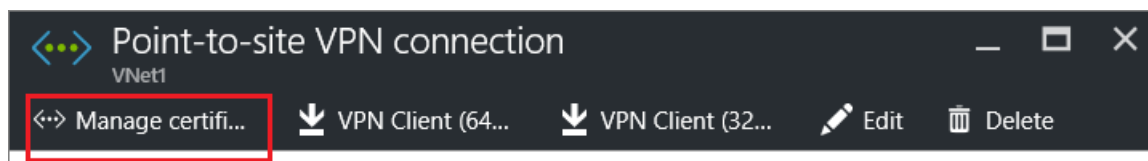
Section 3 - Upload the root certificate .cer file

After the gateway has been created, you can upload the .cer file for a trusted root certificate to Azure. You can upload files for up to 20 root certificates. You do not upload the private key for the root certificate to Azure. Once the .cer file is uploaded, Azure uses it to authenticate clients that connect to the virtual network.

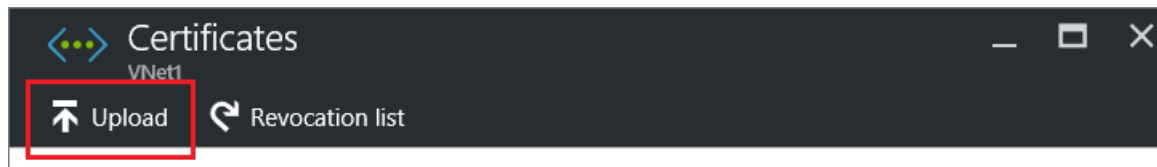
1. On the **VPN connections** section of the blade for your VNet, click the **clients** graphic to open the **Point-to-site VPN connection** blade.



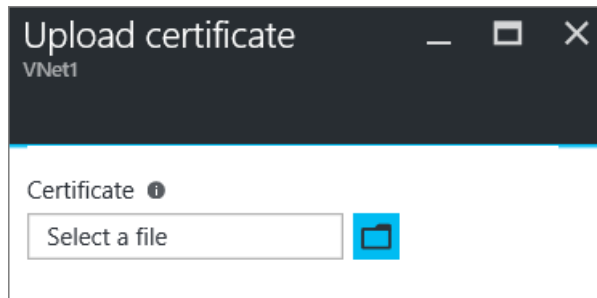
2. On the **Point-to-site connection** blade, click **Manage certificates** to open the **Certificates** blade.



3. On the **Certificates** blade, click **Upload** to open the **Upload certificate** blade.



- Click the folder graphic to browse for the .cer file. Select the file, then click **OK**. Refresh the page to see the uploaded certificate on the **Certificates** blade.



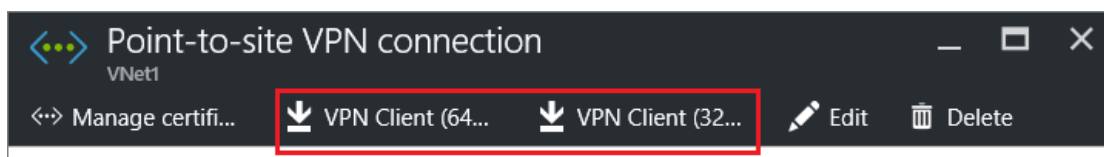
Section 4 - Generate the VPN client configuration package

To connect to the virtual network, you also need to configure a VPN client. The client computer requires both a client certificate and the proper VPN client configuration package in order to connect.

The VPN client package contains configuration information to configure the VPN client software built into Windows. The package does not install additional software. The settings are specific to the virtual network that you want to connect to. For the list of client operating systems that are supported, see the [Point-to-Site connections](#) section of the VPN Gateway FAQ.

To generate the VPN client configuration package

- In the Azure portal, in the **Overview** blade for your VNet, in **VPN connections**, click the client graphic to open the **Point-to-site VPN connection** blade.
- At the top of the **Point-to-site VPN connection** blade, click the download package that corresponds to the client operating system on which it will be installed:
 - For 64-bit clients, select **VPN Client (64-bit)**.
 - For 32-bit clients, select **VPN Client (32-bit)**.



- You will see a message that Azure is generating the VPN client configuration package for the virtual network. After a few minutes, the package is generated and you will see a message on your local computer that the package has been downloaded. Save the configuration package file. You will install this on each client computer that will connect to the virtual network using P2S.

Section 5 - Configure the client computer

Part 1: Install the client certificate

Each client computer must have a client certificate in order to authenticate. When installing the client certificate, you will need the password that was created when the client certificate was exported.

- Copy the .pfx file to the client computer.
- Double-click the .pfx file to install it. Do not modify the installation location.

Part 2: Install the VPN client configuration package

You can use the same VPN client configuration package on each client computer, provided that the version matches the architecture for the client.

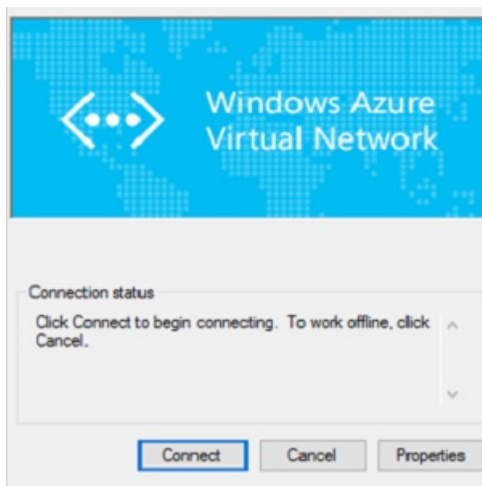
1. Copy the configuration file locally to the computer that you want to connect to your virtual network and double-click the .exe file.
2. Once the package has installed, you can start the VPN connection. The configuration package is not signed by Microsoft. You may want to sign the package using your organization's signing service, or sign it yourself using [SignTool](#). It's OK to use the package without signing. However, if the package isn't signed, a warning appears when you install the package.
3. On the client computer, navigate to **Network Settings** and click **VPN**. You will see the connection listed. It shows the name of the virtual network that it will connect to and will look similar to this:



Section 6 - Connect to Azure

Connect to your VNet

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection should now be established.



Verify the VPN 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 connectivity address range that you specified when you created your VNet. The results should be something 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

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).

Configure a Point-to-Site connection to a VNet using the classic portal

1/17/2017 • 11 min to read • [Edit on GitHub](#)

A Point-to-Site (P2S) configuration lets you create a secure connection from an individual client computer to a virtual network. A P2S connection is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a virtual network.

This article walks you through creating a VNet with a Point-to-Site connection in the **classic deployment model** using the **classic portal**.

Point-to-Site connections do not require a VPN device or a public-facing IP address to work. A VPN connection is established by starting the connection from the client computer. For more information about Point-to-Site connections, see the [VPN Gateway FAQ](#) and [Planning and Design](#).

Deployment models and methods for P2S connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

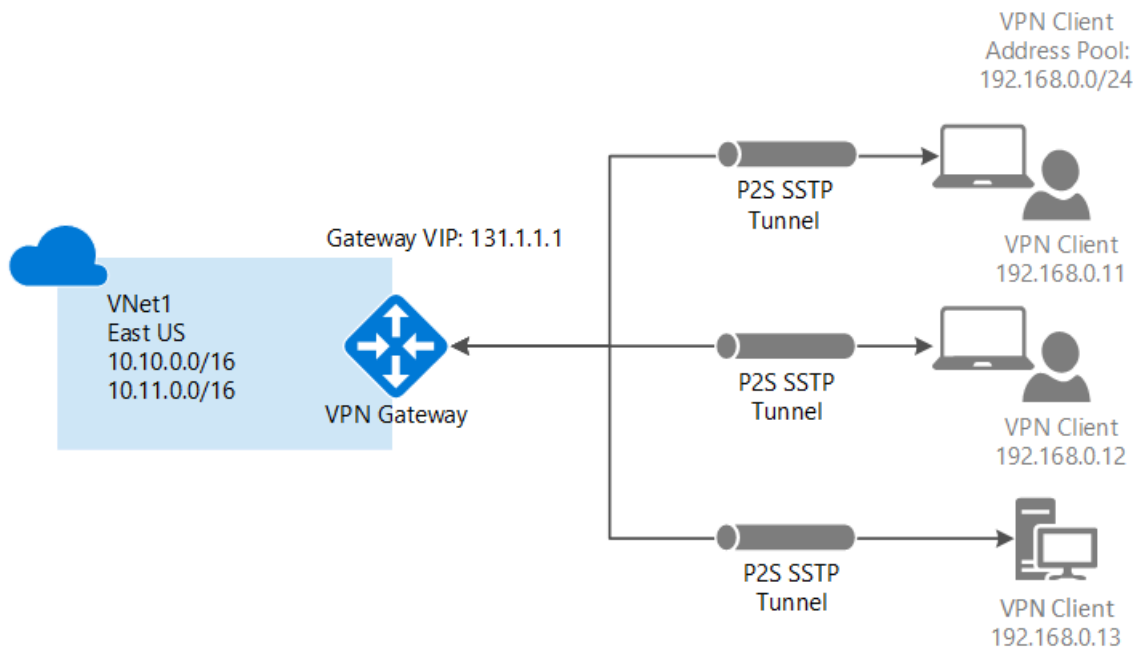
For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the two deployment models and available deployment methods for P2S configurations. When an article with configuration steps is available, we link directly to it from this table.

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

Basic workflow



The following steps walk you through the steps to create a secure Point-to-Site connection to a virtual network.

The configuration for a Point-to-Site connection is broken down into four sections. The order in which you configure each of these sections is important. Don't skip steps or jump ahead.

- **Section 1** Create a virtual network and VPN gateway.
- **Section 2** Create the certificates used for authentication and upload them.
- **Section 3** Export and install your client certificates.
- **Section 4** Configure your VPN client.

Section 1 - Create a virtual network and a VPN gateway

Part 1: Create a virtual network

1. Log in to the [Azure classic portal](#). These steps use the classic portal, not the Azure portal. Currently, you cannot create a P2S connection using the Azure portal.
2. In the lower left corner of the screen, click **New**. In the navigation pane, click **Network Services**, and then click **Virtual Network**. Click **Custom Create** to begin the configuration wizard.
3. On the **Virtual Network Details** page, enter the following information, and then click the next arrow on the lower right.
 - **Name:** Name your virtual network. For example, 'VNet1'. This is the name that you'll refer to when you deploy VMs to this VNet.
 - **Location:** The location is directly related to the physical location (region) where you want your resources (VMs) to reside. For example, if you want the VMs that you deploy to this virtual network to be physically located in East US, select that location. You can't change the region associated with your virtual network after you create it.
4. On the **DNS Servers and VPN Connectivity** page, enter the following information, and then click the next arrow on the lower right.
 - **DNS Servers:** Enter the DNS server name and IP address, or select a previously registered DNS server from the shortcut menu. 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. If you want to use the Azure default name resolution service, leave this section blank.
 - **Configure Point-To-Site VPN:** Select the checkbox.
5. On the **Point-To-Site Connectivity** page, specify the IP address range from which your VPN clients will

receive an IP address when connected. There are a few rules regarding the address ranges that you can specify. It's important to verify that the range that you specify doesn't overlap with any of the ranges located on your on-premises network.

6. Enter the following information, and then click the next arrow.

- **Address Space:** Include the Starting IP and CIDR (Address Count).
- **Add address space:** Add address space only if it is required for your network design.

7. 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 (DIPS) that will be assigned to the VMs and other role instances that you deploy to this virtual network.

It's especially important to select a range that does not overlap with any of the ranges that are used for your on-premises network. You must coordinate with your network administrator, who may need to carve out a range of IP addresses from your on-premises network address space for you to use for your virtual network.

8. Enter the following information, and then click the checkmark to begin creating your virtual network.

- **Address Space:** Add the internal IP address range that you want to use for this virtual network, including Starting IP and Count. It's important to select a range that does not overlap with any of the ranges that are used for your on-premises network.
- **Add subnet:** Additional subnets are not required, but you may want to create a separate subnet for VMs that will have static DIPS. Or you might want to have your VMs in a subnet that's separate from your other role instances.
- **Add gateway subnet:** The gateway subnet is required for a Point-to-Site VPN. Click to add the gateway subnet. The gateway subnet is used only for the virtual network gateway.

9. Once your virtual network has been created, you will see **Created** listed under **Status** on the networks page in the Azure classic portal. Once your virtual network has been created, you can create your dynamic routing gateway.

Part 2: Create a dynamic routing gateway

The gateway type must be configured as dynamic. Static routing gateways do not work with this feature.

1. In the Azure classic portal, on the **Networks** page, click the virtual network that you created, and navigate to the **Dashboard** page.
2. Click **Create Gateway**, located at the bottom of the **Dashboard** page. A message appears asking **Do you want to create a gateway for virtual network "VNet1"**. Click **Yes** to begin creating the gateway. It can take around 15 minutes for the gateway to create.

Section 2 - Generate and upload certificates

Certificates are used to authenticate VPN clients for Point-to-Site VPNs. You can use a root certificate generated by an enterprise certificate solution, or you can use a self-signed certificate. You can upload up to 20 root certificates to Azure. Once the .cer file is uploaded, Azure can use the information contained in it to authenticate clients that have a client certificate installed. The client certificate must be generated from the same certificate that the .cer file represents.

In this section you will do the following:

- Obtain the .cer file for a root certificate. This can either be a self-signed certificate, or you can use your enterprise certificate system.
- Upload the .cer file to Azure.
- Generate client certificates.

Part 1: Obtain the .cer file for the root certificate

If you are using an enterprise certificate system, obtain the .cer file for the root certificate that you want to use. In

[Part 3](#), you generate the client certificates from the root certificate.

If you are not using an enterprise certificate solution, you'll need to generate a self-signed root certificate. For Windows 10 steps, you can refer to [Working with self-signed root certificates for Point-to-Site configurations](#). The article walks you through using makecert to generate a self-signed certificate, and then export the .cer file.

Part 2: Upload the root certificate .cer file to the Azure classic portal

Add a trusted certificate to Azure. When you add a Base64-encoded X.509 (.cer) file to Azure, you are telling Azure to trust the root certificate that the file represents.

1. In the Azure classic portal, on the **Certificates** page for your virtual network, click **Upload a root certificate**.
2. On the **Upload Certificate** page, browse for the .cer root certificate, and then click the checkmark.

Part 3: Generate a client certificate

Next, generate the client certificates. You can either generate a unique certificate for each client that will connect, or you can use the same certificate on multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate if needed. Otherwise, if everyone is using the same client certificate and you find that you need to revoke the certificate for one client, you will need to generate and install new certificates for all of the clients that use the certificate to authenticate.

- If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the NetBIOS 'DOMAIN\username' format.
- If you are using a self-signed certificate, see [Working with self-signed root certificates for Point-to-Site configurations](#) to generate a client certificate.

Section 3 - Export and install the client certificate

Install a client certificate on each computer that you want to connect to the virtual network. A client certificate is required for authentication. You can automate installing the client certificate, or you can install manually. The following steps walk you through exporting and installing the client certificate manually.

1. To export a client certificate, you can use *certmgr.msc*. Right-click the client certificate that you want to export, click **all tasks**, and then click **export**.
2. Export the client certificate with the private key. This is a .pfx file. Make sure to record or remember the password (key) that you set for this certificate.
3. Copy the .pfx file to the client computer. On the client computer, double-click the .pfx file to install it. Enter the password when requested. Do not modify the installation location.

Section 4 - Configure your VPN client

To connect to the virtual network, you also need to configure a VPN client. The client requires both a client certificate and the proper VPN client configuration in order to connect. To configure a VPN client, perform the following steps, in order.

Part 1: Create the VPN client configuration package

1. In the Azure classic portal, on the **Dashboard** page for your virtual network, navigate to the quick glance menu in the right corner. For the list of client operating systems that are supported, see the [Point-to-Site connections](#) section of the VPN Gateway FAQ. The VPN client package contains configuration information to configure the VPN client software built into Windows. The package does not install additional software. The settings are specific to the virtual network that you want to connect to.

Select the download package that corresponds to the client operating system on which it will be installed:

- For 32-bit clients, select **Download the 32-bit Client VPN Package**.
- For 64-bit clients, select **Download the 64-bit Client VPN Package**.

2. It takes a few minutes to create your client package. Once the package has been completed, you can download the file. The .exe file that you download can be safely stored on your local computer.
3. After you generate and download the VPN client package from the Azure classic portal, you can install the client package on the client computer from which you want to connect to your virtual network. If you plan to install the VPN client package to multiple client computers, make sure that they each also have a client certificate installed.

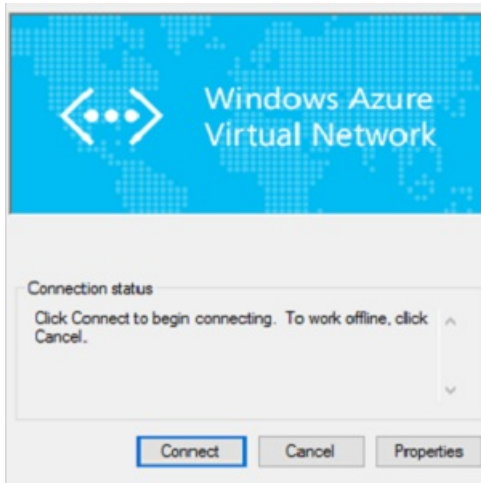
Part 2: Install the VPN configuration package on the client

1. Copy the configuration file locally to the computer that you want to connect to your virtual network and double-click the .exe file.
2. Once the package has installed, you can start the VPN connection. The configuration package is not signed by Microsoft. You may want to sign the package using your organization's signing service, or sign it yourself using [SignTool](#). It's OK to use the package without signing. However, if the package isn't signed, a warning appears when you install the package.
3. On the client computer, navigate to **Network Settings** and click **VPN**. You will see the connection listed. It will show the name of the virtual network that it will connect to and will look similar to this:



Part 3: Connect to Azure

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



3. Your connection should now be established.



Part 4: Verify the VPN 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 connectivity address range that you specified when you created your VNet. The results should be something 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
```

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. For more information, see [Virtual Machines](#).

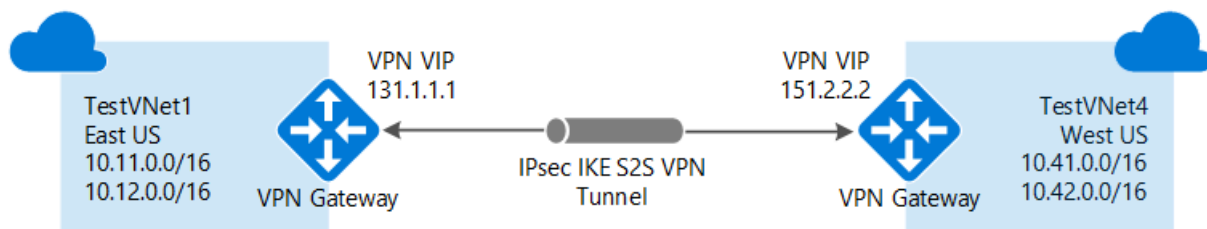
If you want more information about Virtual Networks, see the [Virtual Network Documentation](#) page.

Configure a VNet-to-VNet connection using the Azure portal

1/17/2017 • 15 min to read • [Edit on GitHub](#)

This article walks you through the steps to create a connection between VNets in the Resource Manager deployment model by using VPN Gateway and the Azure portal.

When you use the Azure portal to connect virtual networks, the VNets must be in the same subscription. If your virtual networks are in different subscriptions, you can still connect them by using the [PowerShell](#) steps.



Deployment models and methods for VNet-to-VNet connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for VNet-to-VNet configurations. When an article with configuration steps is available, we link directly to it from this table.

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

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

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

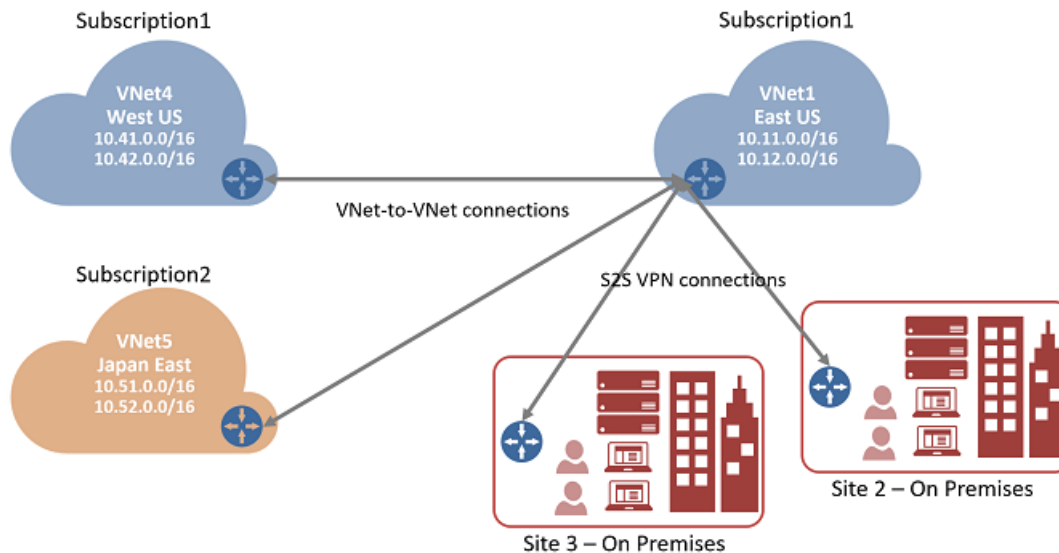
VNet peering

It's also possible to connect VNets without using a VPN gateway. If your VNets are in the same region, you may want to consider connecting them by using VNet peering. For more information, see the [VNet peering](#) article.

About VNet-to-VNet connections

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 an Azure VPN gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in different regions, or in different subscriptions.

You can even 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, as shown in the following diagram:



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 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.

For more information about VNet-to-VNet connections, see the [VNet-to-VNet FAQ](#) at the end of this article.

Example settings

When using these steps as an exercise, you can use the sample configuration values. For example purposes, we use multiple address spaces for each VNet. However, VNet-to-VNet configurations don't require multiple address spaces.

Values for TestVNet1:

- VNet Name: TestVNet1
- Address space: 10.11.0.0/16
 - Subnet name: FrontEnd
 - Subnet address range: 10.11.0.0/24
- Resource Group: TestRG1

- Location: East US
- Address Space: 10.12.0.0/16
 - Subnet name: BackEnd
 - Subnet address range: 10.12.0.0/24
- Gateway Subnet name: GatewaySubnet (this will auto-fill in the portal)
 - Gateway Subnet address range: 10.11.255.0/27
- DNS Server: Use the IP address of your DNS Server
- Virtual Network Gateway Name: TestVNet1GW
- Gateway Type: VPN
- VPN type: Route-based
- SKU: Select the Gateway SKU you want to use
- Public IP address name: TestVNet1GWIP
- Connection values:
 - Name: TestVNet1toTestVNet4
 - Shared key: You can create the shared key yourself. For this example, we'll use abc123. The important thing is that when you create the connection between the VNets, the value must match.

Values for TestVNet4:

- VNet Name: TestVNet4
- Address space: 10.41.0.0/16
 - Subnet name: FrontEnd
 - Subnet address range: 10.41.0.0/24
- Resource Group: TestRG1
- Location: West US
- Address Space: 10.42.0.0/16
 - Subnet name: BackEnd
 - Subnet address range: 10.42.0.0/24
- GatewaySubnet name: GatewaySubnet (this will auto-fill in the portal)
 - GatewaySubnet address range: 10.41.255.0/27
- DNS Server: Use the IP address of your DNS Server
- Virtual Network Gateway Name: TestVNet4GW
- Gateway Type: VPN
- VPN type: Route-based
- SKU: Select the Gateway SKU you want to use
- Public IP address name: TestVNet4GWIP
- Connection values:
 - Name: TestVNet4toTestVNet1
 - Shared key: You can create the shared key yourself. For this example, we'll use abc123. The important thing is that when you create the connection between the VNets, the value must match.

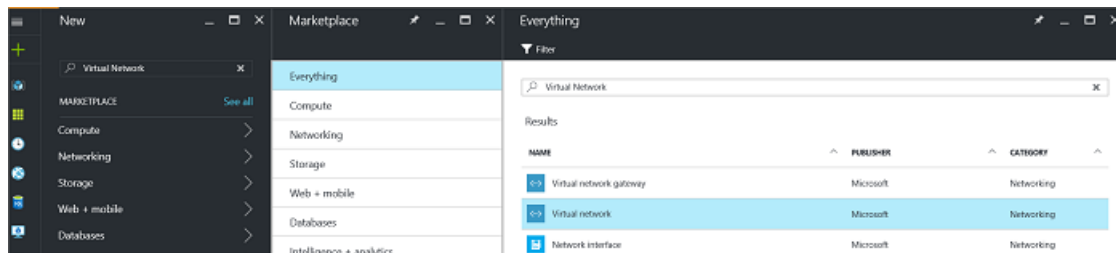
1. Create and configure TestVNet1

If you already have a VNet, verify that the settings are compatible with your VPN gateway design. Pay particular attention to any subnets that may overlap with other networks. If you have overlapping subnets, your connection won't work properly. If your VNet is configured with the correct settings, you can begin the steps in the [Specify a DNS server](#) section.

To create a virtual network

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

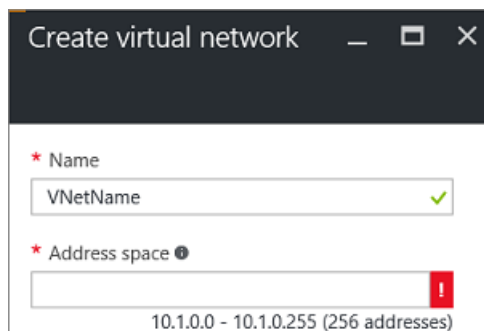
1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **New**. In the **Search the marketplace** field, type "Virtual Network". Locate **Virtual Network** from the returned list and click to open the **Virtual Network** blade.



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



4. On the **Create virtual network** blade, configure the VNet settings. When you fill in the fields, the red exclamation mark will become a green check mark when the characters entered in the field are valid.



5. The **Create virtual network** blade looks similar to the following example. There may be values that are auto-filled. If so, replace the values with your own.

Create virtual network

* Name

* Address space ⓘ
10.1.0.0 - 10.1.0.255 (256 addresses)

* Subnet name

* Subnet address range ⓘ
10.1.0.0 - 10.1.0.255 (256 addresses)

* Subscription
Windows Azure Internal Consumption

* Resource group ⓘ
☒ Create new ☐ Use existing

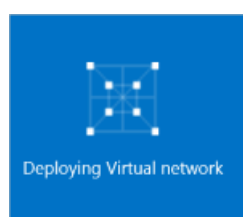
* Location
East US

6. **Name:** Enter the name for your Virtual Network.
7. **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.
8. **Subnet name:** Add the subnet name and subnet address range. You can add additional subnets later, after creating the VNet.
9. **Subscription:** Verify that the Subscription listed is the correct one. You can change subscriptions by using the drop-down.
10. **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](#).
11. **Location:** Select the location for your VNet. The location determines where the resources that you deploy to this VNet will reside.
12. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.

☒ Pin to dashboard

Create

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

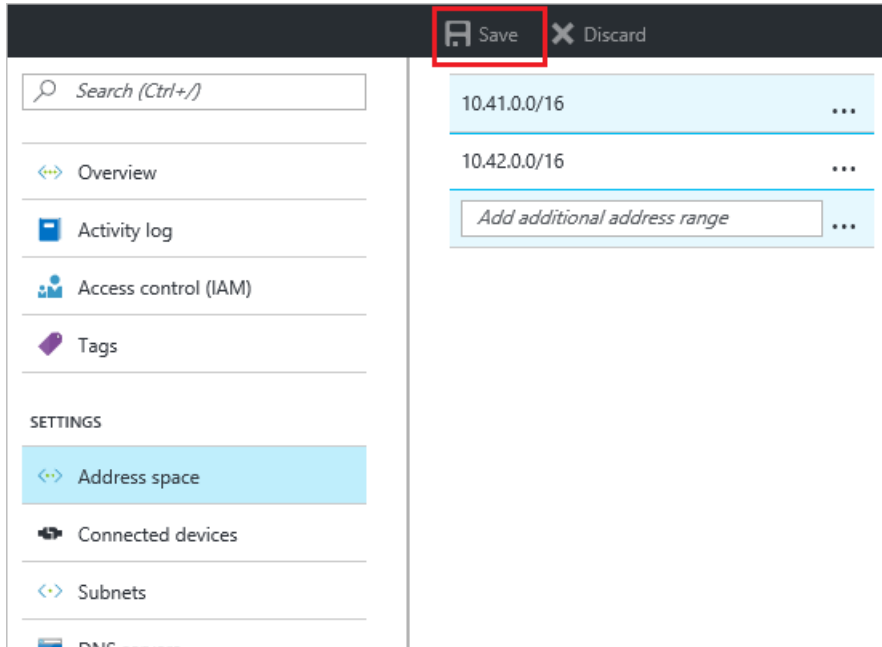


2. Add additional address space and create subnets

You can add additional address space and create subnets once your VNet has been created.

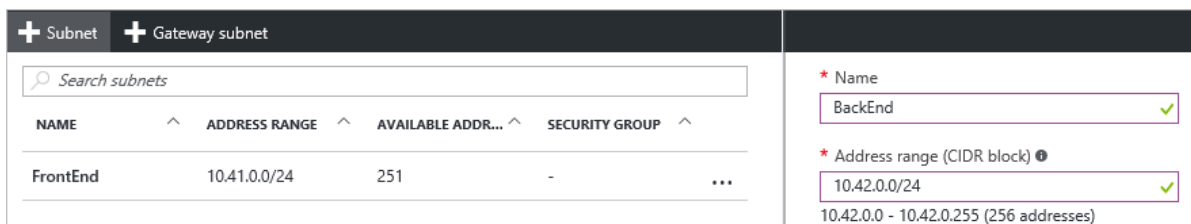
To add address space

1. To add additional address space, under the **Settings** section for your virtual network blade, click **Address space** to open the Address space blade.
2. Add the additional address space, and then click **Save** at the top of the blade.



To create subnets

1. To create subnets, in the **Settings** section of your virtual network blade, click **Subnets** to open the **Subnets** blade.
2. In the Subnets blade, click **+Subnet** to open the **Add subnet** blade. Name your new subnet and specify the address range.



3. Click **OK** at the bottom of the blade to save your changes.



3. Create a gateway subnet

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

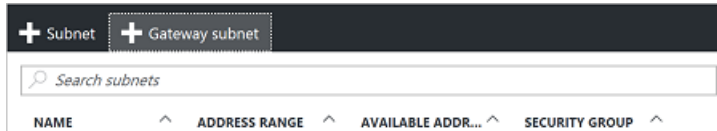
If you are creating this configuration as an exercise, refer to these [Example settings](#) when creating your gateway subnet.

IMPORTANT

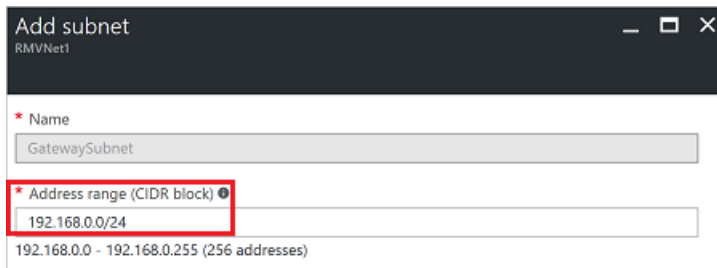
When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your VPN gateway to stop functioning as expected. For more information about network security groups, see [What is a network security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

To create a gateway subnet

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



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



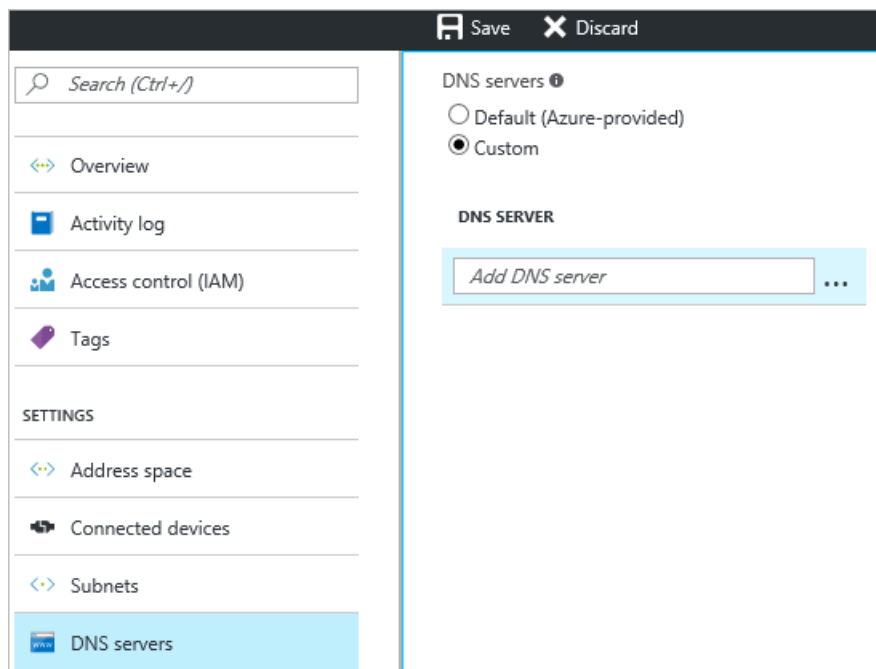
5. Click **OK** at the bottom of the blade to create the subnet.

4. Specify a DNS server (optional)

If you want to have name resolution for virtual machines that are deployed to your VNets, you should specify a DNS server.

This setting allows you to specify the DNS server that you want to use for name resolution for this virtual network. It does not create a DNS server.

1. On the **Settings** page for your virtual network, navigate to **DNS Servers** and click to open the DNS servers blade.
2. On the **DNS Servers** page, under **DNS servers**, select **Custom**.
3. In the **DNS Server** field, in the **Add DNS server** box, enter the IP address of the DNS server that you want to use for name resolution.
4. When you are done adding DNS servers, click **Save** at the top of the blade to save your configuration.

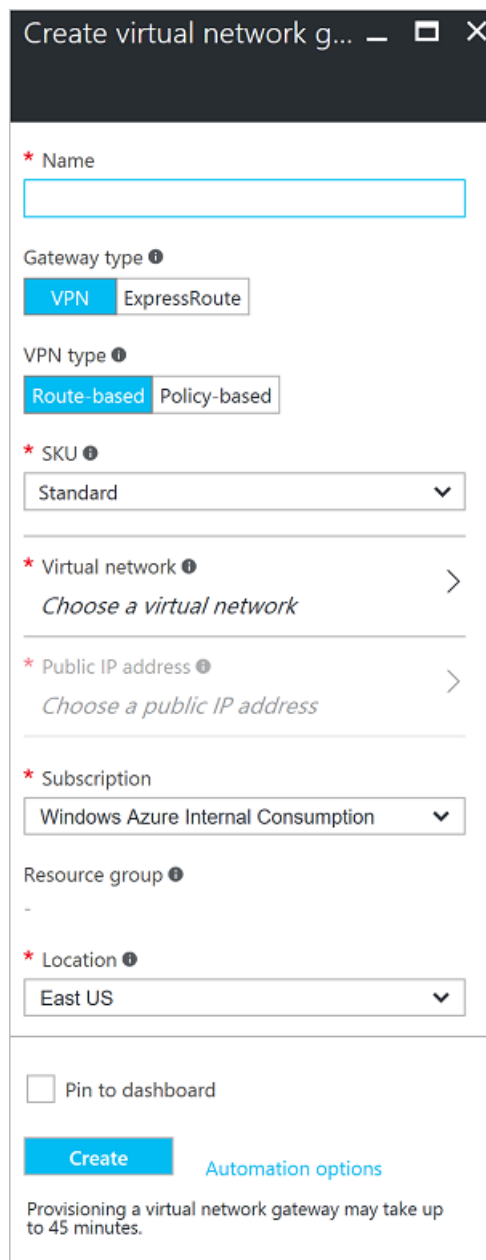


5. Create a virtual network gateway

In this step, you create the virtual network gateway for your VNet. This step can take up to 45 minutes to complete. If you are creating this configuration as an exercise, you can refer to the [Example settings](#).

To create a virtual network gateway

1. In the portal, on the left side, click + and type "Virtual Network Gateway" in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** blade, click **Create** at the bottom of the blade. This opens the **Create virtual network gateway** blade.
2. On the **Create virtual network gateway** blade, fill in the values for your virtual network gateway.



Create virtual network gateway

* Name

Gateway type ⓘ

VPN ExpressRoute

VPN type ⓘ

Route-based Policy-based

* SKU ⓘ

Standard

* Virtual network ⓘ

Choose a virtual network

* Public IP address ⓘ

Choose a public IP address

* Subscription

Windows Azure Internal Consumption

Resource group ⓘ

-

* Location ⓘ

East US

☐ Pin to dashboard

Create Automation options

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

3. **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
4. **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
5. **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.
6. **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select.
7. **Location:** Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, the virtual network will not appear in the 'Choose a virtual network' dropdown.
8. Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the **Choose a virtual network** blade. Select the VNet. If you don't see your VNet, make sure the **Location** field is pointing to the region in which your virtual network is located.
9. Choose a public IP address. Click **Public IP address** to open the **Choose public IP address** blade. Click **+Create New** to open the **Create public IP address** blade. Input a name for your public IP address. This blade creates a public IP address object to which a public IP address will be dynamically assigned. Click **OK** to save your changes to this blade.
10. **Subscription:** Verify that the correct subscription is selected.
11. **Resource group:** This setting is determined by the Virtual Network that you select.

12. Don't adjust the **Location** after you've specified the previous settings.
13. Verify the settings. You can select **Pin to dashboard** at the bottom of the blade if you want your gateway to appear on the dashboard.
14. Click **Create** to begin creating the gateway. The settings will be validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.



15. 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 will appear as a connected device. You can click the connected device (your virtual network gateway) to view more information.

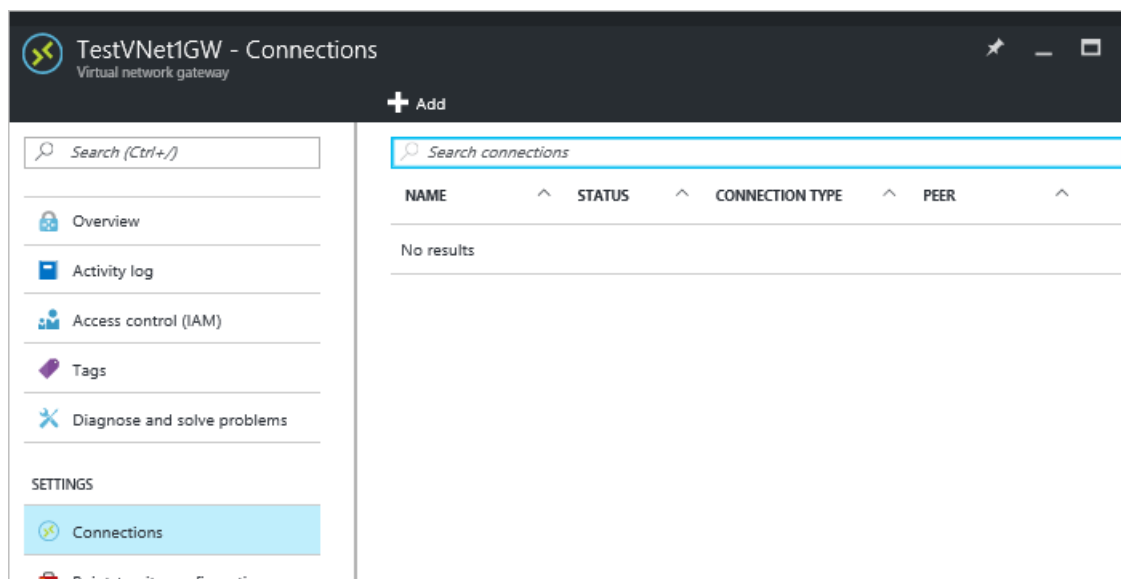
6. Create and configure TestVNet4

Once you've configured TestVNet1, create TestVNet4 by repeating the previous steps, replacing the values with those of TestVNet4. You don't need to wait until the virtual network gateway for TestVNet1 has finished creating before configuring TestVNet4. If you are using your own values, make sure that the address spaces don't overlap with any of the VNets that you want to connect to.

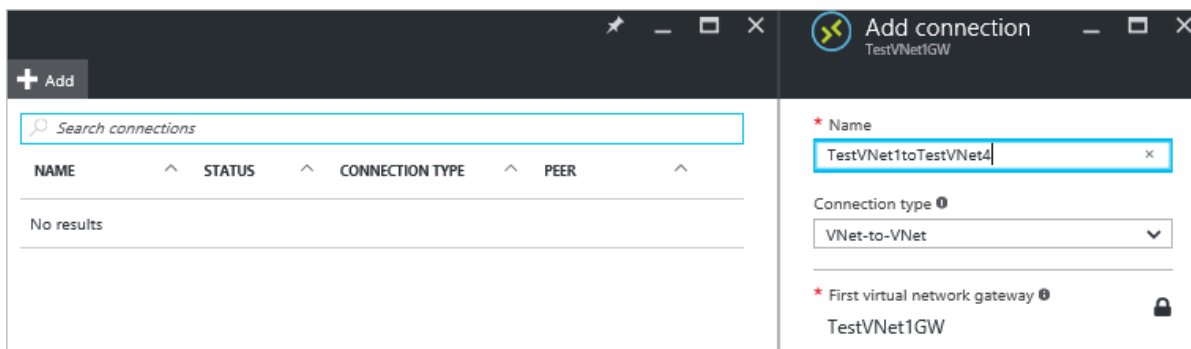
7. Configure the TestVNet1 connection

When the virtual network gateways for both TestVNet1 and TestVNet4 have completed, you can create your virtual network gateway connections. In this section, you will create a connection from VNet1 to VNet4.

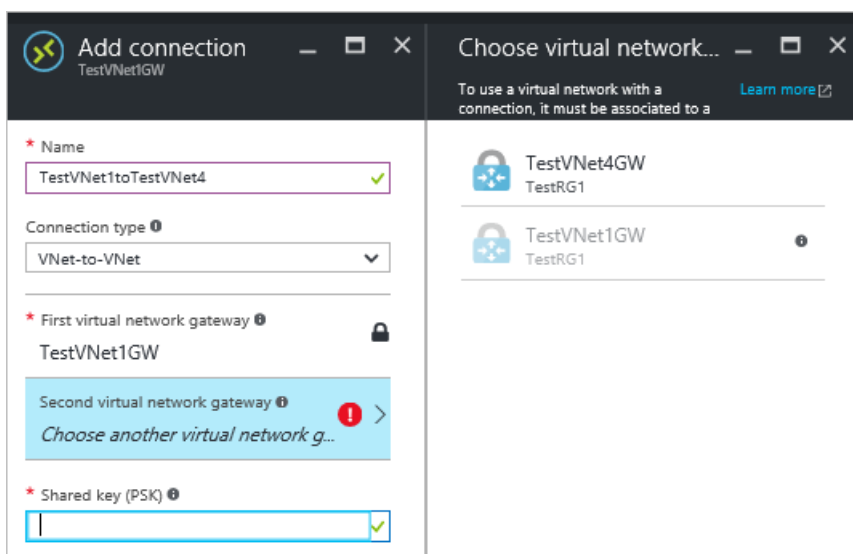
1. In **All resources**, navigate to the virtual network gateway for your VNet. For example, **TestVNet1GW**. Click **TestVNet1GW** to open the virtual network gateway blade.



2. Click **+Add** to open the **Add connection** blade.
3. On the **Add connection** blade, in the name field, type a name for your connection. For example, **TestVNet1toTestVNet4**.



4. For **Connection type**, select **VNet-to-VNet** from the dropdown.
5. The **First virtual network gateway** field value is automatically filled in because you are creating this connection from the specified virtual network gateway.
6. The **Second virtual network gateway** field is the virtual network gateway of the VNet that you want to create a connection to. Click **Choose another virtual network gateway** to open the **Choose virtual network gateway** blade.



7. View the virtual network gateways that are listed on this blade. Notice that only virtual network gateways that are in your subscription are listed. If you want to connect to a virtual network gateway that is not in your subscription, please use the [PowerShell article](#).
8. Click the virtual network gateway that you want to connect to.
9. In the **Shared key** field, type a shared key for your connection. You can generate or create this key yourself. In a site-to-site connection, the key you use would be exactly the same for your on-premises device and your virtual network gateway connection. The concept is similar here, except that rather than connecting to a VPN device, you are connecting to another virtual network gateway.



10. Click **OK** at the bottom of the blade to save your changes.

8. Configure the TestVNet4 connection

Next, create a connection from TestVNet4 to TestVNet1. Use the same method that you used to create the connection from TestVNet1 to TestVNet4. Make sure that you use the same shared key.

9. Verify your connection

Verify the connection. For each virtual network gateway, do the following:

1. Locate the blade for the virtual network gateway. For example, **TestVNet4GW**.
2. On the virtual network gateway blade, click **Connections** to view the connections blade for the virtual network gateway.

View the connections and verify the status. Once the connection is created, you will see **Succeeded** and **Connected** as the Status values.

NAME	STATUS	CONNECTION TYPE	PEER	
TestVNet4toTestVN...	Connected	VNet-to-VNet	TestVNet4GW	...
TestVNet1toTestVN...	Connected	VNet-to-VNet	TestVNet4GW	...

You can double-click each connection separately to view more information about the connection.

The screenshot shows the Azure portal interface for a connection named 'TestVNet4toTestVNet1'. The top bar includes a search icon, the connection name, and a 'Delete' button. Below the top bar is a sidebar with navigation links: 'Overview' (selected), 'Activity log', 'Access control (IAM)', and 'Tags'. The main content area is titled 'Essentials' and displays key information about the connection. On the left, it lists 'Resource group' as 'TestRG1', 'Status' as 'Connected', 'Location' as 'West US', 'Subscription name' as 'Windows Azure Internal Consumption', and 'Subscription ID'. On the right, it shows 'Data in' as '30.74 KB', 'Data out' as '30.51 KB', 'Virtual network' as 'TestVNet1, TestVNet4', 'Virtual network gateway 1' as 'TestVNet4GW (138.91.138.153)', and 'Virtual network gateway 2' as 'TestVNet1GW (40.76.16.43)'.

TestVNet4toTestVNet1 Connection		Delete
Search (Ctrl+ <i>/</i>)		
Overview		
Activity log		
Access control (IAM)		
Tags		
SETTINGS		
Essentials		
Resource group	Data in	
TestRG1	30.74 KB	
Status	Data out	
Connected	30.51 KB	
Location	Virtual network	
West US	TestVNet1, TestVNet4	
Subscription name	Virtual network gateway 1	
Windows Azure Internal Consumption	TestVNet4GW (138.91.138.153)	
Subscription ID	Virtual network gateway 2	
	TestVNet1GW (40.76.16.43)	

VNet-to-VNet FAQ

View the FAQ details for additional information about VNet-to-VNet connections.

- The virtual networks can be in the same or different Azure regions (locations).
- A cloud service or a load balancing endpoint CANNOT span across virtual networks, even if they are connected together.
- Connecting multiple Azure virtual networks together doesn't require any on-premises VPN gateways unless cross-premises connectivity is required.
- VNet-to-VNet supports connecting virtual networks. It does not support connecting virtual machines or cloud services NOT in a virtual network.
- VNet-to-VNet requires Azure VPN gateways with RouteBased (previously called Dynamic Routing) VPN types.
- Virtual network connectivity can be used simultaneously with multi-site VPNs. There is a maximum of 10 (Default/Standard Gateways) or 30 (HighPerformance Gateways) 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 VNet-to-VNet connections to fail.
- Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.
- 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.

- VNet-to-VNet traffic travels across the Microsoft Network, not the Internet.
- VNet-to-VNet traffic within the same region is free for both directions. Cross region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. Please refer to the [pricing page](#) for details.

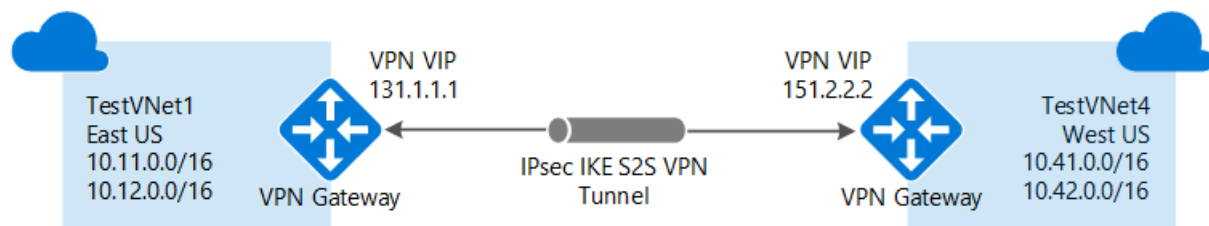
Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See the [Virtual Machines documentation](#) for more information.

Configure a VNet-to-VNet connection for Resource Manager using PowerShell

1/17/2017 • 15 min to read • [Edit on GitHub](#)

This article walks you through the steps to create a connection between VNets in the Resource Manager deployment model by using VPN Gateway. The virtual networks can be in the same or different regions, and from the same or different subscriptions.



Deployment models and methods for VNet-to-VNet connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for VNet-to-VNet configurations. When an article with configuration steps is available, we link directly to it from this table.

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

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

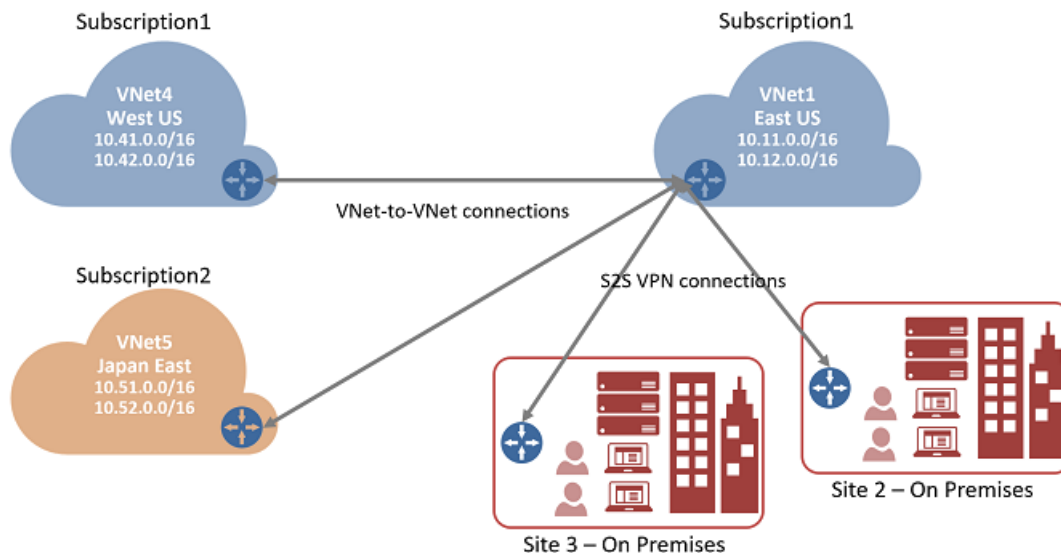
(*) denotes that this deployment method also requires PowerShell.

VNet peering

It's also possible to connect VNets without using a VPN gateway. If your VNets are in the same region, you may want to consider connecting them by using VNet peering. For more information, see the [VNet peering](#) article.

About VNet-to-VNet connections

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 an Azure VPN gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in different regions. Or in different subscriptions. You can even 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, as shown in the following diagram:



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 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 FAQ

- The virtual networks can be in the same or different Azure regions (locations).
- A cloud service or a load balancing endpoint CANNOT span across virtual networks, even if they are connected together.
- Connecting multiple Azure virtual networks together doesn't require any on-premises VPN gateways unless cross-premises connectivity is required.
- VNet-to-VNet supports connecting virtual networks. It does not support connecting virtual machines or cloud services NOT in a virtual network.
- VNet-to-VNet requires Azure VPN gateways with RouteBased (previously called Dynamic Routing) VPN types.
- Virtual network connectivity can be used simultaneously with multi-site VPNs. There is a maximum of 10 (Default/Standard Gateways) or 30 (HighPerformance Gateways) 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.

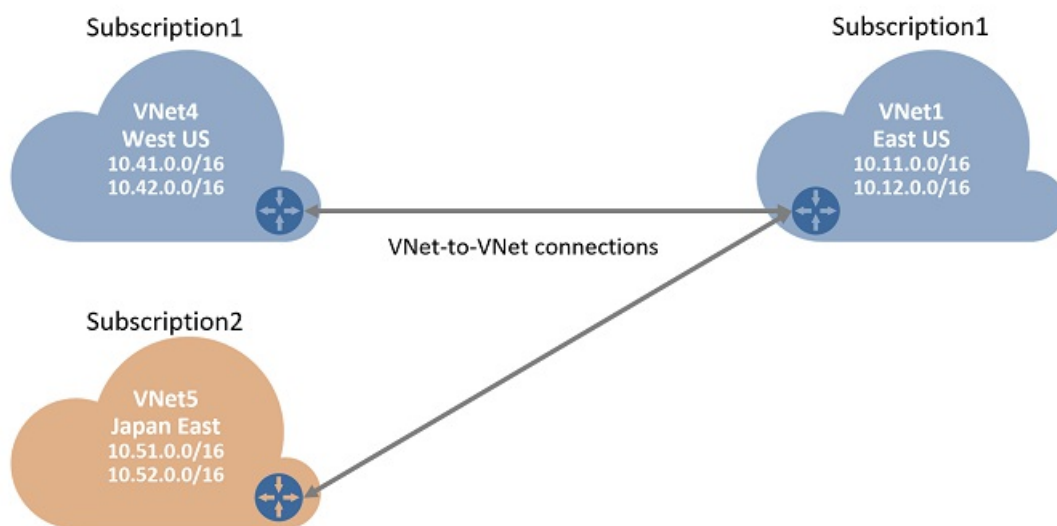
Overlapping address spaces will cause the creation of VNet-to-VNet connections to fail.

- Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.
- 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.
- VNet-to-VNet traffic travels across the Microsoft Network, not the Internet.
- VNet-to-VNet traffic within the same region is free for both directions. Cross region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. Please refer to the [pricing page](#) for details.

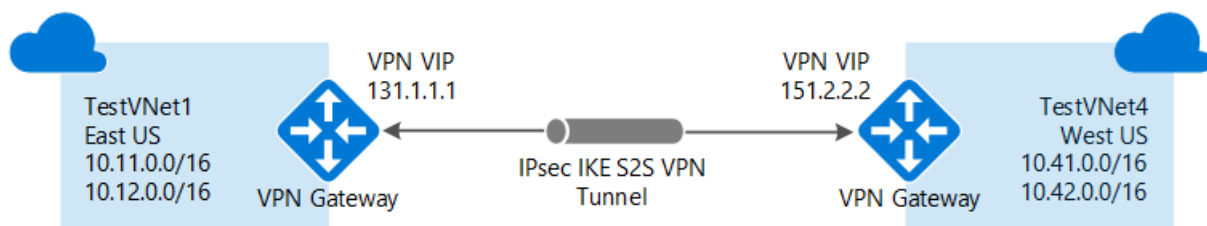
Which set of steps should I use?

In this article, you see two different sets of steps. One set of steps for [VNets that reside in the same subscription](#), and another for [VNets that reside in different subscriptions](#). The key difference between the sets is whether you can create and configure all virtual network and gateway resources within the same PowerShell session.

The steps in this article use variables that are declared at the beginning of each section. If you already are working with existing VNets, modify the variables to reflect the settings in your own environment.



How to connect VNets that are in the same subscription



Before you begin

Before beginning, you need to install the Azure Resource Manager PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Step 1 - Plan your IP address ranges

In the following steps, we create two virtual networks along with their respective gateway subnets and configurations. We 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.

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
- DNS Server: 8.8.8.8
- GatewayName: VNet1GW
- Public IP: VNet1GWIP
- VPNTType: RouteBased
- Connection(1to4): VNet1toVNet4
- Connection(1to5): VNet1toVNet5
- 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
- DNS Server: 8.8.8.8
- GatewayName: VNet4GW
- Public IP: VNet4GWIP
- VPNTType: RouteBased
- Connection: VNet4toVNet1
- ConnectionType: VNet2VNet

Step 2 - Create and configure TestVNet1

1. Declare your variables

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


```

$Sub1 = "Replace_With_Your_Subscription_Name"
$RG1 = "TestRG1"
$Location1 = "East US"
$VNetName1 = "TestVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$VNetPrefix1 = "10.11.0.0/16"
$VNetPrefix2 = "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"
$GWIPName1 = "VNet1GWIP"
$GWIPconfName1 = "gwipconf1"
$Connection14 = "VNet1toVNet4"
$Connection15 = "VNet1toVNet5"

```

2. Connect to your subscription

Switch to PowerShell mode to use the Resource Manager cmdlets. Open your PowerShell console and connect to your account. Use the following example to help you connect:

```
Login-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName $Sub1
```

3. Create a new resource group

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

4. Create the subnet configurations for TestVNet1

This example creates a virtual network named TestVNet1 and three subnets, one called GatewaySubnet, one called FrontEnd, and one called Backend. When substituting values, it's important that you always name your gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation will fail.

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

```

$Fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$Besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$Gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

```

5. Create TestVNet1

```
New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 `
-Location $Location1 -AddressPrefix $VNetPrefix1,$VNetPrefix2 -Subnet $fesub1,$besub1,$gwsb1
```

6. Request a public IP address

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.

```
$gwip1 = New-AzureRmPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 `
-Location $Location1 -AllocationMethod Dynamic
```

7. Create the gateway configuration

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

```
$vnet1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet1
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 `
-Subnet $subnet1 -PublicIpAddress $gwip1
```

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 take a while (45 minutes or more to complete).

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

Step 3 - Create and configure TestVNet4

Once you've configured TestVNet1, create TestVNet4. Follow the steps below, replacing the values with your own when needed. This step can be done within the same PowerShell session because it is in the same subscription.

1. Declare your variables

Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG4 = "TestRG4"
$Location4 = "West US"
$VnetName4 = "TestVNet4"
$FESubName4 = "FrontEnd"
$BESubName4 = "Backend"
$GWSubName4 = "GatewaySubnet"
$VnetPrefix41 = "10.41.0.0/16"
$VnetPrefix42 = "10.42.0.0/16"
$FESubPrefix4 = "10.41.0.0/24"
$BESubPrefix4 = "10.42.0.0/24"
$GWSubPrefix4 = "10.42.255.0/27"
$DNS4 = "8.8.8.8"
$GWName4 = "VNet4GW"
$GWIPName4 = "VNet4GWIP"
$GWIPconfName4 = "gwipconf4"
$Connection41 = "VNet4toVNet1"
```

Before you continue, make sure you are still connected to Subscription 1.

2. Create a new resource group

```
New-AzureRmResourceGroup -Name $RG4 -Location $Location4
```

3. Create the subnet configurations for TestVNet4

```
$fesub4 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName4 -AddressPrefix $FESubPrefix4  
$besub4 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName4 -AddressPrefix $BESubPrefix4  
$gwsb4 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName4 -AddressPrefix $GWSubPrefix4
```

4. Create TestVNet4

```
New-AzureRmVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4 `   
-Location $Location4 -AddressPrefix $VnetPrefix41,$VnetPrefix42 -Subnet $fesub4,$besub4,$gwsb4
```

5. Request a public IP address

```
$gwpip4 = New-AzureRmPublicIpAddress -Name $GWIPName4 -ResourceGroupName $RG4 `   
-Location $Location4 -AllocationMethod Dynamic
```

6. Create the gateway configuration

```
$vnet4 = Get-AzureRmVirtualNetwork -Name $VnetName4 -ResourceGroupName $RG4  
$subnet4 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet4  
$gwipconf4 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName4 -Subnet $subnet4 -PublicIpAddress $gwpip4
```

7. Create the TestVNet4 gateway

```
New-AzureRmVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4 `   
-Location $Location4 -IpConfigurations $gwipconf4 -GatewayType Vpn `   
-VpnType RouteBased -GatewaySku Standard
```

Step 4 - Connect the gateways

1. Get both virtual network gateways

In this example, because both gateways are in the same subscription, this step can be completed in the same PowerShell session.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1  
$vnet4gw = Get-AzureRmVirtualNetworkGateway -Name $GWName4 -ResourceGroupName $RG4
```

2. Create the TestVNet1 to TestVNet4 connection

In this step, you create the connection from TestVNet1 to TestVNet4. You'll see a shared key referenced in the examples. You can use your own values for the shared key. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection14 -ResourceGroupName $RG1 `   
-VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet4gw -Location $Location1 `   
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

3. Create the TestVNet4 to TestVNet1 connection

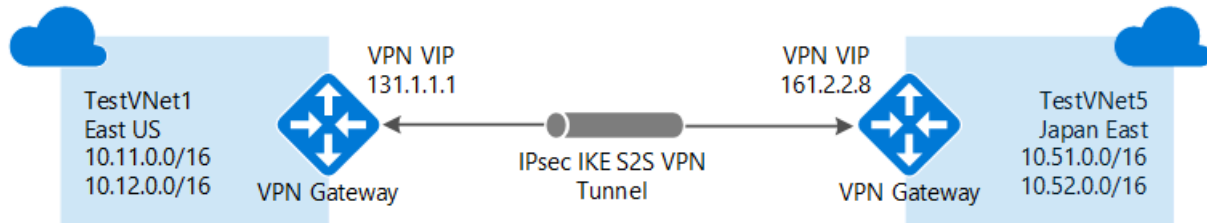
This step is similar to the one above, except you are creating the connection from TestVNet4 to TestVNet1. Make sure the shared keys match.

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection41 -ResourceGroupName $RG4 `
-VirtualNetworkGateway1 $vnet4gw -VirtualNetworkGateway2 $vnet1gw -Location $Location4 `
-ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

The connection should be established after a few minutes.

4. Verify your connection. See the section [How to verify your connection](#).

How to connect VNets that are in different subscriptions



In this scenario, we connect TestVNet1 and TestVNet5. TestVNet1 and TestVNet5 reside in a different subscription. The steps for this configuration add an additional VNet-to-VNet connection in order to connect TestVNet1 to TestVNet5.

The difference here 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.

The instructions continue from the previous steps listed above. You must complete [Step 1](#) and [Step 2](#) to create and configure TestVNet1, and the VPN Gateway for TestVNet1. Once you complete Step 1 and Step 2, continue with Step 5 to create TestVNet5.

Step 5 - Verify the additional 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
- DNS Server: 8.8.8.8
- GatewayName: VNet5GW
- Public IP: VNet5GWIP
- VPNTType: RouteBased
- Connection: VNet5toVNet1
- ConnectionType: VNet2VNet

Additional Values for TestVNet1:

- Connection: VNet1toVNet5

Step 6 - 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"
$DNSS = "8.8.8.8"
$GWName5 = "VNet5GW"
$GWIPName5 = "VNet5GWIP"
$GWIPconfName5 = "gwipconf5"
$Connection51 = "VNet5toVNet1"
```

2. Connect to subscription 5

Open your PowerShell console and connect to your account. Use the following sample to help you connect:

```
Login-AzureRmAccount
```

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName $Sub5
```

3. Create a new resource group

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5
```

4. Create the subnet configurations for TestVNet4

```
$fesub5 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName5 -AddressPrefix $FESubPrefix5
$besub5 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName5 -AddressPrefix $BESubPrefix5
$gwsb5 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName5 -AddressPrefix $GWSubPrefix5
```

5. Create TestVNet5

```
New-AzureRmVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5 -Location $Location5 `
-AddressPrefix $VnetPrefix51,$VnetPrefix52 -Subnet $fesub5,$besub5,$gwsb5
```

6. Request a public IP address

```
$gwpip5 = New-AzureRmPublicIpAddress -Name $GWIPName5 -ResourceGroupName $RG5 `
-Location $Location5 -AllocationMethod Dynamic
```

7. Create the gateway configuration

```
$vnet5 = Get-AzureRmVirtualNetwork -Name $VnetName5 -ResourceGroupName $RG5
$subnet5 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet5
$gwipconf5 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName5 -Subnet $subnet5 -PublicIpAddress $gwpip5
```

8. Create the TestVNet5 gateway

```
New-AzureRmVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5 -Location $Location5 `
-IpConfigurations $gwipconf5 -GatewayType Vpn -VpnType RouteBased -GatewaySku Standard
```

Step 7 - Connecting the gateways

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

Make sure you log in and connect to Subscription 1.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
```

Copy the output of the following elements and send these to the administrator of Subscription 5 via email or another method.

```
$vnet1gw.Name
$vnet1gw.Id
```

These two elements will have values similar to the following example output:

```
PS D:\> $vnet1gw.Name
VNet1GW
PS D:\> $vnet1gw.Id
/subscriptions/b636ca99-6f88-4df4-a7c3-
2f8dc4545509/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW
```

2. **[Subscription 5]** Get the virtual network gateway for Subscription 5

Make sure you log in and connect to Subscription 5.

```
$vnet5gw = Get-AzureRmVirtualNetworkGateway -Name $GWName5 -ResourceGroupName $RG5
```

Copy the output of the following elements and send these to the administrator of Subscription 1 via email or another method.

```
$vnet5gw.Name
$vnet5gw.Id
```

These two elements will have values similar to the following example output:

```
PS C:\> $vnet5gw.Name
VNet5GW
PS C:\> $vnet5gw.Id
/subscriptions/66c8e4f1-ecd6-47ed-9de7-
7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW
```

3. [Subscription 1] Create the TestVNet1 to TestVNet5 connection

In this step, you create the connection from TestVNet1 to TestVNet5. The difference here is that \$vnet5gw cannot be obtained directly because it is in a different subscription. You will need to create a new PowerShell object with the values communicated from Subscription 1 in the steps above. Use the example below. Replace the Name, Id, and shared key with your own values. The important thing is that the shared key must match for both connections. Creating a connection can take a short while to complete.

Make sure you connect to Subscription 1.

```
$vnet5gw = New-Object Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet5gw.Name = "VNet5GW"
$vnet5gw.Id = "/subscriptions/66c8e4f1-ecd6-47ed-9de7-
7e530de23994/resourceGroups/TestRG5/providers/Microsoft.Network/virtualNetworkGateways/VNet5GW"
$Connection15 = "VNet1toVNet5"
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName SRG1 -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.

Make sure you connect to Subscription 5.

```
$vnet1gw = New-Object Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
$vnet1gw.Name = "VNet1GW"
$vnet1gw.Id = "/subscriptions/b636ca99-6f88-4df4-a7c3-
2f8dc4545509/resourceGroups/TestRG1/providers/Microsoft.Network/virtualNetworkGateways/VNet1GW "
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection51 -ResourceGroupName SRG5 -VirtualNetworkGateway1
$vnet5gw -VirtualNetworkGateway2 $vnet1gw -Location $Location5 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3'
```

How to verify a connection

IMPORTANT

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

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.

```
Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1"
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite"
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}
```

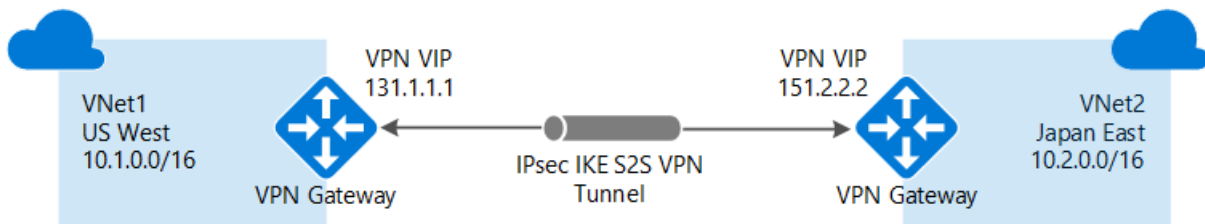
Next steps

- Once your connection is complete, you can add virtual machines to your virtual networks. See the [Virtual Machines documentation](#) for more information.
- For information about BGP, see the [BGP Overview](#) and [How to configure BGP](#).

Configure a VNet-to-VNet connection for the classic deployment model

1/17/2017 • 13 min to read • [Edit on GitHub](#)

This article walks you through the steps to create and connect virtual networks together using the classic deployment model (also known as Service Management). The following steps use the Azure classic portal to create the VNets and gateways, and PowerShell to configure the VNet-to-VNet connection. You cannot configure the connection in the portal.



Deployment models and methods for VNet-to-VNet connections

It's important to understand that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, verify that you are using the instructions for the deployment model that you want to work in. The two models are not completely compatible with each other.

For example, if you are working with a virtual network that was created using the classic deployment model and wanted to add a connection to the VNet, you would use the deployment methods that correspond to the classic deployment model, not Resource Manager. If you are working with a virtual network that was created using the Resource Manager deployment model, you would use the deployment methods that correspond with Resource Manager, not classic.

For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

The following table shows the currently available deployment models and methods for VNet-to-VNet configurations. When an article with configuration steps is available, we link directly to it from this table.

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

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

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

About VNet-to-VNet connections

Connecting a virtual network to another virtual network (VNet-to-VNet) 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.

VNet-to-VNet FAQ 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.

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 settings

VIRTUAL NETWORK	ADDRESS SPACE	REGION	CONNECTS TO LOCAL NETWORK SITE
VNet1	VNet1 (10.1.0.0/16)	US West	VNet2Local (10.2.0.0/16)
VNet2	VNet2 (10.2.0.0/16)	Japan East	VNet1Local (10.1.0.0/16)

Step 2 - Create VNet1

In this step, we create VNet1. When using any of the examples, be sure to substitute your own values. If your VNet already exists, you don't need to do this step. But, you do need to verify that the IP address ranges don't overlap with the ranges for your second VNet, or with any of the other VNets to which you want to connect.

1. Log in to the [Azure classic portal](#). In this article, we use the classic portal because some of the required configuration settings are not yet available in the Azure portal.
2. In the lower left-hand corner of the screen, click **New > Network Services > Virtual Network > Custom Create** to begin the configuration wizard. As you navigate through the wizard, add the specified values to each page.

Virtual Network Details

On the Virtual Network Details page, enter the following information:

CREATE A VIRTUAL NETWORK

Virtual Network Details

NAME

VNET

LOCATION

East Asia
Southeast Asia
North Europe
West Europe
East US
North Central US
South Central US
West US
Japan East
Japan West
Brazil South

- **Name** - Name your virtual network. For example, VNet1.
- **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.

DNS Servers and VPN Connectivity

On the DNS Servers and VPN Connectivity page, enter the following information, and then click the next arrow on the lower right.

DNS Servers and VPN Connectivity

DNS SERVERS ?

DNS110.1.0.10

SELECT OR ENTER NAME

IP ADDRESS

POINT-TO-SITE CONNECTIVITY ?

☐ Configure a point-to-site VPN

SITE-TO-SITE CONNECTIVITY ?

☐ Configure a site-to-site VPN

- **DNS Servers** - Enter the DNS server name and IP address, or select a previously registered DNS server from the dropdown. 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. If you want to have name resolution between your virtual networks, you have to configure your own DNS server, rather than using the name resolution that is provided by Azure.
- Don't select any of the checkboxes for P2S or S2S connectivity. Click the arrow on the lower right to move to the next screen.

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 (DIPS) that will be assigned to the VMs and other role instances that you deploy to this virtual network.

If you are creating a VNet that will also have a connection to your on-premises network, it's especially important to select a range that does not overlap with any of the ranges that are used for your on-premises network. In that case, you need to coordinate with your network administrator. Your network administrator may need to carve out a range of IP addresses from your on-premises network address space for you to use for your VNet.

Virtual Network Address Spaces

ADDRESS SPACE	STARTING IP	CIDR (ADDRESS COUNT)	USABLE ADDRESS RANGE
10.1.0.0/16	10.1.0.0	/16 (65536)	10.1.0.0 - 10.1.255.255
SUBNETS			
Subnet-1	10.1.0.0	/19 (8192)	10.1.0.0 - 10.1.31.255
add subnet			
add address space			

- **Address Space** - including Starting IP and Address Count. Verify that the address spaces you specify don't overlap with any of the address spaces that you have on your on-premises network. For this example, we use 10.1.0.0/16 for VNet1.
- **Add subnet** - including Starting IP and Address Count. Additional subnets are not required, but you may want to create a separate subnet for VMs that will have static DIPS. Or you might want to have your VMs in a subnet that is separate from your other role instances.

Click the checkmark on the lower right of the page and your virtual network will begin to create. When it completes, you will see "Created" listed under Status on the Networks page.

Step 3 - Create VNet2

Next, repeat the preceding steps to create another VNet. In later steps, you will connect the two VNets. You can refer to the [example settings](#) in Step 1. If your VNet already exists, you don't need to do this step. However, you need to verify that the IP address ranges don't overlap with any of the other VNets or on-premises networks that

you want to connect to.

Step 4 - Add the local network sites

When you create a VNet-to-VNet configuration, you need to configure local network sites, which are shown in the **Local Networks** page of the portal. Azure uses the settings specified in each local network site to determine how to route traffic between the VNets. You determine the name you want to use to refer to each local network site. It's best to use something descriptive, as you select the value from a dropdown list in later steps.

For example, VNet1 connects to a local network site that you create named "VNet2Local". The settings for VNet2Local contain the address prefixes for VNet2, and a public IP address for the VNet2 gateway. VNet2 connects to a local network site you create named "VNet1Local" that contains the address prefixes for VNet1 and the public IP address for the VNet1 gateway.

Add the local network site VNet1Local

1. In the lower left-hand corner of the screen, click **New > Network Services > Virtual Network > Add Local Network**.
2. On the **Specify your local network details** page, for **Name**, enter a name that you want to use to represent the network that you want to connect to. In this example, you can use "VNet1Local" to refer to the IP address ranges and gateway for VNet1.
3. For **VPN Device IP address (optional)**, specify any valid public IP address. Typically, you'd use the actual external IP address for a VPN device. For VNet-to-VNet configurations, you use the public IP address that is assigned to the gateway for your VNet. But, given that you've not yet created the gateway, you can 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 gateway IP addresses once Azure generates it. Click the arrow to advance to the next screen.
4. On the **Specify the address page**, enter the IP address range and address count for VNet1. This must correspond exactly to the range that is configured for VNet1. Azure uses the IP address ranges that you specify to route the traffic intended for VNet1. Click the checkmark to create the local network.

Add the local network site VNet2Local

Use the steps above to create the local network site "VNet2Local". You can refer to the values in the [example settings](#) in Step 1, if necessary.

Configure each VNet to point to a local network

Each VNet must point to the respective local network that you want to route traffic to.

For VNet1

1. Navigate to the **Configure** page for virtual network **VNet1**.
2. Under site-to-site connectivity, select "Connect to the local network", and then select **VNet2Local** as the local network from the dropdown.
3. Save your settings.

For VNet2

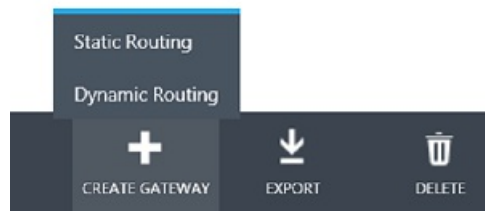
1. Navigate to the **Configure** page for virtual network **VNet2**.
2. Under site-to-site connectivity, select "Connect to the local network", then select **VNet1Local** from the dropdown as the local network.
3. Save your settings.

Step 5 - Configure a gateway for each VNet

Configure a Dynamic Routing gateway for each virtual network. This configuration does not support Static

Routing gateways. If you are using VNets that were previously configured and that already have Dynamic Routing gateways, you don't need to do this step. If your gateways are Static Routing, you need to delete them and recreate them as Dynamic Routing gateways. If you delete a gateway, the public IP address assigned to it gets released, and you need to go back and reconfigure any of your local networks and VPN devices with the new public IP address for the new gateway.

1. On the **Networks** page, verify that the status column for your virtual network is **Created**.
2. In the **Name** column, click the name of your virtual network. For this example, we use "VNet1".
3. On the **Dashboard** page, notice that this VNet doesn't have a gateway configured yet. You'll see this status change as you go through the steps to configure your gateway.
4. At the bottom of the page, click **Create Gateway** and **Dynamic Routing**. When the system prompts you to confirm that you want the gateway created, click Yes.



5. When your gateway is creating, notice the gateway graphic on the page changes to yellow and says "Creating Gateway". It typically takes about 30 minutes for the gateway to create.
6. Repeat the same steps for VNet2. You don't need the first VNet gateway to complete before you begin to create the gateway for your other VNet.
7. When the gateway status changes to "Connecting", the public IP address for each gateway is visible in the Dashboard. Write down the IP address that corresponds to each VNet, taking care not to mix them up. These are the IP addresses that are used when you edit your placeholder IP addresses for the VPN Device for each local network.

Step 6 - Edit the local network

1. On the **Local Networks** page, click the name of the Local Network name that you want to edit, then click **Edit** at the bottom of the page. For **VPN Device IP address**, input the IP address of the gateway that corresponds to the VNet. For example, for VNet1Local, put in the gateway IP address assigned to VNet1. Then click the arrow at the bottom of the page.
2. On the **Specify the address space** page, click the checkmark on the lower right without making any changes.

Step 7 - Create the VPN connection

When all the previous steps have been completed, set the IPsec/IKE pre-shared keys and create the connection. This set of steps uses PowerShell and cannot be configured in the portal. See [How to install and configure Azure PowerShell](#) for more information about installing the Azure PowerShell cmdlets. Make sure to download the latest version of the Service Management (SM) cmdlets.

1. Open Windows PowerShell and log in.

```
Add-AzureAccount
```

2. Select the subscription that your VNets reside in.

```
Get-AzureSubscription | Sort SubscriptionName | Select SubscriptionName  
Select-AzureSubscription -SubscriptionName "<Subscription Name>"
```

3. Create the connections. In the examples, notice that the shared key is exactly the same. The shared key

must always match.

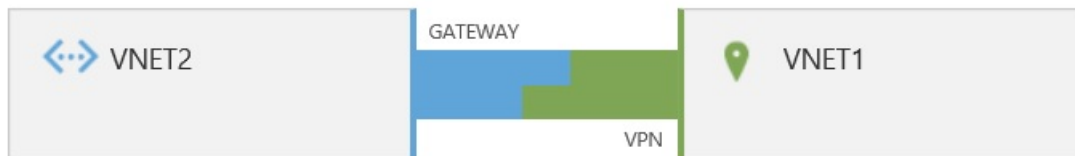
VNet1 to VNet2 connection

```
Set-AzureVNetGatewayKey -VNetName VNet1 -LocalNetworkSiteName VNet2Local -SharedKey A1b2C3D4
```

VNet2 to VNet1 connection

```
Set-AzureVNetGatewayKey -VNetName VNet2 -LocalNetworkSiteName VNet1Local -SharedKey A1b2C3D4
```

4. Wait for the connections to initialize. Once the gateway has initialized, the gateway looks like the following graphic.



IMPORTANT

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

Next steps

You can add virtual machines to your virtual networks. See the [Virtual Machines documentation](#) for more information.

Connect virtual networks from different deployment models in the portal

1/17/2017 • 15 min to read • [Edit on GitHub](#)

Azure currently has two management models: classic and Resource Manager (RM). 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. This article walks you through connecting classic VNets to Resource Manager VNets to allow the resources located in the separate deployment models to communicate with each other over a gateway connection.

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.

Deployment models and methods for VNet-to-VNet connections

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

We update the following table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from the table.

VNet-to-VNet

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

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

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

VNet peering

It's also possible to connect VNets without using a VPN gateway. If your VNets are in the same region, you may want to consider connecting them by using VNet peering. For more information, see the [VNet peering](#) article.

Before beginning

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.

In this article, we use the classic portal, the Azure portal, and PowerShell. Currently, it's not possible to create this configuration using only the Azure portal.

Prerequisites

- Both VNets have already been created.
- The address ranges for the VNets do not overlap with each other, or overlap with any of the ranges for other connections that the gateways may be connected to.
- You have installed the latest PowerShell cmdlets (1.0.2 or later). 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 the example settings as reference.

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

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

Virtual network gateway name = RMGateway

Gateway public IP name = gwpip

Gateway type = VPN

VPN type = Route-based

Local network gateway = ClassicVNetLocal

Section 1: Configure classic VNet settings

In this section, we create the local network and the gateway for your classic VNet. The instructions in this section use the classic portal. Currently, the Azure portal does not offer all the settings that pertain to a classic VNet.

Part 1 - Create a new local network

Open the [classic portal](#) and sign in with your Azure account.

1. On the bottom left corner of the screen, click **NEW > Network Services > Virtual Network > Add local network**.
2. In the **Specify your local network details** window, type a name for the RM VNet you want to connect to. In the **VPN device IP address (optional)** box, type any valid public IP address. This is just a temporary placeholder. You change this IP address later. On the bottom right corner of the window, click the arrow button.
3. On the **Specify the address space** page, in the **Starting IP** text box, type the network prefix and CIDR block for the Resource Manager VNet you want to connect to. This setting is used to specify the address space to

route to the RM VNet.

Part 2 - Associate the local network to your VNet

1. Click **Virtual Networks** at the top of the page to switch to the Virtual Networks screen, then click to select your classic VNet. On the page for your VNet, click **Configure** to navigate to the configuration page.
2. Under the **site-to-site connectivity** connection section, select the **Connect to the local network** checkbox. Then select the local network that you created. If you have multiple local networks that you created, be sure to select the one that you created to represent your Resource Manager VNet from the dropdown.
3. Click **Save** at the bottom of the page.

Part 3 - Create the gateway

1. After saving the settings, click **Dashboard** at the top of the page to change to the Dashboard page. On the bottom of the Dashboard page, click **Create Gateway**, then click **Dynamic Routing**. Click **Yes** to begin creating your gateway. A Dynamic Routing gateway is required for this configuration.
2. Wait for the gateway to be created. This can sometimes take 45 minutes or more to complete.

Part 4 - View the gateway public IP address

After the gateway has been created, you can view the gateway IP address on the **Dashboard** page. This is the public IP address of your gateway. Write down or copy the public IP address. You use it in later steps when you create the local network for your Resource Manager VNet configuration.

Section 2: Configure Resource Manager VNet settings

In this section, we create the virtual network gateway and the local network for your Resource Manager VNet. Don't start the following steps until after you have retrieved the public IP address for the classic VNet's gateway.

The screenshots are provided as examples. Be sure to replace the values with your own. If you are creating this configuration as an exercise, refer to these [values](#).

Part 1 - Create a gateway subnet

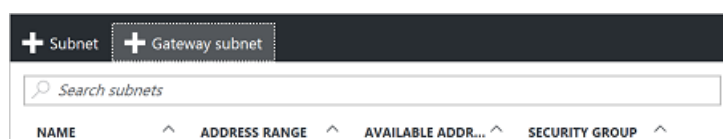
Before connecting your virtual network to a gateway, you first need to create the gateway subnet for the virtual network to which you want to connect. Create a gateway subnet with CIDR count of /28 or larger (/27, /26, etc.)

IMPORTANT

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

From a browser, navigate to the [Azure portal](#) and sign in with your Azure account.

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



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

Add subnet
RMVNet1

* Name
GatewaySubnet

* Address range (CIDR block) ⓘ
192.168.0.0/24

192.168.0.0 - 192.168.0.255 (256 addresses)

5. Click **OK** at the bottom of the blade to create the subnet.

Part 2 - Create a virtual network gateway

1. In the portal, on the left side, click + and type "Virtual Network Gateway" in search. Locate **Virtual network gateway** in the search return and click the entry. On the **Virtual network gateway** blade, click **Create** at the bottom of the blade. This opens the **Create virtual network gateway** blade.
2. On the **Create virtual network gateway** blade, fill in the values for your virtual network gateway.

Create virtual network gateway

* Name
[Empty]

Gateway type ⓘ
VPN ExpressRoute

VPN type ⓘ
Route-based Policy-based

* SKU ⓘ
Standard

* Virtual network ⓘ
Choose a virtual network

* Public IP address ⓘ
Choose a public IP address

* Subscription
Windows Azure Internal Consumption

Resource group ⓘ
-

* Location ⓘ
East US

☐ Pin to dashboard

Create [Automation options](#)

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

3. **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
4. **Gateway type:** Select **VPN**. VPN gateways use the virtual network gateway type **VPN**.
5. **VPN type:** Select the VPN type that is specified for your configuration. Most configurations require a Route-based VPN type.

6. **SKU:** Select the gateway SKU from the dropdown. The SKUs listed in the dropdown depend on the VPN type you select.
7. **Location:** Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, the virtual network will not appear in the 'Choose a virtual network' dropdown.
8. Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the **Choose a virtual network** blade. Select the VNet. If you don't see your VNet, make sure the **Location** field is pointing to the region in which your virtual network is located.
9. Choose a public IP address. Click **Public IP address** to open the **Choose public IP address** blade. Click **+Create New** to open the **Create public IP address blade**. Input a name for your public IP address. This blade creates a public IP address object to which a public IP address will be dynamically assigned. Click **OK** to save your changes to this blade.
10. **Subscription:** Verify that the correct subscription is selected.
11. **Resource group:** This setting is determined by the Virtual Network that you select.
12. Don't adjust the **Location** after you've specified the previous settings.
13. Verify the settings. You can select **Pin to dashboard** at the bottom of the blade if you want your gateway to appear on the dashboard.
14. Click **Create** to begin creating the gateway. The settings will be validated and you'll see the "Deploying Virtual network gateway" tile on the dashboard. Creating a gateway can take up to 45 minutes. You may need to refresh your portal page to see the completed status.



15. 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 will appear as a connected device. You can click the connected device (your virtual network gateway) to view more information.

Part 3 - Create a local network gateway

The 'local network gateway' typically refers to your on-premises location. It tells Azure which IP address ranges to route to the location and the public IP address of the device for that location. However, in this case, it refers to the address range and public IP address associated with your classic VNet and virtual network gateway.

Give the local network gateway a name by which Azure can refer to it. You can create your local network gateway while your virtual network gateway is being created. For this configuration, you use the public IP address that was assigned to your classic VNet gateway in the [previous section](#).

1. In the portal, from **All resources**, click **+Add**. In the **Everything** blade search box, type **Local network gateway**, then click to search. This will return a list. Click **Local network gateway** to open the blade, then click **Create** to open the **Create local network gateway** blade.

Create local network gateway

* Name
LocalNetworkName ✓

* IP address ⓘ
33.2.1.5 ✓

Address space ⓘ
192.168.3.0/24 ...
Add additional address range ...

* Subscription
Windows Azure Internal Consumption ▼

* Resource group ⓘ
☐ Create new ☒ Use existing
TestRG1 ▼

* Location
East US ▼

☐ Pin to dashboard

Create [Automation options](#)

2. On the **Create local network gateway blade**, specify a **Name** for your local network gateway object.
3. Specify a valid public **IP address** for the VPN device or virtual network gateway to which you want to connect.
If this local network represents an on-premises location, this is the public IP address of the VPN device that you want to connect to. It cannot be behind NAT and has to be reachable by Azure.
If this local network represents another VNet, you will specify the public IP address that was assigned to the virtual network gateway for that VNet.
4. **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.
5. For **Subscription**, verify that the correct subscription is showing.
6. For **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.
7. For **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.
8. Click **Create** to create the local network gateway.

Part 4 - Copy the public IP address

Once the virtual network gateway has finished creating, copy the public IP address that is associated with the gateway. You use it when you configure the local network settings for your classic VNet.

Section 3: Modify the local network for the classic VNet

Open the [classic portal](#).

1. In the classic portal, scroll down on the left side and click **Networks**. On the **networks** page, click **Local Networks** at the top of the page.
2. Click to select the local network that you configured in Part 1. At the bottom of the page, click **Edit**.
3. On the **Specify your local network details** page, replace the placeholder IP address with the public IP address for the Resource Manager gateway that you created in the previous section. Click the arrow to move to the next section. Verify that the **Address Space** is correct, and then click the checkmark to accept the changes.

Section 4: Create the connection

In this section, we create the connection between the VNets. The steps for this require PowerShell. You cannot create this connection in either of the portals. Make sure you have downloaded and installed both the classic (SM) and Resource Manager (RM) PowerShell cmdlets.

1. Log in to your Azure account in the PowerShell console. The following cmdlet prompts you for the login credentials for your Azure Account. After logging in, your account settings are downloaded so that they are available to Azure PowerShell.

```
Login-AzureRmAccount
```

Get a list of your Azure subscriptions if you have more than one subscription.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

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

2. Add your Azure Account to use the classic PowerShell cmdlets. To do so, you can use the following command:

```
Add-AzureAccount
```

3. Set your shared key by running the following sample. In this sample, `-VNetName` is the name of the classic VNet and `-LocalNetworkSiteName` is the name you specified for the local network when you configured it in the classic portal. The `-SharedKey` is a value that you can generate and specify. The value you specify here must be the same value that you specify in the next step when you create your connection.

```
Set-AzureVNetGatewayKey -VNetName ClassicVNet `
-LocalNetworkSiteName RmVNetLocal -SharedKey abc123
```

4. Create the VPN connection by running the following commands:

Set the variables

```
$vnet01gateway = Get-AzureRMLocalNetworkGateway -Name ClassicVNetLocal -ResourceGroupName RGI
$vnet02gateway = Get-AzureRmVirtualNetworkGateway -Name RmGateway -ResourceGroupName RGI
```

Create the connection

Note that the `-ConnectionType` is 'IPsec', not 'Vnet2Vnet'. In this sample, `-Name` is the name that you want

to call your connection. The following sample creates a connection named '*rm-to-classic-connection*'.

```
New-AzureRmVirtualNetworkGatewayConnection -Name rm-to-classic-connection -ResourceGroupName RGI `
-Location "East US" -VirtualNetworkGateway1 `
$Vnet02gateway -LocalNetworkGateway2 `
$Vnet01gateway -ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

Verify your connection

You can verify your connection by using the classic portal, the Azure portal, or PowerShell. You can use the following steps to verify your connection. Replace the values with your own.

To verify your connection by using PowerShell

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.

```
Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1",
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cf000-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite",
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}
```

To verify your connection by using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

VNet-to-VNet FAQ

View the FAQ details for additional information about VNet-to-VNet connections.

- The virtual networks can be in the same or different Azure regions (locations).
- A cloud service or a load balancing endpoint CANNOT span across virtual networks, even if they are connected together.
- Connecting multiple Azure virtual networks together doesn't require any on-premises VPN gateways unless cross-premises connectivity is required.
- VNet-to-VNet supports connecting virtual networks. It does not support connecting virtual machines or cloud services NOT in a virtual network.
- VNet-to-VNet requires Azure VPN gateways with RouteBased (previously called Dynamic Routing) VPN types.
- Virtual network connectivity can be used simultaneously with multi-site VPNs. There is a maximum of 10 (Default/Standard Gateways) or 30 (HighPerformance Gateways) 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 VNet-to-VNet connections to fail.
- Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.
- 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.
- VNet-to-VNet traffic travels across the Microsoft Network, not the Internet.
- VNet-to-VNet traffic within the same region is free for both directions. Cross region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. Please refer to the [pricing page](#) for details.

Connect virtual networks from different deployment models using PowerShell

1/17/2017 • 15 min to read • [Edit on GitHub](#)

Azure currently has two management models: classic and Resource Manager (RM). 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. This article walks you through connecting classic VNets to Resource Manager VNets to allow the resources located in the separate deployment models to communicate with each other over a gateway connection.

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.

Deployment models and methods for connecting VNets in different deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

We update the following table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from the table.

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

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

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

VNet peering

It's also possible to connect VNets without using a VPN gateway. If your VNets are in the same region, you may want to consider connecting them by using VNet peering. For more information, see the [VNet peering](#) article.

Before beginning

The following steps walk you through the settings necessary to configure a dynamic or route-based gateway for each VNet and create a VPN connection between the gateways. This configuration does not support static or policy-based gateways.

Prerequisites

- Both VNets have already been created.
- The address ranges for the VNets do not overlap with each other, or overlap with any of the ranges for other connections that the gateways may be connected to.
- You have installed the latest PowerShell cmdlets (1.0.2 or later). 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 the example settings as a reference.

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 = gwpip

Local Network Gateway = ClassicVNetLocal

Virtual Network Gateway name = RMGateway

Gateway IP addressing configuration = gwipconfig

Section 1 - Configure the classic VNet

Part 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. You will be using the SM PowerShell cmdlets to complete this part of the configuration.

```
Add-AzureAccount
```

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. You will edit the file and then import it to Azure.

```
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](#).

Part 2 -Verify the gateway subnet

In the **VirtualNetworkSites** element, add a gateway subnet to your VNet if one has not already been created. When working with the network configuration file, the gateway subnet **MUST** be named "GatewaySubnet" or Azure cannot recognize and use it as a gateway subnet.

IMPORTANT

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

Example:

```
<VirtualNetworkSites>
  <VirtualNetworkSite name="ClassicVNet" Location="West US">
    <AddressSpace>
      <AddressPrefix>10.0.0.0/24</AddressPrefix>
    </AddressSpace>
    <Subnets>
      <Subnet name="Subnet-1">
        <AddressPrefix>10.0.0.0/27</AddressPrefix>
      </Subnet>
      <Subnet name="GatewaySubnet">
        <AddressPrefix>10.0.0.32</AddressPrefix>
      </Subnet>
    </Subnets>
  </VirtualNetworkSite>
</VirtualNetworkSites>
```

Part 3 - Add the local network site

The local network site you add represents the RM VNet to which you want to connect. You will have to 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>
```

Part 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>
```

Part 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 should see something similar to this result showing that the import succeeded.

OperationDescription	OperationId	OperationStatus
Set-AzureVNetConfig	e0ee6e66-9167-cfa7-a746-7casb9	Succeeded

Part 6 - Create the gateway

You can create the VNet gateway either by using the classic portal, or by using PowerShell.

Before running this sample, refer to the network configuration file that you downloaded for the exact names that Azure expects to see. The network configuration file contains the values for your classic virtual networks. Sometimes the names for classic VNets are changed in the network configuration file when creating classic VNet settings in the Azure portal due to the differences in the deployment models. For example, if you used the Azure portal to create a classic VNet named 'Classic VNet' and created it in a resource group named 'ClassicRG', the name that is contained in the network configuration file is converted to 'Group ClassicRG Classic VNet'. When specifying the name of a VNet that contains spaces, use quotation marks around the value.

Use the following example to create a dynamic routing gateway:

```
New-AzureVNetGateway -VNetName ClassicVNet -GatewayType DynamicRouting
```

You can check the status of the gateway by using the `Get-AzureVNetGateway` cmdlet.

Section 2: Configure the RM VNet gateway

To create a VPN gateway for the RM VNet, follow the following instructions. Don't start the steps until after you have retrieved the public IP address for the classic VNet's gateway.

1. **Log in to your Azure account** in the PowerShell console. The following cmdlet prompts you for the login credentials for your Azure Account. After logging in, your account settings are downloaded so that they are available to Azure PowerShell.

```
Login-AzureRmAccount
```

Get a list of your Azure subscriptions if you have more than one subscription.

```
Get-AzureRmSubscription
```

Specify the subscription that you want to use.

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

2. **Create a local network gateway.** In a virtual network, the local network gateway typically refers to your on-premises location. In this case, the local network gateway refers to your Classic VNet. Give it a name by which Azure can refer to it, and also specify the address space prefix. Azure uses the IP address prefix you specify to identify which traffic to send to your on-premises location. If you need to adjust the information here later, before creating your gateway, you can modify the values and run the sample again.

- `-Name` is the name you want to assign to refer to the local network gateway.
- `-AddressPrefix` is the Address Space for your classic VNet.
- `-GatewayIpAddress` is the public IP address of the classic VNet's gateway. Be sure to change the following sample to reflect the correct IP address.

```
New-AzureRmLocalNetworkGateway -Name ClassicVNetLocal `
-Location "West US" -AddressPrefix "10.0.0.0/24" `
-GatewayIpAddress "n.n.n.n" -ResourceGroupName RGI
```

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 will change. The only time the virtual network gateway IP address changes is when the gateway is deleted and recreated. It won't change across resizing, resetting, or other internal maintenance/upgrades of the gateway. In this step, we also set a variable that is used in a later step.

```
$ipaddress = New-AzureRmPublicIpAddress -Name gwpip `
-ResourceGroupName RGI -Location 'EastUS' `
-AllocationMethod Dynamic
```

4. **Verify that your virtual network has a gateway subnet.** If no gateway subnet exists, add one. Make sure the gateway subnet is named *GatewaySubnet*.
5. **Retrieve the subnet** used for the gateway by running the following command. In this step, we also set a variable to be used in the next step.

- `-Name` is the name of your Resource Manager VNet.
- `-ResourceGroupName` is the resource group that the VNet is associated with. The gateway subnet must already exist for this VNet and must be named *GatewaySubnet* to work properly.

```
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet `
-VirtualNetwork (Get-AzureRmVirtualNetwork -Name RMVNet -ResourceGroupName RGI)
```

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 cannot 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-AzureRmVirtualNetworkGatewayIpConfig `
-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 this to complete.

```
New-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName RGI `
-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-AzureRmPublicIpAddress -Name gwpip -ResourceGroupName RGI
```

Section 3: Modify the classic VNet local site settings

In this section, you will work with the classic VNet. You will replace the placeholder IP address that you used when specifying the local site settings that will be used to connect to the Resource Manager VNet gateway.

1. Export the network configuration file.

```
Get-AzureVNetConfig -ExportToFile C:\AzureNet\NetworkConfig.xml
```

2. Using a text editor, modify the value for VPNGatewayAddress. Replace the placeholder IP address with the public IP address of the Resource Manager gateway and then save the changes.

```
<VPNGatewayAddress>13.68.210.16</VPNGatewayAddress>
```

3. Import the modified network configuration file to Azure.

```
Set-AzureVNetConfig -ConfigurationPath C:\AzureNet\NetworkConfig.xml
```

Section 4: Create a connection between the gateways

Creating a connection between the gateways requires PowerShell. You may need to add your Azure Account to use the classic PowerShell cmdlets. To do so, use `Add-AzureAccount`.

1. In the PowerShell console, set your shared key. Before running the cmdlets, refer to the network configuration file that you downloaded for the exact names that Azure expects to see. When specifying the name of a VNet that contains spaces, use single quotation marks around the value.

In following example, `-VNetName` is the name of the classic VNet and `-LocalNetworkSiteName` is the name you specified for the local network site. The `-SharedKey` is a value that you generate and specify. In the example, we used 'abc123', but you can generate and use something more complex. The important thing is that the value you specify here must be the same value that you specify in the next step when you create your connection. The return should show **Status: Successful**.

```
Set-AzureVNetGatewayKey -VNetName 'ClassicVNet' `
-LocalNetworkSiteName RmVNetLocal -SharedKey abc123
```

2. Create the VPN connection by running the following commands.

Set the variables

```
$vnet01gateway = Get-AzureRmLocalNetworkGateway -Name ClassicVNetLocal -ResourceGroupName RGI
$vnet02gateway = Get-AzureRmVirtualNetworkGateway -Name RmGateway -ResourceGroupName RGI
```

Create the connection

Notice that the `-ConnectionType` is IPsec, not Vnet2Vnet.

```
New-AzureRmVirtualNetworkGatewayConnection -Name RM-Classic -ResourceGroupName RGI `
-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

Verify your connection using PowerShell

You can use the `Get-AzureVNetConnection` to verify the connection for a classic virtual network gateway.

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"
```

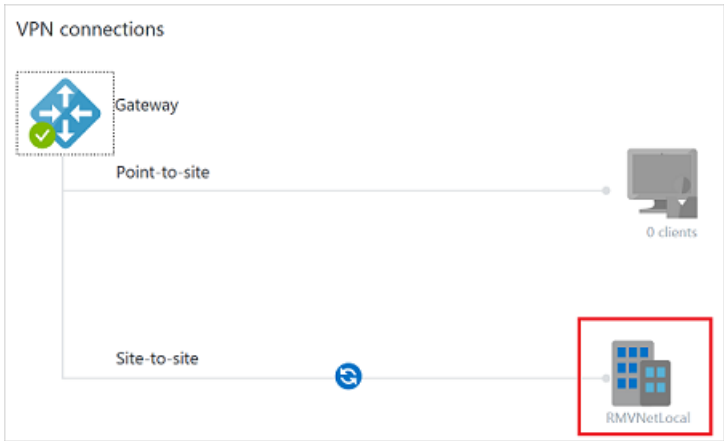
2. After the cmdlet has finished, view the values. In the example below, the Connectivity State shows as 'Connected' and you can see ingress and egress bytes.

```
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
```

Verify your connection using the Azure Portal

In the Azure portal, you can view the connection status for a classic VNet gateway by navigating to the connection. There are multiple ways to do this. The following steps show one way to navigate to your connection and verify.

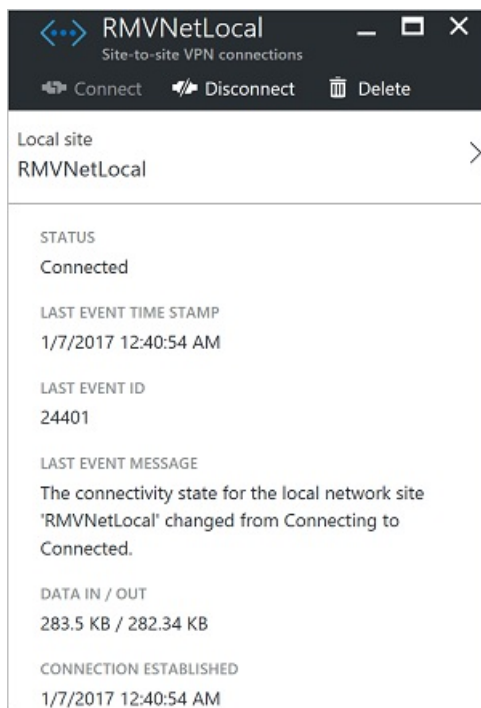
1. In the [Azure portal](#), click **All resources** and navigate to your classic virtual network.
2. On the virtual network blade, click **Overview** to access the **VPN connections** section of the blade.
3. On the VPN connections graphic, click the site.



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

Site-to-site VPN connections			
ClassicVNet			
+ Add VPN Device Scr...			
NAME	STATUS	LAST EVENT	DATA IN / OUT
RMVNetLocal	Connected	1/7/2017 12:40:54 AM	177.97 KB / 177.72 KB

5. To view more information about the connection, click the name of the connection to open the **Site-to-site VPN Connection** blade.



To verify the connection from your Resource Manager VNet to your classic VNet

Verify your connection using PowerShell

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.


```



Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1"
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite"
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}

```

Verify your connection using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	 2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	 RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

VNet-to-VNet considerations

- The virtual networks can be in the same or different Azure regions (locations).
- A cloud service or a load balancing endpoint CANNOT span across virtual networks, even if they are connected together.
- Connecting multiple Azure virtual networks together doesn't require any on-premises VPN gateways unless cross-premises connectivity is required.
- VNet-to-VNet supports connecting virtual networks. It does not support connecting virtual machines or cloud

services NOT in a virtual network.

- VNet-to-VNet requires Azure VPN gateways with RouteBased (previously called Dynamic Routing) VPN types.
- Virtual network connectivity can be used simultaneously with multi-site VPNs. There is a maximum of 10 (Default/Standard Gateways) or 30 (HighPerformance Gateways) 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 VNet-to-VNet connections to fail.
- Redundant tunnels between a pair of virtual networks are supported when one virtual network gateway is configured as active-active.
- 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.
- VNet-to-VNet traffic travels across the Microsoft Network, not the Internet.
- VNet-to-VNet traffic within the same region is free for both directions. Cross region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. Please refer to the [pricing page](#) for details.

Configure ExpressRoute and Site-to-Site coexisting connections for the Resource Manager deployment model

1/17/2017 • 9 min to read • [Edit on GitHub](#)

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. This configuration is not available in the Azure portal.

About Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

IMPORTANT

ExpressRoute circuits must be pre-configured before you follow the instructions below. Make sure that you have followed the guides to [create an ExpressRoute circuit](#) and [configure routing](#) before you follow the steps below.

Limits and limitations

- **Transit routing is not supported.** You cannot route (via Azure) between your local network connected via Site-to-Site VPN and your local network connected via ExpressRoute.
- **Basic SKU gateway is not supported.** You must use a non-Basic SKU gateway for both the [ExpressRoute gateway](#) and the [VPN gateway](#).
- **Only route-based VPN gateway is supported.** You must use a route-based [VPN Gateway](#).
- **Static route should be configured for your VPN gateway.** If your local network is connected to both ExpressRoute and a Site-to-Site VPN, you must have a static route configured in your local network to route the Site-to-Site VPN connection to the public Internet.
- **ExpressRoute gateway must be configured first.** You must create the ExpressRoute gateway first before you add the Site-to-Site VPN gateway.

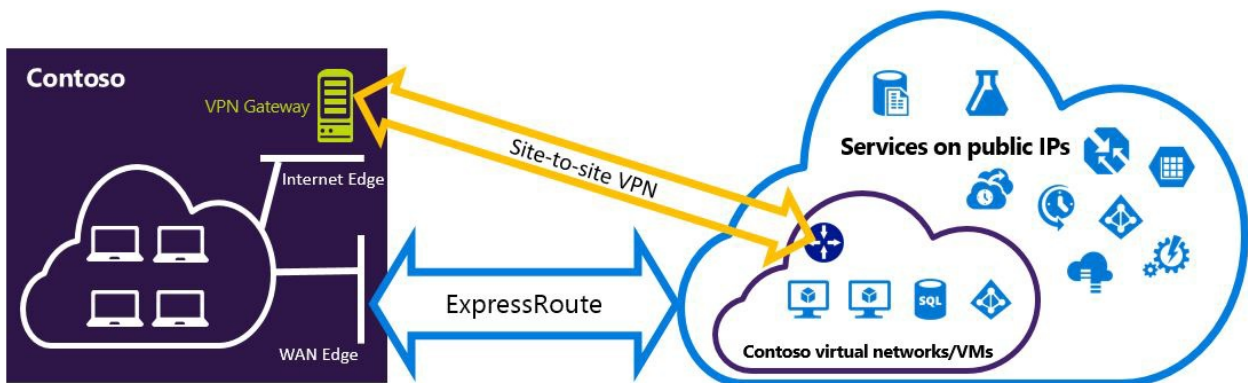
Configuration designs

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 applies only to virtual networks linked to the Azure private peering path. There is no VPN-based failover solution for services accessible through Azure public and Microsoft peerings. The ExpressRoute circuit is always the primary link. Data will flow through the Site-to-Site VPN path only if the ExpressRoute circuit fails.

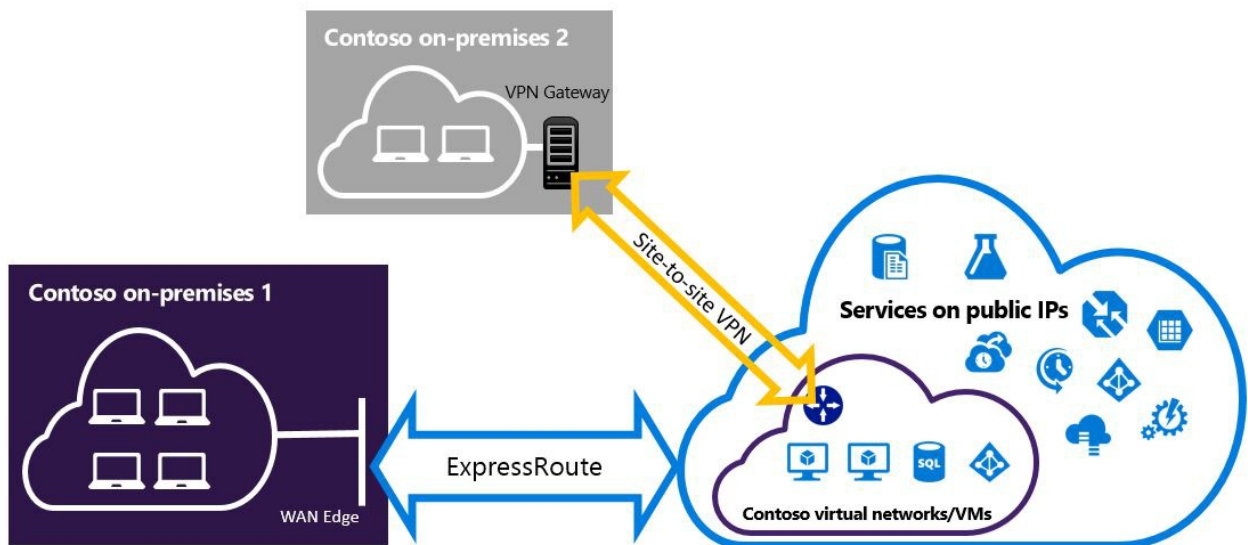
NOTE

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 in order to configure connections that can coexist. The configuration procedure that you select will depend 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 will walk you through creating a new virtual network using Resource Manager deployment model and creating new ExpressRoute and Site-to-Site VPN connections. To configure, follow the steps in the article section [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. The [To configure coexisting connections for an already existing VNet](#) section will walk you through deleting the gateway, and then creating new ExpressRoute and Site-to-Site VPN connections. Note that when creating the new connections, the steps must be completed in a very specific order. Don't use the instructions in other articles to create your gateways and connections.

In this procedure, creating connections that can coexist will require you to delete your gateway, and then configure new gateways. This means you will have downtime for your cross-premises connections while you delete and recreate your gateway and connections, but you will not need to migrate any of your VMs or services to a new virtual network. Your VMs and services will still be able to communicate out through the load balancer while you configure your gateway if they are configured to do so.

To create a new virtual network and coexisting connections

This procedure will walk you through creating a VNet and create Site-to-Site and ExpressRoute connections that will coexist.

1. You'll need to install the latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets. Note that the cmdlets that you'll use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Login your account and set up the environment.

```
login-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName 'yoursubscription'
$location = "Central US"
$resgrp = New-AzureRmResourceGroup -Name "ErVpnCoex" -Location $location
```

3. Create a virtual network including Gateway Subnet. For more information about the virtual network configuration, see [Azure Virtual Network configuration](#).

IMPORTANT

The Gateway Subnet must be /27 or a shorter prefix (such as /26 or /25).

Create a new VNet.

```
$vnet = New-AzureRmVirtualNetwork -Name "CoexVnet" -ResourceGroupName $resgrp.ResourceGroupName -Location $location -
AddressPrefix "10.200.0.0/16"
```

Add subnets.

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

Save the VNet configuration.

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

4. Create an ExpressRoute gateway. For more information about the ExpressRoute gateway configuration, see [ExpressRoute gateway configuration](#). The GatewaySKU must be *Standard*, *HighPerformance*, or

UltraPerformance.

```
$gwSubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzureRmPublicIpAddress -Name "ERGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -Location $location -
AllocationMethod Dynamic
$gwConfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "ERGatewayIpConfig" -SubnetId $gwSubnet.Id -PublicIpAddressId
$gwIP.Id
$gw = New-AzureRmVirtualNetworkGateway -Name "ERGateway" -ResourceGroupName $resgrp.ResourceGroupName -Location $location
-IPConfigurations $gwConfig -GatewayType "ExpressRoute" -GatewaySku Standard
```

5. Link the ExpressRoute gateway to the ExpressRoute circuit. After this step has been completed, the connection between your on-premises network and Azure, through ExpressRoute, is established. For more information about the link operation, see [Link VNets to ExpressRoute](#).

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

6. 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 must be *Standard*, *HighPerformance*, or *UltraPerformance*. The VpnType must *RouteBased*.

```
$gwSubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzureRmPublicIpAddress -Name "VPNGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -Location $location
-AllocationMethod Dynamic
$gwConfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "VPNGatewayIpConfig" -SubnetId $gwSubnet.Id -PublicIpAddressId
$gwIP.Id
New-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -Location $location -
IPConfigurations $gwConfig -GatewayType "Vpn" -VpnType "RouteBased" -GatewaySku "Standard"
```

Azure VPN gateway supports the BGP. You can specify `-EnableBgp` in the following command.

```
$azureVpn = New-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -Location
$location -IPConfigurations $gwConfig -GatewayType "Vpn" -VpnType "RouteBased" -GatewaySku "Standard" -EnableBgp $true
```

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.

7. Create a local site VPN gateway entity. This command doesn't configure your on-premises VPN gateway. Rather, it allows you to provide the local gateway settings, such as the public IP and the on-premises address space, so that the Azure VPN gateway can connect to it.

If your local VPN device only supports static routing, you can configure the static routes in the following way.

```
$MyLocalNetworkAddress = @"(10.100.0.0/16,"10.101.0.0/16","10.102.0.0/16")
$localVpn = New-AzureRmLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -GatewayIpAddress *Public IP* -AddressPrefix $MyLocalNetworkAddress
```

If your local VPN device supports the BGP and you want to enable dynamic routing, you need to know the BGP peering IP and the AS number that your local VPN device uses.

```
$localVPNPublicIP = "<Public IP>"
$localBGPPeeringIP = "<Private IP for the BGP session>"
$localBGPASN = "<ASN>"
$localAddressPrefix = $localBGPPeeringIP + "/32"
$localVpn = New-AzureRmLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -GatewayIpAddress $localVPNPublicIP -AddressPrefix $localAddressPrefix -BgpPeeringAddress $localBGPPeeringIP -
Asn $localBGPASN
```

- Configure your local VPN device to connect to the new Azure VPN gateway. For more information about VPN device configuration, see [VPN Device Configuration](#).
- Link the Site-to-Site VPN gateway on Azure to the local gateway.

```
$azureVpn = Get-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
New-AzureRmVirtualNetworkGatewayConnection -Name "VPNConnection" -ResourceGroupName $resgrp.ResourceGroupName -Location
$location -VirtualNetworkGateway1 $azureVpn -LocalNetworkGateway2 $localVpn -ConnectionType IPsec -SharedKey <yourkey>
```

To configure coexisting connections for an already existing VNet

If you have an existing virtual network, check the gateway subnet size. If the gateway subnet is /28 or /29, you must first delete the virtual network gateway and increase the gateway subnet size. The steps in this section will show you how to do that.

If the gateway subnet is /27 or larger and the virtual network is connected via ExpressRoute, you can skip the steps below and proceed to ["Step 6 - Create a Site-to-Site VPN gateway"](#) in the previous section.

NOTE

When you delete the existing gateway, your local premises will lose the connection to your virtual network while you are working on this configuration.

- You'll need to install the latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets. Note that the cmdlets that you'll 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.
- Delete the existing ExpressRoute or Site-to-Site VPN gateway.

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

- Delete Gateway Subnet.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Remove-AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet
```

- Add a Gateway Subnet that is /27 or larger.

NOTE

If you don't have enough IP addresses left in your virtual network to increase the gateway subnet size, you need to add more IP address space.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Add-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix "10.200.255.0/24"
```

Save the VNet configuration.

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

5. At this point, you'll have a VNet with no gateways. To create new gateways and complete your connections, you can proceed with [Step 4 - Create an ExpressRoute gateway](#), found in the preceding set of steps.

To add point-to-site configuration to the VPN gateway

You can follow the steps below to add Point-to-Site configuration to your VPN gateway in a co-existence setup.

1. Add VPN Client address pool.

```
$azureVpn = Get-AzureRmVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName  
Set-AzureRmVirtualNetworkGatewayVpnClientConfig -VirtualNetworkGateway $azureVpn -VpnClientAddressPool "10.251.251.0/24"
```

2. Upload the VPN root certificate to Azure for your VPN gateway. In this example, it's assumed that the root certificate is stored in the local machine where the following PowerShell cmdlets are run.

```
$p2sCertFullName = "RootErVpnCoexP2S.cer"  
$p2sCertMatchName = "RootErVpnCoexP2S"  
$p2sCertToUpload=get-childitem Cert:\CurrentUser\My | Where-Object {$_.Subject -match $p2sCertMatchName}  
if ($p2sCertToUpload.count -eq 1){  
    write-host "cert found"  
} else {  
    write-host "cert not found"  
    exit  
}  
$p2sCertData = [System.Convert]::ToBase64String($p2sCertToUpload.RawData)  
Add-AzureRmVpnClientRootCertificate -VpnClientRootCertificateName $p2sCertFullName -VirtualNetworkGatewayname $azureVpn.Name  
-ResourceGroupName $resgrp.ResourceGroupName -PublicCertData $p2sCertData
```

For more information on Point-to-Site VPN, see [Configure a Point-to-Site connection](#).

Next steps

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

Configure ExpressRoute and Site-to-Site coexisting connections for the classic deployment model

1/17/2017 • 8 min to read • [Edit on GitHub](#)

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 classic deployment model. This configuration is not available in the portal.

About Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

IMPORTANT

ExpressRoute circuits must be pre-configured before you follow the instructions below. Make sure that you have followed the guides to [create an ExpressRoute circuit](#) and [configure routing](#) before you follow the steps below.

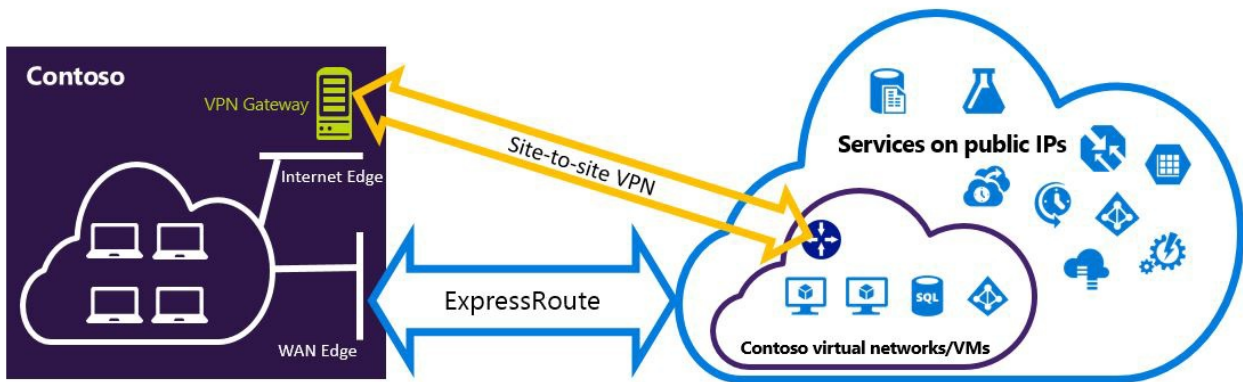
Limits and limitations

- **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.
- **Point-to-site is not supported.** You can't enable point-to-site VPN connections to the same VNet that is connected to ExpressRoute. Point-to-site VPN and ExpressRoute cannot coexist for the same VNet.
- **Forced tunneling cannot be enabled on the Site-to-Site VPN gateway.** You can only "force" all Internet-bound traffic back to your on-premises network via ExpressRoute.
- **Basic SKU gateway is not supported.** You must use a non-Basic SKU gateway for both the [ExpressRoute gateway](#) and the [VPN gateway](#).
- **Only route-based VPN gateway is supported.** You must use a route-based [VPN Gateway](#).
- **Static route should be configured for your VPN gateway.** If your local network is connected to both ExpressRoute and a Site-to-Site VPN, you must have a static route configured in your local network to route the Site-to-Site VPN connection to the public Internet.
- **ExpressRoute gateway must be configured first.** You must create the ExpressRoute gateway first before you add the Site-to-Site VPN gateway.

Configuration designs

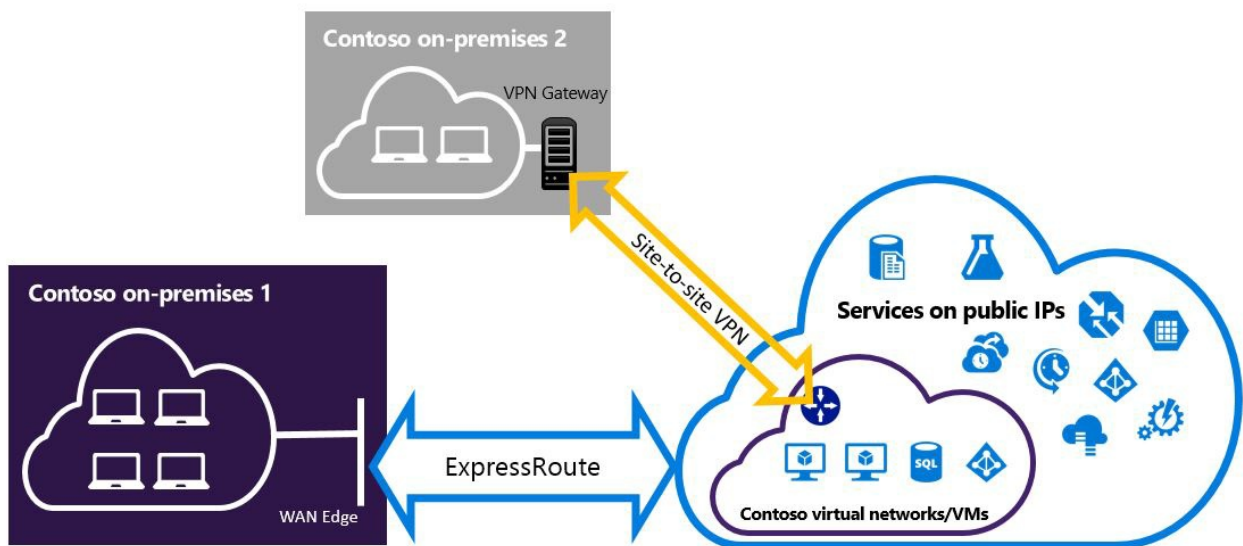
Configure a Site-to-Site VPN as a failover path for ExpressRoute

You can configure a Site-to-Site VPN connection as a backup for ExpressRoute. This applies only to virtual networks linked to the Azure private peering path. There is no VPN-based failover solution for services accessible through Azure public and Microsoft peerings. The ExpressRoute circuit is always the primary link. Data will flow through the Site-to-Site VPN path only if the ExpressRoute circuit fails.



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 in order to configure connections that can coexist. The configuration procedure that you select will depend 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 will walk you through creating a new virtual network using the classic deployment model and creating new ExpressRoute and Site-to-Site VPN connections. To configure, follow the steps in the article section [To create a new virtual network and coexisting connections](#).

- I already have a classic deployment model VNet.

You may already have a virtual network in place with an existing Site-to-Site VPN connection or ExpressRoute connection. The article section [To configure coexisting connections for an already existing](#)

[VNet](#) will walk you through deleting the gateway, and then creating new ExpressRoute and Site-to-Site VPN connections. Note that when creating the new connections, the steps must be completed in a very specific order. Don't use the instructions in other articles to create your gateways and connections.

In this procedure, creating connections that can coexist will require you to delete your gateway, and then configure new gateways. This means you will have downtime for your cross-premises connections while you delete and recreate your gateway and connections, but you will not need to migrate any of your VMs or services to a new virtual network. Your VMs and services will still be able to communicate out through the load balancer while you configure your gateway if they are configured to do so.

To create a new virtual network and coexisting connections

This procedure will walk you through creating a VNet and create Site-to-Site and ExpressRoute connections that will coexist.

1. You'll need to install the latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets. Note that the cmdlets that you'll use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Create a schema for your virtual network. For more information about the configuration schema, see [Azure Virtual Network configuration schema](#).

When you create your schema, make sure you use the following values:

- The gateway subnet for the virtual network must be /27 or a shorter prefix (such as /26 or /25).
- The gateway connection type is "Dedicated".

```
<VirtualNetworkSite name="My Azure VNET" Location="Central US">
  <AddressSpace>
    <AddressPrefix>10.17.159.192</AddressPrefix>
  </AddressSpace>
  <Subnets>
    <Subnet name="Subnet-1">
      <AddressPrefix>10.17.159.192</AddressPrefix>
    </Subnet>
    <Subnet name="GatewaySubnet">
      <AddressPrefix>10.17.159.224</AddressPrefix>
    </Subnet>
  </Subnets>
  <Gateway>
    <ConnectionsToLocalNetwork>
      <LocalNetworkSiteRef name="MyLocalNetwork">
        <Connection type="Dedicated" />
      </LocalNetworkSiteRef>
    </ConnectionsToLocalNetwork>
  </Gateway>
</VirtualNetworkSite>
```

3. After creating and configuring your xml schema file, upload the file. This will create your virtual network.

Use the following cmdlet to upload your file, replacing the value with your own.

```
Set-AzureVNetConfig -ConfigurationPath 'C:\NetworkConfig.xml'
```

4. Create an ExpressRoute gateway. Be sure to specify the GatewaySKU as *Standard*, *HighPerformance*, or *UltraPerformance* and the GatewayType as *DynamicRouting*.

Use the following sample, substituting the values for your own.

```
New-AzureVNetGateway -VNetName MyAzureVNET -GatewayType DynamicRouting -GatewaySKU HighPerformance
```

- 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.

```
New-AzureDedicatedCircuitLink -ServiceKey <service-key> -VNetName MyAzureVNET
```

- Next, create your Site-to-Site VPN gateway. The GatewaySKU must be *Standard*, *HighPerformance*, or *UltraPerformance* and the GatewayType must be *DynamicRouting*.

```
New-AzureVirtualNetworkGateway -VNetName MyAzureVNET -GatewayName S2SVPN -GatewayType DynamicRouting -GatewaySKU HighPerformance
```

To retrieve the virtual network gateway settings, including the gateway ID and the public IP, use the

`Get-AzureVirtualNetworkGateway` cmdlet.

```
Get-AzureVirtualNetworkGateway
```

```
GatewayId      : 348ae011-ffa9-4add-b530-7cb30010565e
GatewayName    : S2SVPN
LastEventData  :
GatewayType    : DynamicRouting
LastEventTimeStamp : 5/29/2015 4:41:41 PM
LastEventMessage : Successfully created a gateway for the following virtual network: GNSDesMoines
LastEventID    : 23002
State          : Provisioned
VIPAddress     : 104.43.xy
DefaultSite    :
GatewaySKU     : HighPerformance
Location       :
VnetId        : 979aabcf-e47f-4136-ab9b-b4780c1e1bd5
SubnetId       :
EnableBgp      : False
OperationDescription : Get-AzureVirtualNetworkGateway
OperationId     : 42773656-85e1-a6b6-8705-35473f1e6f6a
OperationStatus : Succeeded
```

- 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.

IMPORTANT

The local site for the Site-to-Site VPN is not defined in the netcfg. Instead, you must use this cmdlet to specify the local site parameters. You cannot define it using either portal, or the netcfg file.

Use the following sample, replacing the values with your own.

```
New-AzureLocalNetworkGateway -GatewayName MyLocalNetwork -IpAddress <MyLocalGatewayIp> -AddressSpace <MyLocalNetworkAddress>
```

NOTE

If your local network has multiple routes, you can pass them all in as an array. `$MyLocalNetworkAddress = @("10.1.2.0/24","10.1.3.0/24","10.2.1.0/24")`

To retrieve the virtual network gateway settings, including the gateway ID and the public IP, use the `Get-AzureVirtualNetworkGateway` cmdlet. See the following example.

```
Get-AzureLocalNetworkGateway

GatewayId      : 532cb428-8c8c-4596-9a4f-7ae3a9fcd01b
GatewayName    : MyLocalNetwork
IpAddress      : 23.39.xy
AddressSpace    : {10.1.2.0/24}
OperationDescription : Get-AzureLocalNetworkGateway
OperationId     : ddc4bfae-502c-adc7-bd7d-1efbc00b3fe5
OperationStatus : Succeeded
```

8. Configure your local VPN device to connect to the new gateway. Use the information that you retrieved in step 6 when configuring your VPN device. For more information about VPN device configuration, see [VPN Device Configuration](#).
9. Link the Site-to-Site VPN gateway on Azure to the local gateway.

In this example, `connectedEntityId` is the local gateway ID, which you can find by running

`Get-AzureLocalNetworkGateway`. You can find `virtualNetworkGatewayId` by using the `Get-AzureVirtualNetworkGateway` cmdlet. After this step, the connection between your local network and Azure via the Site-to-Site VPN connection is established.

```
New-AzureVirtualNetworkGatewayConnection -connectedEntityId <local-network-gateway-id> -gatewayConnectionName Azure2Local -
gatewayConnectionType IPsec -sharedKey abc123 -virtualNetworkGatewayId <azure-s2s-vpn-gateway-id>
```

To configure coexisting connections for an already existing VNet

If you have an existing virtual network, check the gateway subnet size. If the gateway subnet is /28 or /29, you must first delete the virtual network gateway and increase the gateway subnet size. The steps in this section will show you how to do that.

If the gateway subnet is /27 or larger and the virtual network is connected via ExpressRoute, you can skip the steps below and proceed to "[Step 6 - Create a Site-to-Site VPN gateway](#)" in the previous section.

NOTE

When you delete the existing gateway, your local premises will lose the connection to your virtual network while you are working on this configuration.

1. You'll need to 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. Note that the cmdlets that you'll use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Delete the existing ExpressRoute or Site-to-Site VPN gateway. Use the following cmdlet, replacing the values with your own.

```
Remove-AzureVNetGateway -VnetName MyAzureVNET
```

3. Export the virtual network schema. Use the following PowerShell cmdlet, replacing the values with your own.

```
Get-AzureVNetConfig -ExportToFile "C:\NetworkConfig.xml"
```

4. Edit the network configuration file schema so that the gateway subnet is /27 or a shorter prefix (such as /26 or /25). See the following example.

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. For more information about the configuration schema, see [Azure Virtual Network configuration schema](#).

```
<Subnet name="GatewaySubnet">
  <AddressPrefix>10.17.159.224/27</AddressPrefix>
</Subnet>
```

5. If your previous gateway was a Site-to-Site VPN, you must also change the connection type to **Dedicated**.

```
<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="MyLocalNetwork">
      <Connection type="Dedicated" />
    </LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>
```

6. At this point, you'll have a VNet with no gateways. To create new gateways and complete your connections, you can proceed with [Step 4 - Create an ExpressRoute gateway](#), found in the preceding set of steps.

Next steps

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

Configure forced tunneling using the Azure Resource Manager deployment model

1/17/2017 • 6 min to read • [Edit on GitHub](#)

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

This article walks you through configuring forced tunneling for virtual networks created using the Resource Manager deployment model.

About Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

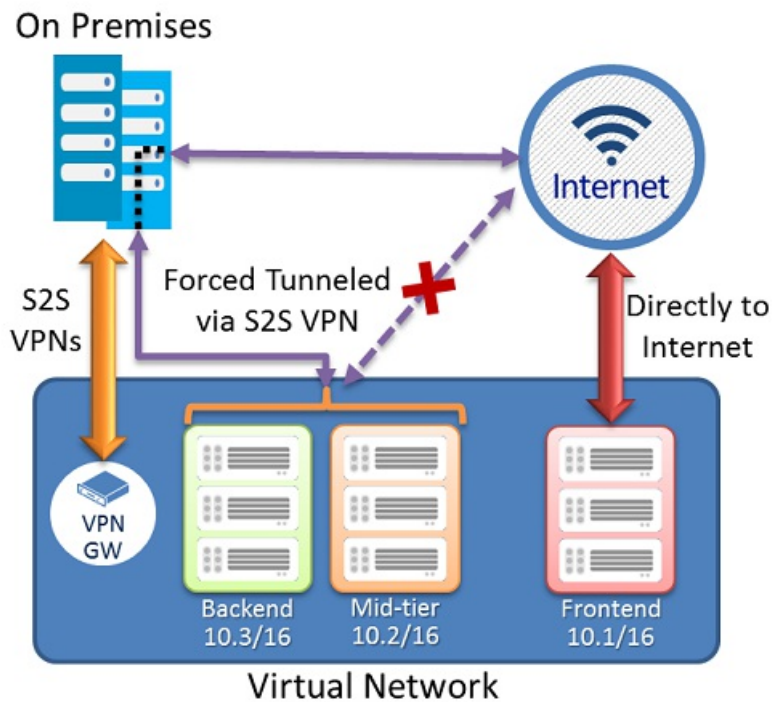
Deployment models and tools for forced tunneling

A forced tunneling connection can be configured for both the classic deployment model and the Resource Manager deployment model. See the following table for more information. We update this table as new articles, new deployment models, and additional tools become available for this configuration. When an article is available, we link directly to it from the table.

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

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. You also can apply forced tunneling to the entire virtual networks if there are no Internet-facing workloads in your 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 will be 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.
- ExpressRoute forced tunneling is not configured via this mechanism, but instead, is enabled by advertising a default route via the ExpressRoute BGP peering sessions. Please see the [ExpressRoute Documentation](#) for more information.

Configuration overview

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 3 subnets: *Frontend*,

Midtier, and *Backend*, with 4 cross-premises connections: *DefaultSiteHQ*, and 3 *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

Verify that you have the following items before beginning your 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](#).
- You'll need to install the latest version of the Azure Resource Manager PowerShell cmdlets (1.0 or later). See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Configure forced tunneling

1. In the PowerShell console, log in to your Azure account. This cmdlet prompts you for the login credentials for your Azure Account. After logging in, it downloads your account settings so they are available to Azure PowerShell.

```
Login-AzureRmAccount
```

2. Get a list of your Azure subscriptions.

```
Get-AzureRmSubscription
```

3. Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

4. Create a resource group.

```
New-AzureRmResourceGroup -Name "ForcedTunneling" -Location "North Europe"
```

5. Create a virtual network and specify subnets.

```
$s1 = New-AzureRmVirtualNetworkSubnetConfig -Name "Frontend" -AddressPrefix "10.1.0.0/24"
$s2 = New-AzureRmVirtualNetworkSubnetConfig -Name "Midtier" -AddressPrefix "10.1.1.0/24"
$s3 = New-AzureRmVirtualNetworkSubnetConfig -Name "Backend" -AddressPrefix "10.1.2.0/24"
$s4 = New-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -AddressPrefix "10.1.200.0/28"
$vn = New-AzureRmVirtualNetwork -Name "MultiTier-VNet" -Location "North Europe" -ResourceGroupName "ForcedTunneling" -
AddressPrefix "10.1.0.0/16" -Subnet $s1,$s2,$s3,$s4
```

6. Create the local network gateways.

```
$lng1 = New-AzureRmLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
GatewayIpAddress "111.111.111.111" -AddressPrefix "192.168.1.0/24"
$lng2 = New-AzureRmLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
GatewayIpAddress "111.111.111.112" -AddressPrefix "192.168.2.0/24"
$lng3 = New-AzureRmLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
GatewayIpAddress "111.111.111.113" -AddressPrefix "192.168.3.0/24"
$lng4 = New-AzureRmLocalNetworkGateway -Name "Branch3" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
GatewayIpAddress "111.111.111.114" -AddressPrefix "192.168.4.0/24"
```

7. Create the route table and route rule.

```
New-AzureRmRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling" -Location "North Europe"
$rt = Get-AzureRmRouteTable -Name "MyRouteTable" -ResourceGroupName "ForcedTunneling"
Add-AzureRmRouteConfig -Name "DefaultRoute" -AddressPrefix "0.0.0.0/0" -NextHopType VirtualNetworkGateway -RouteTable $rt
Set-AzureRmRouteTable -RouteTable $rt
```

8. Associate the route table to the Midtier and Backend subnets.

```
$vnet = Get-AzureRmVirtualNetwork -Name "MultiTier-Vnet" -ResourceGroupName "ForcedTunneling"
Set-AzureRmVirtualNetworkSubnetConfig -Name "MidTier" -VirtualNetwork $vnet -AddressPrefix "10.1.1.0/24" -RouteTable $rt
Set-AzureRmVirtualNetworkSubnetConfig -Name "Backend" -VirtualNetwork $vnet -AddressPrefix "10.1.2.0/24" -RouteTable $rt
Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

9. Create the Gateway with a default site. This step takes some time to complete, sometimes 45 minutes or more, because you are creating and configuring the gateway.

The `-GatewayDefaultSite` is the cmdlet parameter that allows the forced routing configuration to work, so take care to configure this setting properly. This parameter is available in PowerShell 1.0 or later.

```
$pip = New-AzureRmPublicIpAddress -Name "GatewayIP" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
AllocationMethod Dynamic
$gwsubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$ipconfig = New-AzureRmVirtualNetworkGatewayIpConfig -Name "gwIpConfig" -SubnetId $gwsubnet.Id -PublicIpAddressId $pip.Id
New-AzureRmVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling" -Location "North Europe" -
IpConfigurations $ipconfig -GatewayType Vpn -VpnType RouteBased -GatewayDefaultSite $lng1 -EnableBgp $false
```

10. Establish the Site-to-Site VPN connections.

```
$gateway = Get-AzureRmVirtualNetworkGateway -Name "Gateway1" -ResourceGroupName "ForcedTunneling"
$lng1 = Get-AzureRmLocalNetworkGateway -Name "DefaultSiteHQ" -ResourceGroupName "ForcedTunneling"
$lng2 = Get-AzureRmLocalNetworkGateway -Name "Branch1" -ResourceGroupName "ForcedTunneling"
$lng3 = Get-AzureRmLocalNetworkGateway -Name "Branch2" -ResourceGroupName "ForcedTunneling"
$lng4 = Get-AzureRmLocalNetworkGateway -Name "Branch3" -ResourceGroupName "ForcedTunneling"

New-AzureRmVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng1 -ConnectionType IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection2" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng2 -ConnectionType IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection3" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng3 -ConnectionType IPsec -SharedKey "preSharedKey"
New-AzureRmVirtualNetworkGatewayConnection -Name "Connection4" -ResourceGroupName "ForcedTunneling" -Location "North
Europe" -VirtualNetworkGateway1 $gateway -LocalNetworkGateway2 $lng4 -ConnectionType IPsec -SharedKey "preSharedKey"

Get-AzureRmVirtualNetworkGatewayConnection -Name "Connection1" -ResourceGroupName "ForcedTunneling"
```

Configure forced tunneling using the classic deployment model

1/17/2017 • 5 min to read • [Edit on GitHub](#)

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.

This article will walk you through configuring forced tunneling for virtual networks created using the classic deployment model.

About Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

Deployment models and tools for forced tunneling

A forced tunneling connection can be configured for both the classic deployment model and the Resource Manager deployment model. See the following table for more information. We update this table as new articles, new deployment models, and additional tools become available for this configuration. When an article is available, we link directly to it from the table.

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

Requirements and considerations

Forced tunneling in Azure is configured via virtual network user defined routes (UDR). Redirecting traffic to an on-premises site is expressed as a Default Route to the Azure VPN gateway. The following section lists the current limitation of the routing table and routes for an Azure Virtual Network:

- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - **Local VNet routes:** Directly to the destination VMs in the same virtual network
 - **On premises routes:** To the Azure VPN gateway
 - **Default route:** Directly to the Internet. Packets destined to the private IP addresses not covered by the previous two routes will be dropped.
- With the release of user defined routes, you can create a routing table to add a default route, and then

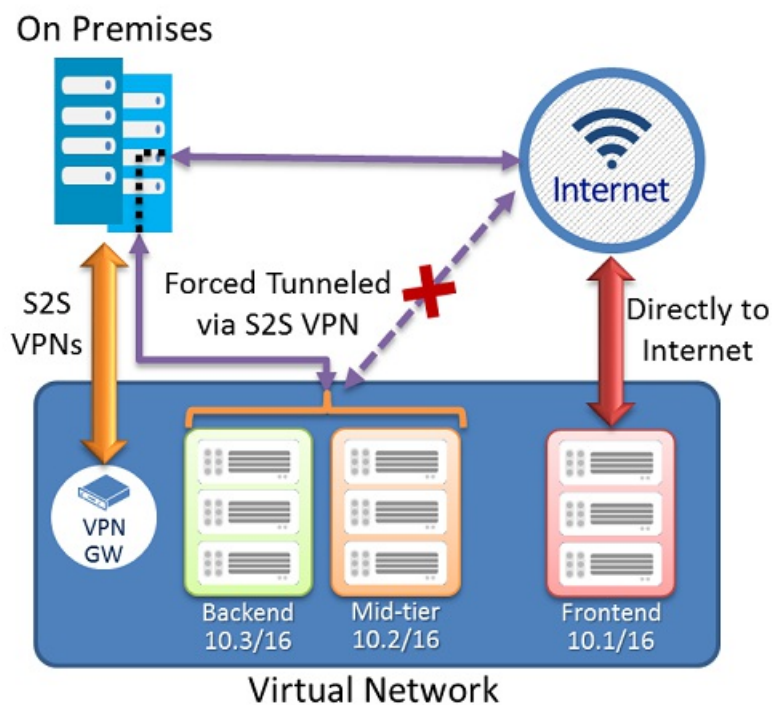
associate the routing table to your VNet subnet(s) to enable forced tunneling on those subnets.

- You need to set a "default site" among the cross-premises local sites connected to the virtual network.
- Forced tunneling must be associated with a VNet that has a dynamic routing VPN gateway (not a static gateway).
- ExpressRoute forced tunneling is not configured via this mechanism, but instead, is enabled by advertising a default route via the ExpressRoute BGP peering sessions. Please see the [ExpressRoute Documentation](#) for more information.

Configuration overview

In the following example, the Frontend subnet is not forced tunneled. The workloads in the Frontend subnet can continue to accept and respond to customer requests from the Internet directly. The Mid-tier and Backend subnets are forced tunneled. Any outbound connections from these two subnets to the Internet will be forced or redirected back to an on-premises site via one of the S2S VPN tunnels.

This allows you to restrict and inspect Internet access from your virtual machines or cloud services in Azure, while continuing to enable your multi-tier service architecture required. You also can apply forced tunneling to the entire virtual networks if there are no Internet-facing workloads in your virtual networks.



Before you begin

Verify that you have the following items before beginning configuration.

- An Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](#) or sign up for a [free account](#).
- A configured virtual network.
- The latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.

Configure forced tunneling

The following procedure will help you specify forced tunneling for a virtual network. The configuration steps correspond to the VNet network configuration file.

```

<VirtualNetworkSite name="MultiTier-VNet" Location="North Europe">
  <AddressSpace>
    <AddressPrefix>10.1.0.0/16</AddressPrefix>
  </AddressSpace>
  <Subnets>
    <Subnet name="Frontend">
      <AddressPrefix>10.1.0.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Midtier">
      <AddressPrefix>10.1.1.0/24</AddressPrefix>
    </Subnet>
    <Subnet name="Backend">
      <AddressPrefix>10.1.2.0/23</AddressPrefix>
    </Subnet>
    <Subnet name="GatewaySubnet">
      <AddressPrefix>10.1.200.0/28</AddressPrefix>
    </Subnet>
  </Subnets>
  <Gateway>
    <ConnectionsToLocalNetwork>
      <LocalNetworkSiteRef name="DefaultSiteHQ">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch1">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch2">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
      <LocalNetworkSiteRef name="Branch3">
        <Connection type="IPsec" />
      </LocalNetworkSiteRef>
    </ConnectionsToLocalNetwork>
  </Gateway>
</VirtualNetworkSite>
</VirtualNetworkSite>

```

In this example, the virtual network "MultiTier-VNet" has three subnets: *Frontend*, *Midtier*, and *Backend* subnets, with four cross premises connections: *DefaultSiteHQ*, and three *Branches*.

The steps will set the *DefaultSiteHQ* as the default site connection for forced tunneling, and configure the *Midtier* and *Backend* subnets to use forced tunneling.

1. Create a routing table. Use the following cmdlet to create your route table.

```
New-AzureRouteTable -Name "MyRouteTable" -Label "Routing Table for Forced Tunneling" -Location "North Europe"
```

2. Add a default route to the routing table.

The following example adds a default route to the routing table created in Step 1. Note that the only route supported is the destination prefix of "0.0.0.0/0" to the "VPNGateway" NextHop.

```
Get-AzureRouteTable -Name "MyRouteTable" | Set-AzureRoute -RouteTable "MyRouteTable" -RouteName "DefaultRoute" -
AddressPrefix "0.0.0.0/0" -NextHopType VPNGateway
```

3. Associate the routing table to the subnets.

After a routing table is created and a route added, use the following example to add or associate the route table to a VNet subnet. The example adds the route table "MyRouteTable" to the *Midtier* and *Backend* subnets of VNet *MultiTier-VNet*.

```
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Midtier" -RouteTableName "MyRouteTable"
```

```
Set-AzureSubnetRouteTable -VirtualNetworkName "MultiTier-VNet" -SubnetName "Backend" -RouteTableName "MyRouteTable"
```

4. Assign a default site for forced tunneling.

In the preceding step, the sample cmdlet scripts created the routing table and associated the route table to two of the VNet subnets. The remaining step is to select a local site among the multi-site connections of the virtual network as the default site or tunnel.

```
$DefaultSite = @("DefaultSiteHQ")
```

```
Set-AzureVNetGatewayDefaultSite -VNetName "MultiTier-VNet" -DefaultSite "DefaultSiteHQ"
```

Additional PowerShell cmdlets

To delete a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To list a route table

```
Get-AzureRouteTable [-Name <routeTableName> [-DetailLevel <detailLevel>]]
```

To delete a route from a route table

```
Remove-AzureRouteTable -Name <routeTableName>
```

To remove a route from a subnet

```
Remove-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To list the route table associated with a subnet

```
Get-AzureSubnetRouteTable -VirtualNetworkName <virtualNetworkName> -SubnetName <subnetName>
```

To remove a default site from a VNet VPN gateway

```
Remove-AzureVnetGatewayDefaultSite -VNetName <virtualNetworkName>
```

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

1/17/2017 • 4 min to read • [Edit on GitHub](#)

This article walks you through using the Azure portal 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.

You can use this article to 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 VNets created using the Resource Manager deployment model that have a RouteBased VPN gateway. These steps do not apply to ExpressRoute/Site-to-Site coexisting connection configurations. See [ExpressRoute/S2S coexisting connections](#) for information about coexisting connections.

Deployment models and methods

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

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	CLASSIC PORTAL	POWERSHELL
Resource Manager	Article	Not Supported	Supported
Classic	Not Supported	Not Supported	Article

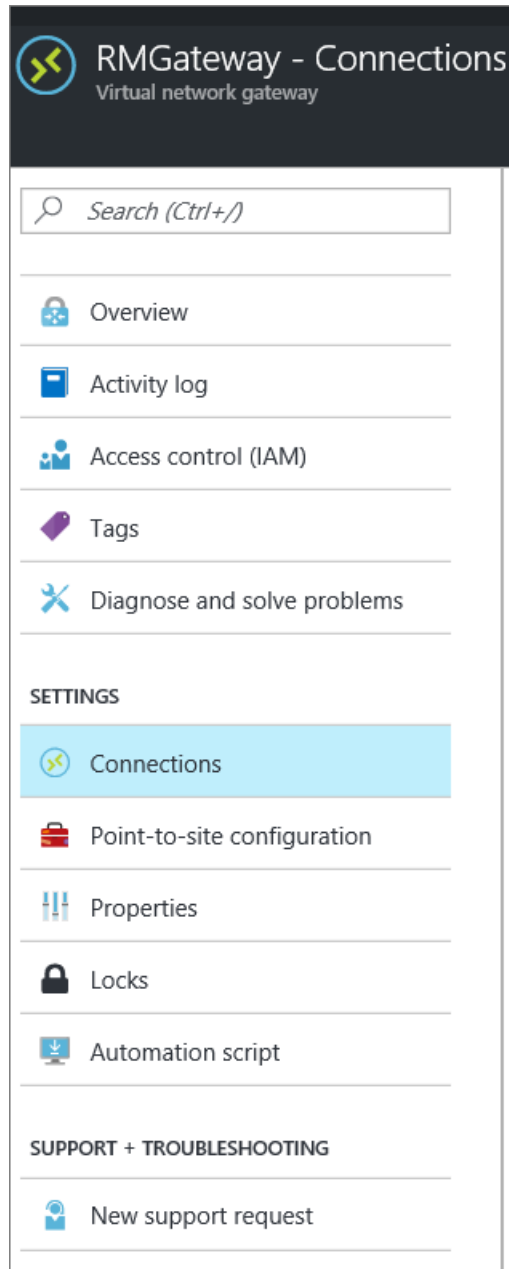
Before you begin

Verify the following items:

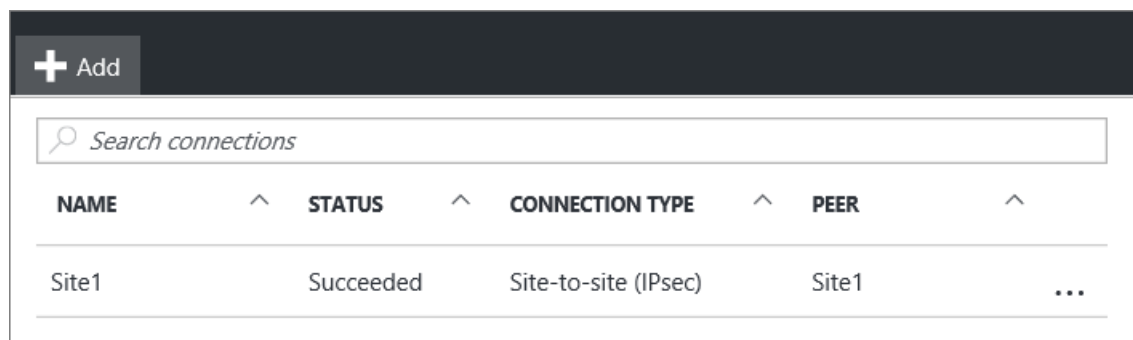
- You are not creating an ExpressRoute/S2S coexisting connection.
- You have a virtual network that was created using the Resource Manager deployment model with an existing connection.
- The virtual network gateway for your VNet is RouteBased. If you have a PolicyBased VPN gateway, you must delete the virtual network gateway and create a new VPN gateway as RoutBased.
- 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

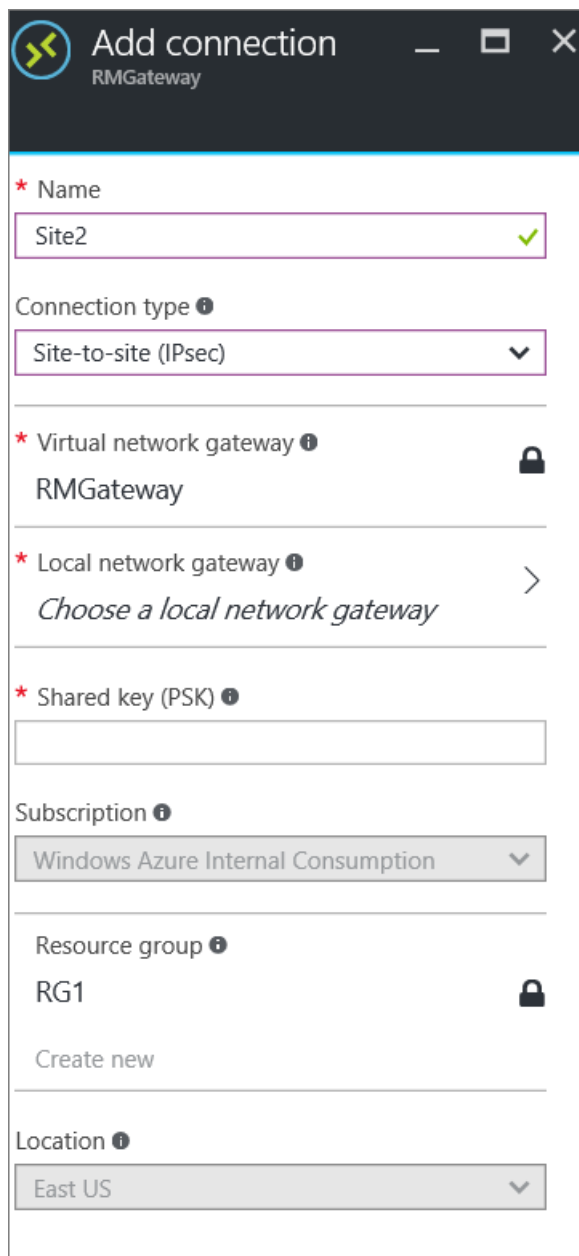
1. From a browser, navigate to the [Azure portal](#) and, if necessary, sign in with your Azure account.
2. Click **All resources** and locate your **virtual network gateway** from the list of resources and click it.
3. On the **Virtual network gateway** blade, click **Connections**.



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



5. On the **Add connection** blade, 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
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** blade.

- Click **Create new** to open the **Create local network gateway** blade.

- On the **Create local network gateway** blade, 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.
- Click **OK** on the **Create local network gateway** blade to save the changes.

Part 3 - Add the shared key and create the connection

- On the **Add connection** blade, 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.

2. At the bottom of the blade, click **OK** to create the connection.

Part 4 - Verify the VPN connection

You can verify your VPN connection either in the portal, or by using PowerShell.

To verify your connection by using PowerShell

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.

```



Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-
f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1"
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite"
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}

```

To verify your connection by using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	 2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	 RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

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.

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

1/17/2017 • 7 min to read • [Edit on GitHub](#)

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.

This article applies 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. See [ExpressRoute/S2S coexisting connections](#) for information about coexisting connections.

Deployment models and methods

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

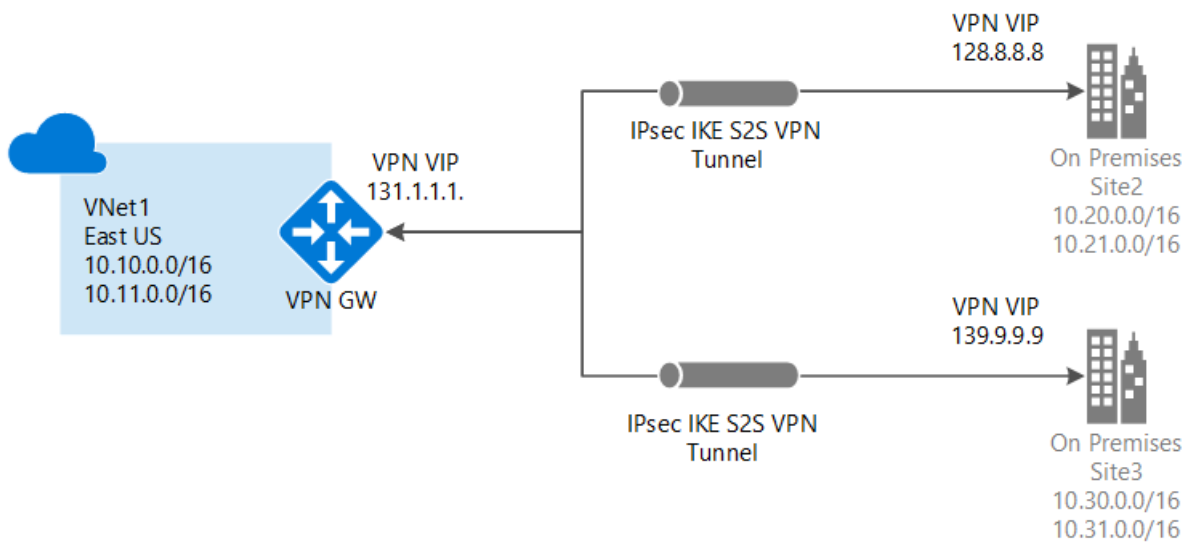
We update this table as new articles and additional tools become available for this configuration. When an article is available, we link directly to it from this table.

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

About connecting

You can connect multiple on-premises sites to a single virtual network. This is especially attractive for building hybrid cloud solutions. Creating a multi-site connection to your Azure virtual network gateway is very 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 Azure Classic Portal to make changes to this virtual network. For this release, you'll need to make changes to the network configuration file instead of using the Azure Classic Portal. If you make changes in the Azure Classic Portal, they'll overwrite your multi-site reference settings for this virtual network.

You should feel pretty 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 Azure Classic 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:

- 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](#).
- Compatible VPN hardware for each on-premises location. Check [About VPN Devices for Virtual Network Connectivity](#) to verify if the device that you want to use is something that is known to be compatible.
- An externally facing public IPv4 IP address for each VPN device. The IP address cannot be located behind a NAT. This is requirement.
- You'll need to install the latest version of the Azure PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets.
- Someone who is proficient at configuring your VPN hardware. You won't be able to use the auto-generated VPN scripts from the Azure Classic Portal to configure your VPN devices. This means 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 Azure Classic Portal or the REST API will reject the configuration being uploaded.

For example, if you have two local network sites that both contain the IP address range 10.2.3.0/24 and you have a package with a destination address 10.2.3.3, Azure wouldn't know which site you want to send the package to because the address ranges are overlapping. To prevent routing issues, Azure doesn't allow you to upload a configuration file that has overlapping ranges.

1. Create a Site-to-Site VPN

If you already have a Site-to-Site VPN with a dynamic routing gateway, great! You can proceed to [Export the virtual network configuration settings](#). If not, do the following:

If you already have a Site-to-Site virtual network, but it has a static (policy-based) routing gateway:

1. Change your gateway type to dynamic routing. A multi-site VPN requires a dynamic (also known as route-based) routing gateway. To change your gateway type, you'll need to first delete the existing gateway, then create a new one. For instructions, see [How to change the VPN routing type for your gateway](#).
2. Configure your new gateway and create your VPN tunnel. For instructions, see [Configure a VPN Gateway in the Azure Classic Portal](#). First, change your gateway type to dynamic routing.

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 in the Azure Classic Portal](#).
2. Configure a dynamic routing gateway using these instructions: [Configure a VPN Gateway](#). Be sure to select **dynamic routing** for your gateway type.

2. Export the network configuration file

Export your network configuration file. The file that you export will be used to configure your new multi-site settings. If you need instructions on how to export a file, see the section in the article: [How to create a VNet using a network configuration file in the Azure Portal](#).

3. Open the network configuration file

Open the network configuration file that you downloaded in the last step. Use any xml editor that you like. The file should look similar to the following:

```

<NetworkConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration">
  <VirtualNetworkConfiguration>
    <LocalNetworkSites>
      <LocalNetworkSite name="Site1">
        <AddressSpace>
          <AddressPrefix>10.0.0.0/16</AddressPrefix>
          <AddressPrefix>10.1.0.0/16</AddressPrefix>
        </AddressSpace>
        <VPNGatewayAddress>131.2.3.4</VPNGatewayAddress>
      </LocalNetworkSite>
      <LocalNetworkSite name="Site2">
        <AddressSpace>
          <AddressPrefix>10.2.0.0/16</AddressPrefix>
          <AddressPrefix>10.3.0.0/16</AddressPrefix>
        </AddressSpace>
        <VPNGatewayAddress>131.4.5.6</VPNGatewayAddress>
      </LocalNetworkSite>
    </LocalNetworkSites>
    <VirtualNetworkSites>
      <VirtualNetworkSite name="VNet1" AffinityGroup="USWest">
        <AddressSpace>
          <AddressPrefix>10.20.0.0/16</AddressPrefix>
          <AddressPrefix>10.21.0.0/16</AddressPrefix>
        </AddressSpace>
        <Subnets>
          <Subnet name="FE">
            <AddressPrefix>10.20.0.0/24</AddressPrefix>
          </Subnet>
          <Subnet name="BE">
            <AddressPrefix>10.20.1.0/24</AddressPrefix>
          </Subnet>
          <Subnet name="GatewaySubnet">
            <AddressPrefix>10.20.2.0/29</AddressPrefix>
          </Subnet>
        </Subnets>
        <Gateway>
          <ConnectionsToLocalNetwork>
            <LocalNetworkSiteRef name="Site1">
              <Connection type="IPsec" />
            </LocalNetworkSiteRef>
          </ConnectionsToLocalNetwork>
        </Gateway>
      </VirtualNetworkSite>
    </VirtualNetworkSites>
  </VirtualNetworkConfiguration>
</NetworkConfiguration>

```

4. Add multiple site references

When you add or remove site reference information, you'll make configuration changes to the `ConnectionsToLocalNetwork/LocalNetworkSiteRef`. Adding a new local site reference triggers Azure to create a new tunnel. In the example below, the network configuration is for a single-site connection. Save the file once you have finished making your changes.


```

<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="Site1"><Connection type="IPsec" /></LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>

```

To add additional site references (create a multi-site configuration), simply add additional "LocalNetworkSiteRef" lines, as shown in the example below:

```

<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="Site1"><Connection type="IPsec" /></LocalNetworkSiteRef>
    <LocalNetworkSiteRef name="Site2"><Connection type="IPsec" /></LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>

```

5. Import the network configuration file

Import the network configuration file. When you import this file with the changes, the new tunnels will be added. The tunnels will use the dynamic gateway that you created earlier. If you need instructions on how to import the file, see the section in the article: [How to create a VNet using a network configuration file in the Azure Portal](#).

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

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 fromNot 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 fromNot 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](#).

How to configure BGP on Azure VPN Gateways using Azure Resource Manager and PowerShell

1/17/2017 • 10 min to read • [Edit on GitHub](#)

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 Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

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.

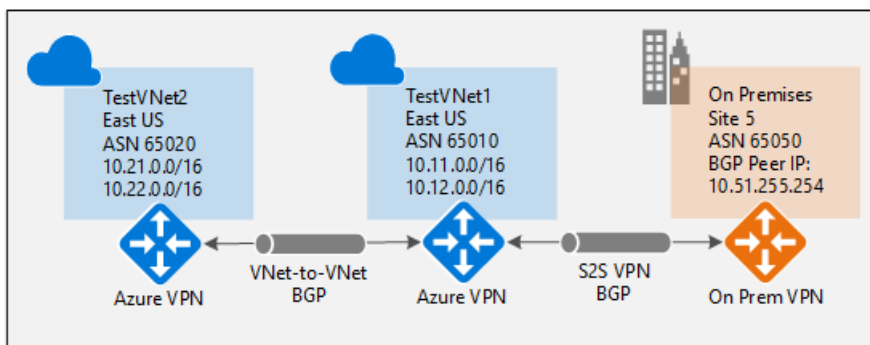
Please see [Overview of BGP with Azure VPN Gateways](#) for more discussion on benefits of BGP and to understand the technical requirements and considerations of using BGP.

Getting started with BGP on Azure VPN gateways

This article will walk 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](#)

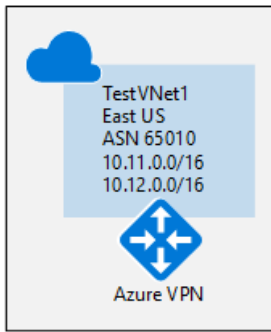
Each part of the instructions forms a basic building block for enabling BGP in your network connectivity. If you complete all three parts, you will build the topology as shown in the following diagram:



You can combine these together to build a more complex, multi-hop, transit network that meet your needs.

Part 1 - Configure BGP on the Azure VPN Gateway

The following configuration steps will setup 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](#).
- You'll need to install the Azure Resource Manager PowerShell cmdlets. See [How to install and configure 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      ="Replace_With_Your_Subscription_Name"
$RG1       ="TestBGPRG1"
$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

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:

```
Login-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $Sub1
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

3. Create TestVNet1

The sample below creates a virtual network named TestVNet1 and three subnets, one called GatewaySubnet, one called FrontEnd, and one called Backend. When substituting values, it's important that you always name your

gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation will fail.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix $VNetPrefix1,$VNetPrefix12 -
Subnet $fesub1,$besub1,$gwsb1
```

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 subnet and IP configurations required.

```
$gwpip1 = New-AzureRmPublicIpAddress -Name $GWIPName1 -ResourceGroupName $RG1 -Location $Location1 -AllocationMethod Dynamic

$vn1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vn1
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName1 -Subnet $subnet1 -PublicIpAddress $gwpip1
```

2. Create the VPN gateway with the AS number

Create the virtual network gateway for TestVNet1. Note that BGP requires a Route-Based VPN gateway, and also the addition parameter, -Asn, to set the ASN (AS Number) for TestVNet1. Creating a gateway can take a while (30 minutes or more to complete).

```
New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -IpConfigurations $gwipconf1 -
GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance -Asn $VNet1ASN
```

3. Obtain the Azure 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.

```
$vn1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vn1gw.BgpSettingsText
```

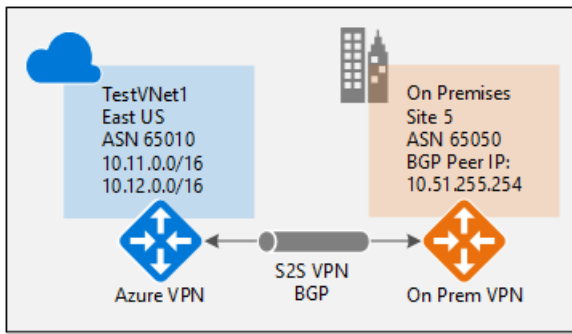
The last command will show the corresponding BGP configurations on the Azure VPN Gateway; for example:

```
$vn1gw.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 will 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 Azure VPN gateway with the local network gateway. The difference between the instructions in this article is the additional properties required to specify the BGP configuration parameters.



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      = "TestBGPRG5"
$Location5 = "East US 2"
$LNGName5  = "Site5"
$LNGPrefix50 = "10.52.255.254/32"
$LNGIP5    = "Your_VPN_Device_IP"
$LNGASN5    = 65050
$BGPPeerIP5 = "10.52.255.254"
```

A couple of things to note regarding the local network gateway parameters:

- The local network gateway can be in the same or different location and resource group as the VPN gateway. This example shows them in different resource groups in different locations.
- The minimum prefix you need to declare for the local network gateway is the host address of your BGP Peer IP address on your VPN device. In this case, it's a /32 prefix of "10.52.255.254/32".
- As a reminder, you must use different BGP ASNs between your on-premises networks and Azure VNet. If they are the same, you need to change your VNet ASN if your on-premises VPN device already use the ASN to peer with other BGP neighbors.

Before you continue, please make sure you are still connected to Subscription 1.

2. Create the local network gateway for Site5

Be sure to create the resource group if it is not created, before you create the local network gateway. Notice the two additional parameters for the local network gateway: Asn and BgpPeerAddress.

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5

New-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress $LNGIP5 -
AddressPrefix $LNGPrefix50 -Asn $LNGASN5 -BgpPeeringAddress $BGPPeerIP5
```

Step 2 - Connect the VNet gateway and local network gateway

1. Get the two gateways

```
$vnetlgw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw = Get-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG5
```

2. Create the TestVNet1 to Site5 connection

In this step, you will create the connection from TestVNet1 to Site5. You must specify "-EnableBGP \$True" to enable BGP for this connection. As discussed earlier, it is possible to have both BGP and non-BGP connections for the same Azure VPN Gateway. Unless BGP is enabled in the connection property, Azure will not enable BGP for this connection even though BGP parameters are already configured on both gateways.

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -EnableBGP $True
```

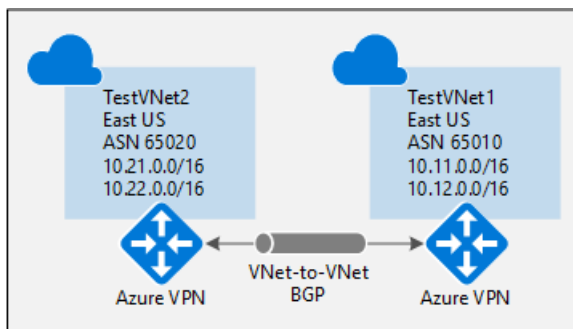
The example below lists the parameters you will enter into the BGP configuration section on your on-premises VPN device for this exercise:

```
- Site5 ASN      : 65050
- Site5 BGP IP   : 10.52.255.254
- 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, and the BGP peering session will start 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 diagram below.



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 setup 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      = "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"
$GWIPIName2  = "VNet2GWIP"
$GWIPIConfName2 = "gwipconf2"
$Connection21 = "VNet2toVNet1"
$Connection12 = "VNet1toVNet2"

```

2. Create TestVNet2 in the new resource group

```

New-AzureRmResourceGroup -Name $RG2 -Location $Location2

$Fesub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$Besub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$Gwsb2 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix $VNetPrefix21,$VNetPrefix22 -
Subnet $Fesub2,$Besub2,$Gwsb2

```

3. Create the VPN gateway for TestVNet2 with BGP parameters

Request a public IP address to be allocated to the gateway you will create for your VNet. You'll also define the subnet and IP configurations required.

```

$gwip2 = New-AzureRmPublicIpAddress -Name $GWIPIName2 -ResourceGroupName $RG2 -Location $Location2 -AllocationMethod Dynamic

$vn2 = Get-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$sb2 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vn2
$gwipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPIConfName2 -Subnet $sb2 -PublicIpAddress $gwip2

```

Create the VPN gateway with the AS number. Note that you must override the default ASN on your Azure VPN gateways. The ASNs for the connected VNets must be different to enable BGP and transit routing.

```

New-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -IpConfigurations $gwipconf2 -
GatewayType Vpn -VpnType RouteBased -GatewaySku Standard -Asn $VNet2ASN

```

Step 2 - Connect the TestVNet1 and TestVNet2 gateways

In this example, both gateways are in the same subscription. You can complete this step in the same PowerShell session.

1. Get both gateways

Make sure you login and connect to Subscription 1.

```

$vn1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vn2gw = Get-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2

```

2. Create both connections

In this step, you will create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1.


```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True
```

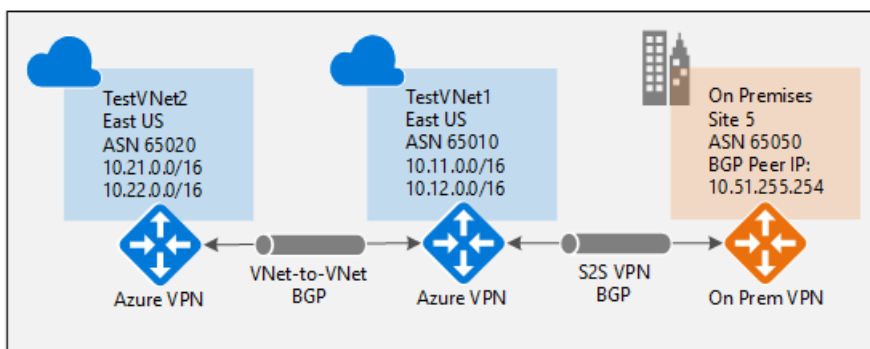
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -VirtualNetworkGateway1 $vnet2gw -VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True
```

IMPORTANT

Be sure to enable BGP for BOTH connections.

After completing these steps, the connection will be established in a few minutes, and the BGP peering session will be up once the VNet-to-VNet connection is completed.

If you have completed all three parts of this exercise, you will have established a network topology as shown below:



Next steps

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 using Azure Resource Manager and PowerShell

1/17/2017 • 13 min to read • [Edit on GitHub](#)

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 Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

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. Please 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](#)
- [Part 4 - Update existing gateway between active-active and active-standby](#)

You can combine these together to build a more complex, highly available network topology that meets your needs.

IMPORTANT

Please note that the active-active mode only works in HighPerformance SKU

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 HighPerformance

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 [How to install and configure 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:

```
Login-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $Sub1
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

3. Create TestVNet1

The sample below creates a virtual network named TestVNet1 and three subnets, one called GatewaySubnet, one called FrontEnd, and one called Backend. When substituting values, it's important that you always name your gateway subnet specifically GatewaySubnet. If you name it something else, your gateway creation will fail.

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1 -AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1 -AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1 -AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix $VNetPrefix11,$VNetPrefix12 -
Subnet $fesub1,$besub1,$gwsb1
```

Step 2 - Create the VPN gateway for TestVNet1 with active-active mode

1. Create the public IP addresses and gateway IP configurations

Request two public IP addresses to be allocated to the gateway you will create for your VNet. You'll also define the subnet and IP configurations required.

```
$gw1pip1 = New-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1 -Location $Location1 -AllocationMethod Dynamic
$gw1pip2 = New-AzureRmPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1 -Location $Location1 -AllocationMethod Dynamic

$vn1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vn1
$gw1ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf1 -Subnet $subnet1 -PublicIpAddress $gw1pip1
$gw1ipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW1IPconf2 -Subnet $subnet1 -PublicIpAddress $gw1pip2
```

2. Create the VPN gateway with active-active configuration

Create the virtual network gateway for TestVNet1. Note that there are two GatewayIpConfig entries, and the EnableActiveActiveFeature flag is set. Active-active mode requires a Route-Based VPN gateway of HighPerformance SKU. Creating a gateway can take a while (30 minutes or more to complete).

```
New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1 -Location $Location1 -IpConfigurations
$gw1ipconf1,$gw1ipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance -Asn $VNet1ASN -
EnableActiveActiveFeature -Debug
```

3. Obtain the gateway public IP addresses and the BGP Peer IP address

Once the gateway is created, you will need to obtain the BGP Peer IP address on the Azure VPN Gateway. This address is needed to configure the Azure VPN Gateway as a BGP Peer for your on-premises VPN devices.

```
$gw1pip1 = Get-AzureRmPublicIpAddress -Name $GW1IPName1 -ResourceGroupName $RG1
$gw1pip2 = Get-AzureRmPublicIpAddress -Name $GW1IPName2 -ResourceGroupName $RG1
$vn1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
```

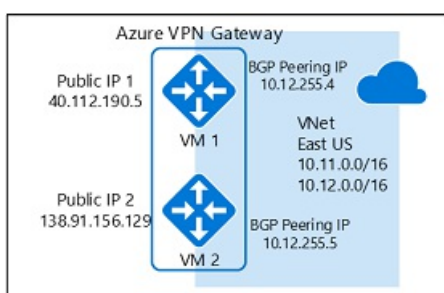
Use the following cmdlets to show the two public IP addresses allocated for your VPN gateway, and their corresponding BGP Peer IP addresses for each gateway instance:

```
PS D:\> $gw1pip1.IpAddress
40.112.190.5

PS D:\> $gw1pip2.IpAddress
138.91.156.129

PS D:\> $vn1gw.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 will 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"
$LNName51 = "Site5_1"
$LNPrefix51 = "10.52.255.253/32"
$LNIP51    = "131.107.72.22"
$LNASN5    = 65050
$BGPPeerIP51 = "10.52.255.253"
```

A couple of things to note regarding the local network gateway parameters:

- The local network gateway can be in the same or different location and resource group as the VPN gateway. This example shows them in different resource groups but in the same Azure location.
- If there is only one on-premises VPN device as shown above, the active-active connection can work with or without BGP protocol. This example uses BGP for the cross-premises connection.
- If BGP is enabled, the prefix you need to declare for the local network gateway is the host address of your BGP Peer IP address on your VPN device. In this case, it's a /32 prefix of "10.52.255.253/32".
- As a reminder, you must use different BGP ASNs between your on-premises networks and Azure VNet. If they are the same, you need to change your VNet ASN if your on-premises VPN device already uses the ASN to peer with other BGP neighbors.

2. Create the local network gateway for Site5

Before you continue, please make sure you are still connected to Subscription 1. Create the resource group if it is not yet created.

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5
New-AzureRmLocalNetworkGateway -Name $LNName51 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress $LNIP51 -AddressPrefix $LNPrefix51 -Asn $LNASN5 -BgpPeeringAddress $BGPPeerIP51
```

Step 2 - Connect the VNet gateway and local network gateway

1. Get the two gateways

```
$vnetlgw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$lng5gw1 = Get-AzureRmLocalNetworkGateway -Name $LNName51 -ResourceGroupName $RG5
```

2. Create the TestVNet1 to Site5 connection

In this step, you will create the connection from TestVNet1 to Site5_1 with "EnableBGP" set to \$True.

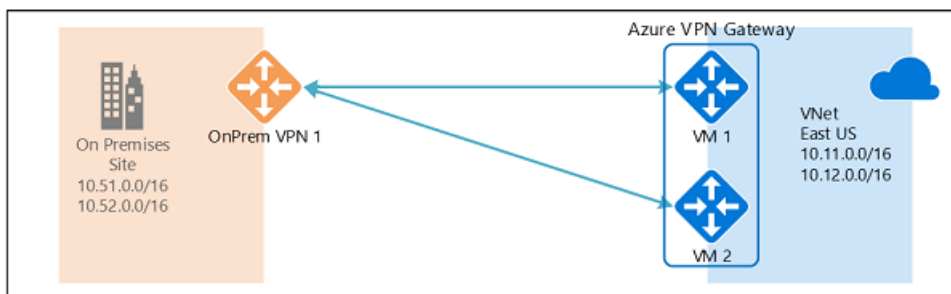
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection151 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw1 -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -EnableBGP True
```

3. VPN and BGP parameters for your on-premises VPN device

The example below lists the parameters you will enter into the BGP configuration section on your on-premises VPN device for this exercise:

```
- Site5 ASN      : 65050
- Site5 BGP IP   : 10.52.255.253
- Prefixes to announce : (for example) 10.51.0.0/16 and 10.52.0.0/16
- Azure VNet ASN   : 65010
- Azure VNet BGP IP 1 : 10.12.255.4 for tunnel to 40.112.190.5
- Azure VNet BGP IP 2 : 10.12.255.5 for tunnel to 138.91.156.129
- Static routes    : Destination 10.12.255.4/32, nexthop the VPN tunnel interface to 40.112.190.5
                    : Destination 10.12.255.5/32, nexthop the VPN tunnel interface to 138.91.156.129
- eBGP Multihop    : Ensure the "multihop" option for eBGP is enabled on your device if needed
```

The connection should be established after a few minutes, and the BGP peering session will start once the IPsec connection is established. This example so far has configured only one on-premises VPN device, resulting in the diagram shown below:



Step 3 - Connect two on-premises VPN devices to the active-active VPN gateway

If you have two VPN devices at the same on-premises network, you can achieve dual redundancy by connecting the Azure VPN gateway to the second VPN device.

1. Create the second local network gateway for Site5

Note that the gateway IP address, address prefix, and BGP peering address for the second local network gateway must not overlap with the previous local network gateway for the same on-premises network.

```
$LNGName52 = "Site5_2"
$LNGPrefix52 = "10.52.255.254/32"
$LNGIP52 = "131.107.72.23"
$BGPPeerIP52 = "10.52.255.254"
```

```
New-AzureRmLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5 -Location $Location5 -GatewayIpAddress $LNGIP52 -AddressPrefix $LNGPrefix52 -Asn $LNGASN5 -BgpPeeringAddress $BGPPeerIP52
```

2. Connect the VNet gateway and the second local network gateway

Create the connection from TestVNet1 to Site5_2 with "EnableBGP" set to \$True

```
$lng5gw2 = Get-AzureRmLocalNetworkGateway -Name $LNGName52 -ResourceGroupName $RG5
```

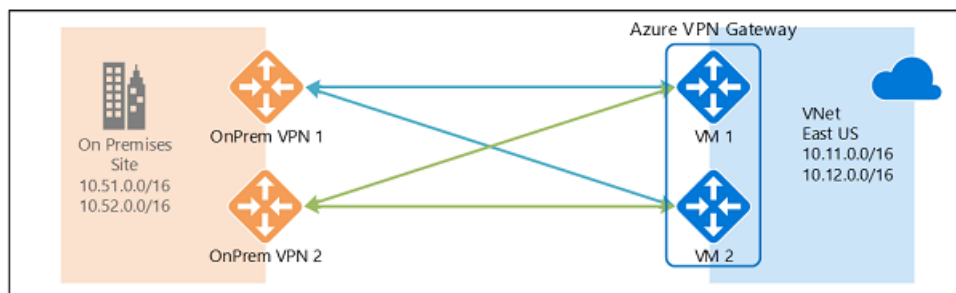
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection152 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2 $lng5gw2 -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3' -EnableBGP True
```

3. VPN and BGP parameters for your second on-premises VPN device

Similarly, below lists the parameters you will enter into the second VPN device:

- Site5 ASN : 65050
- Site5 BGP IP : 10.52.255.254
- Prefixes to announce : (for example) 10.51.0.0/16 and 10.52.0.0/16
- Azure VNet ASN : 65010
- Azure VNet BGP IP 1 : 10.12.255.4 for tunnel to 40.112.190.5
- Azure VNet BGP IP 2 : 10.12.255.5 for tunnel to 138.91.156.129
- Static routes : Destination 10.12.255.4/32, nexthop the VPN tunnel interface to 40.112.190.5
Destination 10.12.255.5/32, nexthop the VPN tunnel interface to 138.91.156.129
- eBGP Multihop : Ensure the "multihop" option for eBGP is enabled on your device if needed

Once the connection (tunnels) are established, you will have dual redundant VPN devices and tunnels connecting your on-premises network and Azure:



Part 3 - Establish an active-active VNet-to-VNet connection

This section creates an active-active VNet-to-VNet connection with BGP.

The instructions below continue from the previous steps listed above. You must complete [Part 1](#) to create and configure TestVNet1 and the VPN Gateway with BGP.

Step 1 - Create TestVNet2 and the VPN gateway

It is important to make sure that the IP address space of the new virtual network, TestVNet2, does not overlap with any of your VNet ranges.

In this example, the virtual networks belong to the same subscription. You can set up VNet-to-VNet connections between different subscriptions; please refer to [Configure a VNet-to-VNet connection](#) to learn more details. Make sure you add the "-EnableBgp \$True" when creating the connections to enable BGP.

1. Declare your variables

Be sure to replace the values with the ones that you want to use for your configuration.

```
$RG2      = "TestAARG2"
$Location2 = "East US"
$VNetName2 = "TestVNet2"
$FESubName2 = "FrontEnd"
$BESubName2 = "Backend"
$GWSubName2 = "GatewaySubnet"
$VNetPrefix21 = "10.21.0.0/16"
$VNetPrefix22 = "10.22.0.0/16"
$FESubPrefix2 = "10.21.0.0/24"
$BESubPrefix2 = "10.22.0.0/24"
$GWSubPrefix2 = "10.22.255.0/27"
$VNet2ASN    = 65020
$DNS2       = "8.8.8.8"
$GWName2    = "VNet2GW"
$GW2IPName1 = "VNet2GWIP1"
$GW2IPConf1 = "gw2ipconf1"
$GW2IPName2 = "VNet2GWIP2"
$GW2IPConf2 = "gw2ipconf2"
$Connection21 = "VNet2toVNet1"
$Connection12 = "VNet1toVNet2"
```

2. Create TestVNet2 in the new resource group

```
New-AzureRmResourceGroup -Name $RG2 -Location $Location2

$Fesub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName2 -AddressPrefix $FESubPrefix2
$Besub2 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName2 -AddressPrefix $BESubPrefix2
$Gwsb2 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName2 -AddressPrefix $GWSubPrefix2

New-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2 -Location $Location2 -AddressPrefix $VNetPrefix21,$VNetPrefix22 -
Subnet $fesub2,$besub2,$gwsb2
```

3. Create the active-active VPN gateway for TestVNet2

Request two public IP addresses to be allocated to the gateway you will create for your VNet. You'll also define the subnet and IP configurations required.

```
$gw2pip1 = New-AzureRmPublicIpAddress -Name $GW2IPName1 -ResourceGroupName $RG2 -Location $Location2 -AllocationMethod Dynamic
$gw2pip2 = New-AzureRmPublicIpAddress -Name $GW2IPName2 -ResourceGroupName $RG2 -Location $Location2 -AllocationMethod Dynamic

$vn2 = Get-AzureRmVirtualNetwork -Name $VNetName2 -ResourceGroupName $RG2
$subnet2 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vn2
$gw2ipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW2IPconf1 -Subnet $subnet2 -PublicIpAddress $gw2pip1
$gw2ipconf2 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GW2IPconf2 -Subnet $subnet2 -PublicIpAddress $gw2pip2
```

Create the VPN gateway with the AS number and the "EnableActiveActiveFeature" flag. Note that you must override the default ASN on your Azure VPN gateways. The ASNs for the connected VNets must be different to enable BGP and transit routing.

```
New-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2 -Location $Location2 -IpConfigurations
$gw2ipconf1,$gw2ipconf2 -GatewayType Vpn -VpnType RouteBased -GatewaySku HighPerformance -Asn $VNet2ASN -
EnableActiveActiveFeature
```

Step 2 - Connect the TestVNet1 and TestVNet2 gateways

In this example, both gateways are in the same subscription. You can complete this step in the same PowerShell session.

1. Get both gateways

Make sure you log in and connect to Subscription 1.

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
$vnet2gw = Get-AzureRmVirtualNetworkGateway -Name $GWName2 -ResourceGroupName $RG2
```

2. Create both connections

In this step, you will create the connection from TestVNet1 to TestVNet2, and the connection from TestVNet2 to TestVNet1.

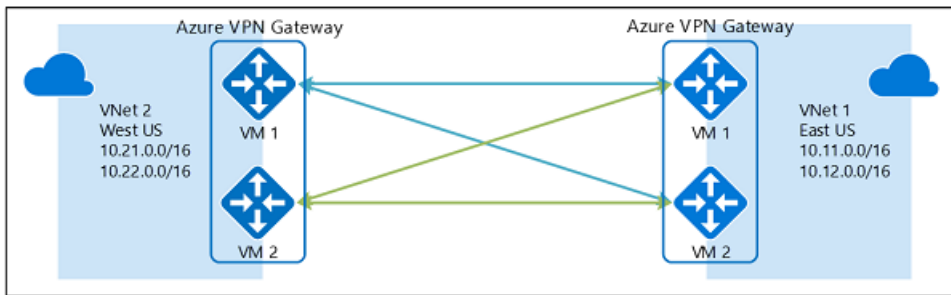
```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection12 -ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -
VirtualNetworkGateway2 $vnet2gw -Location $Location1 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True

New-AzureRmVirtualNetworkGatewayConnection -Name $Connection21 -ResourceGroupName $RG2 -VirtualNetworkGateway1 $vnet2gw -
VirtualNetworkGateway2 $vnet1gw -Location $Location2 -ConnectionType Vnet2Vnet -SharedKey 'AzureA1b2C3' -EnableBgp $True
```

IMPORTANT

Be sure to enable BGP for BOTH connections.

After completing these steps, the connection will be establish in a few minutes, and the BGP peering session will be up once the VNet-to-VNet connection is completed with dual redundancy:



Part 4 - Update existing gateway between active-active and active-standby

The last section will describe how you can configure an existing Azure VPN gateway from active-standby to active-active mode, or vice versa.

IMPORTANT

Please note that the active-active mode only works in HighPerformance SKU

Configure an active-standby gateway to active-active gateway

1. Gateway parameters

The following example converts an active-standby gateway into an active-active gateway. You need to create another public IP address, then add a second Gateway IP configuration. Below shows the parameters used:

```
$GWName = "TestVNetAA1GW"
$VNetName = "TestVNetAA1"
$RG = "TestVPNActiveActive01"
$GWIPName2 = "gwpip2"
$GWIPconf2 = "gw1ipconf2"

$Vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG
$Subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $Vnet
$Gw = Get-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
$Location = $Gw.Location
```

2. Create the public IP address, then add the second gateway IP configuration

```
$gwpip2 = New-AzureRmPublicIpAddress -Name $GWIPName2 -ResourceGroupName $RG -Location $Location -AllocationMethod Dynamic
Add-AzureRmVirtualNetworkGatewayIpConfig -VirtualNetworkGateway $Gw -Name $GWIPconf2 -Subnet $Subnet -PublicIpAddress $gwpip2
```

3. Enable active-active mode and update the gateway

You must set the gateway object in PowerShell to trigger the actual update. The SKU of the gateway object must also be changed to HighPerformance since it was created previously as Standard.

```
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $Gw -EnableActiveActiveFeature -GatewaySku HighPerformance
```

This update can take 30 to 45 minutes.

Configure an active-active gateway to active-standby gateway

1. Gateway parameters

Use the same parameters as above, get the name of the IP configuration you want to remove.

```
$GWName = "TestVNetAA1GW"
$RG      = "TestVPNActiveActive01"

$gw      = Get-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
$ipconfname = $gw.IpConfigurations[1].Name
```

2. Remove the gateway IP configuration and disable the active-active mode

Similarly, you must set the gateway object in PowerShell to trigger the actual update.

```
Remove-AzureRmVirtualNetworkGatewayIpConfig -Name $ipconfname -VirtualNetworkGateway $gw
Set-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -DisableActiveActiveFeature
```

This update can take up to 30 to 45 minutes.

Next steps

Once your connection is complete, you can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Modify local network gateway settings using PowerShell

1/17/2017 • 4 min to read • [Edit on GitHub](#)

Sometimes the settings for your local network gateway AddressPrefix or GatewayIPAddress change. The instructions below will help you modify your local network gateway settings. You can also modify these settings in the Azure portal.

Before you begin

You'll need to 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.

To modify IP address prefixes

How to add or remove prefixes - no gateway connection

- **To add** additional address prefixes to a local network gateway that you created, but that doesn't yet have a gateway connection, use the example below. Be sure to change the values to your own.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @('10.0.0.0/24','20.0.0.0/24','30.0.0.0/24')
```

- **To remove** an address prefix from a local network gateway that doesn't have a VPN connection, use the example below. Leave out the prefixes that you no longer need. In this example, we no longer need prefix 20.0.0.0/24 (from the previous example), so we will update the local network gateway and exclude that prefix.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @('10.0.0.0/24','30.0.0.0/24')
```

How to add or remove prefixes - existing gateway connection

If you have created your gateway connection and want to add or remove the IP address prefixes contained in your local network gateway, you'll need to do the following steps in order. This will result in some downtime for your VPN connection. When updating your prefixes, you'll first remove the connection, modify the prefixes, and then create a new connection. In the examples below, be sure to change the values to your own.

IMPORTANT

Don't delete the VPN gateway. If you do so, you'll have to go back through the steps to recreate it, as well as reconfigure your on-premises router with the new settings.

1. Remove the connection.

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName -ResourceGroupName MyRGName
```

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

Set the variable for the LocalNetworkGateway.

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName
```

Modify the prefixes.

```
Set-AzureRmLocalNetworkGateway -LocalNetworkGateway $local `
-AddressPrefix @( '10.0.0.0/24','20.0.0.0/24','30.0.0.0/24' )
```

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

Set the variable for the VirtualNetworkGateway.

```
$gateway1 = Get-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName MyRGName
```

Create the connection. Note that this sample uses the variable \$local that you set in the preceding step.

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

To modify the gateway IP address

To modify the gateway IP address, use the `New-AzureRmVirtualNetworkGatewayConnection` cmdlet. As long as you keep the name of the local network gateway exactly the same as the existing name, the settings will overwrite. At this time, the "Set" cmdlet does not support modifying the gateway IP address.

How to modify the gateway IP address - no gateway connection

To update the gateway IP address for your local network gateway that doesn't yet have a connection, use the example below. You can also update the address prefixes at the same time. The settings you specify will overwrite the existing settings. Be sure to use the existing name of your local network gateway. If you don't, you'll be creating a new local network gateway, not overwriting the existing one.

Use the following example, replacing the values for your own.

```
New-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName `
-Location "West US" -AddressPrefix @( '10.0.0.0/24','20.0.0.0/24','30.0.0.0/24' ) `
-GatewayIpAddress "5.4.3.2" -ResourceGroupName MyRGName
```

How to modify the gateway IP address - existing gateway connection

If a gateway connection already exists, you'll first need to remove the connection. Then, you can modify the gateway IP address and recreate a new connection. This will result in some downtime for your VPN connection.

IMPORTANT

Don't delete the VPN gateway. If you do so, you'll have to go back through the steps to recreate it, as well as reconfigure your on-premises router with the IP address that will be assigned to the newly created gateway.

1. Remove the connection. You can find the name of your connection by using the

```
Get-AzureRmVirtualNetworkGatewayConnection cmdlet.
```

```
Remove-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName `
-ResourceGroupName MyRGName
```

2. Modify the GatewayIpAddress value. You can also modify your address prefixes at this time, if necessary. Note that this will overwrite the existing local network gateway settings. Use the existing name of your local network gateway when modifying so that the settings will overwrite. If you don't, you'll be creating a new local network gateway, not modifying the existing one.

```
New-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName `
-Location "West US" -AddressPrefix @( '10.0.0.0/24','20.0.0.0/24','30.0.0.0/24' ) `
-GatewayIpAddress "104.40.81.124" -ResourceGroupName MyRGName
```

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

```
Get-AzureRmVirtualNetworkGateway cmdlet.
```

Set the variables:

```
$local = Get-AzureRmLocalNetworkGateway -Name MyLocalNetworkGWName -ResourceGroupName MyRGName `
$vnnetgw = Get-AzureRmVirtualNetworkGateway -Name RMGateway -ResourceGroupName MyRGName
```

Create the connection:

```
New-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnectionName -ResourceGroupName MyRGName `
-Location "West US" `
-VirtualNetworkGateway1 $vnnetgw `
-LocalNetworkGateway2 $local `
-ConnectionType IPsec -RoutingWeight 10 -SharedKey 'abc123'
```

Next steps

You can verify your gateway connection. See [Verify a gateway connection](#).

Verify a gateway connection

1/17/2017 • 1 min to read • [Edit on GitHub](#)

You can verify your gateway connection in a few different ways. This article will show you how to verify the status of a Resource Manager gateway connection by using the Azure portal and by using PowerShell.

Verify using PowerShell

You'll need to install the latest version of the Azure Resource Manager PowerShell cmdlets. For information on installing PowerShell cmdlets, see [How to install and configure Azure PowerShell](#). For more information about using Resource Manager cmdlets, see [Using Windows PowerShell with Resource Manager](#).

Step 1: Log in to your Azure account

1. Open your PowerShell console with elevated privileges and connect to your account.

```
Login-AzureRmAccount
```

2. Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

3. Specify the subscription that you want to use.

```
Select-AzureRmSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Step 2: Verify your connection

You can verify that your connection succeeded by using the `Get-AzureRmVirtualNetworkGatewayConnection` cmdlet, with or without `-Debug`.

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

```
Get-AzureRmVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

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.

```



Body:
{
  "name": "MyGWConnection",
  "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/connections/MyGWConnection",
  "properties": {
    "provisioningState": "Succeeded",
    "resourceGuid": "1c484f82-23ec-47e2-8cd8-231107450446b",
    "virtualNetworkGateway1": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/virtualNetworkGateways/vnetgw1",
    },
    "localNetworkGateway2": {
      "id":
"/subscriptions/086cfaa0-0d1d-4b1c-94544-f8e3da2a0c7789/resourceGroups/MyRG/providers/Microsoft.Network/localNetworkGateways/LocalSite",
    },
    "connectionType": "IPsec",
    "routingWeight": 10,
    "sharedKey": "abc123",
    "connectionStatus": "Connected",
    "ingressBytesTransferred": 33509044,
    "egressBytesTransferred": 4142431
  }
}

```

Verify using the Azure portal

In the Azure portal, you can view the connection status by navigating to the connection. There are multiple ways to do this. 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
RG1	 2.35 KB
Status	Data out
Connected	3.14 KB
Location	Virtual network
East US	RMVNet
Subscription name	Virtual network gateway
Windows Azure Internal Consumption	 RMGateway (40.114.5.29)
Subscription ID	Local network gateway
	Site2 (40.76.7.127)

Next steps

- You can add virtual machines to your virtual networks. See [Create a Virtual Machine](#) for steps.

Reset an Azure VPN Gateway using PowerShell

1/17/2017 • 2 min to read • [Edit on GitHub](#)

This article walks you through resetting your Azure VPN Gateway using PowerShell cmdlets. These instructions include both the classic deployment model and the Resource Manager deployment model.

Resetting the Azure VPN gateway is helpful if you lose cross-premises VPN connectivity on one or more S2S 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.

Each Azure VPN gateway is composed of two VM instances running in an active-standby configuration. When you use the PowerShell cmdlet to 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.

Once the command is issued, the current active instance of the Azure VPN gateway is rebooted immediately. There will be a brief gap during the failover from the active instance (being rebooted), to the standby instance. The gap should be less than one minute.

If the connection is not restored after the first reboot, issue the same command again to reboot the second VM instance (the new active gateway). If the two reboots are requested back to back, there will be a slightly longer period where both VM instances (active and standby) are being rebooted. This will cause a longer gap on the VPN connectivity, up to 2 to 4 minutes for VMs to complete the reboots.

After two reboots, if you are still experiencing cross-premises connectivity problems, please open a support request from the Azure portal.

Before you begin

Before you reset your gateway, verify the key items listed below for each IPsec Site-to-Site (S2S) VPN tunnel. Any mismatch in the items will result in the disconnect of S2S VPN tunnels. Verifying and correcting the configurations for your on-premises and Azure VPN gateways saves you from unnecessary reboots and disruptions for the other working connections on the gateways.

Verify the following items before resetting your gateway:

- The Internet IP addresses (VIPs) for both the Azure VPN gateway and the on-premises VPN gateway are configured correctly in both the Azure and the on-premises VPN policies.
- The pre-shared key must be the same on both Azure and on-premises VPN gateways.
- If you apply specific IPsec/IKE configuration, such as encryption, hashing algorithms, and PFS (Perfect Forward Secrecy), ensure both the Azure and on-premises VPN gateways have the same configurations.

Reset a VPN Gateway using the Resource Management deployment model

The PowerShell Resource Manager cmdlet for resetting gateway is `Reset-AzureRmVirtualNetworkGateway`. The following example resets the Azure VPN gateway, "VNet1GW", in resource group "TestRG1".

```
$gw = Get-AzureRmVirtualNetworkGateway -Name VNet1GW -ResourceGroup TestRG1
Reset-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw
```


Reset a VPN Gateway using the classic deployment model

The PowerShell cmdlet for resetting Azure VPN gateway is `Reset-AzureVNetGateway`. The following example resets the Azure VPN gateway for the virtual network called "ContosoVNet".

```
Reset-AzureVNetGateway -VnetName "ContosoVNet"
```

Result:

```
Error      :  
HttpStatusCode : OK  
Id          : f1600632-c819-4b2f-ac0e-f4126bec1ff8  
Status      : Successful  
RequestId   : 9ca273de2c4d01e986480ce1ffa4d6d9  
StatusCode  : OK
```

Next steps

See the [PowerShell Service Management cmdlet reference](#) and the [PowerShell Resource Manager cmdlet reference](#) for more information.

Working with self-signed certificates for Point-to-Site connections

1/17/2017 • 5 min to read • [Edit on GitHub](#)

This article helps you create a self-signed certificate using **makecert**, and then generate client certificates from it. The steps are written for makecert on Windows 10. Makecert has been validated to create certificates that are compatible with P2S connections.

For P2S connections, the preferred method for certificates is to use your enterprise certificate solution, making sure to issue the client certificates with the common name value format 'name@yourdomain.com', rather than the 'NetBIOS domain name\username' format.

If you don't have an enterprise solution, a self-signed certificate is necessary to allow P2S clients to connect to a virtual network. We are aware that makecert has been deprecated, but it is still a valid method for creating self-signed certificates that are compatible with P2S connections. We're working on another solution for creating self-signed certificates, but at this time, makecert is the preferred method.

Create a self-signed certificate

Makecert is one way of creating a self-signed certificate. The following steps walk you through creating a self-signed certificate using makecert. These steps are not deployment-model specific. They are valid for both Resource Manager and classic.

To create a self-signed certificate

1. From a computer running Windows 10, download and install the [Windows Software Development Kit \(SDK\) for Windows 10](#).
2. After installation, you can find the makecert.exe utility under this path: C:\Program Files (x86)\Windows Kits\10\bin<arch>.

Example: `C:\Program Files (x86)\Windows Kits\10\bin\x64`

3. Next, 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. Run the following command, as administrator. Replace *ARMP2SRootCert* and *ARMP2SRootCert.cer* with the name that you want to use for the certificate.

The certificate will be located in your Certificates - Current User\Personal\Certificates.

```
makecert -sky exchange -r -n "CN=ARMP2SRootCert" -pe -a sha1 -len 2048 -ss My "ARMP2SRootCert.cer"
```

To obtain the public key

As part of the VPN Gateway configuration for Point-to-Site connections, the public key for the root certificate is uploaded to Azure.

1. To obtain a .cer file from the certificate, open **certmgr.msc**. Right-click the self-signed root certificate, click **all tasks**, and then click **export**. This opens the **Certificate Export Wizard**.
2. In the Wizard, click **Next**, select **No, do not export the private key**, and then click **Next**.
3. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)**. Then, click **Next**.
4. 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**.

5. Click **Finish** to export the certificate.

Export the self-signed certificate (optional)

You may want to export the self-signed 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. Any computer with a client certificate installed and that is also configured with the proper VPN client settings can connect to your virtual network via P2S. For that reason, you want to make sure that client certificates are generated and installed only when needed and that the self-signed certificate is stored safely.

To export the self-signed certificate as a .pfx, select the root certificate and use the same steps as described in [Export a client certificate](#) to export.

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.

Part 1 - Generate a client certificate from a self-signed certificate

The following steps walk you through one way to generate a client certificate from a self-signed certificate. You may generate multiple client certificates from the same certificate. Each client certificate can then be exported and installed on the client computer.

1. On the same computer that you used to create the self-signed certificate, open a command prompt as administrator.
2. In this example, "ARMP2SRootCert" refers to the self-signed certificate that you generated.
 - Change "ARMP2SRootCert" to the name of the self-signed root that you are generating the client certificate from.
 - Change *ClientCertificateName* to the name you want to generate a client certificate to be.

Modify and run the sample to generate a client certificate. If you run the following example without modifying it, the result is a client certificate named ClientCertificateName in your Personal certificate store that was generated from root certificate ARMP2SRootCert.

```
makecert.exe -n "CN=ClientCertificateName" -pe -sky exchange -m 96 -ss My -in  
"ARMP2SRootCert" -is my -a sha1
```

3. All certificates are stored in your 'Certificates - Current User\Personal\Certificates' store on your computer. You can generate as many client certificates as needed based on this procedure.

Part 2 - Export a client certificate

1. To export a client certificate, open **certmgr.msc**. Right-click the client certificate that you want to export, click **all tasks**, and then click **export**. This opens the **Certificate Export Wizard**.
2. In the Wizard, click **Next**, then select **Yes, export the private key**, and then click **Next**.
3. On the **Export File Format** page, you can leave the defaults selected. Then click **Next**.
4. 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**.
5. 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**.
6. Click **Finish** to export the certificate.

Part 3 - Install a client certificate

Each client that you want to connect to your virtual network by using a Point-to-Site connection must have a

client certificate installed. This certificate is in addition to the required VPN configuration package. The following steps walk you through installing the client certificate manually.

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**, 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 if you used one, or verify that the security principal that is installing the certificate 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**. The certificate is now successfully imported.

Next steps

Continue with your Point-to-Site configuration.

- For **Resource Manager** deployment model steps, see [Configure a Point-to-Site connection to a VNet using PowerShell](#).
- For **classic** deployment model steps, see [Configure a Point-to-Site VPN connection to a VNet using the classic portal](#).

Configure a VPN gateway for the classic deployment model

1/17/2017 • 8 min to read • [Edit on GitHub](#)

If you want to create a secure cross-premises connection between Azure and your on-premises location, you need to configure a VPN gateway connection. In the classic deployment model, a gateway can be one of two VPN routing types: static, or dynamic. The type you choose depends on both your network design plan, and the on-premises VPN device you want to use.

For example, some connectivity options, such as a point-to-site connection, require a dynamic routing gateway. If you want to configure your gateway to support both point-to-site (P2S) connections and a site-to-site (S2S) connection, you have to configure a dynamic routing gateway even though site-to-site can be configured with either gateway VPN routing type.

Additionally, must make sure that the device you want to use for your connection supports the VPN routing type that you want to create. See [About VPN Devices](#).

About this article

This article was written for the classic deployment model using the [classic portal](#) (not the Azure portal).

About Azure deployment models

It's important to know that Azure currently works with two deployment models: Resource Manager and classic. Before you begin your configuration, make sure that you understand the deployment models and tools. You'll need to know which model that you want to work in. Not all networking features are supported yet for both models. For information about the deployment models, see [Understanding Resource Manager deployment and classic deployment](#).

Configuration overview

The following steps walk you through configuring your VPN gateway in the Azure classic portal. These steps apply to gateways for virtual networks that were created using the classic deployment model. Currently, not all of the configuration settings for gateways are available in the Azure portal. When they are, we will create a new set of instructions that apply to the Azure portal.

1. [Create a VPN gateway for your VNet](#)
2. [Gather information for your VPN device configuration](#)
3. [Configure your VPN device](#)
4. [Verify your local network ranges and VPN gateway IP address](#)

Before you begin

Before you configure your gateway, you first need to create your virtual network. For steps to create a virtual network for cross-premises connectivity, see [Configure a virtual network with a site-to-site VPN connection](#), or [Configure a virtual network with a point-to-site VPN connection](#). Then, use the following steps to configure the VPN gateway and gather the information you need to configure your VPN device.

If you already have a VPN gateway and you want to change the VPN routing type, see [How to change the VPN routing type for your gateway](#).

Create a VPN gateway

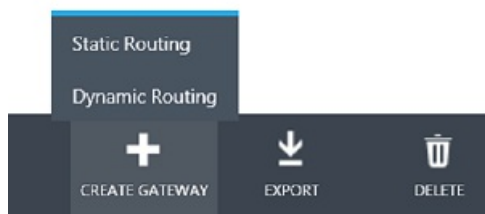
1. In the [Azure classic portal](#), on the **Networks** page, verify that the status column for your virtual network is **Created**.
2. In the **Name** column, click the name of your virtual network.
3. On the **Dashboard** page, notice that this VNet doesn't have a gateway configured yet. You'll see this status as you go through the steps to configure your gateway.

 DASHBOARD  CONFIGURE  CERTIFICATES

virtual network



Next, at the bottom of the page, click **Create Gateway**. You can select either *Static Routing* or *Dynamic Routing*. The VPN routing type you select depends on few factors. For example, what your VPN device supports and whether you need to support point-to-site connections. Check [About VPN Devices for Virtual Network Connectivity](#) to verify the VPN routing type that you need. Once the gateway has been created, you can't change between gateway VPN routing types without deleting and re-creating the gateway. When the system prompts you to confirm that you want the gateway created, click **Yes**.



When your gateway is creating, notice the gateway graphic on the page changes to yellow and says *Creating Gateway*. It may take up to 45 minutes for the gateway to create. Wait until the gateway is complete before you can move forward with other configuration settings.

virtual network



When the gateway changes to *Connecting*, you can gather the information you'll need for your VPN device.

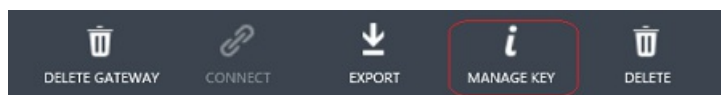


Gather information for your VPN device configuration

After the gateway has been created, gather information for your VPN device configuration. This information is located on the **Dashboard** page for your virtual network:

1. **Gateway IP address** - The IP address can be found on the **Dashboard** page. You won't be able to see it until after your gateway has finished creating.
2. **Shared key** - Click **Manage Key** at the bottom of the screen. Click the icon next to the key to copy it to your clipboard, and then paste and save the key. This button only works when there is a single S2S VPN tunnel. If

you have multiple S2S VPN tunnels, please use the *Get Virtual Network Gateway Shared Key* API or PowerShell cmdlet.



Configure your VPN device

After completing the previous steps, you or your network administrator will need to configure the VPN device in order to create the connection. See [About VPN Devices for Virtual Network Connectivity](#) for more information about VPN devices.

After the VPN device has been configured, you can view your updated connection information on the Dashboard page for your VNet.

You can also run one of the following commands to test your connection:

	CISCO ASA	CISCO ISR/ASR	JUNIPER SSG/ISG	JUNIPER SRX/J
Check main mode SAs	show crypto isakmp sa	show crypto isakmp sa	get ike cookie	show security ike security-association
Check quick mode SAs	show crypto ipsec sa	show crypto ipsec sa	get sa	show security ipsec security-association

Verify your local network ranges and VPN gateway IP address

Verify your VPN gateway IP address

For gateway to connect properly, the IP address for your VPN device must be correctly configured for the Local Network that you specified for your cross-premises configuration. Typically, this is configured during the site-to-site configuration process. However, if you previously used this local network with a different device, or the IP address has changed for this local network, edit the settings to specify the correct Gateway IP address.

1. To verify your gateway IP address, click **Networks** on the left portal pane and then select **Local Networks** at the top of the page. You'll see the VPN Gateway Address for each local network that you have created. To edit the IP address, select the VNet and click **Edit** at the bottom of the page.
2. On the **Specify your local network details** page, edit the IP address, and then click the next arrow at the bottom of the page.
3. On the **Specify the address space** page, click the checkmark on the lower right to save your settings.

Verify the address ranges for your local networks

For the correct traffic to flow through the gateway to your on-premises location, you need to verify that each IP address range is specified. Each range must be listed in your Azure **Local Networks** configuration. Depending on the network configuration of your on-premises location, this can be a somewhat large task. Traffic that is bound for an IP address that is contained within the listed ranges will be sent through the virtual network VPN gateway. The ranges that you list don't have to be private ranges, although you will want to verify that your on-premises configuration can receive the inbound traffic.

To add or edit the ranges for a Local Network, use the following steps.

1. To edit the IP address ranges for a local network, click **Networks** on the left portal pane and then select **Local Networks** at the top of the page. In the portal, the easiest way to view the ranges that you've listed is on the **Edit** page. To see your ranges, select the VNet and click **Edit** at the bottom of the page.
2. On the **Specify your local network details** page, don't make any changes. Click the next arrow at the bottom

of the page.

3. On the **Specify the address space** page, make your network address space changes. Then click the checkmark to save your configuration.

How to view gateway traffic

You can view your gateway and gateway traffic from your Virtual Network **Dashboard** page.

On the **Dashboard** page you can view the following:

- The amount of data that is flowing through your gateway, both data in and data out.
- The names of the DNS servers that are specified for your virtual network.
- The connection between your gateway and your VPN device.
- The shared key that is used to configure your gateway connection to your VPN device.

How to change the VPN routing type for your gateway

Because some connectivity configurations are only available for certain gateway routing types, you may find that you need to change the gateway VPN routing type of an existing VPN gateway. For example, you may want to add point-to-site connectivity to an already existing site-to-site connection that has a static gateway. Point-to-site connections require a dynamic gateway. This means to configure a P2S connection, you have to change your gateway VPN routing type from static to dynamic.

If you need to change a gateway VPN routing type, you'll delete the existing gateway, and then create a new gateway with the new routing type. You don't need to delete the entire virtual network to change the gateway routing type.

Before changing your gateway VPN routing type, be sure to verify that your VPN device supports the routing type that you want to use. To download new routing configuration samples and check VPN device requirements, see [About VPN Devices for Virtual Network Connectivity](#).

IMPORTANT

When you delete a virtual network VPN gateway, the VIP assigned to the gateway is released. When you recreate the gateway, a new VIP is assigned to it.

1. Delete the existing VPN gateway.

On the **Dashboard** page for your virtual network, navigate to the bottom of the page and click **Delete Gateway**. Wait for the notification that the gateway has been deleted. Once you receive the notification on the screen that your gateway has been deleted, you can create a new gateway.

2. Create a new VPN gateway.

Use the procedure at the top of the page to create a new gateway: [Create a VPN gateway](#).

Next steps

You can add virtual machines to your virtual network. See [How to create a custom virtual machine](#).

If you want to configure a point-to-site VPN connection, see [Configure a point-to-site VPN connection](#).