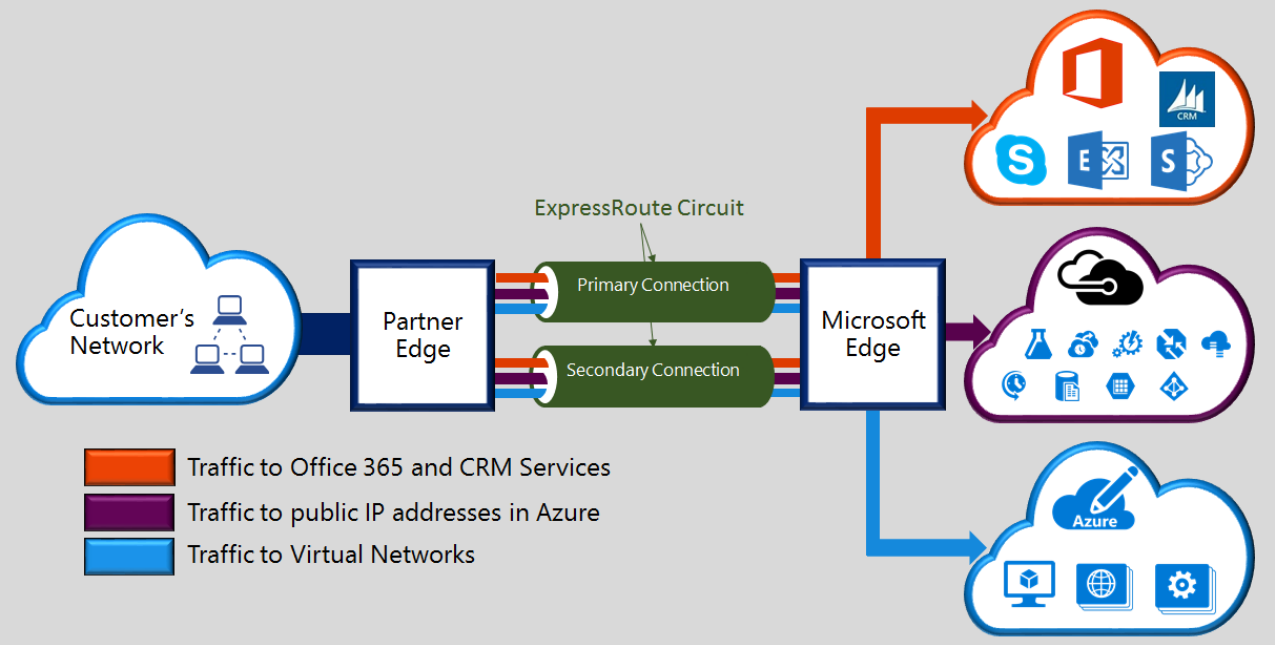
Table of Contents

ExpressRoute technical overview

1/17/2017 • 6 min to read • Edit on GitHub

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.



Key benefits include:

- Layer 3 connectivity between your on-premises network and the Microsoft Cloud through a connectivity provider. Connectivity can be from an any-to-any (IPVPN) network, a point-to-point Ethernet connection, or through a virtual cross-connection via an Ethernet exchange.
- Connectivity to Microsoft cloud services across all regions in the geopolitical region.
- Global connectivity to Microsoft services across all regions with ExpressRoute premium add-on.
- Dynamic routing between your network and Microsoft over industry standard protocols (BGP).
- Built-in redundancy in every peering location for higher reliability.
- Connection uptime SLA.
- QoS and support for multiple classes of service for special applications, such as Skype for Business.

See the ExpressRoute FAQ for more details.

How can I connect my network to Microsoft using ExpressRoute?

You can create a connection between your on-premises network and the Microsoft cloud in three different ways:

Co-located at a cloud exchange

If you are co-located in a facility with a cloud exchange, you can order virtual cross-connections to the Microsoft cloud through the co-location provider's Ethernet exchange. Co-location providers can offer either Layer 2 cross-

connections, or managed Layer 3 cross-connections between your infrastructure in the co-location facility and the Microsoft cloud.

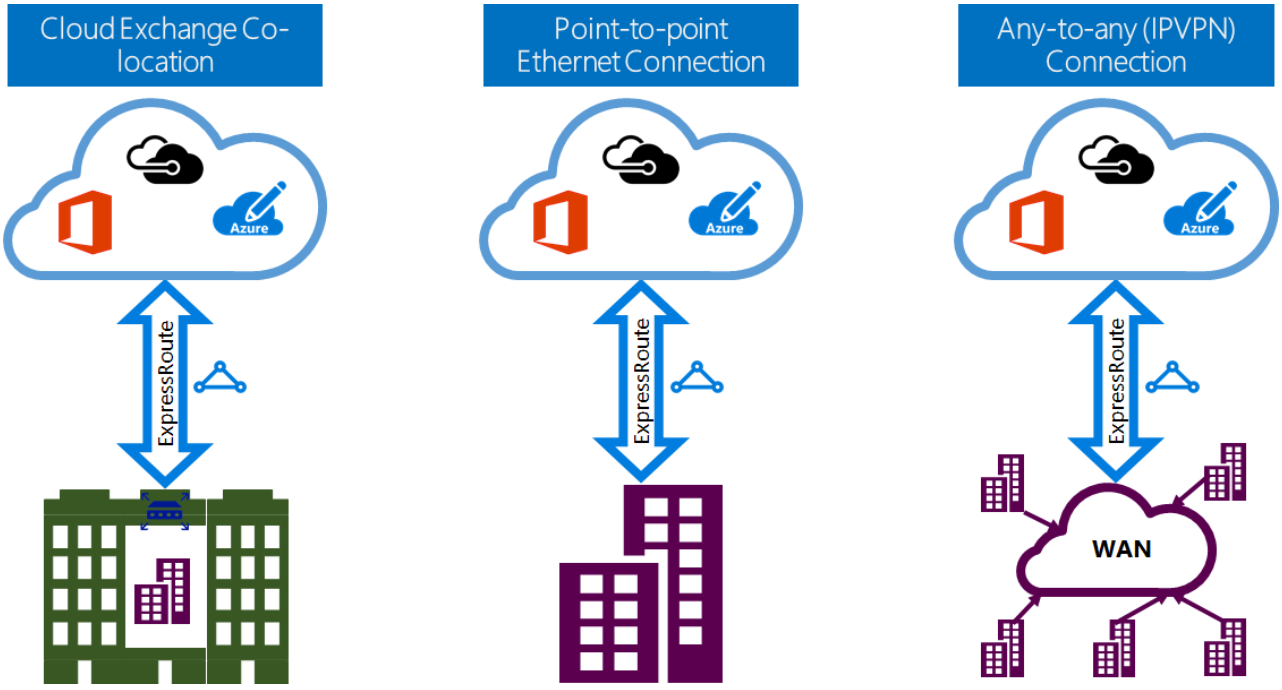
Point-to-point Ethernet connections

You can connect your on-premises datacenters/offices to the Microsoft cloud through point-to-point Ethernet links. Point-to-point Ethernet providers can offer Layer 2 connections, or managed Layer 3 connections between your site and the Microsoft cloud.

Any-to-any (IPVPN) networks

You can integrate your WAN with the Microsoft cloud. IPVPN providers (typically MPLS VPN) offer any-to-any connectivity between your branch offices and datacenters. The Microsoft cloud can be interconnected to your WAN to make it look just like any other branch office. WAN providers typically offer managed Layer 3 connectivity. ExpressRoute capabilities and features are all identical across all of the above connectivity models.

Connectivity providers can offer one or more connectivity models. You can work with your connectivity provider to pick the model that works best for you.



ExpressRoute features

ExpressRoute supports the following features and capabilities:

Layer 3 connectivity

Microsoft uses industry standard dynamic routing protocol (BGP) to exchange routes between your on-premises network, your instances in Azure, and Microsoft public addresses. We establish multiple BGP sessions with your network for different traffic profiles. More details can be found in the ExpressRoute circuit and routing domains article.

Redundancy

Each ExpressRoute circuit consists of two connections to two Microsoft Enterprise edge routers (MSEEs) from the connectivity provider / your network edge. Microsoft will require dual BGP connection from the connectivity provider / your side – one to each MSEE. You may choose not to deploy redundant devices / Ethernet circuits at your end. However, connectivity providers use redundant devices to ensure that your connections are handed off to Microsoft in a redundant manner. A redundant Layer 3 connectivity configuration is a requirement for our SLA to be valid.

Connectivity to Microsoft cloud services

ExpressRoute provides private network connectivity to Microsoft cloud services. Infrastructure and platform services running in Azure often benefit by addressing network architecture and performance considerations. Therefore we recommend enterprises use ExpressRoute for Azure.

Software as a Service offerings, like Office 365 and Dynamics 365, were created to be accessed securely and reliably via the Internet. Therefore, we only recommend ExpressRoute for these applications in specific scenarios.

> **IMPORTANT**
> Using ExpressRoute to access Azure is **recommended** for all enterprises. For guidance on using ExpressRoute to access Office 365 visit http://aka.ms/ExpressRouteOffice365.

ExpressRoute connections enable access to the following services:

- Microsoft Azure services
- Microsoft Office 365 services
- Microsoft CRM Online services

You can visit the ExpressRoute FAQ page for a detailed list of services supported over ExpressRoute.

Connectivity to all regions within a geopolitical region

You can connect to Microsoft in one of our peering locations and have access to all regions within the geopolitical region.

For example, if you connected to Microsoft in Amsterdam through ExpressRoute, you will have access to all Microsoft cloud services hosted in Northern Europe and Western Europe. Refer to the ExpressRoute partners and peering locations article for an overview of the geopolitical regions, associated Microsoft cloud regions, and corresponding ExpressRoute peering locations.

Global connectivity with ExpressRoute premium add-on

You can enable the ExpressRoute premium add-on feature to extend connectivity across geopolitical boundaries. For example, if you are connected to Microsoft in Amsterdam through ExpressRoute, you will have access to all Microsoft cloud services hosted in all regions across the world (national clouds are excluded). You can access services deployed in South America or Australia the same way you access the North and West Europe regions.

Rich connectivity partner ecosystem

ExpressRoute has a constantly growing ecosystem of connectivity providers and SI partners. You can refer to the ExpressRoute providers and locations article for the latest information.

Connectivity to national clouds

Microsoft operates isolated cloud environments for special geopolitical regions and customer segments. Refer to the ExpressRoute providers and locations page for a list of national clouds and providers.

Supported bandwidth options

You can purchase ExpressRoute circuits for a wide range of bandwidths. The list of supported bandwidths is listed below. Be sure to check with your connectivity provider to determine the list of supported bandwidths they provide.

- 50 Mbps
- 100 Mbps
- 200 Mbps
- 500 Mbps
- 1 Gbps
- 2 Gbps
- 5 Gbps

- 10 Gbps

Dynamic scaling of bandwidth

You have the ability to increase the ExpressRoute circuit bandwidth (on a best effort basis) without having to tear down your connections.

Flexible billing models

You can pick a billing model that works best for you. Choose between the billing models listed below. Refer to the ExpressRoute FAQ page for more details.

- **Unlimited data**. The ExpressRoute circuit is charged based on a monthly fee, and all inbound and outbound data transfer is included free of charge.
- **Metered data**. The ExpressRoute circuit is charged based on a monthly fee. All inbound data transfer is free of charge. Outbound data transfer is charged per GB of data transfer. Data transfer rates vary by region.
- **ExpressRoute premium add-on**. The ExpressRoute premium is an add-on over the ExpressRoute circuit. The ExpressRoute premium add-on provides the following capabilities:
  - Increased route limits for Azure public and Azure private peering from 4,000 routes to 10,000 routes.
  - Global connectivity for services. An ExpressRoute circuit created in any region (excluding national clouds) will have access to resources across any other region in the world. For example, a virtual network created in West Europe can be accessed through an ExpressRoute circuit provisioned in Silicon Valley.
  - Increased number of VNet links per ExpressRoute circuit from 10 to a larger limit, depending on the bandwidth of the circuit.

Next steps

- Learn about ExpressRoute connections and routing domains. See ExpressRoute circuits and routing domains.
- Find a service provider. See ExpressRoute partners and peering locations.
- Ensure that all prerequisites are met. See ExpressRoute prerequisites.
- Refer to the requirements for Routing, NAT and QoS.
- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit
  - Configure routing
  - Link a VNet to an ExpressRoute circuit

ExpressRoute FAQ

1/17/2017 • 14 min to read • Edit on GitHub

What is ExpressRoute?

ExpressRoute is an Azure service that lets you create private connections between Microsoft datacenters and infrastructure that's on your premises or in a colocation facility. ExpressRoute connections do not go over the public Internet, and offer higher security, reliability and speeds with lower latencies than typical connections over the Internet.

What are the benefits of using ExpressRoute and private network connections?

ExpressRoute connections do not go over the public Internet, and offer higher security, reliability and speeds with lower and consistent latencies than typical connections over the Internet. In some cases, using ExpressRoute connections to transfer data between on-premises devices and Azure can yield significant cost benefits.

What Microsoft cloud services are supported over ExpressRoute?

ExpressRoute supports most Microsoft Azure services today including Office 365. Look for updates on general availability soon.

Where is the service available?

See this page for service location and availability: ExpressRoute partners and locations.

How can I use ExpressRoute to connect to Microsoft if I don't have partnerships with one of the ExpressRoute-carrier partners?

You can select a regional carrier and land Ethernet connections to one of the supported exchange provider locations. You can then peer with Microsoft at the provider location. Check the last section of ExpressRoute partners and locations to see if your service provider is present in any of the exchange locations. You can then order an ExpressRoute circuit through the service provider to connect to Azure.

How much does ExpressRoute cost?

Check pricing details for pricing information.

If I pay for an ExpressRoute circuit of a given bandwidth, does the VPN connection I purchase from my network service provider have to be the same speed?

No. You can purchase a VPN connection of any speed from your service provider. However, your connection to Azure will be limited to the ExpressRoute circuit bandwidth that you purchase.

If I pay for an ExpressRoute circuit of a given bandwidth, do I have the ability to burst up to higher speeds if required?

Yes. ExpressRoute circuits are configured to support cases where you can burst up to two times the bandwidth limit you procured for no additional cost. Check with your service provider if they support this capability.

Can I use the same private network connection with Virtual Network and other Azure services simultaneously?

Yes. An ExpressRoute circuit, once setup will allow you to access services within a virtual network and other Azure services simultaneously. You will connect to virtual networks over the private peering path and other services over the public peering path.

Does ExpressRoute offer a Service Level Agreement (SLA)?

Please refer to the ExpressRoute SLA page for more information.

Supported services

ExpressRoute supports three routing domains for various types of services.

Private peering

- Virtual Networks, including all virtual machines and cloud services

Public peering

- Most of the Azure services with a few exceptions below
- Power BI
- Dynamics 365 for Operations (formerly known as Dynamics AX Online)

Microsoft peering

- Office 365
- Most of the Dynamics 365 services (formerly known as CRM Online)
  - Dynamics 365 for Sales
  - Dynamics 365 for Customer Service
  - Dynamics 365 for Field Service
  - Dynamics 365 for Project Service

The following Azure services are not supported on ExpressRoute

- CDN
- Visual Studio Team Services Load Testing
- Multi-factor Authentication
- Traffic Manager

Data and connections

Are there limits on the amount of data that I can transfer using ExpressRoute?

We do not set a limit on the amount of data transfer. Refer to pricing details for information on bandwidth rates.

What connection speeds are supported by ExpressRoute?

Supported bandwidth offers:

|50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1Gbps, 2 Gbps, 5 Gbps, 10Gbps|

Which service providers are available?

See ExpressRoute partners and locations for the list of service providers and locations.

Technical details

What are the technical requirements for connecting my on-premises location to Azure?

Please see ExpressRoute prerequisites page for requirements.

Are connections to ExpressRoute redundant?

Yes. Each Express Route circuit has a redundant pair of cross connections configured to provide high availability.

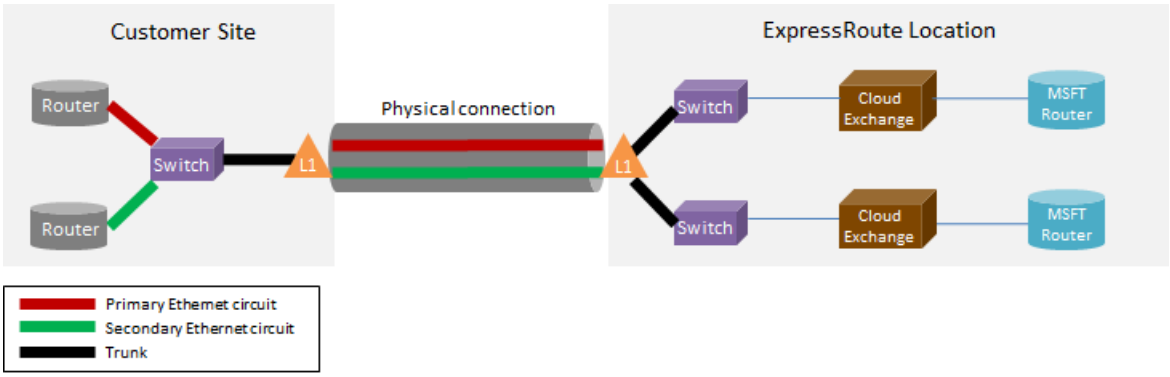
Will I lose connectivity if one of my ExpressRoute links fail?

You will not lose connectivity if one of the cross connections fails. A redundant connection is available to

support the load of your network. You can additionally create multiple circuits in a different peering location to achieve failure resilience.

If I'm not co-located at a cloud exchange and my service provider offers point-to-point connection, do I need to order two physical connections between my on-premises network and Microsoft?

No, you only need one physical connection if your service provider can establish two Ethernet virtual circuits over the physical connection. The physical connection (e.g. an optical fiber) is terminated on a layer 1 (L1) device (see image below). The two Ethernet virtual circuits are tagged with different VLAN IDs, one for the primary circuit and one for the secondary. Those VLAN IDs are in the outer 802.1Q Ethernet header. The inner 802.1Q Ethernet header (not shown) is mapped to a specific ExpressRoute routing domain.



Can I extend one of my VLANs to Azure using ExpressRoute?

No. We do not support layer 2 connectivity extensions into Azure.

Can I have more than one ExpressRoute circuit in my subscription?

Yes. You can have more than one ExpressRoute circuit in your subscription. The default limit on the number of dedicated circuits is set to 10. You can contact Microsoft Support to increase the limit if needed.

Can I have ExpressRoute circuits from different service providers?

Yes. You can have ExpressRoute circuits with many service providers. Each ExpressRoute circuit will be associated with one service provider only.

How do I connect my virtual networks to an ExpressRoute circuit

The basic steps are outlined below.

- You must establish an ExpressRoute circuit and have the service provider enable it.
- You or the provider must configure the BGP peering(s).
- You must link the virtual network to the ExpressRoute circuit.

See ExpressRoute workflows for circuit provisioning and circuit states for more information.

Are there connectivity boundaries for my ExpressRoute circuit?

Yes. ExpressRoute partners and locations page provides an overview of the connectivity boundaries for an ExpressRoute circuit. Connectivity for an ExpressRoute circuit is limited to a single geopolitical region. Connectivity can be expanded to cross geopolitical regions by enabling the ExpressRoute premium feature.

Can I link to more than one virtual network to an ExpressRoute circuit?

Yes. You can link up to 10 virtual networks to an ExpressRoute circuit.

I have multiple Azure subscriptions that contain virtual networks. Can I connect virtual networks that are in separate subscriptions to a single ExpressRoute circuit?

Yes. You can authorize up to 10 other Azure subscriptions to use a single ExpressRoute circuit. This limit can be increased by enabling the ExpressRoute premium feature.

For more details, see Sharing an ExpressRoute circuit across multiple subscriptions.

Are virtual networks connected to the same circuit isolated from each other?

No. All virtual networks linked to the same ExpressRoute circuit are part of the same routing domain and are not isolated from each other from a routing perspective. If you need route isolation, you'll need to create a separate ExpressRoute circuit.

Can I have one virtual network connected to more than one ExpressRoute circuit?

Yes. You can link a single virtual network with up to 4 ExpressRoute circuits. They must be ordered through 4 different ExpressRoute locations.

Can I access the internet from my virtual networks connected to ExpressRoute circuits?

Yes. If you have not advertised default routes (0.0.0.0/0) or internet route prefixes through the BGP session, you will be able to connect to the internet from a virtual network linked to an ExpressRoute circuit.

Can I block internet connectivity to virtual networks connected to ExpressRoute circuits?

Yes. You can advertise default routes (0.0.0.0/0) to block all internet connectivity to virtual machines deployed within a virtual network and route all traffic out through the ExpressRoute circuit. Note that if you advertise default routes, we will force traffic to services offered over public peering (such as Azure storage and SQL DB) back to your premises. You will have to configure your routers to return traffic to Azure through the public peering path or over the internet.

Can virtual networks linked to the same ExpressRoute circuit talk to each other?

Yes. Virtual machines deployed in virtual networks connected to the same ExpressRoute circuit can communicate with each other.

Can I use site-to-site connectivity for virtual networks in conjunction with ExpressRoute?

Yes. ExpressRoute can coexist with site-to-site VPNs.

Can I move a virtual network from site-to-site / point-to-site configuration to use ExpressRoute?

Yes. You will have to create an ExpressRoute gateway within your virtual network. There will be a small downtime associated with the process.

What do I need to connect to Azure storage over ExpressRoute?

You must establish an ExpressRoute circuit and configure routes for public peering.

Are there limits on the number of routes I can advertise?

Yes. We accept up to 4000 route prefixes for private peering and 200 each for public peering and Microsoft peering. You can increase this to 10,000 routes for private peering if you enable the ExpressRoute premium feature.

Are there restrictions on IP ranges I can advertise over the BGP session?

We do not accept private prefixes (RFC1918) in the Public and Microsoft peering BGP session.

What happens if I exceed the BGP limits?

BGP sessions will be dropped. They will be reset once the prefix count goes below the limit.

What is the ExpressRoute BGP hold time? Can it be adjusted?

The hold time is 180. The keep-alive messages are sent every 60 seconds. These are fixed settings on the Microsoft side that cannot be changed.

After I advertise the default route (0.0.0.0/0) to my virtual networks, I can't activate Windows running on my Azure VMs. How to I fix this?

The following steps will help Azure recognize the activation request:

1. Establish the public peering for your ExpressRoute circuit.
2. Perform a DNS lookup and find the IP address of **kms.core.windows.net**

3. Then do one of the following two items so that the Key Management Service will recognize that the activation request comes from Azure and will honor the request.
   - On your on-premises network, route the traffic destined for the IP address (obtained in step 2) back to Azure via the public peering.
   - Have your NSP provider hair-pin the traffic back to Azure via the public peering.

Can I change the bandwidth of an ExpressRoute circuit?

Yes. You can increase the bandwidth of an ExpressRoute circuit without having to tear it down. You will have to follow up with your connectivity provider to ensure that they update the throttles within their networks to support the bandwidth increase. You will however not be able to reduce the bandwidth of an ExpressRoute circuit. Having to lower the bandwidth will mean a tear down and recreation of an ExpressRoute circuit.

How do I change the bandwidth of an ExpressRoute circuit?

You can update the bandwidth of the ExpressRoute circuit using the update dedicated circuit API and PowerShell cmdlet.

ExpressRoute Premium

What is ExpressRoute premium?

ExpressRoute premium is a collection of features listed below.

- Increased routing table limit from 4000 routes to 10,000 routes for private peering.
- Increased number of VNets that can be connected to the ExpressRoute circuit (default is 10). See table below for more details.
- Global connectivity over the Microsoft core network. You will now be able to link a VNet in one geopolitical region with an ExpressRoute circuit in another region. **Example:** You can link a VNet created in Europe West to an ExpressRoute circuit created in Silicon Valley.
- Connectivity to Office 365 services and CRM Online.

How many VNets can I link to an ExpressRoute circuit if I enabled ExpressRoute premium?

The tables below show the ExpressRoute limits and the number of VNets per ExpressRoute circuit.

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
Maximum number of routes for Azure public peering with ExpressRoute premium add-on	200

RESOURCE	DEFAULT LIMIT
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

How do I enable ExpressRoute premium?

ExpressRoute premium features can be enabled when the feature is enabled and can be shut down by updating the circuit state. You can enable ExpressRoute premium at circuit creation time or can call the update dedicated circuit API / PowerShell cmdlet to enable ExpressRoute premium.

How do I disable ExpressRoute premium?

You can disable ExpressRoute premium by calling the update dedicated circuit API / PowerShell cmdlet You must ensure that you have scaled your connectivity needs to meet the default limits before you disable ExpressRoute premium. We will fail request to disable ExpressRoute premium if your utilization scales beyond the default limits.

Can I pick and choose the features I want from the premium feature set?

No. You will not be able to pick the features you need. We enable all features when you turn on ExpressRoute premium.

How much does ExpressRoute premium cost?

Refer to pricing details for cost.

Do I pay for ExpressRoute premium in addition to standard ExpressRoute charges?

Yes. ExpressRoute premium charges apply on top of ExpressRoute circuit charges and charges required by the connectivity provider.

ExpressRoute and Office 365 Services and CRM Online

ExpressRoute provides private network connectivity to Microsoft cloud services. Infrastructure and platform services running in Azure often benefit by addressing network architecture and performance considerations. Therefore we recommend enterprises use ExpressRoute for Azure.

Software as a Service offerings, like Office 365 and Dynamics 365, were created to be accessed securely and reliably via the Internet. Therefore, we only recommend ExpressRoute for these applications in specific scenarios.

> **IMPORTANT**
>
> Using ExpressRoute to access Azure is **recommended** for all enterprises. For guidance on using ExpressRoute to access Office 365 visit http://aka.ms/ExpressRouteOffice365.

How do I create an ExpressRoute circuit to connect to Office 365 services and CRM Online?

1. Review the ExpressRoute prerequisites page page to make sure you meet the requirements.
2. Review the list of service providers and locations at ExpressRoute partners and locations to ensure that your connectivity needs are met.
3. Plan your capacity requirements by reviewing Network planning and performance tuning for Office 365.
4. Follow the steps listed in the workflows below to setup connectivity ExpressRoute workflows for circuit provisioning and circuit states.

> **IMPORTANT**
>
> Ensure that you have enabled ExpressRoute premium add-on when configuring connectivity to Office 365 services and CRM Online.

Do I need to enable Azure Public Peering to connect to Office 365 services and CRM Online?

No, you only need to enable Microsoft Peering. Authentication traffic to Azure AD will be sent through Microsoft Peering.

Can my existing ExpressRoute circuits support connectivity to Office 365 services and CRM Online?

Yes. Your existing ExpressRoute circuit can be configured to support connectivity to Office 365 services. Ensure that you have sufficient capacity to connect to Office 365 services and make sure that you have enabled premium add-on. Network planning and performance tuning for Office 365 will help you plan your connectivity needs. Also, see Create and modify an ExpressRoute circuit.

What Office 365 services can be accessed over an ExpressRoute connection?

Refer to Office 365 URLs and IP address ranges page for an up to date list of services supported over ExpressRoute.

How much does ExpressRoute for Office 365 services and CRM Online cost?

Office 365 services and CRM Online requires premium add-on to be enabled. The pricing details page provides details of costs for ExpressRoute.

What regions is ExpressRoute for Office 365 supported in?

Refer to ExpressRoute partners and locations for more information on the list of partners and locations where ExpressRoute is supported.

Can I access Office 365 over the internet even if ExpressRoute was configured for my organization?

Yes. Office 365 service endpoints are reachable through the internet even though ExpressRoute has been configured for your network. If you are in a location that is configured to connect to Office 365 services through

ExpressRoute, you will connect through ExpressRoute.

Can Dynamics 365 for Operations (formerly known as Dynamics AX Online) be accessed over an ExpressRoute connection?

Yes. Dynamics 365 for Operations is hosted on Azure. You can enable Azure public peering on your ExpressRoute circuit to connect to it.

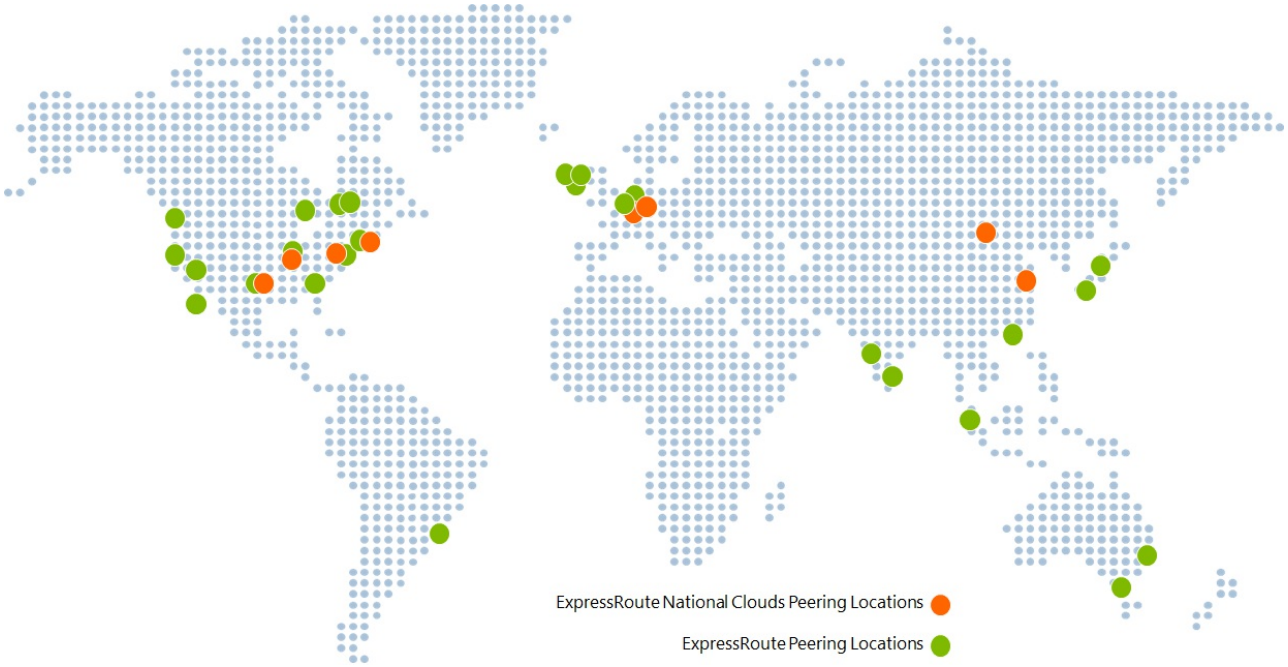
ExpressRoute partners and peering locations

1/17/2017 • 5 min to read • Edit on GitHub

The tables in this article provide information on ExpressRoute connectivity providers, ExpressRoute geographical coverage, Microsoft cloud services supported over ExpressRoute, and ExpressRoute System Integrators (SIs).

ExpressRoute connectivity providers

ExpressRoute is supported across all Azure regions and locations. The following map provides a list of Azure regions and ExpressRoute locations. ExpressRoute locations refer to those where Microsoft peers with several service providers.



You will have access to Azure services across all regions within a geopolitical region if you connected to at least one ExpressRoute location within the geopolitical region.

Azure regions to ExpressRoute locations within a geopolitical region.

The following table provides a map of Azure regions to ExpressRoute locations within a geopolitical region.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
North America	East US, West US, East US 2, West US 2, Central US, South Central US, North Central US, West Central US, Canada Central, Canada East	Atlanta, Chicago, Dallas, Las Vegas, Los Angeles, New York, Seattle, Silicon Valley, Washington DC, Montreal+, Quebec City+, Toronto
South America	Brazil South	Sao Paulo
Europe	North Europe, West Europe, UK West, UK South	Amsterdam, Dublin, London, Newport(Wales), Paris
Asia	East Asia, Southeast Asia	Hong Kong, Singapore
Japan	Japan West, Japan East	Osaka, Tokyo

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
Australia	Australia Southeast, Australia East	Melbourne, Sydney
India	India West, India Central, India South	Chennai, Mumbai
South Korea	Korea Central, Korea South	Busan, Seoul

Regions and geopolitical boundaries for national clouds

The table below provides information on regions and geopolitical boundaries for national clouds.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
US Government cloud	US Gov Iowa, US Gov Virginia, US DoD Central+, US DoD East+	Chicago, Dallas, New York, Silicon Valley+, Washington DC
China	China North, China East	Beijing, Shanghai
Germany	Germany Central, Germany East	Berlin, Frankfurt

Connectivity across geopolitical regions is not supported on the standard ExpressRoute SKU. You will need to enable the ExpressRoute premium add-on to support global connectivity. Connectivity to national cloud environments is not supported. You can work with your connectivity provider if such a need arises.

Connectivity provider locations

Production Azure

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
AARNet	Supported	Supported	Melbourne, Sydney
Aryaka Networks	Supported	Supported	Amsterdam, Dallas, Silicon Valley, Singapore, Tokyo, Washington DC
AT&T NetBond	Supported	Supported	Amsterdam, Chicago, Dallas, London, Silicon Valley, Singapore, Sydney, Washington DC
Bell Canada	Supported	Supported	Toronto
British Telecom	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
CenturyLink	Coming soon	Coming soon	Silicon Valley
China Telecom Global	Supported	Not Supported	Hong Kong
Cologix	Supported	Supported	Dallas, Montreal, Toronto

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
Colt	Supported	Supported	Amsterdam, Dublin, London, Tokyo
Comcast	Supported	Supported	Chicago, Silicon Valley, Washington DC
Console	Supported	Supported	Silicon Valley
CoreSite	Supported	Supported	Los Angeles , New York
Equinix	Supported	Supported	Amsterdam, Atlanta, Chicago, Dallas, Hong Kong, London, Los Angeles, Melbourne, New York, Osaka, Paris+, Sao Paulo, Seattle, Silicon Valley, Singapore, Sydney, Tokyo, Toronto, Washington DC
euNetworks	Supported	Supported	Amsterdam
GÉANT	Supported	Supported	Amsterdam
Internet Initiative Japan Inc. - IIJ	Supported	Supported	Osaka, Tokyo
InterCloud	Supported	Supported	Amsterdam, London, Singapore, Washington DC
Internet Solutions - Cloud Connect	Supported	Supported	Amsterdam, London
Interxion	Supported	Supported	Amsterdam, London, Paris
Jisc	Supported	Supported	London
KPN	Supported	Supported	Amsterdam
Level 3 Communications	Supported	Supported	Amsterdam, Chicago, Dallas, Las Vegas+, London, Seattle, Silicon Valley, Singapore, Washington DC
Megaport	Supported	Supported	Dallas, Hong Kong, Las Vegas, Los Angeles, Melbourne, New York, Seattle, Singapore, Sydney, Toronto, Washington DC
MTN	Supported	Supported	London
Next Generation Data	Supported	Supported	Newport(Wales)
NEXTDC	Supported	Supported	Melbourne, Sydney

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
NTT Communications	Supported	Supported	London, Los Angeles, Osaka, Singapore, Tokyo, Washington DC
Orange	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Washington DC
PCCW Global Limited	Supported	Supported	Hong Kong
SIFY	Supported	Supported	Chennai
SingTel	Supported	Supported	Singapore
Softbank	Supported	Supported	Osaka, Tokyo
Tata Communications	Supported	Supported	Amsterdam, Chennai, Hong Kong, London, Mumbai, Silicon Valley, Singapore, Washington DC
TeleCity Group	Supported	Supported	Amsterdam, Dublin, London
Telefonica	Supported	Supported	Sao Paulo
Telenor	Supported	Supported	Amsterdam, London
Telstra Corporation	Supported	Supported	Melbourne, Sydney
Verizon	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
Vodafone	Supported	Not Supported	London
Zayo Group	Supported	Supported	Chicago, Los Angeles, New York, Silicon Valley, Toronto, Washington DC

+ denotes coming soon

National cloud environment

US Government cloud

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
AT&T NetBond	Supported	Supported	Chicago, Washington DC
Equinix	Supported	Supported	Chicago, Dallas, New York, Silicon Valley+, Washington DC

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Level 3 Communications	Supported	Supported	Chicago, New York+, Washington DC
Megaport	Supported	Supported	Dallas
Verizon	Supported	Supported	Chicago, Dallas, New York, Washington DC

China

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
China Telecom	Supported	Not Supported	Beijing, Shanghai

To learn more, see ExpressRoute in China.

Germany

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Colt	Supported	Not Supported	Berlin+, Frankfurt
Equinix	Supported	Not Supported	Frankfurt
e-shelter	Supported	Not Supported	Berlin
Interxion	Supported	Not Supported	Frankfurt
Megaport	Supported	Not Supported	Berlin+

Connectivity through service providers not listed

If your connectivity provider is not listed in previous sections, you can still create a connection.

- Check with your connectivity provider to see if they are connected to any of the exchanges in the table above. You can check the following links to gather more information about services offered by exchange providers. Several connectivity providers are already connected to Ethernet exchanges.
  - Cologix
  - CoreSite
  - Equinix Cloud Exchange
  - Interxion
  - Megaport
  - NextDC
  - TeleCity CloudIX
- Have your connectivity provider extend your network to the peering location of choice.
  - Ensure that your connectivity provider extends your connectivity in a highly available manner so that there are no single points of failure.
- Order an ExpressRoute circuit with the exchange as your connectivity provider to connect to Microsoft.
  - Follow steps in Create an ExpressRoute circuit to set up connectivity.

CONNECTIVITY PROVIDER	EXCHANGE	LOCATIONS
1CLOUDSTAR	Equinix	Singapore
Arteria Networks Corporation	Equinix	Tokyo
Alaska Communications	Equinix	Seattle
Eurofiber	Equinix	Amsterdam
Exponential E	Equinix	London
HSO	Equinix	London, Slough
Lightower	Equinix	New York, Washington DC
Macquarie Telecom Group	Megaport	Sydney
Masergy	Equinix	Washington DC
Nianet	Telecity	Amsterdam, Frankfurt
Transtelco	Equinix	Dallas, Los Angeles
QSC AG	Interxion	Frankfurt
Windstream	Equinix	Chicago, Silicon Valley, Washington DC
XO Communications	Equinix	Silicon Valley
Zertia	Level 3	Madrid

ExpressRoute system integrators

Enabling private connectivity to fit your needs can be challenging, based on the scale of your network. You can work with any of the system integrators listed in the following table to assist you with onboarding to ExpressRoute.

SYSTEM INTEGRATOR	CONTINENT
Avanade Inc.	Asia, Europe, US
Dotnet Solutions	Europe
Equinix Professional Services	US
IT Consultancy	Australia
MSG Services	Europe (Germany)
Nelite	Europe
OneAs1a	Asia

SYSTEM INTEGRATOR	CONTINENT
Perficient	US
Project Leadership	US

Next steps

- For more information about ExpressRoute, see the ExpressRoute FAQ.
- Ensure that all prerequisites are met. See ExpressRoute prerequisites.

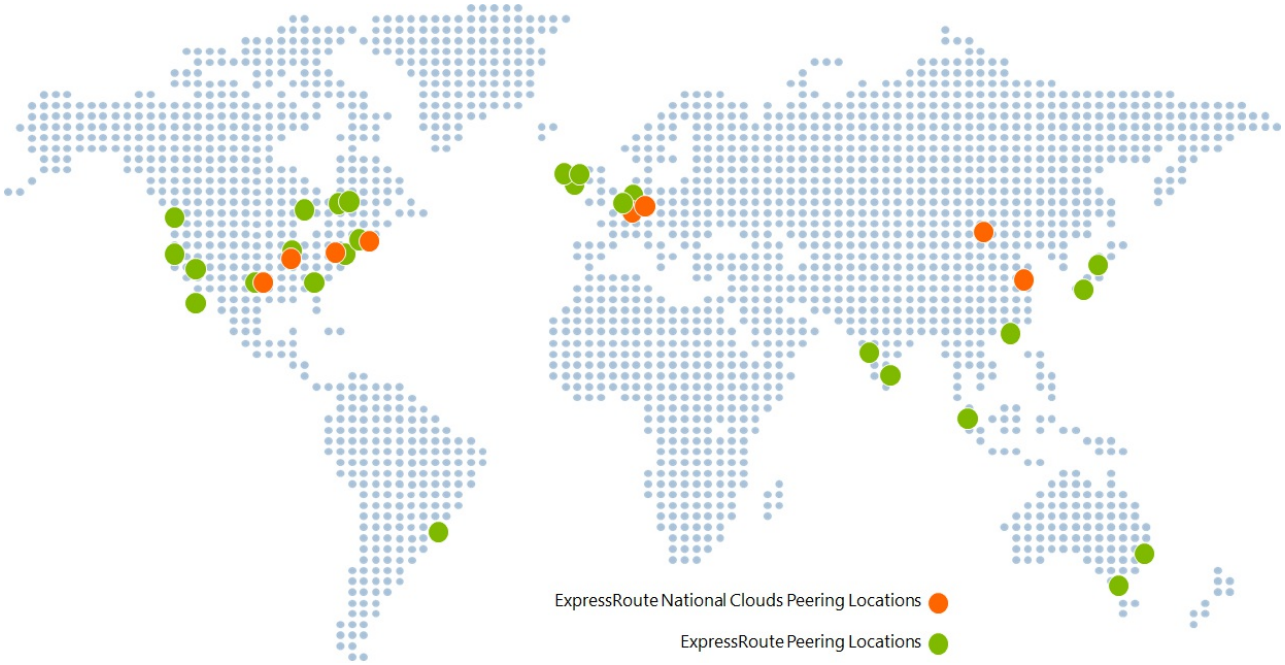
ExpressRoute partners and peering locations

1/17/2017 • 4 min to read • Edit on GitHub

The tables in this article provide information on ExpressRoute connectivity providers, ExpressRoute geographical coverage, Microsoft cloud services supported over ExpressRoute, and ExpressRoute System Integrators (SIs).

ExpressRoute connectivity providers

ExpressRoute is supported across all Azure regions and locations. The following map provides a list of Azure regions and ExpressRoute locations. ExpressRoute locations refer to those where Microsoft peers with several service providers.



You will have access to Azure services across all regions within a geopolitical region if you connected to at least one ExpressRoute location within the geopolitical region.

Azure regions to ExpressRoute locations within a geopolitical region

The following table provides a map of Azure regions to ExpressRoute locations within a geopolitical region.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
North America	East US, West US, East US 2, West US 2, Central US, South Central US, North Central US, West Central US, Canada Central, Canada East	Atlanta, Chicago, Dallas, Las Vegas, Los Angeles, New York, Seattle, Silicon Valley, Washington DC, Montreal+, Quebec City+, Toronto
South America	Brazil South	Sao Paulo
Europe	North Europe, West Europe, UK West, UK South	Amsterdam, Dublin, London, Newport(Wales), Paris
Asia	East Asia, Southeast Asia	Hong Kong, Singapore
Japan	Japan West, Japan East	Osaka, Tokyo

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
Australia	Australia Southeast, Australia East	Melbourne, Sydney
India	India West, India Central, India South	Chennai, Mumbai
South Korea	Korea Central, Korea South	Busan, Seoul

Regions and geopolitical boundaries for national clouds

The table below provides information on regions and geopolitical boundaries for national clouds.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
US Government cloud	US Gov Iowa, US Gov Virginia, US DoD Central+, US DoD East+	Chicago, Dallas, New York, Silicon Valley+, Washington DC
China	China North, China East	Beijing, Shanghai
Germany	Germany Central, Germany East	Berlin, Frankfurt

Connectivity across geopolitical regions is not supported on the standard ExpressRoute SKU. You will need to enable the ExpressRoute premium add-on to support global connectivity. Connectivity to national cloud environments is not supported. You can work with your connectivity provider if such a need arises.

Connectivity provider locations

Production Azure

LOCATION	SERVICE PROVIDERS
Amsterdam	Aryaka Networks, AT&T NetBond, British Telecom, Colt, Equinix, euNetworks, GÉANT, InterCloud, Internet Solutions - Cloud Connect, Interxion, KPN, Level 3 Communications, Orange, Tata Communications, TeleCity Group, Telenor, Verizon
Atlanta	Equinix
Chennai	SIFY, Tata Communications
Chicago	AT&T NetBond, Comcast, Equinix, Level 3 Communications, Zayo Group
Dallas	Aryaka Networks, AT&T NetBond, Cologix, Equinix, Level 3 Communications, Megaport
Dublin	Colt, Telecity Group
Hong Kong	British Telecom, China Telecom Global, Equinix, Megaport, Orange, PCCW Global Limited, Tata Communications, Verizon

LOCATION	SERVICE PROVIDERS
London	AT&T NetBond, British Telecom, Colt, Equinix, InterCloud, Internet Solutions - Cloud Connect, Interxion, Jisc, Level 3 Communications, MTN, NTT Communications, Orange, Tata Communications, Telecity Group, Telenor, Verizon, Vodafone
Las Vegas	Level 3 Communications+, Megaport
Los Angeles	CoreSite, Equinix, Megaport, NTT, Zayo Group
Melbourne	AARNet, Equinix, Megaport, NEXTDC, Telstra Corporation
New York	Coresite, Equinix, Megaport, Zayo Group
Newport(Wales)	Next Generation Data
Montreal	Cologix
Mumbai	Tata Communications
Osaka	Equinix, Internet Initiative Japan Inc. - IIJ, NTT Communications, Softbank
Paris	Interxion, Equinix+
Sao Paulo	Equinix, Telefonica
Seattle	Equinix, Level 3 Communications, Megaport
Silicon Valley	Aryaka Networks, AT&T NetBond, British Telecom, CenturyLink+, Comcast, Console, Equinix, Level 3 Communications, Orange, Tata Communications, Verizon, Zayo Group
Singapore	Aryaka Networks, AT&T NetBond, British Telecom, Equinix, InterCloud, Level 3 Communications, Megaport, NTT Communications, Orange, SingTel, Tata Communications, Verizon
Sydney	AARNet, AT&T NetBond, British Telecom, Equinix, Megaport, NEXTDC, Orange, Telstra Corporation, Verizon
Tokyo	Aryaka Networks, British Telecom, Colt, Equinix, Internet Initiative Japan Inc. - IIJ, NTT Communications, Softbank, Verizon
Toronto	Bell Canada, Cologix, Equinix, Megaport, Zayo Group
Washington DC	Aryaka Networks, AT&T NetBond, British Telecom, Comcast, Equinix, InterCloud, Level 3 Communications, Megaport, NTT Communications, Orange, Tata Communications, Verizon, Zayo Group

+ denotes coming soon

National cloud environments

US Government cloud

LOCATION	SERVICE PROVIDERS
Chicago	AT&T NetBond, Equinix, Level 3 Communications, Verizon
Dallas	Equinix, Megaport, Verizon
New York	Equinix, Level 3 Communications+, Verizon
Silicon Valley	Equinix+
Washington DC	AT&T NetBond, Equinix, Level 3 Communications, Verizon

China

LOCATION	SERVICE PROVIDERS
Beijing	China Telecom
Shanghai	China Telecom

To learn more, see ExpressRoute in China

Germany

LOCATION	SERVICE PROVIDERS
Berlin	Colt+, e-shelter, Megaport+
Frankfurt	Colt, Equinix, Interxion

Connectivity through service providers not listed

If your connectivity provider is not listed in previous sections, you can still create a connection.

- Check with your connectivity provider to see if they are connected to any of the exchanges in the table above. You can check the following links to gather more information about services offered by exchange providers. Several connectivity providers are already connected to Ethernet exchanges.
  - Cologix
  - CoreSite
  - Equinix Cloud Exchange
  - InterXion
  - NextDC
  - Megaport
  - TeleCity CloudIX
- Have your connectivity provider extend your network to the peering location of choice.
  - Ensure that your connectivity provider extends your connectivity in a highly available manner so that there are no single points of failure.
- Order an ExpressRoute circuit with the exchange as your connectivity provider to connect to Microsoft.
  - Follow steps in Create an ExpressRoute circuit to set up connectivity.

LOCATION	EXCHANGE	CONNECTIVITY PROVIDERS
Amsterdam	Equinix	Eurofiber
London	Equinix	Exponential E, HSO
New York	Equinix	Lightower
Seattle	Equinix	Alaska Communications
Silicon Valley	Equinix	XO Communications
Singapore	Equinix	1CLOUDSTAR
Sydney	Megaport	Macquarie Telecom Group
Tokyo	Equinix	ARTERIA Networks Corporation
Washington DC	Equinix	Lightower, Masergy

ExpressRoute system integrators

Enabling private connectivity to fit your needs can be challenging, based on the scale of your network. You can work with any of the system integrators listed in the following table to assist you with onboarding to ExpressRoute.

CONTINENT	SYSTEM INTEGRATORS
Asia	Avanade Inc., OneAs1a
Australia	IT Consultancy
Europe	Avanade Inc., Dotnet Solutions , MSG Services, Nelite
US	Avanade Inc., Equinix Professional Services, Perficient, Project Leadership

Next steps

- For more information about ExpressRoute, see the ExpressRoute FAQ.
- Ensure that all prerequisites are met. See ExpressRoute prerequisites.

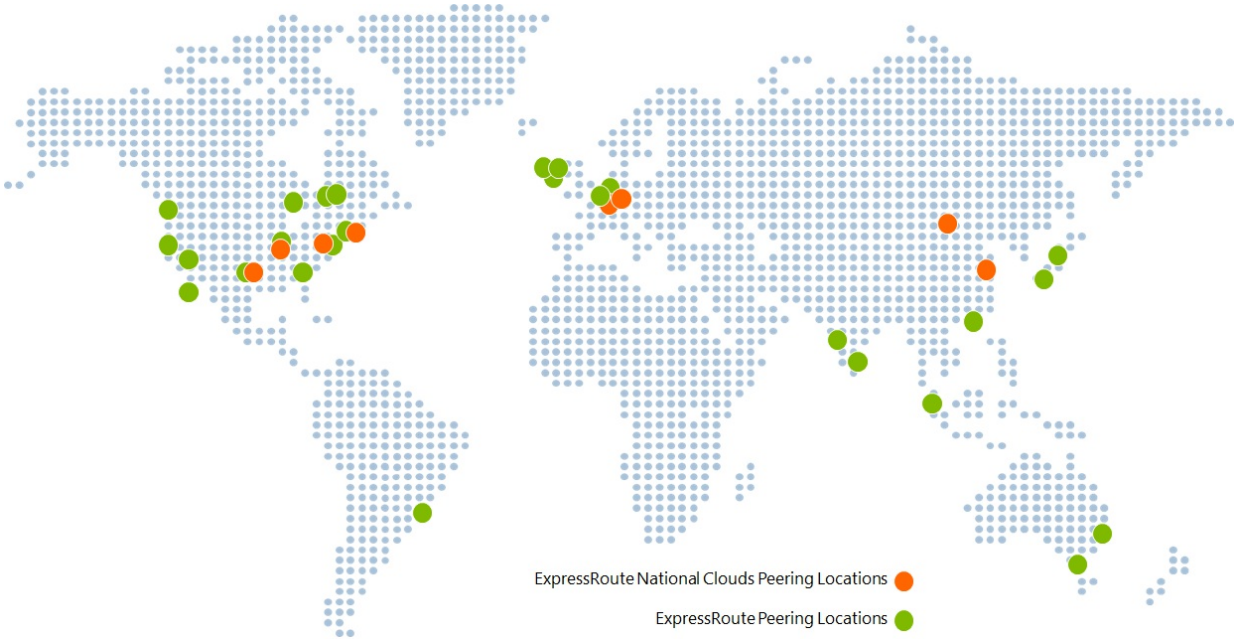
ExpressRoute partners and peering locations

1/17/2017 • 5 min to read • Edit on GitHub

The tables in this article provide information on ExpressRoute connectivity providers, ExpressRoute geographical coverage, Microsoft cloud services supported over ExpressRoute, and ExpressRoute System Integrators (SIs).

ExpressRoute connectivity providers

ExpressRoute is supported across all Azure regions and locations. The following map provides a list of Azure regions and ExpressRoute locations. ExpressRoute locations refer to those where Microsoft peers with several service providers.



You will have access to Azure services across all regions within a geopolitical region if you connected to at least one ExpressRoute location within the geopolitical region.

Azure regions to ExpressRoute locations within a geopolitical region.

The following table provides a map of Azure regions to ExpressRoute locations within a geopolitical region.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
North America	East US, West US, East US 2, West US 2, Central US, South Central US, North Central US, West Central US, Canada Central, Canada East	Atlanta, Chicago, Dallas, Las Vegas, Los Angeles, New York, Seattle, Silicon Valley, Washington DC, Montreal+, Quebec City+, Toronto
South America	Brazil South	Sao Paulo
Europe	North Europe, West Europe, UK West, UK South	Amsterdam, Dublin, London, Newport(Wales), Paris
Asia	East Asia, Southeast Asia	Hong Kong, Singapore

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
Japan	Japan West, Japan East	Osaka, Tokyo
Australia	Australia Southeast, Australia East	Melbourne, Sydney
India	India West, India Central, India South	Chennai, Mumbai
South Korea	Korea Central, Korea South	Busan, Seoul

Regions and geopolitical boundaries for national clouds

The table below provides information on regions and geopolitical boundaries for national clouds.

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
US Government cloud	US Gov Iowa, US Gov Virginia, US DoD Central+, US DoD East+	Chicago, Dallas, New York, Silicon Valley+, Washington DC
China	China North, China East	Beijing, Shanghai
Germany	Germany Central, Germany East	Berlin, Frankfurt

Connectivity across geopolitical regions is not supported on the standard ExpressRoute SKU. You will need to enable the ExpressRoute premium add-on to support global connectivity. Connectivity to national cloud environments is not supported. You can work with your connectivity provider if such a need arises.

Connectivity provider locations

Production Azure

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
AARNet	Supported	Supported	Melbourne, Sydney
Aryaka Networks	Supported	Supported	Amsterdam, Dallas, Silicon Valley, Singapore, Tokyo, Washington DC
AT&T NetBond	Supported	Supported	Amsterdam, Chicago, Dallas, London, Silicon Valley, Singapore, Sydney, Washington DC
Bell Canada	Supported	Supported	Toronto
British Telecom	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
CenturyLink	Coming soon	Coming soon	Silicon Valley
China Telecom Global	Supported	Not Supported	Hong Kong

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
Cologix	Supported	Supported	Dallas, Montreal, Toronto
Colt	Supported	Supported	Amsterdam, Dublin, London, Tokyo
Comcast	Supported	Supported	Chicago, Silicon Valley, Washington DC
Console	Supported	Supported	Silicon Valley
CoreSite	Supported	Supported	Los Angeles , New York
Equinix	Supported	Supported	Amsterdam, Atlanta, Chicago, Dallas, Hong Kong, London, Los Angeles, Melbourne, New York, Osaka, Paris+, Sao Paulo, Seattle, Silicon Valley, Singapore, Sydney, Tokyo, Toronto, Washington DC
euNetworks	Supported	Supported	Amsterdam
GÉANT	Supported	Supported	Amsterdam
Internet Initiative Japan Inc. - IIJ	Supported	Supported	Osaka, Tokyo
InterCloud	Supported	Supported	Amsterdam, London, Singapore, Washington DC
Internet Solutions - Cloud Connect	Supported	Supported	Amsterdam, London
Interxion	Supported	Supported	Amsterdam, London, Paris
Jisc	Supported	Supported	London
KPN	Supported	Supported	Amsterdam
Level 3 Communications	Supported	Supported	Amsterdam, Chicago, Dallas, Las Vegas+, London, Seattle, Silicon Valley, Singapore, Washington DC
Megaport	Supported	Supported	Dallas, Hong Kong, Las Vegas, Los Angeles, Melbourne, New York, Seattle, Singapore, Sydney, Toronto, Washington DC
MTN	Supported	Supported	London
Next Generation Data	Supported	Supported	Newport(Wales)

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365 AND CRM ONLINE	LOCATIONS
NEXTDC	Supported	Supported	Melbourne, Sydney
NTT Communications	Supported	Supported	London, Los Angeles, Osaka, Singapore, Tokyo, Washington DC
Orange	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Washington DC
PCCW Global Limited	Supported	Supported	Hong Kong
SIFY	Supported	Supported	Chennai
SingTel	Supported	Supported	Singapore
Softbank	Supported	Supported	Osaka, Tokyo
Tata Communications	Supported	Supported	Amsterdam, Chennai, Hong Kong, London, Mumbai, Silicon Valley, Singapore, Washington DC
TeleCity Group	Supported	Supported	Amsterdam, Dublin, London
Telefonica	Supported	Supported	Sao Paulo
Telenor	Supported	Supported	Amsterdam, London
Telstra Corporation	Supported	Supported	Melbourne, Sydney
Verizon	Supported	Supported	Amsterdam, Hong Kong, London, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
Vodafone	Supported	Not Supported	London
Zayo Group	Supported	Supported	Chicago, Los Angeles, New York, Silicon Valley, Toronto, Washington DC

+ denotes coming soon

National cloud environment

US Government cloud

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
AT&T NetBond	Supported	Supported	Chicago, Washington DC

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Equinix	Supported	Supported	Chicago, Dallas, New York, Silicon Valley+, Washington DC
Level 3 Communications	Supported	Supported	Chicago, New York+, Washington DC
Megaport	Supported	Supported	Dallas
Verizon	Supported	Supported	Chicago, Dallas, New York, Washington DC

China

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
China Telecom	Supported	Not Supported	Beijing, Shanghai

To learn more, see ExpressRoute in China.

Germany

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Colt	Supported	Not Supported	Berlin+, Frankfurt
Equinix	Supported	Not Supported	Frankfurt
e-shelter	Supported	Not Supported	Berlin
Interxion	Supported	Not Supported	Frankfurt
Megaport	Supported	Not Supported	Berlin+

Connectivity through service providers not listed

If your connectivity provider is not listed in previous sections, you can still create a connection.

- Check with your connectivity provider to see if they are connected to any of the exchanges in the table above. You can check the following links to gather more information about services offered by exchange providers. Several connectivity providers are already connected to Ethernet exchanges.
  - Cologix
  - CoreSite
  - Equinix Cloud Exchange
  - Interxion
  - Megaport
  - NextDC
  - TeleCity CloudIX
- Have your connectivity provider extend your network to the peering location of choice.
  - Ensure that your connectivity provider extends your connectivity in a highly available manner so that there are no single points of failure.

- Order an ExpressRoute circuit with the exchange as your connectivity provider to connect to Microsoft.
  - Follow steps in Create an ExpressRoute circuit to set up connectivity.

CONNECTIVITY PROVIDER	EXCHANGE	LOCATIONS
1CLOUDSTAR	Equinix	Singapore
Arteria Networks Corporation	Equinix	Tokyo
Alaska Communications	Equinix	Seattle
Eurofiber	Equinix	Amsterdam
Exponential E	Equinix	London
HSO	Equinix	London, Slough
Lightower	Equinix	New York, Washington DC
Macquarie Telecom Group	Megaport	Sydney
Masergy	Equinix	Washington DC
Nianet	Telecity	Amsterdam, Frankfurt
Transtelco	Equinix	Dallas, Los Angeles
QSC AG	Interxion	Frankfurt
Windstream	Equinix	Chicago, Silicon Valley, Washington DC
XO Communications	Equinix	Silicon Valley
Zertia	Level 3	Madrid

ExpressRoute system integrators

Enabling private connectivity to fit your needs can be challenging, based on the scale of your network. You can work with any of the system integrators listed in the following table to assist you with onboarding to ExpressRoute.

SYSTEM INTEGRATOR	CONTINENT
Avanade Inc.	Asia, Europe, US
Dotnet Solutions	Europe
Equinix Professional Services	US
IT Consultancy	Australia

SYSTEM INTEGRATOR	CONTINENT
MSG Services	Europe (Germany)
Nelite	Europe
OneAs1a	Asia
Perficient	US
Project Leadership	US

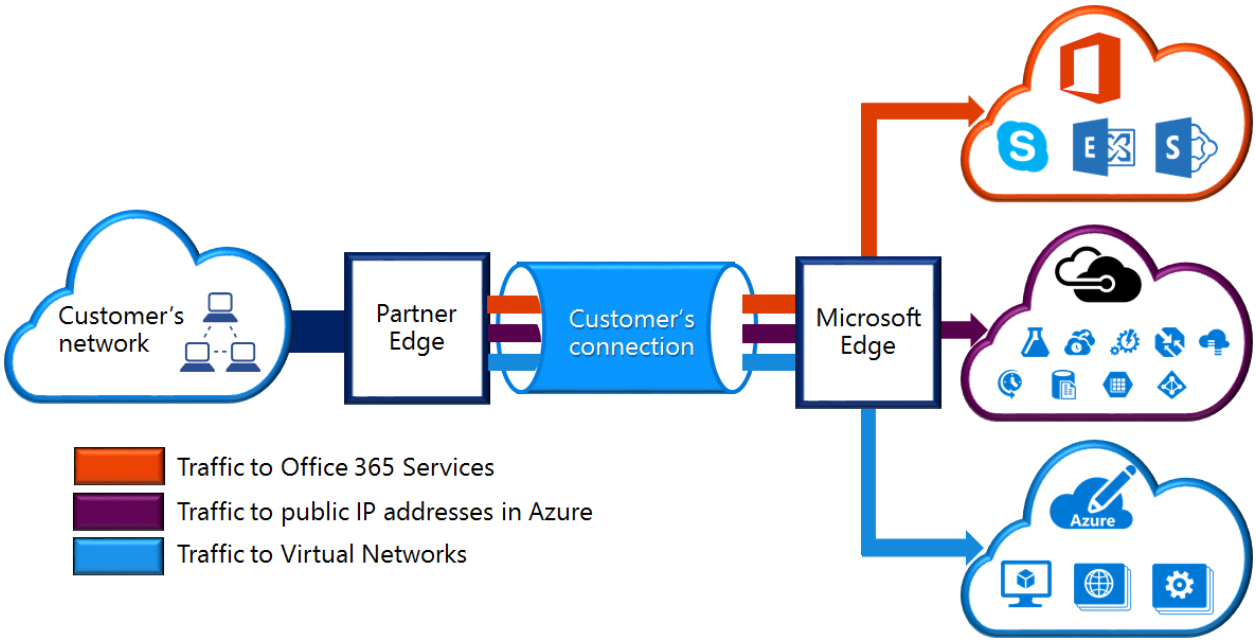
Next steps

- For more information about ExpressRoute, see the ExpressRoute FAQ.
- Ensure that all prerequisites are met. See ExpressRoute prerequisites.

ExpressRoute circuits and routing domains

1/17/2017 • 6 min to read • Edit on GitHub

You must order an *ExpressRoute circuit* to connect your on-premises infrastructure to Microsoft through a connectivity provider. The figure below provides a logical representation of connectivity between your WAN and Microsoft.



ExpressRoute circuits

An *ExpressRoute circuit* represents a logical connection between your on-premises infrastructure and Microsoft cloud services through a connectivity provider. You can order multiple ExpressRoute circuits. Each circuit can be in the same or different regions, and can be connected to their premises through different connectivity providers.

ExpressRoute circuits do not map to any physical entities. A circuit is uniquely identified by a standard GUID called as a service key (s-key). The service key is the only piece of information exchanged between Microsoft, the connectivity provider, and you. The s-key is not a secret for security purposes. There is a 1:1 mapping between an ExpressRoute circuit and the s-key.

An ExpressRoute circuit can have up to three independent peerings: Azure public, Azure private, and Microsoft. Each peering is a pair of independent BGP sessions each of them configured redundantly for high availability. There is a 1:N (1 <= N <= 3) mapping between an ExpressRoute circuit and routing domains. An ExpressRoute circuit can have any one, two, or all three peerings enabled per ExpressRoute circuit.

Each circuit has a fixed bandwidth (50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 10 Gbps) and is mapped to a connectivity provider and a peering location. The bandwidth you select is be shared across all the peerings for the circuit.

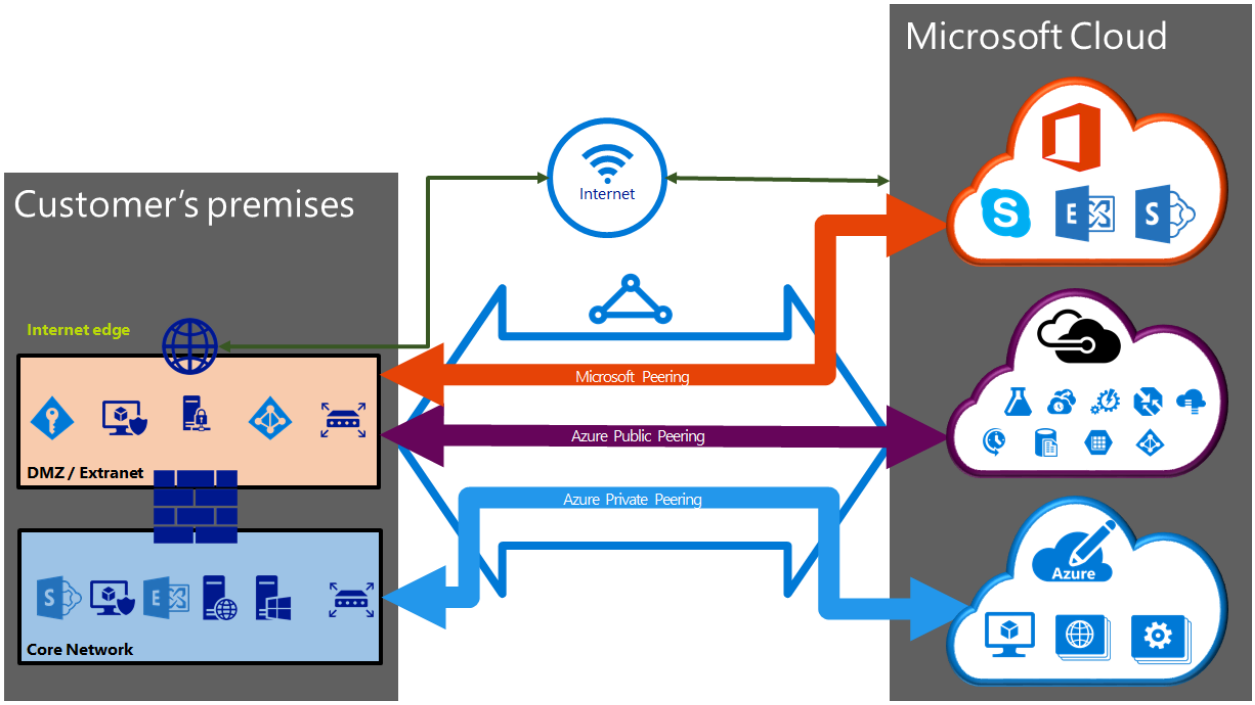
Quotas, limits, and limitations

Default quotas and limits apply for every ExpressRoute circuit. Refer to the Azure Subscription and Service Limits, Quotas, and Constraints page for up-to-date information on quotas.

ExpressRoute routing domains

An ExpressRoute circuit has multiple routing domains associated with it: Azure public, Azure private, and

Microsoft. Each of the routing domains is configured identically on a pair of routers (in active-active or load sharing configuration) for high availability. Azure services are categorized as *Azure public* and *Azure private* to represent the IP addressing schemes.



Private peering

Azure compute services, namely virtual machines (IaaS) and cloud services (PaaS), that are deployed within a virtual network can be connected through the private peering domain. The private peering domain is considered to be a trusted extension of your core network into Microsoft Azure. You can set up bi-directional connectivity between your core network and Azure virtual networks (VNets). This peering lets you connect to virtual machines and cloud services directly on their private IP addresses.

You can connect more than one virtual network to the private peering domain. Review the FAQ page for information on limits and limitations. You can visit the Azure Subscription and Service Limits, Quotas, and Constraints page for up-to-date information on limits. Refer to the Routing page for detailed information on routing configuration.

Public peering

Services such as Azure Storage, SQL databases, and Websites are offered on public IP addresses. You can privately connect to services hosted on public IP addresses, including VIPs of your cloud services, through the public peering routing domain. You can connect the public peering domain to your DMZ and connect to all Azure services on their public IP addresses from your WAN without having to connect through the internet.

Connectivity is always initiated from your WAN to Microsoft Azure services. Microsoft Azure services will not be able to initiate connections into your network through this routing domain. Once public peering is enabled, you will be able to connect to all Azure services. We do not allow you to selectively pick services for which we advertise routes to. You can review the list of prefixes we advertise to you through this peering on the Microsoft Azure Datacenter IP Ranges page. The page is updated weekly.

You can define custom route filters within your network to consume only the routes you need. Refer to the Routing page for detailed information on routing configuration. You can define custom route filters within your network to consume only the routes you need.

See the FAQ page for more information on services supported through the public peering routing domain.

Microsoft peering

ExpressRoute provides private network connectivity to Microsoft cloud services. Infrastructure and platform

services running in Azure often benefit by addressing network architecture and performance considerations. Therefore we recommend enterprises use ExpressRoute for Azure.

Software as a Service offerings, like Office 365 and Dynamics 365, were created to be accessed securely and reliably via the Internet. Therefore, we only recommend ExpressRoute for these applications in specific scenarios.

> **IMPORTANT**
>
> Using ExpressRoute to access Azure is **recommended** for all enterprises. For guidance on using ExpressRoute to access Office 365 visit http://aka.ms/ExpressRouteOffice365.

Connectivity to all other Microsoft online services (such as Office 365 services) will be through the Microsoft peering. We enable bi-directional connectivity between your WAN and Microsoft cloud services through the Microsoft peering routing domain. You must connect to Microsoft cloud services only over public IP addresses that are owned by you or your connectivity provider and you must adhere to all the defined rules. See the ExpressRoute prerequisites page for more information.

See the FAQ page for more information on services supported, costs, and configuration details. See the ExpressRoute Locations page for information on the list of connectivity providers offering Microsoft peering support.

Routing domain comparison

The table below compares the three routing domains.

	PRIVATE PEERING	PUBLIC PEERING	MICROSOFT PEERING
Max. # prefixes supported per peering	4000 by default, 10,000 with ExpressRoute Premium	200	200
IP address ranges supported	Any valid IPv4 address within your WAN.	Public IPv4 addresses owned by you or your connectivity provider.	Public IPv4 addresses owned by you or your connectivity provider.
AS Number requirements	Private and public AS numbers. You must own the public AS number if you choose to use one.	Private and public AS numbers. However, you must prove ownership of public IP addresses.	Private and public AS numbers. However, you must prove ownership of public IP addresses.
Routing Interface IP addresses	RFC1918 and public IP addresses	Public IP addresses registered to you in routing registries.	Public IP addresses registered to you in routing registries.
MD5 Hash support	Yes	Yes	Yes

You can choose to enable one or more of the routing domains as part of their ExpressRoute circuit. You can choose to have all the routing domains put on the same VPN if you want to combine them into a single routing domain. You can also put them on different routing domains, similar to the diagram. The recommended configuration is that private peering is connected directly to the core network, and the public and Microsoft peering links are connected to your DMZ.

If you choose to have all three peering sessions, you must have three pairs of BGP sessions (one pair for each peering type). The BGP session pairs provide a highly available link. If you are connecting through layer 2 connectivity providers, you will be responsible for configuring and managing routing. You can learn more by reviewing the workflows for setting up ExpressRoute.

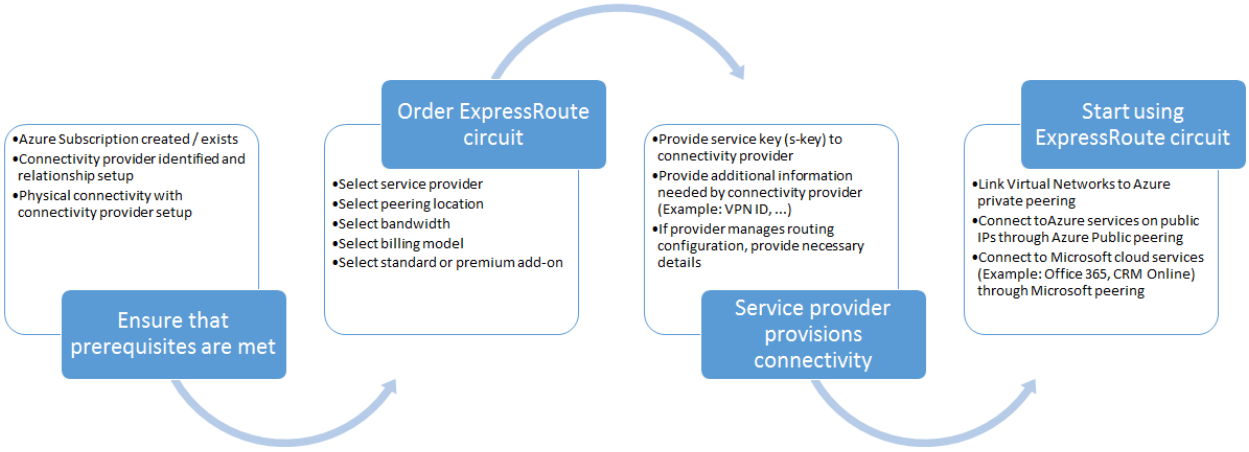
Next steps

- Find a service provider. See ExpressRoute service providers and locations.
- Ensure that all prerequisites are met. See ExpressRoute prerequisites.
- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit
  - Configure routing (circuit peerings)
  - Link a VNet to an ExpressRoute circuit

ExpressRoute workflows for circuit provisioning and circuit states

1/17/2017 • 4 min to read • Edit on GitHub

This page walks you through the service provisioning and routing configuration workflows at a high level.

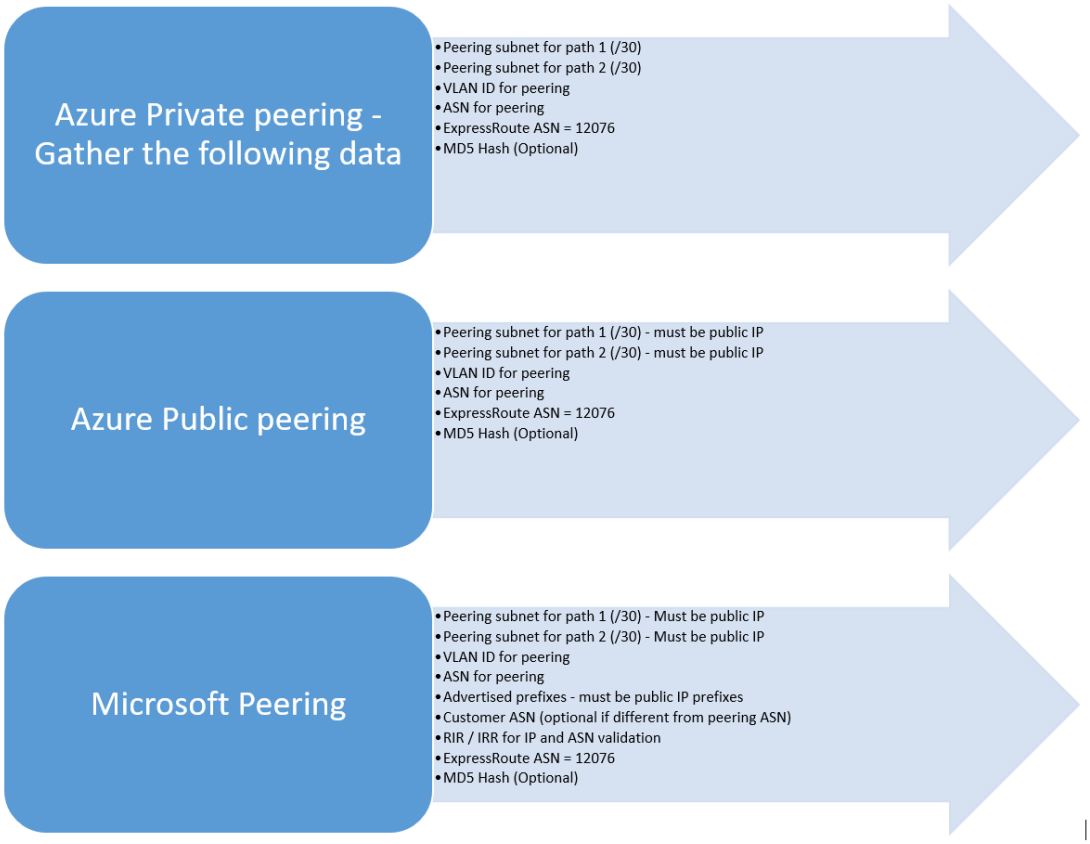


The following figure and corresponding steps show the tasks you must follow in order to have an ExpressRoute circuit provisioned end-to-end.

1. Use PowerShell to configure an ExpressRoute circuit. Follow the instructions in the Create ExpressRoute circuits article for more details.
2. Order connectivity from the service provider. This process varies. Contact your connectivity provider for more details about how to order connectivity.
3. Ensure that the circuit has been provisioned successfully by verifying the ExpressRoute circuit provisioning state through PowerShell.
4. Configure routing domains. If your connectivity provider manages Layer 3 for you, they will configure routing for your circuit. If your connectivity provider only offers Layer 2 services, you must configure routing per guidelines described in the routing requirements and routing configuration pages.

    - Enable Azure private peering - You must enable this peering to connect to VMs / cloud services deployed within virtual networks.
    - Enable Azure public peering - You must enable Azure public peering if you wish to connect to Azure services hosted on public IP addresses. This is a requirement to access Azure resources if you have chosen to enable default routing for Azure private peering.
    - Enable Microsoft peering - You must enable this to access Office 365 and CRM online services.

> **IMPORTANT**
> You must ensure that you use a separate proxy / edge to connect to Microsoft than the one you use for the Internet. Using the same edge for both ExpressRoute and the Internet will cause asymmetric routing and cause connectivity outages for your network.



5. Linking virtual networks to ExpressRoute circuits - You can link virtual networks to your ExpressRoute circuit. Follow instructions to link VNets to your circuit. These VNets can either be in the same Azure subscription as the ExpressRoute circuit, or can be in a different subscription.

ExpressRoute circuit provisioning states

Each ExpressRoute circuit has two states:

- Service provider provisioning state
- Status

Status represents Microsoft's provisioning state. This property is set to Enabled when you create an Expressroute circuit

The connectivity provider provisioning state represents the state on the connectivity provider's side. It can either be *NotProvisioned*, *Provisioning*, or *Provisioned*. The ExpressRoute circuit must be in Provisioned state for you to be able to use it.

Possible states of an ExpressRoute circuit

This section lists out the possible states for an ExpressRoute circuit.

At creation time

You will see the ExpressRoute circuit in the following state as soon as you run the PowerShell cmdlet to create the ExpressRoute circuit.

```
ServiceProviderProvisioningState : NotProvisioned
Status                           : Enabled
```

When connectivity provider is in the process of provisioning the circuit

You will see the ExpressRoute circuit in the following state as soon as you pass the service key to the connectivity provider and they have started the provisioning process.

```
ServiceProviderProvisioningState : Provisioning
Status                           : Enabled
```

When connectivity provider has completed the provisioning process

You will see the ExpressRoute circuit in the following state as soon as the connectivity provider has completed the provisioning process.

```
ServiceProviderProvisioningState : Provisioned
Status                           : Enabled
```

Provisioned and Enabled is the only state the circuit can be in for you to be able to use it. If you are using a Layer 2 provider, you can configure routing for your circuit only when it is in this state.

When connectivity provider is deprovisioning the circuit

If you requested the service provider to deprovision the ExpressRoute circuit, you will see the circuit set to the following state after the service provider has completed the deprovisioning process.

```
ServiceProviderProvisioningState : NotProvisioned
Status                           : Enabled
```

You can choose to re-enable it if needed, or run PowerShell cmdlets to delete the circuit.

> **IMPORTANT**
> If you run the PowerShell cmdlet to delete the circuit when the ServiceProviderProvisioningState is Provisioning or Provisioned the operation will fail. Please work with your connectivity provider to deprovision the ExpressRoute circuit first and then delete the circuit. Microsoft will continue to bill the circuit until you run the PowerShell cmdlet to delete the circuit.

Routing session configuration state

The BGP provisioning state lets you know if the BGP session has been enabled on the Microsoft edge. The state must be enabled for you to be able to use the peering.

It is important to check the BGP session state especially for Microsoft peering. In addition to the BGP provisioning state, there is another state called *advertised public prefixes state*. The advertised public prefixes state must be in *configured* state, both for the BGP session to be up and for your routing to work end-to-end.

If the advertised public prefix state is set to a *validation needed* state, the BGP session is not enabled, as the advertised prefixes did not match the AS number in any of the routing registries.

IMPORTANT

If the advertised public prefixes state is in *manual validation* state, you must open a support ticket with Microsoft support and provide evidence that you own the IP addresses advertised along with the associated Autonomous System number.

Next steps

- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit
  - Configure routing
  - Link a VNet to an ExpressRoute circuit

ExpressRoute prerequisites & checklist

1/17/2017 • 2 min to read • Edit on GitHub

To connect to Microsoft cloud services using ExpressRoute, you need to verify that the following requirements listed in the following sections have been met.

ExpressRoute provides private network connectivity to Microsoft cloud services. Infrastructure and platform services running in Azure often benefit by addressing network architecture and performance considerations. Therefore we recommend enterprises use ExpressRoute for Azure.

Software as a Service offerings, like Office 365 and Dynamics 365, were created to be accessed securely and reliably via the Internet. Therefore, we only recommend ExpressRoute for these applications in specific scenarios.

> **IMPORTANT**
> Using ExpressRoute to access Azure is **recommended** for all enterprises. For guidance on using ExpressRoute to access Office 365 visit http://aka.ms/ExpressRouteOffice365.

Azure account

- A valid and active Microsoft Azure account. This account is required to set up the ExpressRoute circuit. ExpressRoute circuits are resources within Azure subscriptions. An Azure subscription is a requirement even if connectivity is limited to non-Azure Microsoft cloud services, such as Office 365 services and CRM online.
- An active Office 365 subscription (if using Office 365 services). For more information, see the Office 365 specific requirements section of this article.

Connectivity provider

- You can work with an ExpressRoute connectivity partner to connect to the Microsoft cloud. You can set up a connection between your on-premises network and Microsoft in three ways.
- If your provider is not an ExpressRoute connectivity partner, you can still connect to the Microsoft cloud through a cloud exchange provider.

Network requirements

- **Redundant connectivity**: there is no redundancy requirement on physical connectivity between you and your provider. Microsoft does require redundant BGP sessions to be set up between Microsoft's routers and the peering routers, even when you have just one physical connection to a cloud exchange.
- **Routing**: depending on how you connect to the Microsoft Cloud, you or your provider need to set up and manage the BGP sessions for routing domains. Some Ethernet connectivity provider or cloud exchange provider may offer BGP management as a value-add service.
- **NAT**: Microsoft only accepts public IP addresses through Microsoft peering. If you are using private IP addresses in your on-premises network, you or your provider need to translate the private IP addresses to the public IP addresses using the NAT.
- **QoS**: Skype for Business has various services (for example; voice, video, text) that require differentiated QoS treatment. You and your provider should follow the QoS requirements.
- **Network Security**: consider network security when connecting to the Microsoft Cloud via ExpressRoute.

Office 365

If you plan to enable Office 365 on ExpressRoute, review the following documents for more information about Office 365 requirements.

- Overview of ExpressRoute for Office 365
- Routing with ExpressRoute for Office 365
- Office 365 URLs and IP address ranges
- Network planning and performance tuning for Office 365
- Network bandwidth calculators and tools
- Office 365 integration with on-premises environments
- ExpressRoute on Office 365 advanced training videos

CRM Online

If you plan to enable CRM Online on ExpressRoute, review the following documents for more information about CRM Online

- CRM Online URLs and IP address ranges

Next steps

- For more information about ExpressRoute, see the ExpressRoute FAQ.
- Find an ExpressRoute connectivity provider. See ExpressRoute partners and peering locations.
- Refer to requirements for Routing, NAT, and QoS.
- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit
  - Configure routing
  - Link a VNet to an ExpressRoute circuit

ExpressRoute routing requirements

1/17/2017 • 9 min to read • Edit on GitHub

To connect to Microsoft cloud services using ExpressRoute, you'll need to set up and manage routing. Some connectivity providers offer setting up and managing routing as a managed service. Check with your connectivity provider to see if they offer this service. If they don't, you must adhere to the following requirements.

Refer to the Circuits and routing domains article for a description of the routing sessions that need to be set up in to facilitate connectivity.

> **NOTE**
> Microsoft does not support any router redundancy protocols (e.g., HSRP, VRRP) for high availability configurations. We rely on a redundant pair of BGP sessions per peering for high availability.

IP addresses used for peerings

You need to reserve a few blocks of IP addresses to configure routing between your network and Microsoft's Enterprise edge (MSEEs) routers. This section provides a list of requirements and describes the rules regarding how these IP addresses must be acquired and used.

IP addresses used for Azure private peering

You can use either private IP addresses or public IP addresses to configure the peerings. The address range used for configuring routes must not overlap with address ranges used to create virtual networks in Azure.

- You must reserve a /29 subnet or two /30 subnets for routing interfaces.
- The subnets used for routing can be either private IP addresses or public IP addresses.
- The subnets must not conflict with the range reserved by the customer for use in the Microsoft cloud.
- If a /29 subnet is used, it will be split into two /30 subnets.
  - The first /30 subnet will be used for the primary link and the second /30 subnet will be used for the secondary link.
  - For each of the /30 subnets, you must use the first IP address of the /30 subnet on your router. Microsoft will use the second IP address of the /30 subnet to set up a BGP session.
  - You must set up both BGP sessions for our availability SLA to be valid.

Example for private peering

If you choose to use a.b.c.d/29 to set up the peering, it will be split into two /30 subnets. In the example below, we will look at how the a.b.c.d/29 subnet is used.

a.b.c.d/29 will be split to a.b.c.d/30 and a.b.c.d+4/30 and passed down to Microsoft through the provisioning APIs. You will use a.b.c.d+1 as the VRF IP for the Primary PE and Microsoft will consume a.b.c.d+2 as the VRF IP for the primary MSEE. You will use a.b.c.d+5 as the VRF IP for the secondary PE and Microsoft will use a.b.c.d+6 as the VRF IP for the secondary MSEE.

Consider a case where you select 192.168.100.128/29 to set up private peering. 192.168.100.128/29 includes addresses from 192.168.100.128 to 192.168.100.135, among which:

- 192.168.100.128/30 will be assigned to link1, with provider using 192.168.100.129 and Microsoft using 192.168.100.130.

- 192.168.100.132/30 will be assigned to link2, with provider using 192.168.100.133 and Microsoft using 192.168.100.134.

IP addresses used for Azure public and Microsoft peering

You must use public IP addresses that you own for setting up the BGP sessions. Microsoft must be able to verify the ownership of the IP addresses through Routing Internet Registries and Internet Routing Registries.

- You must use a unique /29 subnet or two /30 subnets to set up the BGP peering for each peering per ExpressRoute circuit (if you have more than one).
- If a /29 subnet is used, it will be split into two /30 subnets.
  - The first /30 subnet will be used for the primary link and the second /30 subnet will be used for the secondary link.
  - For each of the /30 subnets, you must use the first IP address of the /30 subnet on your router. Microsoft will use the second IP address of the /30 subnet to set up a BGP session.
  - You must set up both BGP sessions for our availability SLA to be valid.

Public IP address requirement

Private Peering

You can choose to use public or private IPv4 addresses for private peering. We provide end-to-end isolation of your traffic so overlapping of addresses with other customers is not possible in case of private peering. These addresses are not advertised to Internet.

Public Peering

The Azure public peering path enables you to connect to all services hosted in Azure over their public IP addresses. These include services listed in the ExpessRoute FAQ and any services hosted by ISVs on Microsoft Azure. Connectivity to Microsoft Azure services on public peering is always initiated from your network into the Microsoft network. You must use Public IP addresses for the traffic destined to Microsoft network.

Microsoft Peering

The Microsoft peering path lets you connect to Microsoft cloud services that are not supported through the Azure public peering path. The list of services includes Office 365 services, such as Exchange Online, SharePoint Online, Skype for Business, and CRM Online. Microsoft supports bi-directional connectivity on the Microsoft peering. Traffic destined to Microsoft cloud services must use valid public IPv4 addresses before they enter the Microsoft network.

Make sure that your IP address and AS number are registered to you in one of the registries listed below.

- ARIN
- APNIC
- AFRINIC
- LACNIC
- RIPENCC
- RADB
- ALTDB

> **IMPORTANT**
> Public IP addresses advertised to Microsoft over ExpressRoute must not be advertised to the Internet. This may break connectivity to other Microsoft services. However, Public IP addresses used by servers in your network that communicate with O365 endpoints within Microsoft may be advertised over ExpressRoute.

Dynamic route exchange

Routing exchange will be over eBGP protocol. EBGP sessions are established between the MSEEs and your routers. Authentication of BGP sessions is not a requirement. If required, an MD5 hash can be configured. See the Configure routing and Circuit provisioning workflows and circuit states for information about configuring BGP sessions.

Autonomous System numbers

Microsoft will use AS 12076 for Azure public, Azure private and Microsoft peering. We have reserved ASNs from 65515 to 65520 for internal use. Both 16 and 32 bit AS numbers are supported.

There are no requirements around data transfer symmetry. The forward and return paths may traverse different router pairs. Identical routes must be advertised from either sides across multiple circuit pairs belonging you. Route metrics are not required to be identical.

Route aggregation and prefix limits

We support up to 4000 prefixes advertised to us through the Azure private peering. This can be increased up to 10,000 prefixes if the ExpressRoute premium add-on is enabled. We accept up to 200 prefixes per BGP session for Azure public and Microsoft peering.

The BGP session will be dropped if the number of prefixes exceeds the limit. We will accept default routes on the private peering link only. Provider must filter out default route and private IP addresses (RFC 1918) from the Azure public and Microsoft peering paths.

Transit routing and cross-region routing

ExpressRoute cannot be configured as transit routers. You will have to rely on your connectivity provider for transit routing services.

Advertising default routes

Default routes are permitted only on Azure private peering sessions. In such a case, we will route all traffic from the associated virtual networks to your network. Advertising default routes into private peering will result in the internet path from Azure being blocked. You must rely on your corporate edge to route traffic from and to the internet for services hosted in Azure.

To enable connectivity to other Azure services and infrastructure services, you must make sure one of the following items is in place:

- Azure public peering is enabled to route traffic to public endpoints
- You use user-defined routing to allow internet connectivity for every subnet requiring Internet connectivity.

> **NOTE**
> Advertising default routes will break Windows and other VM license activation. Follow instructions here to work around this.

Support for BGP communities

This section provides an overview of how BGP communities will be used with ExpressRoute. Microsoft will advertise routes in the public and Microsoft peering paths with routes tagged with appropriate community values. The rationale for doing so and the details on community values are described below. Microsoft, however,

will not honor any community values tagged to routes advertised to Microsoft.

If you are connecting to Microsoft through ExpressRoute at any one peering location within a geopolitical region, you will have access to all Microsoft cloud services across all regions within the geopolitical boundary.

For example, if you connected to Microsoft in Amsterdam through ExpressRoute, you will have access to all Microsoft cloud services hosted in North Europe and West Europe.

Refer to the ExpressRoute partners and peering locations page for a detailed list of geopolitical regions, associated Azure regions, and corresponding ExpressRoute peering locations.

You can purchase more than one ExpressRoute circuit per geopolitical region. Having multiple connections offers you significant benefits on high availability due to geo-redundancy. In cases where you have multiple ExpressRoute circuits, you will receive the same set of prefixes advertised from Microsoft on the public peering and Microsoft peering paths. This means you will have multiple paths from your network into Microsoft. This can potentially cause sub-optimal routing decisions to be made within your network. As a result, you may experience sub-optimal connectivity experiences to different services. You can rely on the community values to make appropriate routing decisions to offer optimal routing to users.

MICROSOFT AZURE REGION	BGP COMMUNITY VALUE
North America	
East US	12076:51004
East US 2	12076:51005
West US	12076:51006
West US 2	12076:51026
West Central US	12076:51027
North Central US	12076:51007
South Central US	12076:51008
Central US	12076:51009
Canada Central	12076:51020
Canada East	12076:51021
South America	
Brazil South	12076:51014
Europe	
North Europe	12076:51003
West Europe	12076:51002
UK South	12076:51024

MICROSOFT AZURE REGION	BGP COMMUNITY VALUE
UK West	12076:51025
Asia Pacific	
East Asia	12076:51010
Southeast Asia	12076:51011
Japan	
Japan East	12076:51012
Japan West	12076:51013
Australia	
Australia East	12076:51015
Australia Southeast	12076:51016
India	
India South	12076:51019
India West	12076:51018
India Central	12076:51017

All routes advertised from Microsoft will be tagged with the appropriate community value.

> **IMPORTANT**
> Global prefixes will be tagged with an appropriate community value and will be advertised only when ExpressRoute premium add-on is enabled.

In addition to the above, Microsoft will also tag prefixes based on the service they belong to. This applies only to the Microsoft peering. The table below provides a mapping of service to BGP community value.

SERVICE	BGP COMMUNITY VALUE
Exchange Online	12076:5010
SharePoint Online	12076:5020
Skype For Business Online	12076:5030
CRM Online	12076:5040
Other Office 365 Online services	12076:5100

> **NOTE**
> Microsoft does not honor any BGP community values that you set on the routes advertised to Microsoft.

BGP Community support in National Clouds (Preview)

NATIONAL CLOUDS AZURE REGION	BGP COMMUNITY VALUE
US Government	
US Gov Iowa	12076:51109
US Gov Virginia	12076:51105

SERVICE IN NATIONAL CLOUDS	BGP COMMUNITY VALUE
US Government	
Exchange Online	12076:5110
SharePoint Online	12076:5120
Skype For Business Online	12076:5130
CRM Online	12076:5140
Other Office 365 Online services	12076:5200

Next steps

- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit for the classic deployment model or Create and modify an ExpressRoute circuit using Azure Resource Manager
  - Configure routing for the classic deployment model or Configure routing for the Resource Manager deployment model
  - Link a classic VNet to an ExpressRoute circuit or Link a Resource Manager VNet to an ExpressRoute circuit

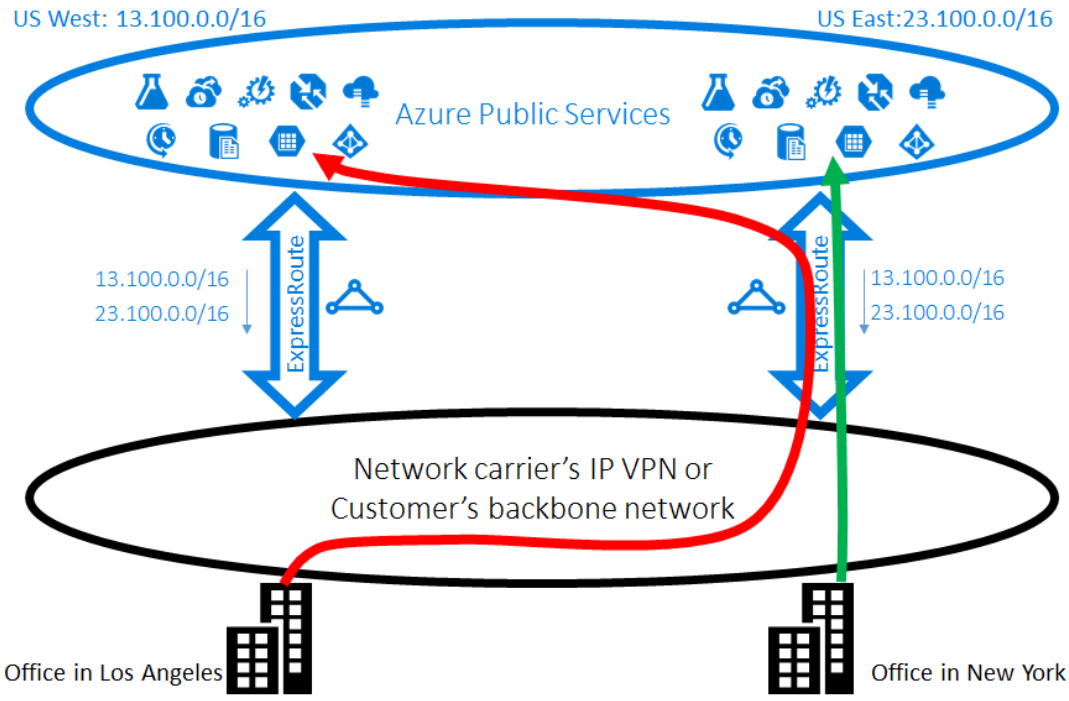
Optimize ExpressRoute Routing

1/17/2017 • 5 min to read • Edit on GitHub

When you have multiple ExpressRoute circuits, you have more than one path to connect to Microsoft. As a result, suboptimal routing may happen - that is, your traffic may take a longer path to reach Microsoft, and Microsoft to your network. The longer the network path, the higher the latency. Latency has direct impact on application performance and user experience. This article will illustrate this problem and explain how to optimize routing using the standard routing technologies.

Suboptimal routing case 1

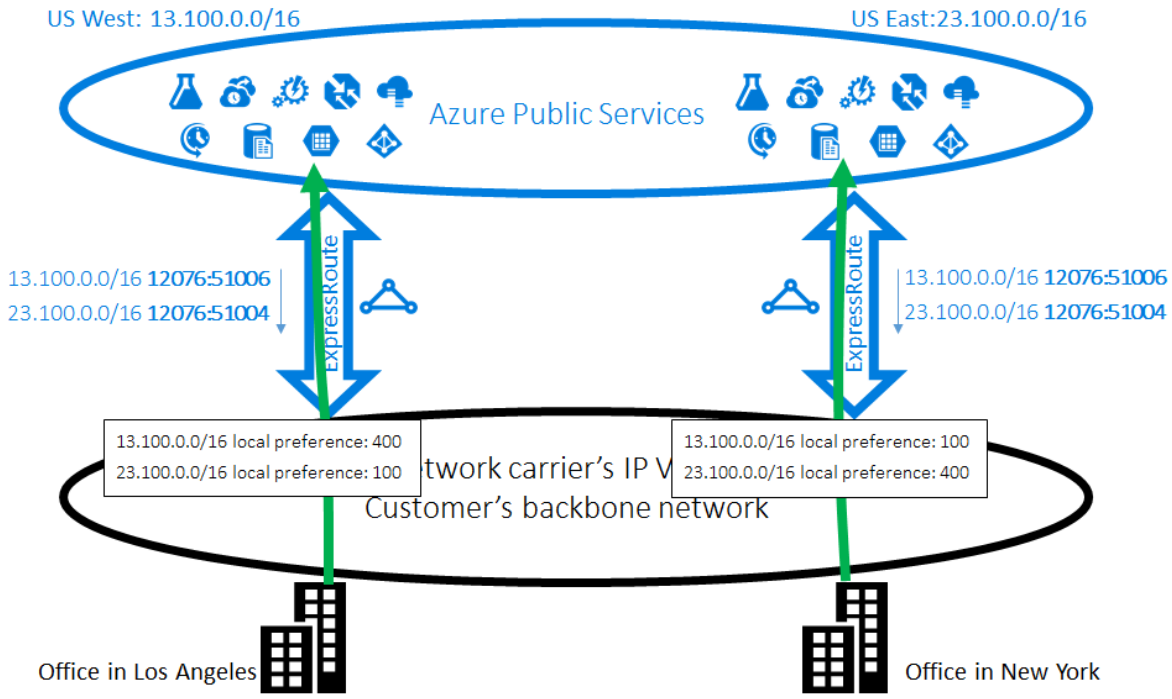
Let's take a close look at the routing problem by an example. Imagine you have two offices in the US, one in Los Angeles and one in New York. Your offices are connected on a Wide Area Network (WAN), which can be either your own backbone network or your service provider's IP VPN. You have two ExpressRoute circuits, one in US West and one in US East, that are also connected on the WAN. Obviously, you have two paths to connect to the Microsoft network. Now imagine you have Azure deployment (e.g. Azure App Service) in both US West and US East. Your intention is to connect your users in Los Angeles to Azure US West and your users in New York to Azure US East because your service admin advertises that users in each office access the nearby Azure services for optimal experiences. Unfortunately, the plan works out well for the east coast users but not for the west coast users. The cause of the problem is the following. On each ExpressRoute circuit, we advertise to you both the prefix in Azure US East (23.100.0.0/16) and the prefix in Azure US West (13.100.0.0/16). If you don't know which prefix is from which region, you are not able to treat it differently. Your WAN network may think both of the prefixes are closer to US East than US West and therefore route both office users to the ExpressRoute circuit in US East. In the end, you will have many unhappy users in the Los Angeles office.



Solution: use BGP Communities

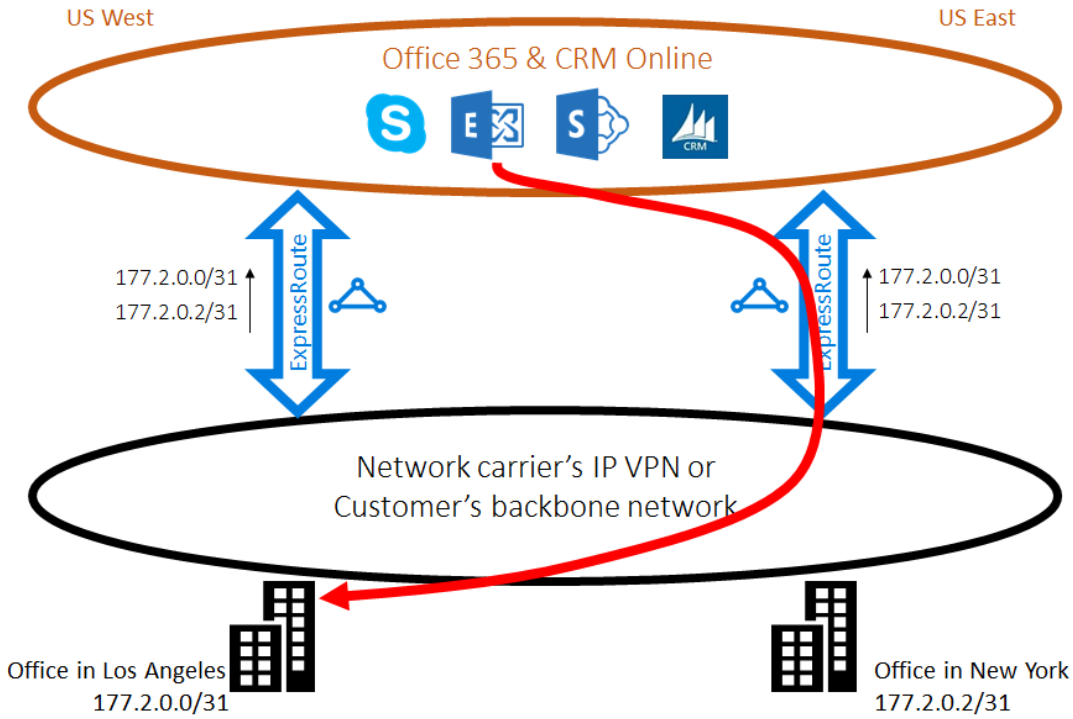
To optimize routing for both office users, you need to know which prefix is from Azure US West and which from Azure US East. We encode this information by using BGP Community values. We've assigned a unique BGP Community value to each Azure region, e.g. "12076:51004" for US East, "12076:51006" for US West. Now that you

know which prefix is from which Azure region, you can configure which ExpressRoute circuit should be preferred. Because we use the BGP to exchange routing info, you can use BGP's Local Preference to influence routing. In our example, you can assign a higher local preference value to 13.100.0.0/16 in US West than in US East, and similarly, a higher local preference value to 23.100.0.0/16 in US East than in US West. This configuration will make sure that, when both paths to Microsoft are available, your users in Los Angeles will take the ExpressRoute circuit in US West to connect to Azure US West whereas your users in New York take the ExpressRoute in US East to Azure US East. Routing is optimized on both sides.



Suboptimal routing case 2

Here is another example where connections from Microsoft take a longer path to reach your network. In this case, you use on-premises Exchange servers and Exchange Online in a hybrid environment. Your offices are connected to a WAN. You advertise the prefixes of your on-premises servers in both of your offices to Microsoft through the two ExpressRoute circuits. Exchange Online will initiate connections to the on-premises servers in cases such as mailbox migration. Unfortunately, the connection to your Los Angeles office is routed to the ExpressRoute circuit in US East before traversing the entire continent back to the west coast. The cause of the problem is similar to the first one. Without any hint, the Microsoft network can't tell which customer prefix is close to US East and which one is close to US West. It happens to pick the wrong path to your office in Los Angeles.

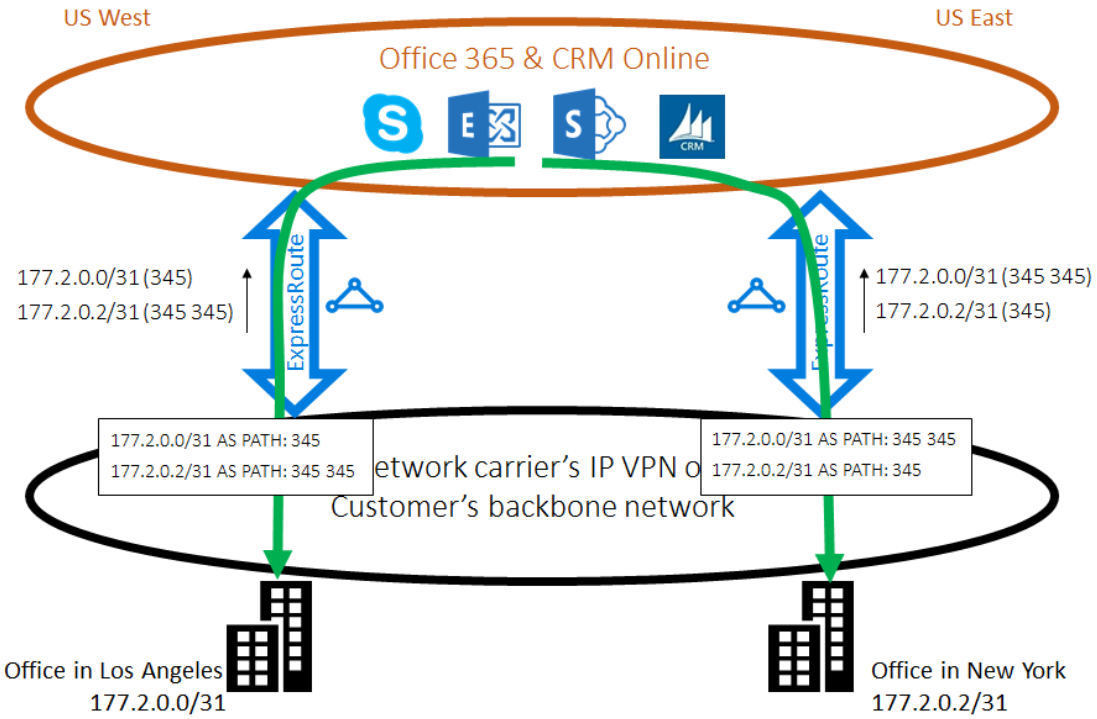


Solution: use AS PATH prepending

There are two solutions to the problem. The first one is that you simply advertise your on-premises prefix for your Los Angeles office, 177.2.0.0/31, on the ExpressRoute circuit in US West and your on-premises prefix for your New York office, 177.2.0.2/31, on the ExpressRoute circuit in US East. As a result, there is only one path for Microsoft to connect to each of your offices. There is no ambiguity and routing is optimized. With this design, you need to think about your failover strategy. In the event that the path to Microsoft via ExpressRoute is broken, you need to make sure that Exchange Online can still connect to your on-premises servers.

The second solution is that you continue to advertise both of the prefixes on both ExpressRoute circuits, and in addition you give us a hint of which prefix is close to which one of your offices. Because we support BGP AS Path prepending, you can configure the AS Path for your prefix to influence routing. In this example, you can lengthen the AS PATH for 172.2.0.0/31 in US East so that we will prefer the ExpressRoute circuit in US West for traffic destined for this prefix (as our network will think the path to this prefix is shorter in the west). Similarly you can lengthen the AS PATH for 172.2.0.2/31 in US West so that we'll prefer the ExpressRoute circuit in US East. Routing is optimized for both offices. With this design, if one ExpressRoute circuit is broken, Exchange Online can still reach you via another ExpressRoute circuit and your WAN.

> **IMPORTANT**
> We remove private AS numbers in the AS PATH for the prefixes received on Microsoft Peering. You need to append public AS numbers in the AS PATH to influence routing for Microsoft Peering.



IMPORTANT

While the examples given here are for Microsoft and Public peerings, we do support the same capabilities for the Private peering. Also, the AS Path prepending works within one single ExpressRoute circuit, to influence the selection of the primary and secondary paths.

ExpressRoute NAT requirements

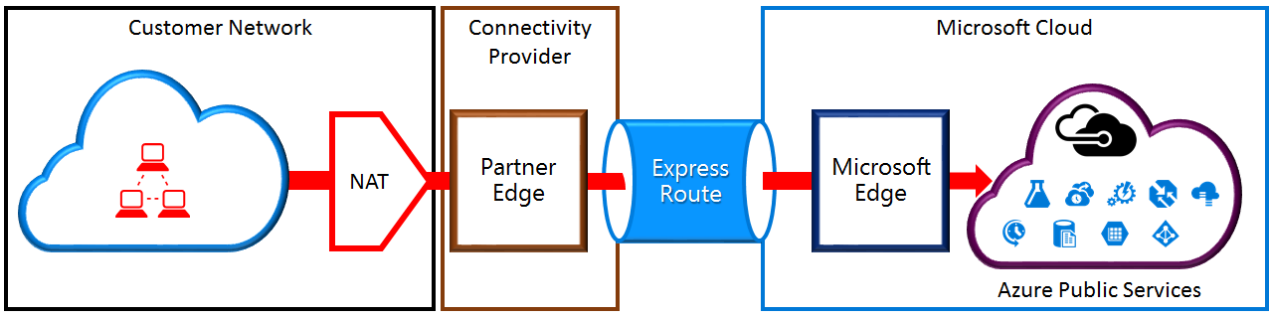
1/17/2017 • 3 min to read • Edit on GitHub

To connect to Microsoft cloud services using ExpressRoute, you'll need to set up and manage NATs. Some connectivity providers offer setting up and managing NAT as a managed service. Check with your connectivity provider to see if they offer such a service. If not, you must adhere to the requirements described below.

Review the ExpressRoute circuits and routing domains page to get an overview of the various routing domains. To meet the public IP address requirements for Azure public and Microsoft peering, we recommend that you set up NAT between your network and Microsoft. This section provides a detailed description of the NAT infrastructure you need to set up.

NAT requirements for Azure public peering

The Azure public peering path enables you to connect to all services hosted in Azure over their public IP addresses. These include services listed in the ExpessRoute FAQ and any services hosted by ISVs on Microsoft Azure. Connectivity to Microsoft Azure services on public peering is always initiated from your network into the Microsoft network. Traffic destined to Microsoft Azure on public peering must be SNATed to valid public IPv4 addresses before they enter the Microsoft network. The figure below provides a high-level picture of how the NAT could be set up to meet the above requirement.



NAT IP pool and route advertisements

You must ensure that traffic is entering the Azure public peering path with valid public IPv4 address. Microsoft must be able to validate the ownership of the IPv4 NAT address pool against a regional routing Internet registry (RIR) or an Internet routing registry (IRR). A check will be performed based on the AS number being peered with and the IP addresses used for the NAT. Refer to the ExpressRoute routing requirements page for information on routing registries.

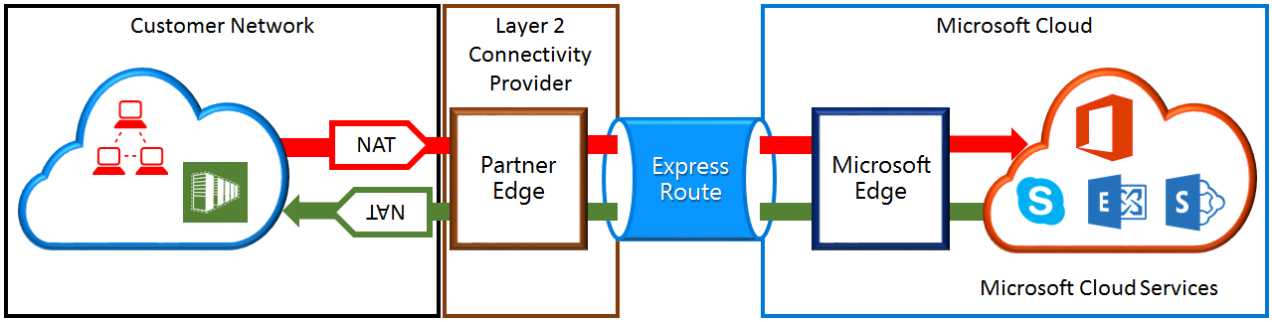
There are no restrictions on the length of the NAT IP prefix advertised through this peering. You must monitor the NAT pool and ensure that you are not starved of NAT sessions.

> **IMPORTANT**
> The NAT IP pool advertised to Microsoft must not be advertised to the Internet. This will break connectivity to other Microsoft services.

NAT requirements for Microsoft peering

The Microsoft peering path lets you connect to Microsoft cloud services that are not supported through the Azure public peering path. The list of services includes Office 365 services, such as Exchange Online, SharePoint Online, Skype for Business, and CRM Online. Microsoft expects to support bi-directional connectivity on the Microsoft

peering. Traffic destined to Microsoft cloud services must be SNATed to valid public IPv4 addresses before they enter the Microsoft network. Traffic destined to your network from Microsoft cloud services must be SNATed before they enter your network. The figure below provides a high-level picture of how the NAT should be setup for Microsoft peering.



Traffic originating from your network destined to Microsoft

- You must ensure that traffic is entering the Microsoft peering path with a valid public IPv4 address. Microsoft must be able to validate the owner of the IPv4 NAT address pool against the regional routing internet registry (RIR) or an internet routing registry (IRR). A check will be performed based on the AS number being peered with and the IP addresses used for the NAT. Refer to the ExpressRoute routing requirements page for information on routing registries.
- IP addresses used for the Azure public peering setup and other ExpressRoute circuits must not be advertised to Microsoft through the BGP session. There is no restriction on the length of the NAT IP prefix advertised through this peering.

> **IMPORTANT**
> The NAT IP pool advertised to Microsoft must not be advertised to the Internet. This will break connectivity to other Microsoft services.

Traffic originating from Microsoft destined to your network

- Certain scenarios require Microsoft to initiate connectivity to service endpoints hosted within your network. A typical example of the scenario would be connectivity to ADFS servers hosted in your network from Office 365. In such cases, you must leak appropriate prefixes from your network into the Microsoft peering.
- You must SNAT traffic destined to IP addresses within your network from Microsoft.

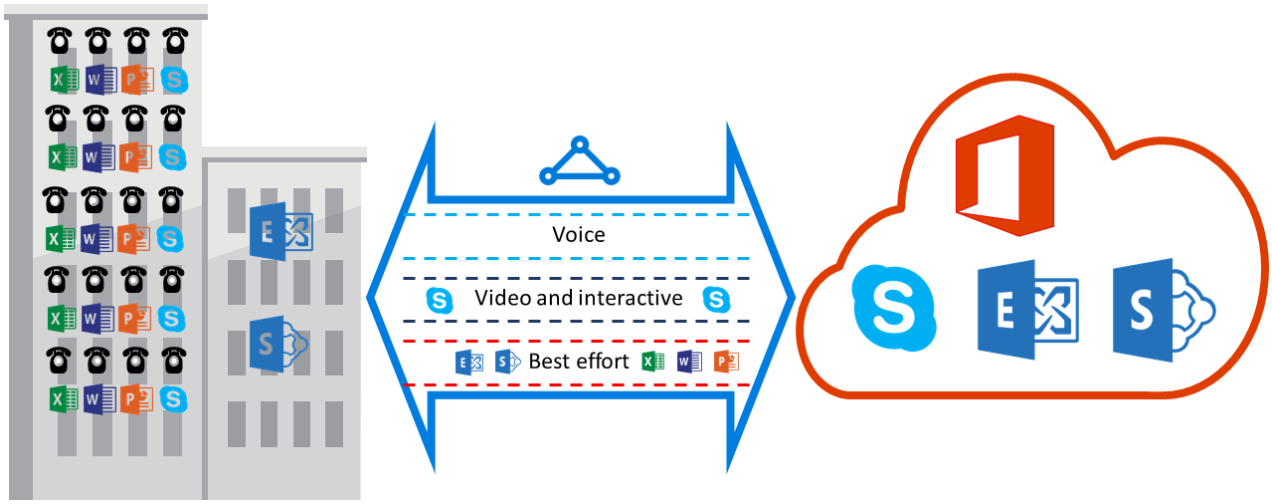
Next steps

- Refer to the requirements for Routing and QoS.
- For workflow information, see ExpressRoute circuit provisioning workflows and circuit states.
- Configure your ExpressRoute connection.
  - Create an ExpressRoute circuit
  - Configure routing
  - Link a VNet to an ExpressRoute circuit

ExpressRoute QoS requirements

1/17/2017 • 1 min to read • Edit on GitHub

Skype for Business has various workloads that require differentiated QoS treatment. If you plan to consume voice services through ExpressRoute, you should adhere to the requirements described below.



NOTE

QoS requirements apply to the Microsoft peering only. The DSCP values in your network traffic received on Azure public peering and Azure private peering will be reset to 0.

The following table provides a list of DSCP markings used by Skype for Business. Refer to Managing QoS for Skype for Business for more information.

TRAFFIC CLASS	TREATMENT (DSCP MARKING)	SKYPE FOR BUSINESS WORKLOADS
Voice	EF (46)	Skype / Lync voice
Interactive	AF41 (34)	Video
AF21 (18)	App sharing	
Default	AF11 (10)	File transfer
CS0 (0)	Anything else	

- You should classify the workloads and mark the right DSCP values. Follow the guidance provided here on how to set DSCP markings in your network.
- You should configure and support multiple QoS queues within your network. Voice must be a standalone class and receive the EF treatment specified in RFC 3246.
- You can decide the queuing mechanism, congestion detection policy, and bandwidth allocation per traffic class. But, the DSCP marking for Skype for Business workloads must be preserved. If you are using DSCP markings not listed above, e.g. AF31 (26), you must rewrite this DSCP value to 0 before sending the packet to Microsoft. Microsoft only sends packets marked with the DSCP value shown in the above table.

Next steps

- Refer to the requirements for Routing and NAT.
- See the following links to configure your ExpressRoute connection.
    - Create an ExpressRoute circuit
    - Configure routing
    - Link a VNet to an ExpressRoute circuit

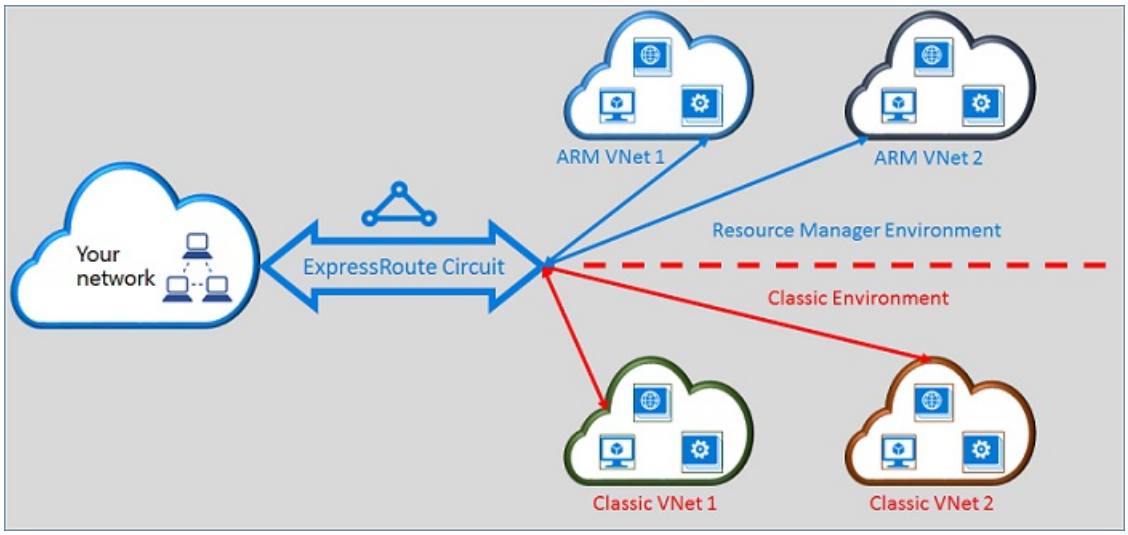
Moving ExpressRoute circuits from the classic to the Resource Manager deployment model

1/17/2017 • 6 min to read • Edit on GitHub

This article provides an overview of what it means to move an Azure ExpressRoute circuit from the classic to the Azure Resource Manager deployment model.

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.

You can use a single ExpressRoute circuit to connect to virtual networks that are deployed both in the classic and the Resource Manager deployment models. An ExpressRoute circuit, regardless of how it is created, can now link to virtual networks across both deployment models.



ExpressRoute circuits that are created in the classic deployment model

ExpressRoute circuits that are created in the classic deployment model need to be moved to the Resource Manager deployment model first to enable connectivity to both the classic and the Resource Manager deployment models. There isn't connectivity loss or disruption when a connection is being moved. All circuit-to-virtual network links in the classic deployment model (within the same subscription and cross-subscription) are preserved.

After the move is completed successfully, the ExpressRoute circuit looks, performs, and feels exactly like an ExpressRoute circuit that was created in the Resource Manager deployment model. You can now create connections to virtual networks in the Resource Manager deployment model.

After an ExpressRoute circuit has been moved to the Resource Manager deployment model, you can manage the life cycle of the ExpressRoute circuit only by using the Resource Manager deployment model. This means that you can perform operations like adding/updating/deleting peerings, updating circuit properties (such as bandwidth, SKU, and billing type), and deleting circuits only in the Resource Manager deployment model. Refer to the section below on circuits that are created in the Resource Manager deployment model for further details on how you can manage access to both deployment models.

You do not have to involve your connectivity provider to perform the move.

ExpressRoute circuits that are created in the Resource Manager deployment model

You can enable ExpressRoute circuits that are created in the Resource Manager deployment model to be accessible from both deployment models. Any ExpressRoute circuit in your subscription can be enabled to be accessed from both deployment models.

- ExpressRoute circuits that were created in the Resource Manager deployment model do not have access to the classic deployment model by default.
- ExpressRoute circuits that have been moved from the classic deployment model to the Resource manager deployment model are accessible from both deployment models by default.
- An ExpressRoute circuit always has access to the Resource Manager deployment model, regardless of whether it was created in the Resource Manager or classic deployment model. This means that you can create connections to virtual networks created in the Resource Manager deployment model by following instructions on how to link virtual networks.
- Access to the classic deployment model is controlled by the **allowClassicOperations** parameter in the ExpressRoute circuit.

> **IMPORTANT**
> All quotas that are documented on the service limits page apply. As an example, a standard circuit can have at most 10 virtual network links/connections across both the classic and the Resource Manager deployment models.

Controlling access to the classic deployment model

You can enable a single ExpressRoute circuit to link to virtual networks in both deployment models by setting the **allowClassicOperations** parameter of the ExpressRoute circuit.

Setting **allowClassicOperations** to TRUE enables you to link virtual networks from both deployment models to the ExpressRoute circuit. You can link to virtual networks in the classic deployment model by following guidance on how to link virtual networks in the classic deployment model. You can link to virtual networks in the Resource Manager deployment model by following guidance on how to link virtual networks in the Resource Manager deployment model.

Setting **allowClassicOperations** to FALSE blocks access to the circuit from the classic deployment model. However, all virtual network links in the classic deployment model are preserved. In this case, the ExpressRoute circuit is not visible in the classic deployment model.

Supported operations in the classic deployment model

The following classic operations are supported on an ExpressRoute circuit when **allowClassicOperations** is set to TRUE:

- Get ExpressRoute circuit information
- Create/update/get/delete virtual network links to classic virtual networks
- Create/update/get/delete virtual network link authorizations for cross-subscription connectivity

You cannot perform the following classic operations when **allowClassicOperations** is set to TRUE:

- Create/update/get/delete Border Gateway Protocol (BGP) peerings for Azure private, Azure public, and Microsoft peerings
- Delete ExpressRoute circuits

Communication between the classic and the Resource Manager deployment models

The ExpressRoute circuit acts like a bridge between the classic and the Resource Manager deployment models. Traffic between virtual machines in virtual networks in the classic deployment model and those in virtual networks in the Resource Manager deployment model flows through ExpressRoute if both virtual networks are linked to the same ExpressRoute circuit.

Aggregate throughput is limited by the throughput capacity of the virtual network gateway. Traffic does not enter the connectivity provider's networks or your networks in such cases. Traffic flow between the virtual networks is fully contained within the Microsoft network.

Access to Azure public and Microsoft peering resources

You can continue to access resources that are typically accessible through Azure public peering and Microsoft peering without any disruption.

What's supported

This section describes what's supported for ExpressRoute circuits:

- You can use a single ExpressRoute circuit to access virtual networks that are deployed in the classic and the Resource Manager deployment models.
- You can move an ExpressRoute circuit from the classic to the Resource Manager deployment model. After it is moved, the ExpressRoute circuit looks, feels, and performs like any other ExpressRoute circuit that is created in the Resource Manager deployment model.
- You can move only the ExpressRoute circuit. Circuit links, virtual networks, and VPN gateways cannot be moved through this operation.
- After an ExpressRoute circuit has been moved to the Resource Manager deployment model, you can manage the life cycle of the ExpressRoute circuit only by using the Resource Manager deployment model. This means that you can perform operations like adding/updating/deleting peerings, updating circuit properties (such as bandwidth, SKU, and billing type), and deleting circuits only in the Resource Manager deployment model.
- The ExpressRoute circuit acts like a bridge between the classic and the Resource Manager deployment models. Traffic between virtual machines in virtual networks in the classic deployment model and those in virtual networks in the Resource Manager deployment model flows through ExpressRoute if both virtual networks are linked to the same ExpressRoute circuit.
- Cross-subscription connectivity is supported in both the classic and the Resource Manager deployment models.

What's not supported

This section describes what's not supported for ExpressRoute circuits:

- Moving circuit links, gateways, and virtual networks from the classic to the Resource Manager deployment model.
- Managing the life cycle of an ExpressRoute circuit from the classic deployment model.
- Role-Based Access Control (RBAC) support for the classic deployment model. You cannot perform RBAC controls to a circuit in the classic deployment model. Any administrator/coadministrator of the subscription can link or unlink virtual networks to the circuit.

Configuration

Follow the instructions that are described in Move an ExpressRoute circuit from the classic to the Resource Manager deployment model.

Next steps

- For workflow information, see ExpressRoute circuit provisioning workflows and circuit states.
- To configure your ExpressRoute connection:
  - Create an ExpressRoute circuit
  - Configure routing
  - Link a virtual network to an ExpressRoute circuit

About virtual network gateways for ExpressRoute

1/17/2017 • 2 min to read • Edit on GitHub

A virtual network gateway is used to send network traffic between Azure virtual networks and on-premises locations. When you configure an ExpressRoute connection, you must create and configure a virtual network gateway and a virtual network gateway connection.

When you create a virtual network gateway, you specify several settings. One of the required settings specifies whether the gateway will be used for ExpressRoute or Site-to-Site VPN traffic. In the Resource Manager deployment model, the setting is '-GatewayType'.

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 the public Internet, 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 Vpn, and one that uses -GatewayType ExpressRoute. This article focuses on the ExpressRoute virtual network gateway.

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.

ExpressRoute virtual network gateways can use the following SKUs:

- Standard
- HighPerformance
- UltraPerformance

If you want to upgrade your gateway to a more powerful gateway SKU, in most cases you can use the 'Resize-AzureRmVirtualNetworkGateway' PowerShell cmdlet. This will work for upgrades to Standard and HighPerformance SKUs. However, to upgrade to the UltraPerformance SKU, you will need to recreate the gateway.

Estimated aggregate throughput by gateway SKU

The following table shows the gateway types and the estimated aggregate throughput. This table applies to both the Resource Manager and classic deployment models.

	EXPRESSROUTE GATEWAY THROUGHPUT	VPN GATEWAY AND EXPRESSROUTE COEXIST
Basic SKU (deprecated)	500 Mbps	No
Standard SKU	1000 Mbps	Yes
High Performance SKU	2000 Mbps	Yes
Ultra Performance SKU	10000 Mbps	Yes

REST APIs and PowerShell cmdlets

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

CLASSIC	RESOURCE MANAGER
PowerShell	PowerShell
REST API	REST API

Next steps

See ExpressRoute Overview for more information about available connection configurations.

Create and modify an ExpressRoute circuit

1/17/2017 • 5 min to read • Edit on GitHub

This article describes how to create an Azure ExpressRoute circuit by using the Azure portal and the Azure Resource Manager deployment model. The following steps also show you how to check the status of the circuit, update it, or delete and deprovision it.

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.

Before you begin

- Review the prerequisites and workflows before you begin configuration.
- Ensure that you have access to the Azure portal.
- Ensure that you have permissions to create new networking resources. Contact your account administrator if you do not have the right permissions.
- You can view a video before beginning in order to better understand the steps.

Create and provision an ExpressRoute circuit

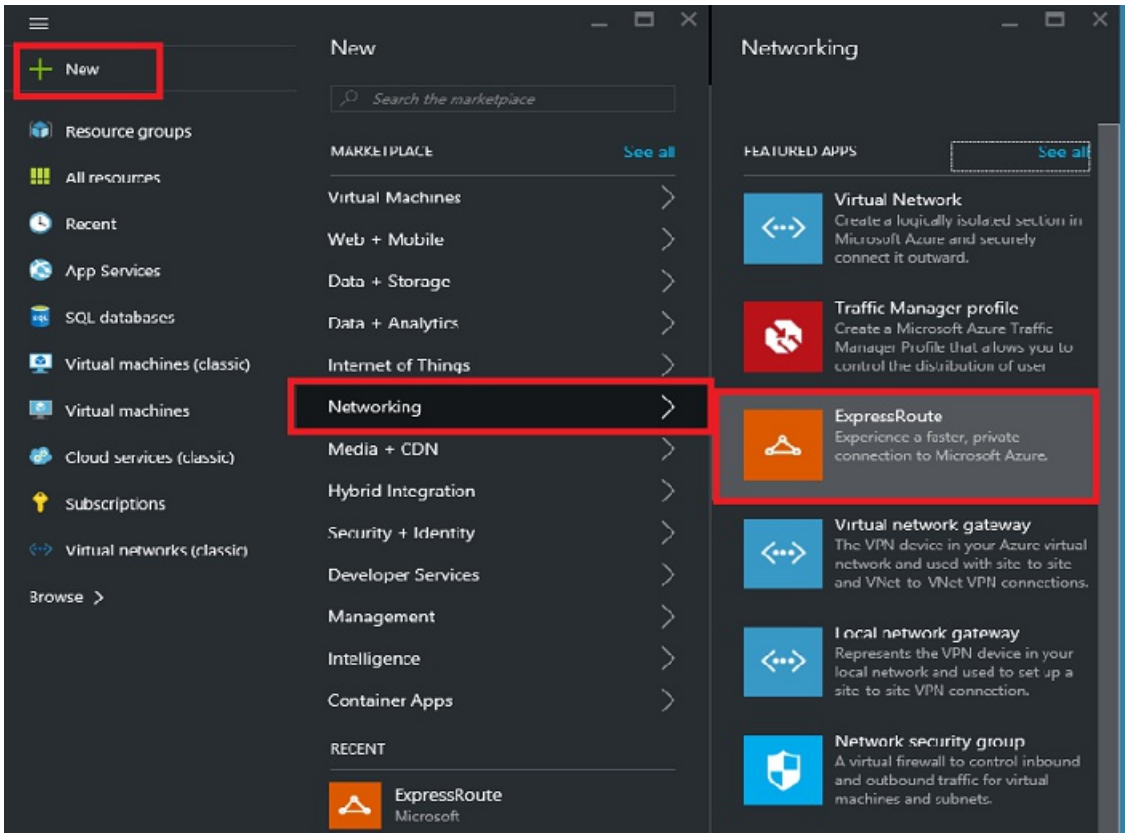
1. Sign in to the Azure portal

From a browser, navigate to the Azure portal and sign in with your Azure account.

2. Create a new ExpressRoute circuit

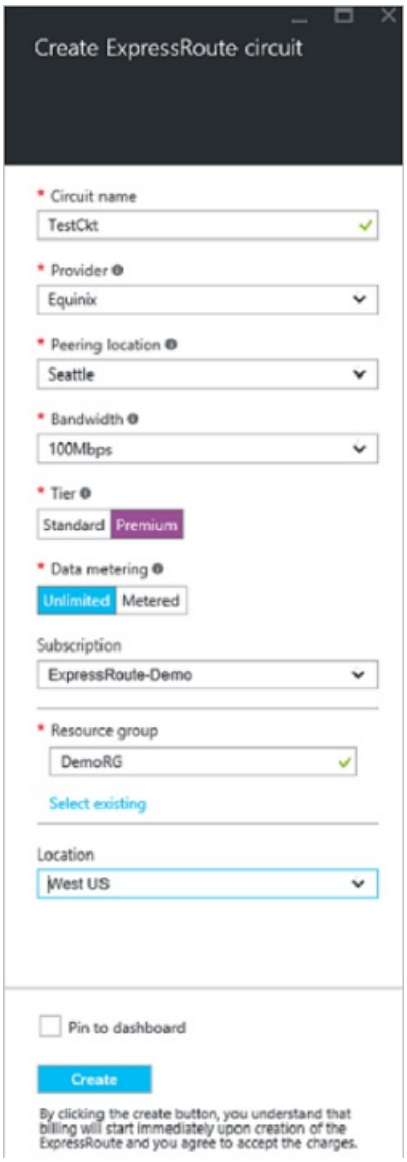
> **IMPORTANT**
> Your ExpressRoute circuit will be billed from the moment a service key is issued. Ensure that you perform this operation when the connectivity provider is ready to provision the circuit.

1. You can create an ExpressRoute circuit by selecting the option to create a new resource. Click **New** > **Networking** > **ExpressRoute**, as shown in the following image:



2. After you click **ExpressRoute**, you'll see the **Create ExpressRoute circuit** blade. When you're filling in the values on this blade, make sure that you specify the correct SKU tier and data metering.

   - **Tier** determines whether an ExpressRoute standard or an ExpressRoute premium add-on is enabled. You can specify **Standard** to get the standard SKU or **Premium** for the premium add-on.
   - **Data metering** determines the billing type. You can specify **Metered** for a metered data plan and **Unlimited** for an unlimited data plan. Note that you can change the billing type from **Metered** to **Unlimited**, but you can't change the type from **Unlimited** to **Metered**.



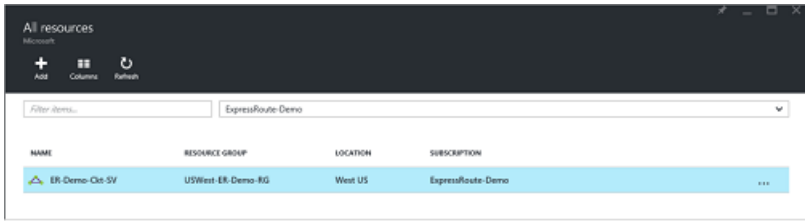
IMPORTANT

Please be aware that the Peering Location indicates the physical location where you are peering with Microsoft. This is **not** linked to "Location" property, which refers to the geography where the Azure Network Resource Provider is located. While they are not related, it is a good practice to choose a Network Resource Provider geographically close to the Peering Location of the circuit.

3. View the circuits and properties

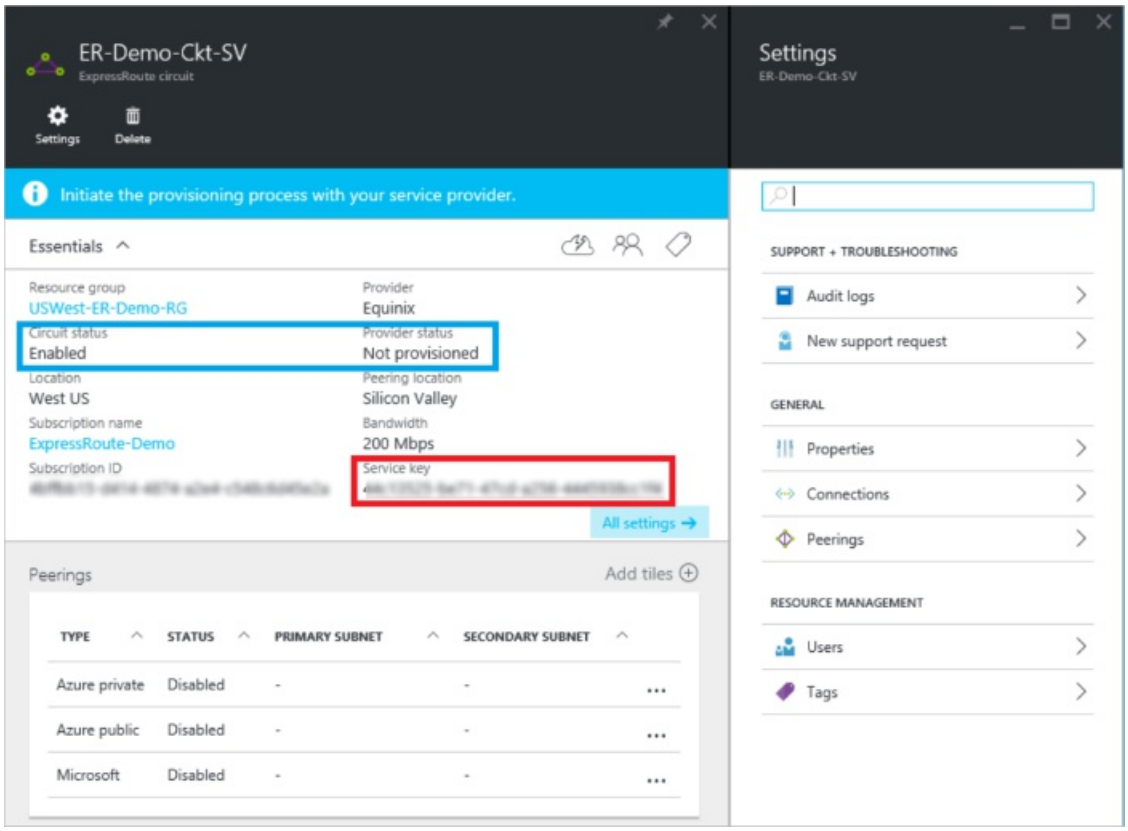
View all the circuits

You can view all the circuits that you created by selecting **All resources** on the left-side menu.



View the properties

```
You can view the properties of the circuit by selecting it. On this blade, note the service key for the circuit.
You must copy the circuit key for your circuit and pass it down to the service provider to complete the
provisioning process. The circuit key is specific to your circuit.
```

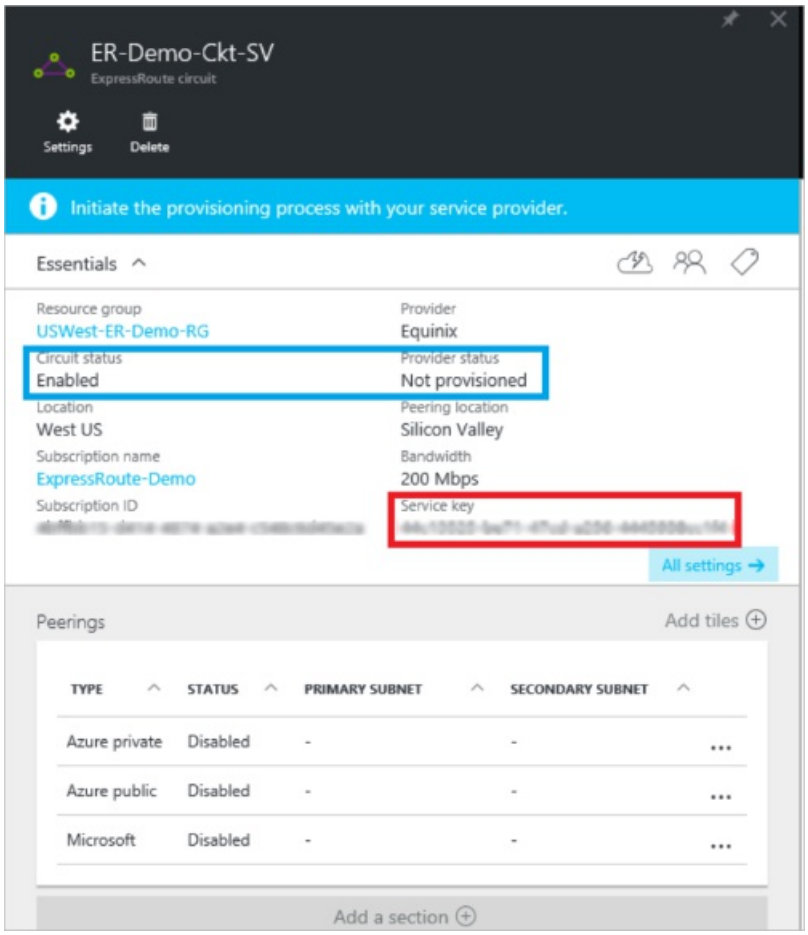


4. Send the service key to your connectivity provider for provisioning

On this blade, **Provider status** provides information on the current state of provisioning on the service-provider side. **Circuit status** provides the state on the Microsoft side. For more information about circuit provisioning states, see the Workflows article.

When you create a new ExpressRoute circuit, the circuit will be in the following state:

Provider status: Not provisioned
Circuit status: Enabled



The circuit will change to the following state when the connectivity provider is in the process of enabling it for you:

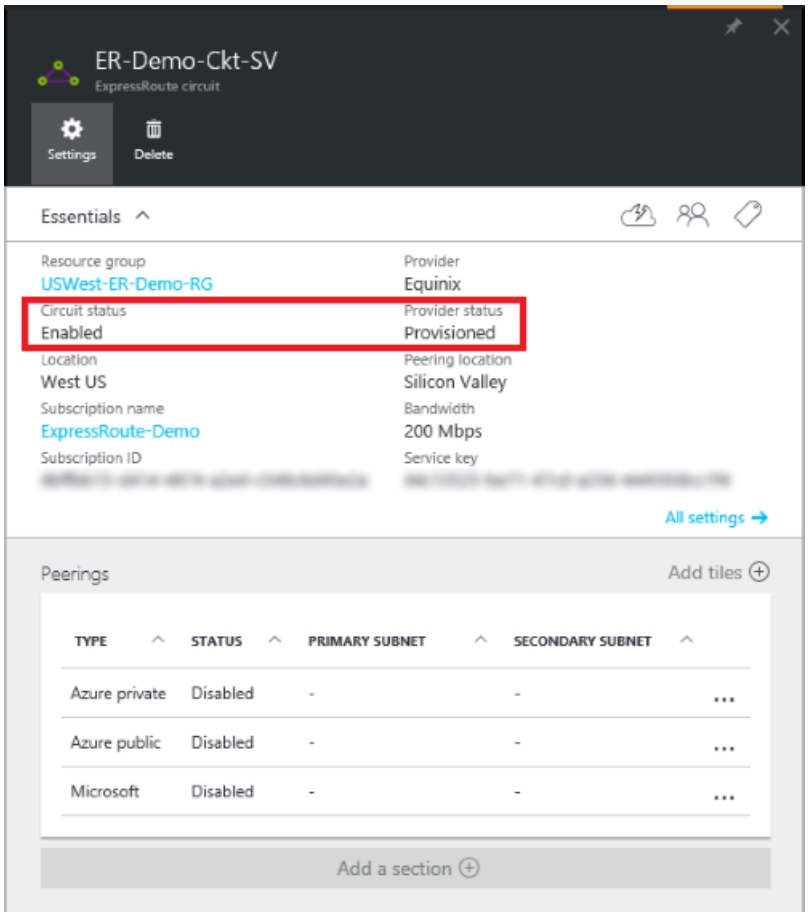
Provider status: Provisioning
Circuit status: Enabled

For you to be able to use an ExpressRoute circuit, it must be in the following state:

Provider status: Provisioned
Circuit status: Enabled

5. Periodically check the status and the state of the circuit key

You can view the properties of the circuit that you're interested in by selecting it. Check the **Provider status** and ensure that it has moved to **Provisioned** before you continue.



6. Create your routing configuration

For step-by-step instructions, refer to the ExpressRoute circuit routing configuration article to create and modify circuit peerings.

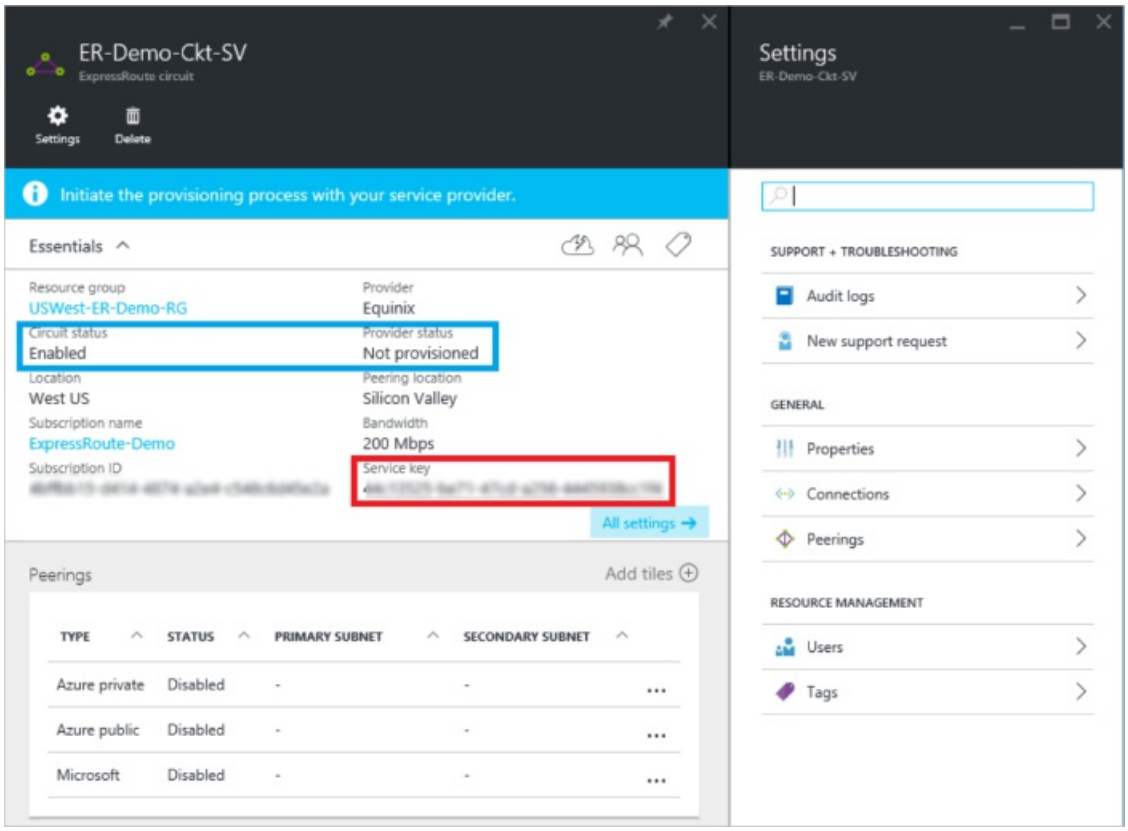
> **IMPORTANT**
>
> These instructions only apply to circuits that are created with service providers that offer layer 2 connectivity services. If you're using a service provider that offers managed layer 3 services (typically an IP VPN, like MPLS), your connectivity provider will configure and manage routing for you.

7. Link a virtual network to an ExpressRoute circuit

Next, link a virtual network to your ExpressRoute circuit. Use the Linking virtual networks to ExpressRoute circuits article when you work with the Resource Manager deployment model.

Getting the status of an ExpressRoute circuit

You can view the status of a circuit by selecting it.



Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity. At this time, you cannot modify ExpressRoute circuit properties by using the Azure portal. However, you can use PowerShell to modify circuit properties. For more information, see the section Modifying an ExpressRoute circuit by using PowerShell.

You can do the following with no downtime:

- Enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit. Note that downgrading the bandwidth of a circuit is not supported.
- Change the metering plan from Metered Data to Unlimited Data. Note that changing the metering plan from Unlimited Data to Metered Data is not supported.
- You can enable and disable **Allow Classic Operations**.

For more information on limits and limitations, refer to the ExpressRoute FAQ.

Deprovisioning and deleting an ExpressRoute circuit

You can delete your ExpressRoute circuit by selecting the **delete** icon. Note the following:

- You must unlink all virtual networks from the ExpressRoute circuit. If this operation fails, check whether any virtual networks are linked to the circuit.
- If the ExpressRoute circuit service provider provisioning state is **Provisioning** or **Provisioned** you must work with your service provider to deprovision the circuit on their side. We will continue to reserve resources and bill you until the service provider completes deprovisioning the circuit and notifies us.
- If the service provider has deprovisioned the circuit (the service provider provisioning state is set to **Not provisioned**) you can then delete the circuit. This will stop billing for the circuit

Next steps

After you create your circuit, make sure that you do the following:

- Create and modify routing for your ExpressRoute circuit
- Link your virtual network to your ExpressRoute circuit

Create and modify an ExpressRoute circuit

1/17/2017 • 9 min to read • Edit on GitHub

This article describes how to create an Azure ExpressRoute circuit by using Windows PowerShell cmdlets and the Azure Resource Manager deployment model. This article will also show you how to check the status of the circuit, update it, or delete and deprovision it.

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.

Before you begin

- Obtain the latest version of the Azure PowerShell modules (at least version 1.0). For step-by-step guidance on how to configure your computer to use the PowerShell modules, follow the instructions in How to install and configure Azure PowerShell.
- Review the prerequisites and workflows before you begin configuration.

Create and provision an ExpressRoute circuit

1. Sign in to your Azure account and select your subscription

To begin your configuration, sign in to your Azure account. For more information about PowerShell, see Using Windows PowerShell with Resource Manager. Use the following examples to help you connect:

```
Login-AzureRmAccount
```

Check the subscriptions for the account:

```
Get-AzureRmSubscription
```

Select the subscription that you want to create an ExpressRoute circuit for:

```
Select-AzureRmSubscription -SubscriptionId "<subscription ID>"
```

2. Get the list of supported providers, locations, and bandwidths

Before you create an ExpressRoute circuit, you need the list of supported connectivity providers, locations, and bandwidth options.

The PowerShell cmdlet `Get-AzureRmExpressRouteServiceProvider` returns this information, which you'll use in later steps:

```
Get-AzureRmExpressRouteServiceProvider
```

Check to see if your connectivity provider is listed there. Make a note of the following information because you'll need it later when you create a circuit:

- Name
- PeeringLocations
- BandwidthsOffered

You're now ready to create an ExpressRoute circuit.

3. Create an ExpressRoute circuit

If you don't already have a resource group, you must create one before you create your ExpressRoute circuit. You can do so by running the following command:

```
New-AzureRmResourceGroup -Name "ExpressRouteResourceGroup" -Location "West US"
```

The following example shows how to create a 200-Mbps ExpressRoute circuit through Equinix in Silicon Valley. If you're using a different provider and different settings, substitute that information when you make your request. The following is an example request for a new service key:

```
New-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup" -
Location "West US" -SkuTier Standard -SkuFamily MeteredData -ServiceProviderName "Equinix" -PeeringLocation
"Silicon Valley" -BandwidthInMbps 200
```

Make sure that you specify the correct SKU tier and SKU family:

- SKU tier determines whether an ExpressRoute standard or an ExpressRoute premium add-on is enabled. You can specify *Standard* to get the standard SKU or *Premium* for the premium add-on.
- SKU family determines the billing type. You can specify *Metereddata* for a metered data plan and *Unlimiteddata* for an unlimited data plan. Note that you can change the billing type from *Metereddata* to *Unlimiteddata*, but you can't change the type from *Unlimiteddata* to *Metereddata*.

> **IMPORTANT**
> Your ExpressRoute circuit will be billed from the moment a service key is issued. Ensure that you perform this operation when the connectivity provider is ready to provision the circuit.

The response contains the service key. You can get detailed descriptions of all the parameters by running the following:

```
get-help New-AzureRmExpressRouteCircuit -detailed
```

4. List all ExpressRoute circuits

To get a list of all the ExpressRoute circuits that you created, run the `Get-AzureRmExpressRouteCircuit` command:

```
Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup"
```

The response will look similar to the following example:

```
Name                             : ExpressRouteARMCircuit
ResourceGroupName                : ExpressRouteResourceGroup
Location                         : westus
Id                               :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Networ
k/expressRouteCircuits/ExpressRouteARMCircuit
Etag                             : W/"################################"
ProvisioningState                : Succeeded
Sku                              : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                     }
CircuitProvisioningState          : Enabled
ServiceProviderProvisioningState  : NotProvisioned
ServiceProviderNotes              :
ServiceProviderProperties         : {
                                      "ServiceProviderName": "Equinix",
                                      "PeeringLocation": "Silicon Valley",
                                      "BandwidthInMbps": 200
                                    }
ServiceKey                       : ************************************
Peerings                         : []
```

You can retrieve this information at any time by using the `Get-AzureRmExpressRouteCircuit` cmdlet. Making the call with no parameters lists all the circuits. Your service key will be listed in the *ServiceKey* field:

```
Get-AzureRmExpressRouteCircuit
```

The response will look similar to the following example:

```
Name                             : ExpressRouteARMCircuit
ResourceGroupName                : ExpressRouteResourceGroup
Location                         : westus
Id                               :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Networ
k/expressRouteCircuits/ExpressRouteARMCircuit
Etag                             : W/"################################"
ProvisioningState                : Succeeded
Sku                              : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                     }
CircuitProvisioningState         : Enabled
ServiceProviderProvisioningState : NotProvisioned
ServiceProviderNotes             :
ServiceProviderProperties        : {
                                     "ServiceProviderName": "Equinix",
                                     "PeeringLocation": "Silicon Valley",
                                     "BandwidthInMbps": 200
                                     }
ServiceKey                       : ************************************
Peerings                         : []
```

You can get detailed descriptions of all the parameters by running the following:

```
get-help Get-AzureRmExpressRouteCircuit -detailed
```

5. Send the service key to your connectivity provider for provisioning

ServiceProviderProvisioningState provides information about the current state of provisioning on the service-provider side. Status provides the state on the Microsoft side. For more information about circuit provisioning

states, see the Workflows article.

When you create a new ExpressRoute circuit, the circuit will be in the following state:

```
ServiceProviderProvisioningState : NotProvisioned
CircuitProvisioningState         : Enabled
```

The circuit will change to the following state when the connectivity provider is in the process of enabling it for you:

```
ServiceProviderProvisioningState : Provisioning
Status                           : Enabled
```

For you to be able to use an ExpressRoute circuit, it must be in the following state:

```
ServiceProviderProvisioningState : Provisioned
CircuitProvisioningState         : Enabled
```

6. Periodically check the status and the state of the circuit key

Checking the status and the state of the circuit key lets you know when your provider has enabled your circuit. After the circuit has been configured, *ServiceProviderProvisioningState* appears as *Provisioned*, as shown in the following example:

```
Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup"
```

The response will look similar to the following example:

```
Name                             : ExpressRouteARMCircuit
ResourceGroupName                : ExpressRouteResourceGroup
Location                         : westus
Id                               :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Networ
k/expressRouteCircuits/ExpressRouteARMCircuit
Etag                             : W/"################################"
ProvisioningState                : Succeeded
Sku                              : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                   }
CircuitProvisioningState         : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes             :
ServiceProviderProperties        : {
                                     "ServiceProviderName": "Equinix",
                                     "PeeringLocation": "Silicon Valley",
                                     "BandwidthInMbps": 200
                                   }
ServiceKey                       : ************************************
Peerings                         : []
```

7. Create your routing configuration

For step-by-step instructions, see the ExpressRoute circuit routing configuration article to create and modify circuit peerings.

> **IMPORTANT**
> These instructions only apply to circuits that are created with service providers that offer layer 2 connectivity services. If you're using a service provider that offers managed layer 3 services (typically an IP VPN, like MPLS), your connectivity provider will configure and manage routing for you.

8. Link a virtual network to an ExpressRoute circuit

Next, link a virtual network to your ExpressRoute circuit. Use the Linking virtual networks to ExpressRoute circuits article when you work with the Resource Manager deployment model.

Getting the status of an ExpressRoute circuit

You can retrieve this information at any time by using the `Get-AzureRmExpressRouteCircuit` cmdlet. Making the call with no parameters lists all the circuits.

```
Get-AzureRmExpressRouteCircuit
```

The response will be similar to the following example:

```
Name                            : ExpressRouteARMCircuit
ResourceGroupName               : ExpressRouteResourceGroup
Location                        : westus
Id                              :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Networ
k/expressRouteCircuits/ExpressRouteARMCircuit
Etag                            : W/"################################"
ProvisioningState               : Succeeded
Sku                             : {
                                    "Name": "Standard_MeteredData",
                                    "Tier": "Standard",
                                    "Family": "MeteredData"
                                  }
CircuitProvisioningState        : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes            :
ServiceProviderProperties       : {
                                      "ServiceProviderName": "Equinix",
                                      "PeeringLocation": "Silicon Valley",
                                      "BandwidthInMbps": 200
                                   }
ServiceKey                      : ************************************
Peerings                        : []
```

You can get information on a specific ExpressRoute circuit by passing the resource group name and circuit name as a parameter to the call:

```
Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup"
```

The response will look similar to the following example:

```
Name                              : ExpressRouteARMCircuit
ResourceGroupName                 : ExpressRouteResourceGroup
Location                          : westus
Id                                :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Networ
k/expressRouteCircuits/ExpressRouteARMCircuit
Etag                              : W/"################################"
ProvisioningState                 : Succeeded
Sku                               : {
                                      "Name": "Standard_MeteredData",
                                         "Tier": "Standard",
                                         "Family": "MeteredData"
                                     }
CircuitProvisioningState          : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes              :
ServiceProviderProperties         : {
                                      "ServiceProviderName": "Equinix",
                                      "PeeringLocation": "Silicon Valley",
                                      "BandwidthInMbps": 200
                                     }
ServiceKey                        : ************************************
Peerings                          : []
```

You can get detailed descriptions of all the parameters by running the following:

```
get-help get-azurededicatedcircuit -detailed
```

Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity.

You can do the following with no downtime:

- Enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit. Note that downgrading the bandwidth of a circuit is not supported.
- Change the metering plan from Metered Data to Unlimited Data. Note that changing the metering plan from Unlimited Data to Metered Data is not supported.
- You can enable and disable *Allow Classic Operations*.

For more information on limits and limitations, refer to the ExpressRoute FAQ.

To enable the ExpressRoute premium add-on

You can enable the ExpressRoute premium add-on for your existing circuit by using the following PowerShell snippet:

```
$ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

$ckt.Sku.Tier = "Premium"
$ckt.sku.Name = "Premium_MeteredData"

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The circuit will now have the ExpressRoute premium add-on features enabled. Note that we will begin billing you for the premium add-on capability as soon as the command has successfully run.

To disable the ExpressRoute premium add-on

> **IMPORTANT**
> This operation can fail if you're using resources that are greater than what is permitted for the standard circuit.

Note the following:

- Before you downgrade from premium to standard, you must ensure that the number of virtual networks that are linked to the circuit is less than 10. If you don't do this, your update request fails, and we will bill you at premium rates.
- You must unlink all virtual networks in other geopolitical regions. If you don't do this, your update request will fail, and we will bill you at premium rates.
- Your route table must be less than 4,000 routes for private peering. If your route table size is greater than 4,000 routes, the BGP session drops and won't be reenabled until the number of advertised prefixes goes below 4,000.

You can disable the ExpressRoute premium add-on for the existing circuit by using the following PowerShell cmdlet:

```
$ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

$ckt.Sku.Tier = "Standard"
$ckt.sku.Name = "Standard_MeteredData"

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To update the ExpressRoute circuit bandwidth

For supported bandwidth options for your provider, check the ExpressRoute FAQ. You can pick any size greater than the size of your existing circuit.

> **IMPORTANT**
> You cannot reduce the bandwidth of an ExpressRoute circuit without disruption. Downgrading bandwidth requires you to deprovision the ExpressRoute circuit and then reprovision a new ExpressRoute circuit.

After you decide what size you need, use the following command to resize your circuit:

```
$ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

$ckt.ServiceProviderProperties.BandwidthInMbps = 1000

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Your circuit will be sized up on the Microsoft side. Then you must contact your connectivity provider to update configurations on their side to match this change. After you make this notification, we will begin billing you for the updated bandwidth option.

To move the SKU from metered to unlimited

You can change the SKU of an ExpressRoute circuit by using the following PowerShell snippet:

```
$ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

$ckt.Sku.Family = "UnlimitedData"
$ckt.sku.Name = "Premium_UnlimitedData"

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To control access to the classic and Resource Manager environments

Review the instructions in Move ExpressRoute circuits from the classic to the Resource Manager deployment model.

Deprovisioning and deleting an ExpressRoute circuit

Note the following:

- You must unlink all virtual networks from the ExpressRoute circuit. If this operation fails, check to see if any virtual networks are linked to the circuit.
- If the ExpressRoute circuit service provider provisioning state is **Provisioning** or **Provisioned** you must work with your service provider to deprovision the circuit on their side. We will continue to reserve resources and bill you until the service provider completes deprovisioning the circuit and notifies us.
- If the service provider has deprovisioned the circuit (the service provider provisioning state is set to **Not provisioned**) you can then delete the circuit. This will stop billing for the circuit

You can delete your ExpressRoute circuit by running the following command:

```
Remove-AzureRmExpressRouteCircuit -ResourceGroupName "ExpressRouteResourceGroup" -Name
"ExpressRouteARMCircuit"
```

Next steps

After you create your circuit, make sure that you do the following:

- Create and modify routing for your ExpressRoute circuit
- Link your virtual network to your ExpressRoute circuit

Create and modify an ExpressRoute circuit

1/17/2017 • 9 min to read • Edit on GitHub

This article walks you through the steps to create an Azure ExpressRoute circuit by using PowerShell cmdlets and the classic deployment model. This article will also show you how to check the status, update, or delete and deprovision an ExpressRoute circuit.

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.

Before you begin

1. Review the prerequisites and workflow articles

Make sure that you have reviewed the prerequisites and workflows before you begin configuration.

2. Install the latest versions of the Azure PowerShell modules

Follow the instructions in How to install and configure Azure PowerShell for step-by-step guidance on how to configure your computer to use the Azure PowerShell modules.

3. Log in to your Azure account and select a subscription

1. Run the following cmdlet using an elevated Windows PowerShell prompt:

```
Add-AzureAccount
```

2. In the sign-in screen that appears, sign in to your account.
3. Get a list of your subscriptions.

```
Get-AzureSubscription
```

4. Select the subscription that you want to use.

```
Select-AzureSubscription -SubscriptionName "mysubscriptionname"
```

Create and provision an ExpressRoute circuit

1. Import the PowerShell modules for ExpressRoute

If you have not already done so, you must import the Azure and ExpressRoute modules into the PowerShell session in order to start using the ExpressRoute cmdlets. You import the modules from the location that they were installed to on your local computer. Depending on the method you used to install the modules, the location may be different than the following example shows. Modify the example if necessary.

```
Import-Module 'C:\Program Files (x86)\Microsoft SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1'
Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\ExpressRoute\ExpressRoute.psd1'
```

2. Get the list of supported providers, locations, and bandwidths

Before you create an ExpressRoute circuit, you need the list of supported connectivity providers, locations, and bandwidth options.

The PowerShell cmdlet `Get-AzureDedicatedCircuitServiceProvider` returns this information, which you'll use in later steps:

```
Get-AzureDedicatedCircuitServiceProvider
```

Check to see if your connectivity provider is listed there. Make a note of the following information because you'll need it later when you create a circuit:

- Name
- PeeringLocations
- BandwidthsOffered

You're now ready to create an ExpressRoute circuit.

3. Create an ExpressRoute circuit

The following example shows how to create a 200-Mbps ExpressRoute circuit through Equinix in Silicon Valley. If you're using a different provider and different settings, substitute that information when you make your request.

> **IMPORTANT**
> Your ExpressRoute circuit will be billed from the moment a service key is issued. Ensure that you perform this operation when the connectivity provider is ready to provision the circuit.

The following is an example request for a new service key:

```
$Bandwidth = 200
$CircuitName = "MyTestCircuit"
$ServiceProvider = "Equinix"
$Location = "Silicon Valley"

New-AzureDedicatedCircuit -CircuitName $CircuitName -ServiceProviderName $ServiceProvider -Bandwidth
$Bandwidth -Location $Location -sku Standard -BillingType MeteredData
```

Or, if you want to create an ExpressRoute circuit with the premium add-on, use the following example. Refer to the ExpressRoute FAQ for more details about the premium add-on.

```
New-AzureDedicatedCircuit -CircuitName $CircuitName -ServiceProviderName $ServiceProvider -Bandwidth
$Bandwidth -Location $Location -sku Premium - BillingType MeteredData
```

The response will contain the service key. You can get detailed descriptions of all the parameters by running the following:

```
get-help new-azurededicatedcircuit -detailed
```

4. List all the ExpressRoute circuits

You can run the `Get-AzureDedicatedCircuit` command to get a list of all the ExpressRoute circuits that you created:

```
Get-AzureDedicatedCircuit
```

The response will be something similar to the following example:

```
Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : NotProvisioned
Sku                             : Standard
Status                          : Enabled
```

You can retrieve this information at any time by using the `Get-AzureDedicatedCircuit` cmdlet. Making the call without any parameters lists all the circuits. Your service key will be listed in the *ServiceKey* field.

```
Get-AzureDedicatedCircuit

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : NotProvisioned
Sku                             : Standard
Status                          : Enabled
```

You can get detailed descriptions of all the parameters by running the following:

```
get-help get-azurededicatedcircuit -detailed
```

5. Send the service key to your connectivity provider for provisioning

ServiceProviderProvisioningState provides information on the current state of provisioning on the service-provider side. *Status* provides the state on the Microsoft side. For more information about circuit provisioning states, see the Workflows article.

When you create a new ExpressRoute circuit, the circuit will be in the following state:

```
ServiceProviderProvisioningState : NotProvisioned
Status                          : Enabled
```

The circuit will go to the following state when the connectivity provider is in the process of enabling it for you:

```
ServiceProviderProvisioningState : Provisioning
Status                          : Enabled
```

An ExpressRoute circuit must be in the following state for you to be able to use it:

```
ServiceProviderProvisioningState : Provisioned
Status                          : Enabled
```

6. Periodically check the status and the state of the circuit key

This lets you know when your provider has enabled your circuit. After the circuit has been configured, *ServiceProviderProvisioningState* will display as *Provisioned* as shown in the following example:

```
Get-AzureDedicatedCircuit

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

7. Create your routing configuration

Refer to the ExpressRoute circuit routing configuration (create and modify circuit peerings) article for step-by-step instructions.

> **IMPORTANT**
> These instructions only apply to circuits that are created with service providers that offer layer 2 connectivity services. If you're using a service provider that offers managed layer 3 services (typically an IP VPN, like MPLS), your connectivity provider will configure and manage routing for you.

8. Link a virtual network to an ExpressRoute circuit

Next, link a virtual network to your ExpressRoute circuit. Refer to Linking ExpressRoute circuits to virtual networks for step-by-step instructions. If you need to create a virtual network by using the classic deployment model for ExpressRoute, see Create a virtual network for ExpressRoute for instructions.

Getting the status of an ExpressRoute circuit

You can retrieve this information at any time by using the `Get-AzureCircuit` cmdlet. Making the call without any parameters lists all the circuits.

```
Get-AzureDedicatedCircuit

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled

Bandwidth                       : 1000
CircuitName                     : MyAsiaCircuit
Location                        : Singapore
ServiceKey                      : ################################
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

You can get information on a specific ExpressRoute circuit by passing the service key as a parameter to the call:

```
Get-AzureDedicatedCircuit -ServiceKey "********************************"

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

You can get detailed descriptions of all the parameters by running the following:

```
get-help get-azurededicatedcircuit -detailed
```

Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity.

You can do the following with no downtime:

- Enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit. Note that downgrading the bandwidth of a circuit is not supported.
- Change the metering plan from Metered Data to Unlimited Data. Note that changing the metering plan from Unlimited Data to Metered Data is not supported.
- You can enable and disable *Allow Classic Operations*.

Refer to the ExpressRoute FAQ for more information on limits and limitations.

To enable the ExpressRoute premium add-on

You can enable the ExpressRoute premium add-on for your existing circuit by using the following PowerShell cmdlet:

```
Set-AzureDedicatedCircuitProperties -ServiceKey "********************************" -Sku Premium

Bandwidth                       : 1000
CircuitName                     : TestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Premium
Status                          : Enabled
```

Your circuit will now have the ExpressRoute premium add-on features enabled. Note that we will start billing you for the premium add-on capability as soon as the command has successfully run.

To disable the ExpressRoute premium add-on

> **IMPORTANT**
> This operation can fail if you're using resources that are greater than what is permitted for the standard circuit.

Note the following:

- You must ensure that the number of virtual networks linked to the circuit is less than 10 before you downgrade from premium to standard. If you don't do this, your update request will fail, and you'll be billed

the premium rates.

- You must unlink all virtual networks in other geopolitical regions. If you don't do this, your update request will fail, and you'll be billed the premium rates.
- Your route table must be less than 4,000 routes for private peering. If your route table size is greater than 4,000 routes, the BGP session will drop and won't be reenabled until the number of advertised prefixes goes below 4,000.

You can disable the ExpressRoute premium add-on for your existing circuit by using the following PowerShell cmdlet:

```
Set-AzureDedicatedCircuitProperties -ServiceKey "**********************************" -Sku Standard

Bandwidth                       : 1000
CircuitName                     : TestCircuit
Location                        : Silicon Valley
ServiceKey                      : **********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

To update the ExpressRoute circuit bandwidth

Check the ExpressRoute FAQ for supported bandwidth options for your provider. You can pick any size that is greater than the size of your existing circuit as long as the physical port (on which your circuit is created) allows.

> **IMPORTANT**
> You cannot reduce the bandwidth of an ExpressRoute circuit without disruption. Downgrading bandwidth will require you to deprovision the ExpressRoute circuit and then reprovision a new ExpressRoute circuit.

After you decide what size you need, you can use the following command to resize your circuit:

```
Set-AzureDedicatedCircuitProperties -ServiceKey ********************************** -Bandwidth 1000

Bandwidth                       : 1000
CircuitName                     : TestCircuit
Location                        : Silicon Valley
ServiceKey                      : **********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

Your circuit will have been sized up on the Microsoft side. You must contact your connectivity provider to update configurations on their side to match this change. Note that we will start billing you for the updated bandwidth option from this point on.

If you see the following error when increasing the circuit bandwidth, it means there is no sufficient bandwidth left on the physical port where your existing circuit is created. You have to delete this circuit and create a new circuit of the size you need.

```
Set-AzureDedicatedCircuitProperties : InvalidOperation : Insufficient bandwidth available to perform this
circuit
update operation
At line:1 char:1
+ Set-AzureDedicatedCircuitProperties -ServiceKey ********************* ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : CloseError: (:) [Set-AzureDedicatedCircuitProperties], CloudException
    + FullyQualifiedErrorId :
Microsoft.WindowsAzure.Commands.ExpressRoute.SetAzureDedicatedCircuitPropertiesCommand
```

Deprovisioning and deleting an ExpressRoute circuit

Note the following:

- You must unlink all virtual networks from the ExpressRoute circuit for this operation to succeed. Check to see if you have any virtual networks that are linked to the circuit if this operation fails.
- If the ExpressRoute circuit service provider provisioning state is **Provisioning** or **Provisioned** you must work with your service provider to deprovision the circuit on their side. We will continue to reserve resources and bill you until the service provider completes deprovisioning the circuit and notifies us.
- If the service provider has deprovisioned the circuit (the service provider provisioning state is set to **Not provisioned**) you can then delete the circuit. This will stop billing for the circuit.

You can delete your ExpressRoute circuit by running the following command:

```
Remove-AzureDedicatedCircuit -ServiceKey "*********************************"
```

Next steps

After you create your circuit, make sure that you do the following:

- Create and modify routing for your ExpressRoute circuit
- Link your virtual network to your ExpressRoute circuit

Create and modify routing for an ExpressRoute circuit

1/17/2017 • 6 min to read • Edit on GitHub

This article walks you through the steps to create and manage routing configuration for an ExpressRoute circuit using the Azure portal and 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.

Configuration prerequisites

- Make sure that you have reviewed the prerequisites page, the routing requirements page, and the workflows page before you begin configuration.
- You must have an active ExpressRoute circuit. Follow the instructions to Create an ExpressRoute circuit and have the circuit enabled by your connectivity provider before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state for you to be able to run the cmdlets described below.

These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider offering managed Layer 3 services (typically an IPVPN, like MPLS), your connectivity provider will configure and manage routing for you.

> **IMPORTANT**
> We currently do not advertise peerings configured by service providers through the service management portal. We are working on enabling this capability soon. Please check with your service provider before configuring BGP peerings.

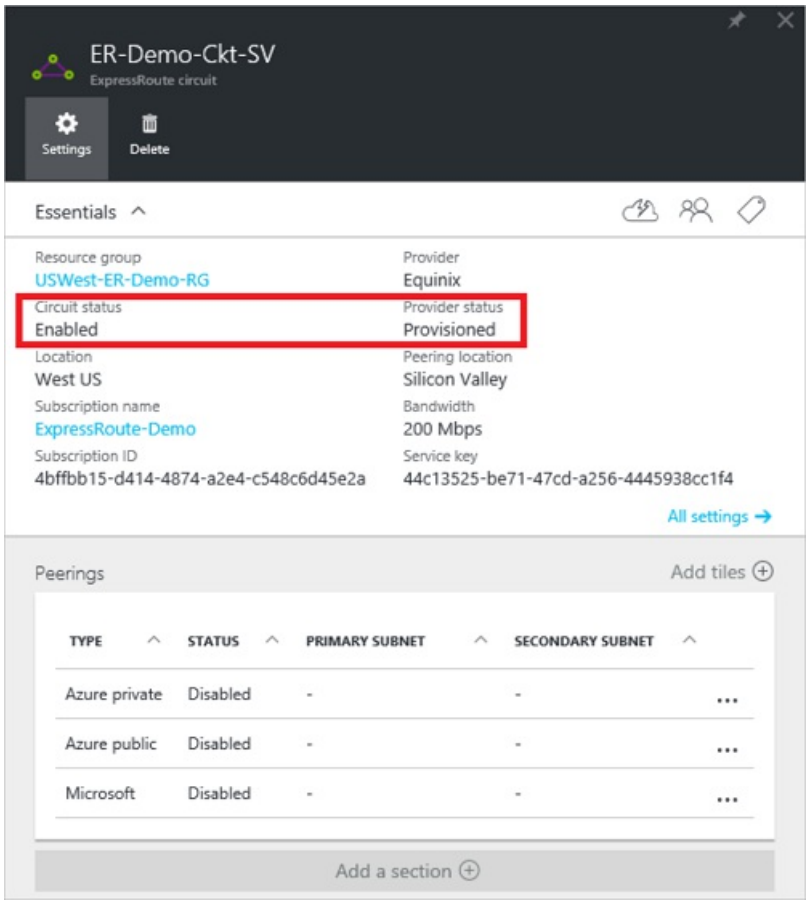
You can configure one, two, or all three peerings (Azure private, Azure public and Microsoft) for an ExpressRoute circuit. You can configure peerings in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time.

Azure private peering

This section provides instructions on how to create, get, update, and delete the Azure private peering configuration for an ExpressRoute circuit.

To create Azure private peering

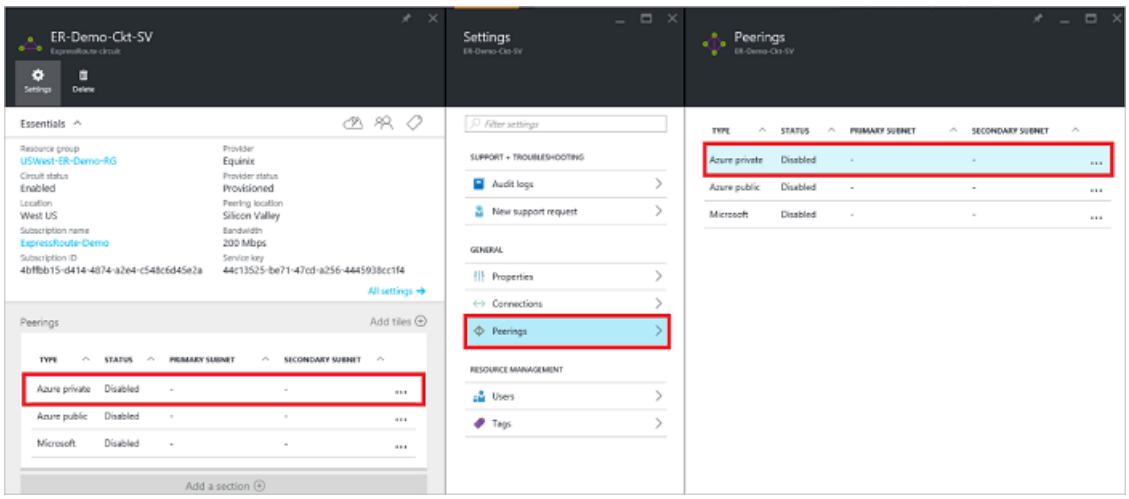
1. Configure the ExpressRoute circuit. Ensure that the circuit is fully provisioned by the connectivity provider before continuing.



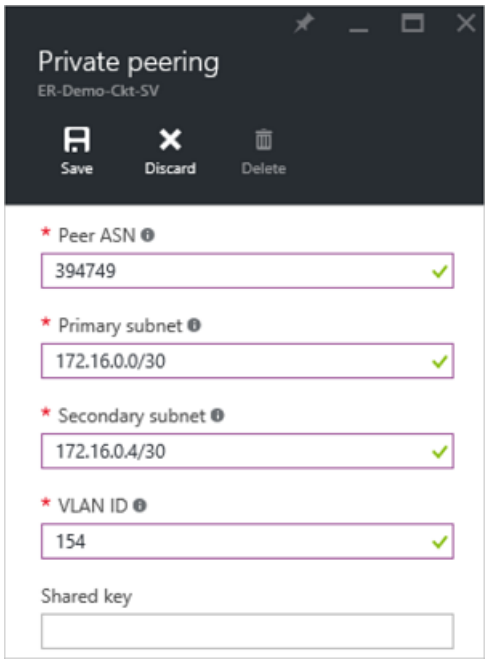
2. Configure Azure private peering for the circuit. Make sure that you have the following items before you proceed with the next steps:

   - A /30 subnet for the primary link. This must not be part of any address space reserved for virtual networks.
   - A /30 subnet for the secondary link. This must not be part of any address space reserved for virtual networks.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers. You can use a private AS number for this peering. Ensure that you are not using 65515.
   - An MD5 hash if you choose to use one. **This is optional**.

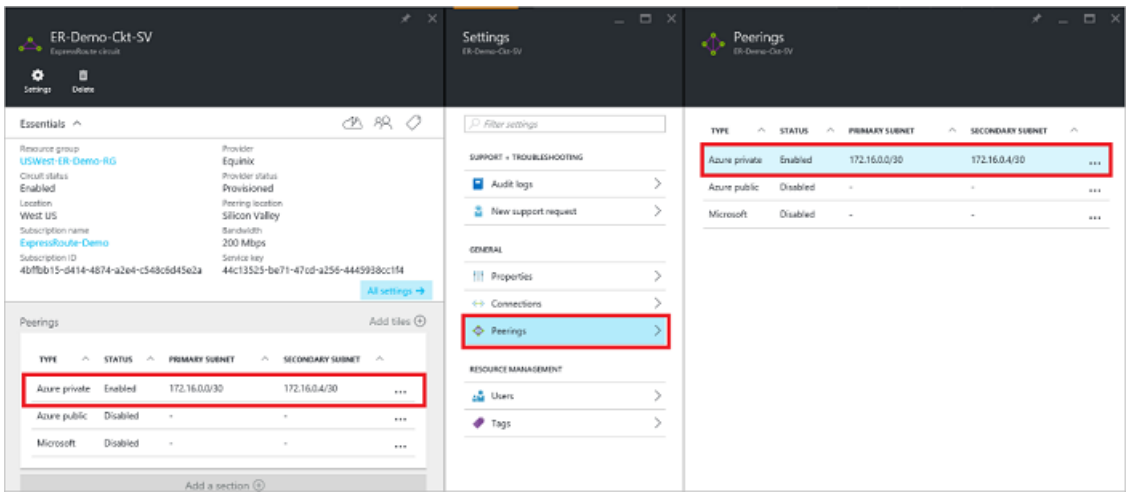
3. Select the Azure Private peering row, as shown below.



4. Configure private peering. The image below shows a configuration example.

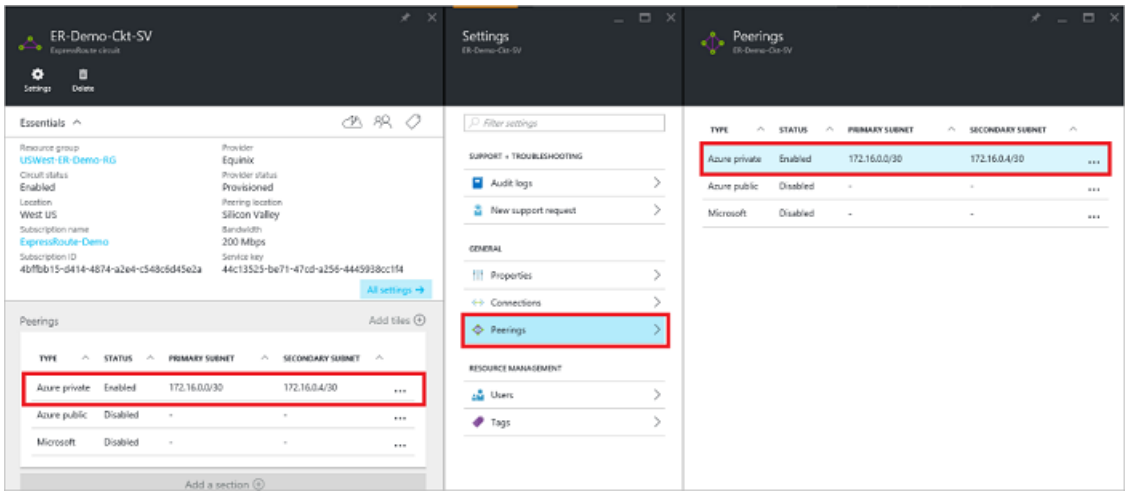


5. Save the configuration once you have specified all parameters. Once the configuration has been accepted successfully, you will see something similar to the example below.



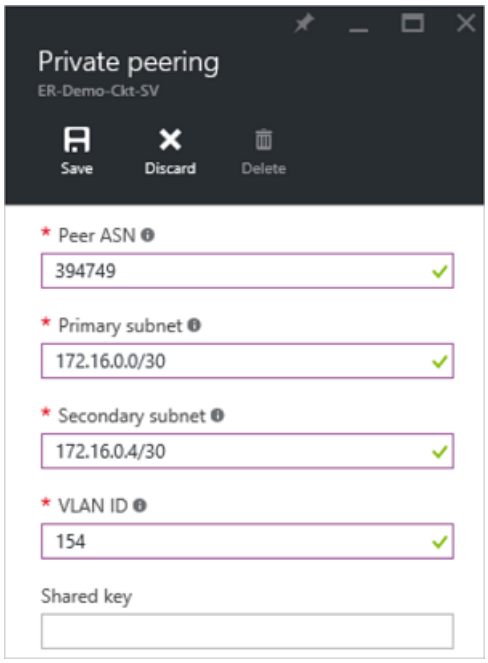
To view Azure private peering details

You can view the properties of Azure private peering by selecting the peering.



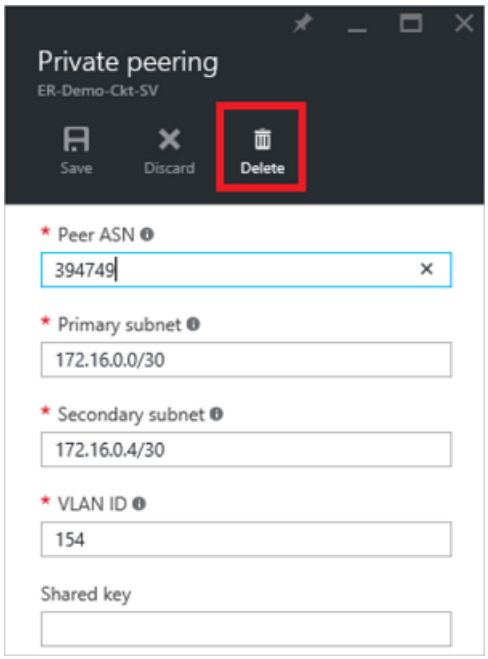
To update Azure private peering configuration

You can select the row for peering and modify the peering properties.



To delete Azure private peering

You can remove your peering configuration by selecting the delete icon as shown below.

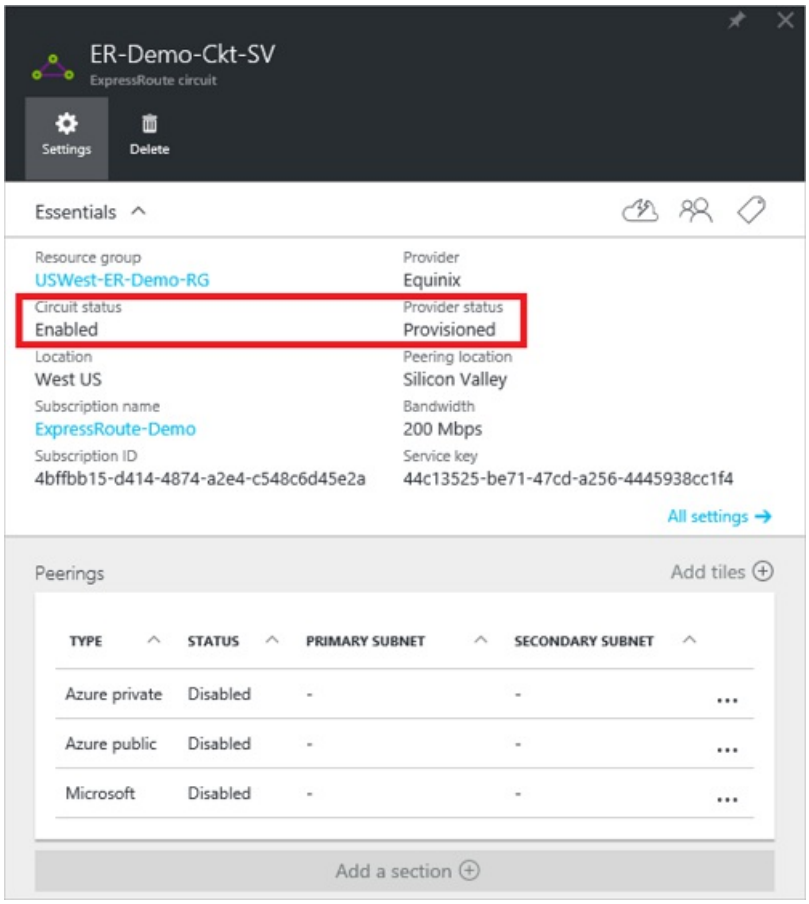


Azure public peering

This section provides instructions on how to create, get, update, and delete the Azure public peering configuration for an ExpressRoute circuit.

To create Azure public peering

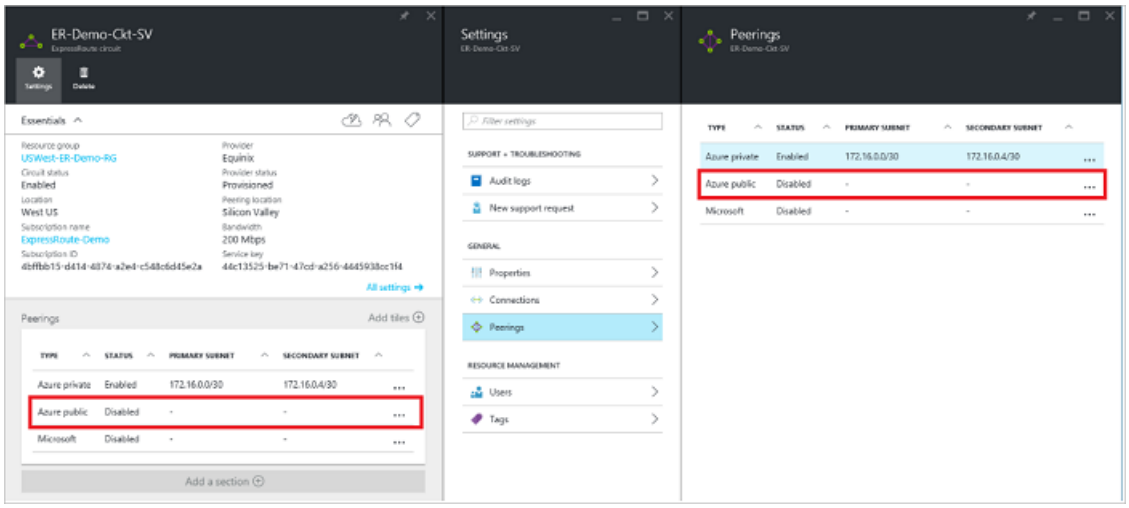
1. Configure ExpressRoute circuit. Ensure that the circuit is fully provisioned by the connectivity provider before continuing further.



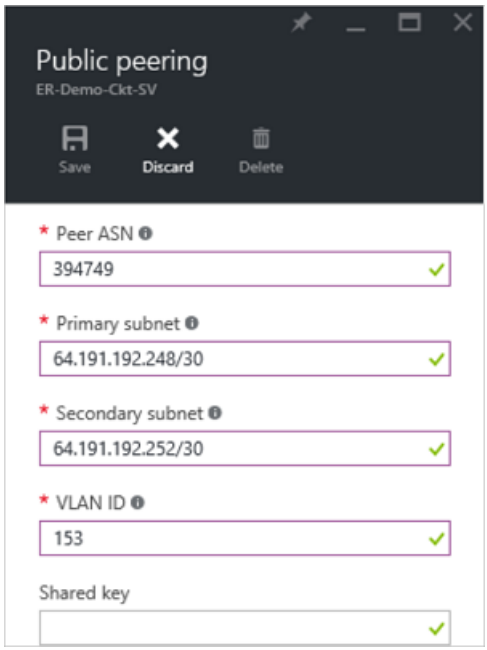
2. Configure Azure public peering for the circuit. Make sure that you have the following items before you proceed with the next steps:

   - A /30 subnet for the primary link.
   - A /30 subnet for the secondary link.
   - All IP addresses used to setup this peering must be valid public IPv4 addresses.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.
   - An MD5 hash if you choose to use one. **This is optional**.

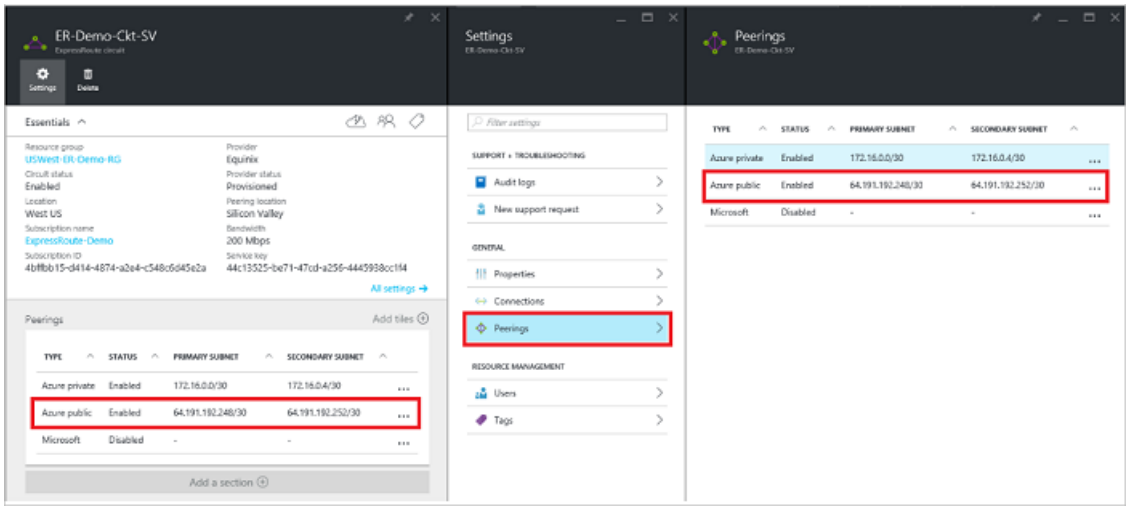
3. Select the Azure public peering row, as shown below.



4. Configure public peering. The image below shows a configuration example.

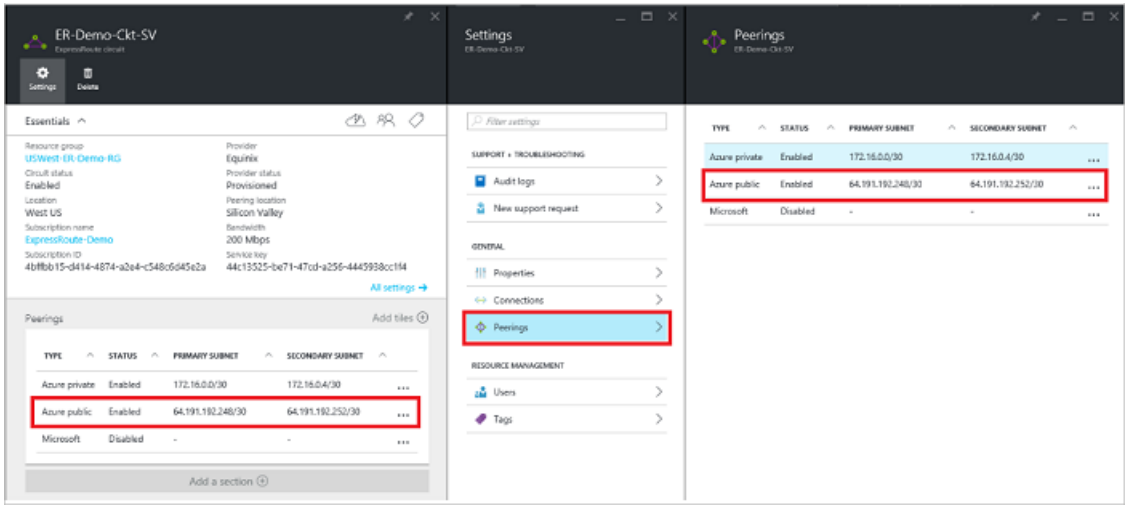


5. Save the configuration once you have specified all parameters. Once the configuration has been accepted successfully, you will see something similar to the example below.



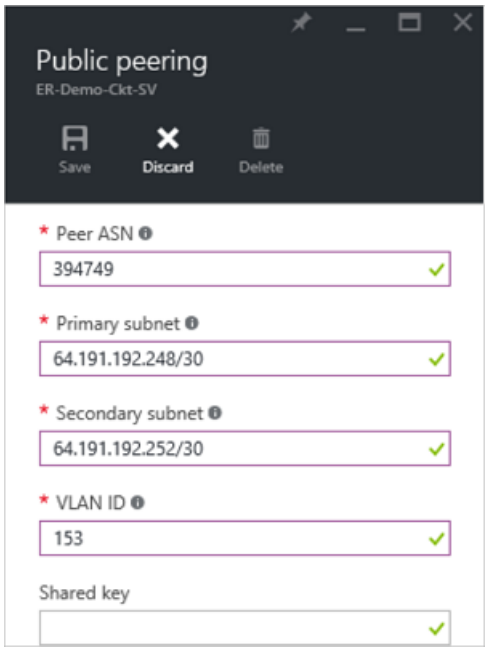
To view Azure public peering details

You can view the properties of Azure public peering by selecting the peering.



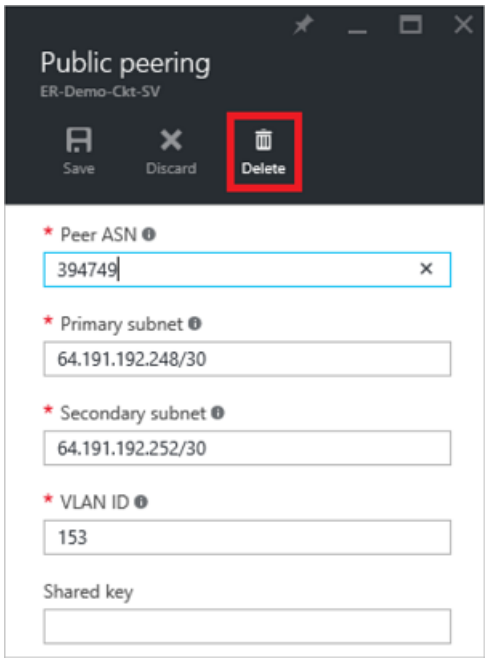
To update Azure public peering configuration

You can select the row for peering and modify the peering properties.



To delete Azure public peering

You can remove your peering configuration by selecting the delete icon as shown below.

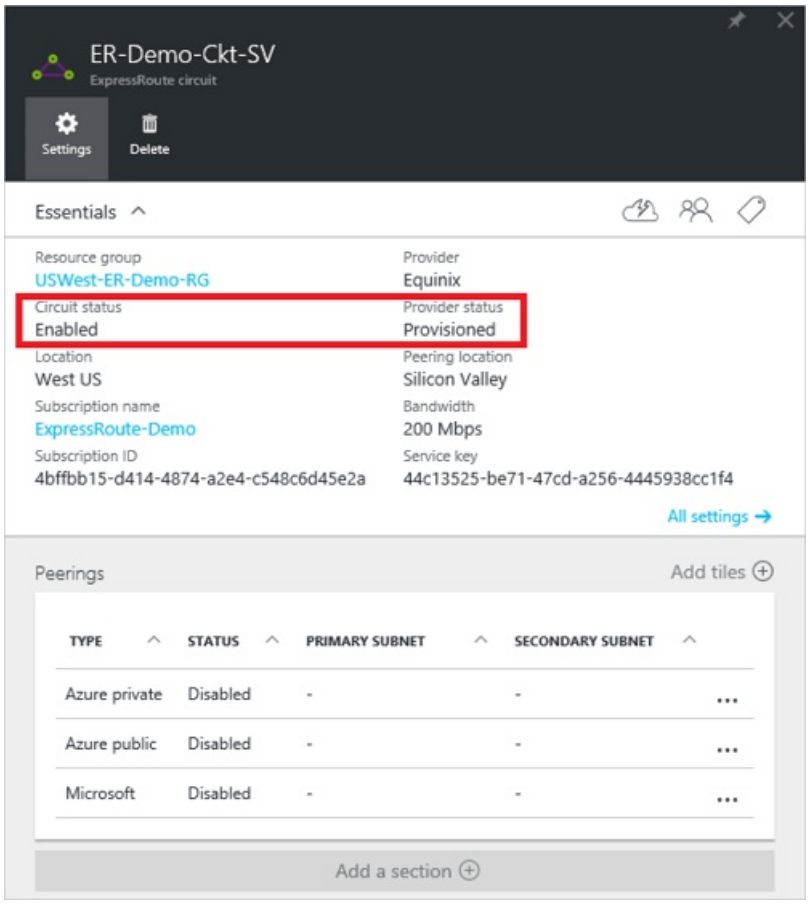


Microsoft peering

This section provides instructions on how to create, get, update, and delete the Microsoft peering configuration for an ExpressRoute circuit.

To create Microsoft peering

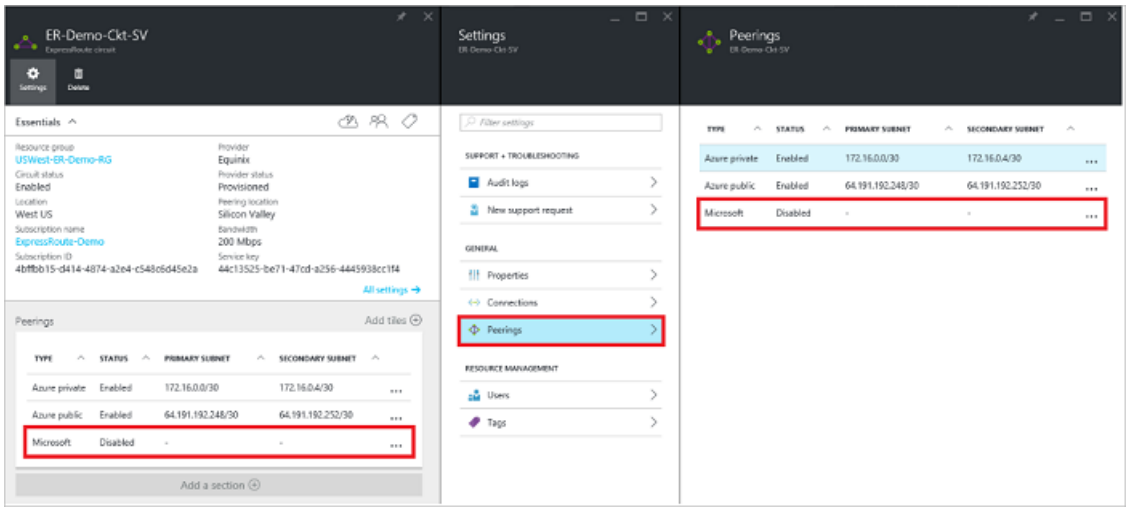
1. Configure ExpressRoute circuit. Ensure that the circuit is fully provisioned by the connectivity provider before continuing further.



2. Configure Microsoft peering for the circuit. Make sure that you have the following information before you proceed.

   - A /30 subnet for the primary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.
   - **Advertised prefixes:** You must provide a list of all prefixes you plan to advertise over the BGP session. Only public IP address prefixes are accepted. You can send a comma separated list if you plan to send a set of prefixes. These prefixes must be registered to you in an RIR / IRR.
   - **Customer ASN:** If you are advertising prefixes that are not registered to the peering AS number, you can specify the AS number to which they are registered. **This is optional**.
   - **Routing Registry Name:** You can specify the RIR / IRR against which the AS number and prefixes are registered. **This is optional.**
   - An MD5 hash, if you choose to use one. **This is optional.**

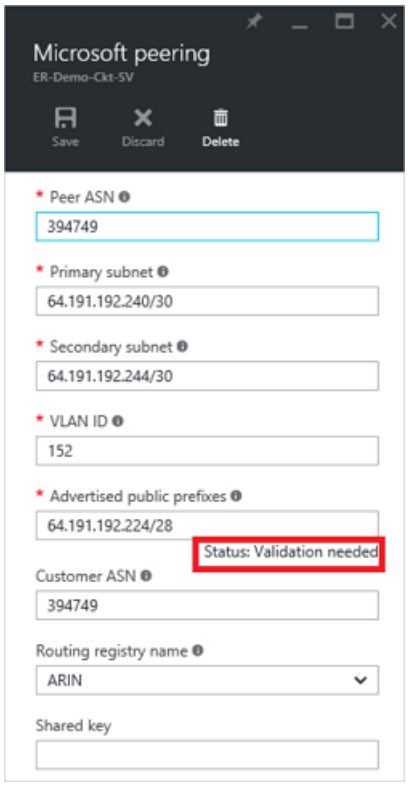
3. You can select the peering you wish to configure as shown below. Select the Microsoft peering row.



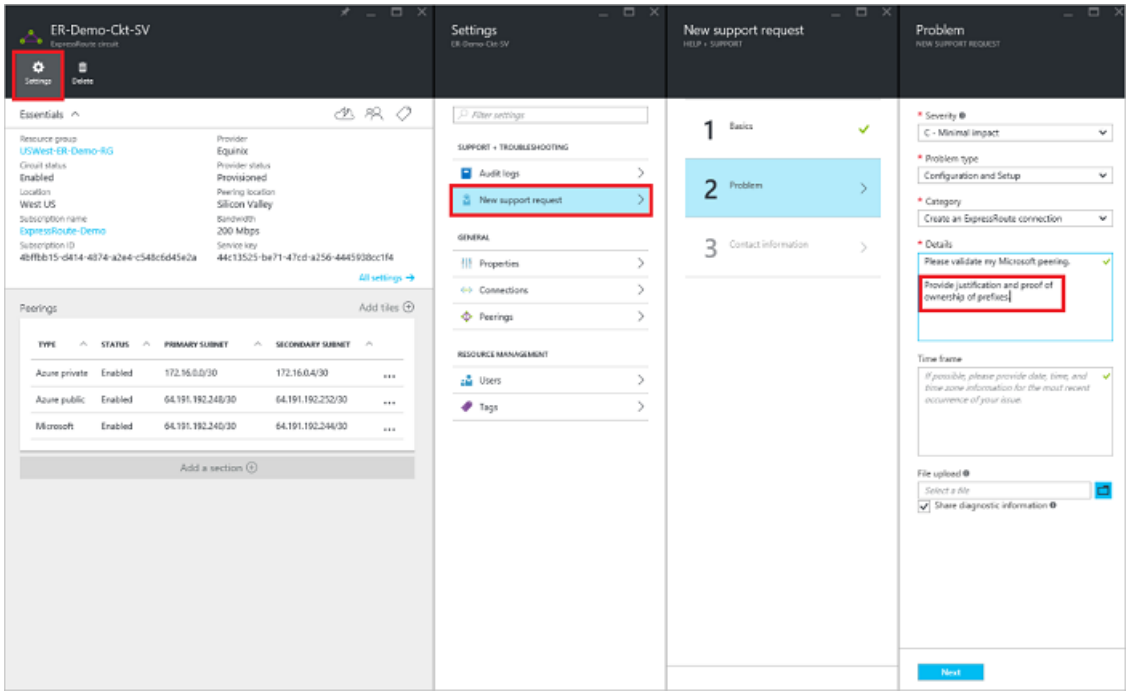
4. Configure Microsoft peering. The image below shows a configuration example.

5. Save the configuration once you have specified all parameters.

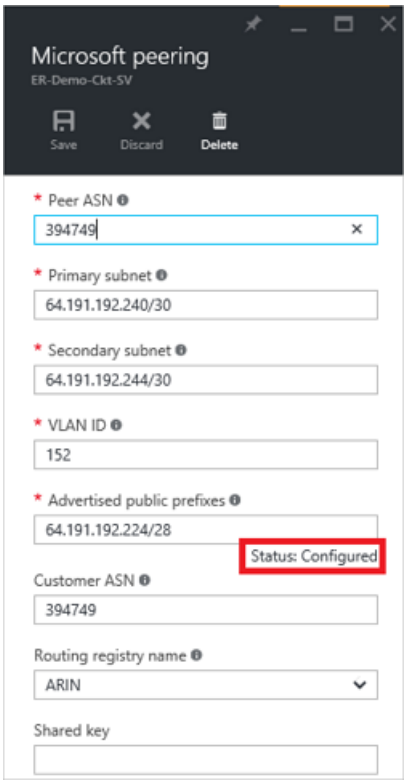
   If your circuit gets to a validation needed state (as shown below), you must open a support ticket to show proof of ownership of the prefixes to our support team.



You can open a support ticket directly from the portal as shown below

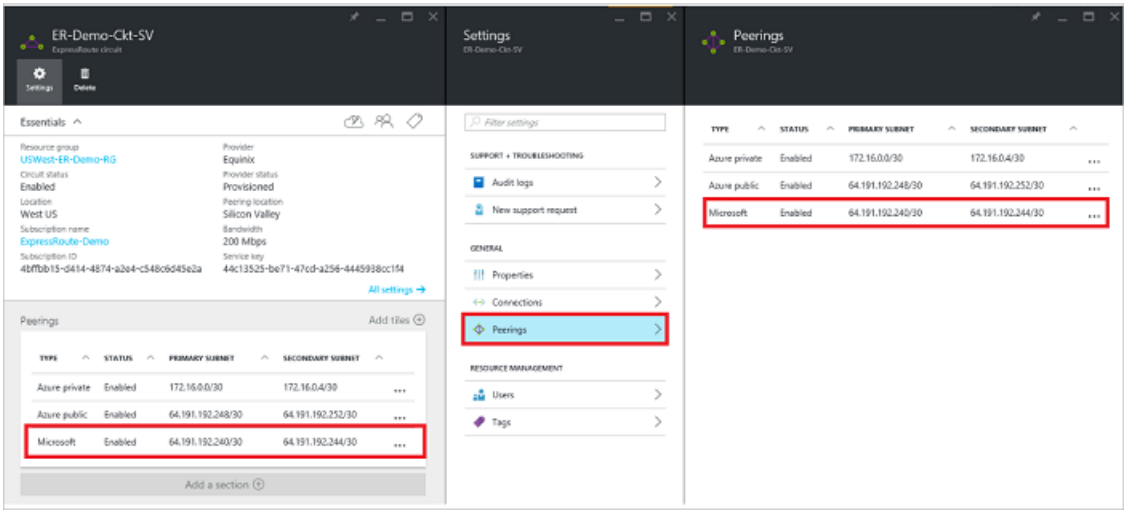


6. Once the configuration has been accepted successfully, you will see something similar to the example below.



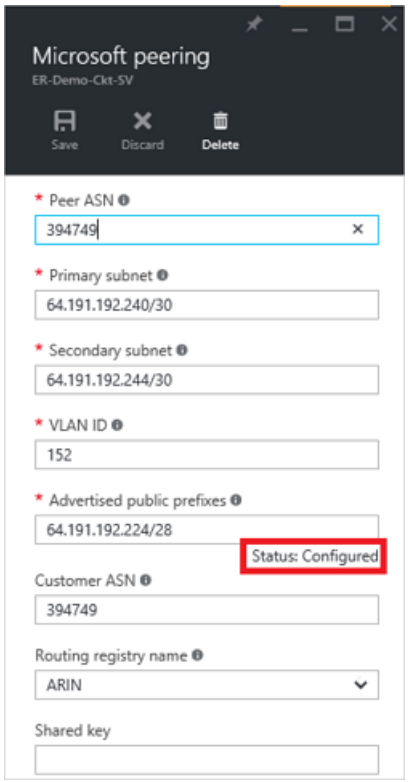
To view Microsoft peering details

You can view the properties of Azure public peering by selecting the peering.



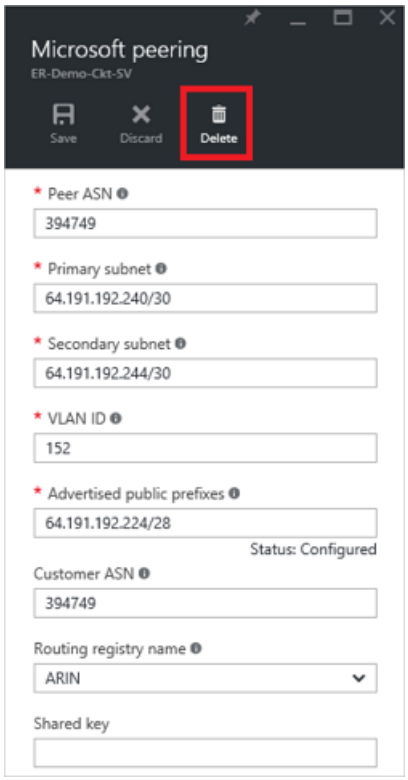
To update Microsoft peering configuration

You can select the row for peering and modify the peering properties.



To delete Microsoft peering

You can remove your peering configuration by selecting the delete icon as shown below.



Next steps

Next step, Link a VNet to an ExpressRoute circuit.

- For more information about ExpressRoute workflows, see ExpressRoute workflows.
- For more information about circuit peering, see ExpressRoute circuits and routing domains.
- For more information about working with virtual networks, see Virtual network overview.

Create and modify routing for an ExpressRoute circuit

1/17/2017 • 10 min to read • Edit on GitHub

This article walks you through the steps to create and manage routing configuration for an ExpressRoute circuit using PowerShell and the Azure Resource Manager deployment model. The steps below will also show you how to check the status, update, or delete and deprovision peerings for an ExpressRoute circuit.

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 prerequisites

- You will need the latest version of the Azure PowerShell modules, version 1.0 or later.
- Make sure that you have reviewed the prerequisites page, the routing requirements page, and the workflows page before you begin configuration.
- You must have an active ExpressRoute circuit. Follow the instructions to Create an ExpressRoute circuit and have the circuit enabled by your connectivity provider before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state for you to be able to run the cmdlets described below.

These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider offering managed Layer 3 services (typically an IPVPN, like MPLS), your connectivity provider will configure and manage routing for you.

> **IMPORTANT**
> We currently do not advertise peerings configured by service providers through the service management portal. We are working on enabling this capability soon. Please check with your service provider before configuring BGP peerings.

You can configure one, two, or all three peerings (Azure private, Azure public and Microsoft) for an ExpressRoute circuit. You can configure peerings in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time.

Azure private peering

This section provides instructions on how to create, get, update, and delete the Azure private peering configuration for an ExpressRoute circuit.

To create Azure private peering

1. Import the PowerShell module for ExpressRoute.

   You must install the latest PowerShell installer from PowerShell Gallery and import the Azure Resource Manager modules into the PowerShell session in order to start using the ExpressRoute cmdlets. You will need to run PowerShell as an Administrator.

```
Install-Module AzureRM

Install-AzureRM
```

Import all of the AzureRM.* modules within the known semantic version range

```
Import-AzureRM
```

You can also just import a select module within the known semantic version range

```
Import-Module AzureRM.Network
```

Logon to your account

```
Login-AzureRmAccount
```

Select the subscription you want to create ExpressRoute circuit

```
Select-AzureRmSubscription -SubscriptionId "<subscription ID>"
```

2. Create an ExpressRoute circuit.

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider.

   If your connectivity provider offers managed Layer 3 services, you can request your connectivity provider to enable Azure private peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. Check the ExpressRoute circuit to ensure it is provisioned.

   You must first check to see if the ExpressRoute circuit is Provisioned and also Enabled. See the example below.

   ```
   Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
  "ExpressRouteResourceGroup"
   ```

   The response will be something similar to the example below:

```
Name                             : ExpressRouteARMCircuit
ResourceGroupName                : ExpressRouteResourceGroup
Location                         : westus
Id                               :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.
Network/expressRouteCircuits/ExpressRouteARMCircuit
Etag                             : W/"################################"
ProvisioningState                : Succeeded
Sku                              : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                   }
CircuitProvisioningState         : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes             :
ServiceProviderProperties        : {
                                     "ServiceProviderName": "Equinix",
                                     "PeeringLocation": "Silicon Valley",
                                     "BandwidthInMbps": 200
                                   }
ServiceKey                       : ************************************
Peerings                         : []
```

4. Configure Azure private peering for the circuit.

   Make sure that you have the following items before you proceed with the next steps:

   - A /30 subnet for the primary link. This must not be part of any address space reserved for virtual networks.
   - A /30 subnet for the secondary link. This must not be part of any address space reserved for virtual networks.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers. You can use a private AS number for this peering. Ensure that you are not using 65515.
   - An MD5 hash if you choose to use one. **This is optional**.

     You can run the following cmdlet to configure Azure private peering for your circuit.

     Add-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -Circuit $ckt -PeeringType AzurePrivatePeering -PeerASN 100 -PrimaryPeerAddressPrefix "10.0.0.0/30" -SecondaryPeerAddressPrefix "10.0.0.4/30" -VlanId 200

     Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt

     You can use the cmdlet below if you choose to use an MD5 hash.

     Add-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -Circuit $ckt -PeeringType AzurePrivatePeering -PeerASN 100 -PrimaryPeerAddressPrefix "10.0.0.0/30" -SecondaryPeerAddressPrefix "10.0.0.4/30" -VlanId 200 -SharedKey "A1B2C3D4"

     Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt

     > **IMPORTANT**
     > Ensure that you specify your AS number as peering ASN, not customer ASN.

To view Azure private peering details

You can get configuration details using the following cmdlet

```
    $ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

    Get-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -Circuit $ckt
```

To update Azure private peering configuration

You can update any part of the configuration using the following cmdlet. In the example below, the VLAN ID of the circuit is being updated from 100 to 500.

```
Set-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -ExpressRouteCircuit $ckt -PeeringType
AzurePrivatePeering -PeerASN 100 -PrimaryPeerAddressPrefix "10.0.0.0/30" -SecondaryPeerAddressPrefix
"10.0.0.4/30" -VlanId 200

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To delete Azure private peering

You can remove your peering configuration by running the following cmdlet.

> **WARNING**
> You must ensure that all virtual networks are unlinked from the ExpressRoute circuit before running this cmdlet.

```
Remove-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -ExpressRouteCircuit $ckt
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Azure public peering

This section provides instructions on how to create, get, update and delete the Azure public peering configuration for an ExpressRoute circuit.

To create Azure public peering

1. Import the PowerShell module for ExpressRoute.

   You must install the latest PowerShell installer from PowerShell Gallery and import the Azure Resource Manager modules into the PowerShell session in order to start using the ExpressRoute cmdlets. You will need to run PowerShell as an Administrator.

   ```
   Install-Module AzureRM

   Install-AzureRM
   ```

   Import all of the AzureRM.* modules within the known semantic version range

   ```
   Import-AzureRM
   ```

   You can also just import a select module within the known semantic version range

   ```
   Import-Module AzureRM.Network
   ```

   Logon to your account

```
Login-AzureRmAccount
```

Select the subscription you want to create ExpressRoute circuit

```
Select-AzureRmSubscription -SubscriptionId "<subscription ID>"
```

2. Create an ExpressRoute circuit.

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider.

   If your connectivity provider offers managed Layer 3 services, you can request your connectivity provider to enable Azure public peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. Check ExpressRoute circuit to ensure it is provisioned.

   You must first check to see if the ExpressRoute circuit is Provisioned and also Enabled. See the example below.

```
 Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"
```

   The response will be something similar to the example below:

```
 Name                            : ExpressRouteARMCircuit
 ResourceGroupName               : ExpressRouteResourceGroup
 Location                        : westus
 Id                              :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.
Network/expressRouteCircuits/ExpressRouteARMCircuit
 Etag                            : W/"################################"
 ProvisioningState               : Succeeded
 Sku                             : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                   }
 CircuitProvisioningState        : Enabled
 ServiceProviderProvisioningState : Provisioned
 ServiceProviderNotes            :
 ServiceProviderProperties       : {
                                     "ServiceProviderName": "Equinix",
                                     "PeeringLocation": "Silicon Valley",
                                     "BandwidthInMbps": 200
                                   }
 ServiceKey                      : ************************************
 Peerings                        : []
```

4. Configure Azure public peering for the circuit.

   Ensure that you have the following information before you proceed further.

   - A /30 subnet for the primary link. This must be a valid public IPv4 prefix.
   - A /30 subnet for the secondary link. This must be a valid public IPv4 prefix.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.

- An MD5 hash if you choose to use one. **This is optional**.

  You can run the following cmdlet to configure Azure public peering for your circuit

  Add-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt -PeeringType AzurePublicPeering -PeerASN 100 -PrimaryPeerAddressPrefix "12.0.0.0/30" -SecondaryPeerAddressPrefix "12.0.0.4/30" -VlanId 100

  Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt

  You can use the cmdlet below if you choose to use an MD5 hash

  Add-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt -PeeringType AzurePublicPeering -PeerASN 100 -PrimaryPeerAddressPrefix "12.0.0.0/30" -SecondaryPeerAddressPrefix "12.0.0.4/30" -VlanId 100 -SharedKey "A1B2C3D4"

  Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt

  > **IMPORTANT**
  > Ensure that you specify your AS number as peering ASN, not customer ASN.

To view Azure public peering details

You can get configuration details using the following cmdlet

```
    $ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

    Get-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -Circuit $ckt
```

To update Azure public peering configuration

You can update any part of the configuration using the following cmdlet

```
Set-AzureRmExpressRouteCircuitPeeringConfig  -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType
MicrosoftPeering -PeerASN 100 -PrimaryPeerAddressPrefix "123.0.0.0/30" -SecondaryPeerAddressPrefix
"123.0.0.4/30" -VlanId 600

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The VLAN ID of the circuit is being updated from 200 to 600 in the above example.

To delete Azure public peering

You can remove your peering configuration by running the following cmdlet

```
Remove-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Microsoft peering

This section provides instructions on how to create, get, update and delete the Microsoft peering configuration for an ExpressRoute circuit.

To create Microsoft peering

1. Import the PowerShell module for ExpressRoute.

   You must install the latest PowerShell installer from PowerShell Gallery and import the Azure Resource Manager modules into the PowerShell session in order to start using the ExpressRoute cmdlets. You will

need to run PowerShell as an Administrator.

```
Install-Module AzureRM

Install-AzureRM
```

Import all of the AzureRM.* modules within the known semantic version range

```
Import-AzureRM
```

You can also just import a select module within the known semantic version range

```
Import-Module AzureRM.Network
```

Logon to your account

```
Login-AzureRmAccount
```

Select the subscription you want to create ExpressRoute circuit

```
Select-AzureRmSubscription -SubscriptionId "<subscription ID>"
```

2. Create an ExpressRoute circuit.

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider.

   If your connectivity provider offers managed Layer 3 services, you can request your connectivity provider to enable Azure private peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. Check ExpressRoute circuit to ensure it is provisioned.

   You must first check to see if the ExpressRoute circuit is Provisioned and also Enabled. See the example below.

   ```
   Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
  "ExpressRouteResourceGroup"
   ```

   The response will be something similar to the example below:

```
 Name                            : ExpressRouteARMCircuit
 ResourceGroupName               : ExpressRouteResourceGroup
 Location                        : westus
 Id                              :
/subscriptions/**************************/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.
Network/expressRouteCircuits/ExpressRouteARMCircuit
 Etag                            : W/"################################"
 ProvisioningState               : Succeeded
 Sku                             : {
                                     "Name": "Standard_MeteredData",
                                     "Tier": "Standard",
                                     "Family": "MeteredData"
                                   }
 CircuitProvisioningState        : Enabled
 ServiceProviderProvisioningState : Provisioned
 ServiceProviderNotes            :
 ServiceProviderProperties       : {
                                     "ServiceProviderName": "Equinix",
                                     "PeeringLocation": "Silicon Valley",
                                     "BandwidthInMbps": 200
                                   }
 ServiceKey                      : ************************************
 Peerings                        : []
```

4. Configure Microsoft peering for the circuit.

   Make sure that you have the following information before you proceed.

   - A /30 subnet for the primary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.
   - Advertised prefixes: You must provide a list of all prefixes you plan to advertise over the BGP session. Only public IP address prefixes are accepted. You can send a comma separated list if you plan to send a set of prefixes. These prefixes must be registered to you in an RIR / IRR.
   - Customer ASN: If you are advertising prefixes that are not registered to the peering AS number, you can specify the AS number to which they are registered. **This is optional**.
   - Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
   - A MD5 hash, if you choose to use one. **This is optional.**

     You can run the following cmdlet to configure Microsoft peering for your circuit

     Add-AzureRmExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType MicrosoftPeering -PeerASN 100 -PrimaryPeerAddressPrefix "123.0.0.0/30" -SecondaryPeerAddressPrefix "123.0.0.4/30" -VlanId 300 -MicrosoftConfigAdvertisedPublicPrefixes "123.1.0.0/24" -MicrosoftConfigCustomerAsn 23 -MicrosoftConfigRoutingRegistryName "ARIN"

     Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt

To get Microsoft peering details

You can get configuration details using the following cmdlet.

```
    $ckt = Get-AzureRmExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName
"ExpressRouteResourceGroup"

    Get-AzureRmExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt
```

To update Microsoft peering configuration

You can update any part of the configuration using the following cmdlet.

```
    Set-AzureRmExpressRouteCircuitPeeringConfig  -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -
PeeringType MicrosoftPeering -PeerASN 100 -PrimaryPeerAddressPrefix "123.0.0.0/30" -SecondaryPeerAddressPrefix
"123.0.0.4/30" -VlanId 300 -MicrosoftConfigAdvertisedPublicPrefixes "124.1.0.0/24" -MicrosoftConfigCustomerAsn
23 -MicrosoftConfigRoutingRegistryName "ARIN"

    Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To delete Microsoft peering

You can remove your peering configuration by running the following cmdlet.

```
Remove-AzureRmExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt

Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Next steps

Next step, Link a VNet to an ExpressRoute circuit.

- For more information about ExpressRoute workflows, see ExpressRoute workflows.
- For more information about circuit peering, see ExpressRoute circuits and routing domains.
- For more information about working with virtual networks, see Virtual network overview.

Create and modify routing for an ExpressRoute circuit

1/17/2017 • 10 min to read • Edit on GitHub

This article walks you through the steps to create and manage routing configuration for an ExpressRoute circuit using PowerShell and the classic deployment model. The steps below will also show you how to check the status, update, or delete and deprovision peerings for an ExpressRoute circuit.

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 prerequisites

- You will need the latest version of the Azure PowerShell modules. You can download the latest PowerShell module from the PowerShell section of the Azure Downloads page. Follow the instructions in the How to install and configure Azure PowerShell page for step-by-step guidance on how to configure your computer to use the Azure PowerShell modules.
- Make sure that you have reviewed the prerequisites page, the routing requirements page, and the workflows page before you begin configuration.
- You must have an active ExpressRoute circuit. Follow the instructions to create an ExpressRoute circuit and have the circuit enabled by your connectivity provider before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state for you to be able to run the cmdlets described below.

> **IMPORTANT**
> These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider offering managed Layer 3 services (typically an IPVPN, like MPLS), your connectivity provider will configure and manage routing for you.

You can configure one, two, or all three peerings (Azure private, Azure public and Microsoft) for an ExpressRoute circuit. You can configure peerings in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time.

Azure private peering

This section provides instructions on how to create, get, update, and delete the Azure private peering configuration for an ExpressRoute circuit.

To create Azure private peering

1. **Import the PowerShell module for ExpressRoute.**

   You must import the Azure and ExpressRoute modules into the PowerShell session in order to start using the ExpressRoute cmdlets. Run the following commands to import the Azure and ExpressRoute modules into the PowerShell session.

```
 Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1'
 Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\ExpressRoute\ExpressRoute.psd1'
```

2. **Create an ExpressRoute circuit.**

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider. If your connectivity provider offers managed Layer 3 services, you can request your connectivity provider to enable Azure private peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. **Check the ExpressRoute circuit to ensure it is provisioned.**

   You must first check to see if the ExpressRoute circuit is Provisioned and also Enabled. See the example below.

```
PS C:\> Get-AzureDedicatedCircuit -ServiceKey "**********************************"

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

   Make sure that the circuit shows as Provisioned and Enabled. If it doesn't, work with your connectivity provider to get your circuit to the required state and status.

```
ServiceProviderProvisioningState : Provisioned
Status                          : Enabled
```

4. **Configure Azure private peering for the circuit.**

   Make sure that you have the following items before you proceed with the next steps:

   - A /30 subnet for the primary link. This must not be part of any address space reserved for virtual networks.
   - A /30 subnet for the secondary link. This must not be part of any address space reserved for virtual networks.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers. You can use a private AS number for this peering. Ensure that you are not using 65515.
   - An MD5 hash if you choose to use one. **This is optional**.

     You can run the following cmdlet to configure Azure private peering for your circuit.

     New-AzureBGPPeering -AccessType Private -ServiceKey "**********************************" -PrimaryPeerSubnet "10.0.0.0/30" -SecondaryPeerSubnet "10.0.0.4/30" -PeerAsn 1234 -VlanId 100

     You can use the cmdlet below if you choose to use an MD5 hash.

     New-AzureBGPPeering -AccessType Private -ServiceKey "**********************************" -

PrimaryPeerSubnet "10.0.0.0/30" -SecondaryPeerSubnet "10.0.0.4/30" -PeerAsn 1234 -VlanId 100 -SharedKey "A1B2C3D4"

> **IMPORTANT**
> Ensure that you specify your AS number as peering ASN, not customer ASN.

To view Azure private peering details

You can get configuration details using the following cmdlet

```
Get-AzureBGPPeering -AccessType Private -ServiceKey "*********************************"

AdvertisedPublicPrefixes       :
AdvertisedPublicPrefixesState  : Configured
AzureAsn                       : 12076
CustomerAutonomousSystemNumber :
PeerAsn                        : 1234
PrimaryAzurePort               :
PrimaryPeerSubnet              : 10.0.0.0/30
RoutingRegistryName            :
SecondaryAzurePort             :
SecondaryPeerSubnet            : 10.0.0.4/30
State                          : Enabled
VlanId                         : 100
```

To update Azure private peering configuration

You can update any part of the configuration using the following cmdlet. In the example below, the VLAN ID of the circuit is being updated from 100 to 500.

```
Set-AzureBGPPeering -AccessType Private -ServiceKey "*********************************" -PrimaryPeerSubnet
"10.0.0.0/30" -SecondaryPeerSubnet "10.0.0.4/30" -PeerAsn 1234 -VlanId 500 -SharedKey "A1B2C3D4"
```

To delete Azure private peering

You can remove your peering configuration by running the following cmdlet.

> **WARNING**
> You must ensure that all virtual networks are unlinked from the ExpressRoute circuit before running this cmdlet.

```
Remove-AzureBGPPeering -AccessType Private -ServiceKey "*********************************"
```

Azure public peering

This section provides instructions on how to create, get, update and delete the Azure public peering configuration for an ExpressRoute circuit.

To create Azure public peering

1. **Import the PowerShell module for ExpressRoute.**

   You must import the Azure and ExpressRoute modules into the PowerShell session in order to start using the ExpressRoute cmdlets. Run the following commands to import the Azure and ExpressRoute modules into the PowerShell session.

```
 Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1'
 Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\ExpressRoute\ExpressRoute.psd1'
```

2. **Create an ExpressRoute circuit**

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider. If your connectivity provider offers managed Layer 3 services, you can request your connectivity provider to enable Azure public peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. **Check ExpressRoute circuit to ensure it is provisioned**

   You must first check to see if the ExpressRoute circuit is Provisioned and also Enabled. See the example below.

```
PS C:\> Get-AzureDedicatedCircuit -ServiceKey "**********************************"

Bandwidth                       : 200
CircuitName                     : MyTestCircuit
Location                        : Silicon Valley
ServiceKey                      : ********************************
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Sku                             : Standard
Status                          : Enabled
```

   Make sure that the circuit shows as Provisioned and Enabled. If it doesn't, work with your connectivity provider to get your circuit to the required state and status.

```
ServiceProviderProvisioningState : Provisioned
Status                          : Enabled
```

4. **Configure Azure public peering for the circuit**

   Ensure that you have the following information before you proceed further.

   - A /30 subnet for the primary link. This must be a valid public IPv4 prefix.
   - A /30 subnet for the secondary link. This must be a valid public IPv4 prefix.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.
   - An MD5 hash if you choose to use one. **This is optional**.

     You can run the following cmdlet to configure Azure public peering for your circuit

     New-AzureBGPPeering -AccessType Public -ServiceKey "**********************************" -PrimaryPeerSubnet "131.107.0.0/30" -SecondaryPeerSubnet "131.107.0.4/30" -PeerAsn 1234 -VlanId 200

     You can use the cmdlet below if you choose to use an MD5 hash

     New-AzureBGPPeering -AccessType Public -ServiceKey "**********************************" -PrimaryPeerSubnet "131.107.0.0/30" -SecondaryPeerSubnet "131.107.0.4/30" -PeerAsn 1234 -VlanId 200 -SharedKey "A1B2C3D4"

> **IMPORTANT**
> Ensure that you specify your AS number as peering ASN and not customer ASN.

To view Azure public peering details

You can get configuration details using the following cmdlet

```
Get-AzureBGPPeering -AccessType Public -ServiceKey "*********************************"

AdvertisedPublicPrefixes       :
AdvertisedPublicPrefixesState  : Configured
AzureAsn                       : 12076
CustomerAutonomousSystemNumber :
PeerAsn                        : 1234
PrimaryAzurePort               :
PrimaryPeerSubnet              : 131.107.0.0/30
RoutingRegistryName            :
SecondaryAzurePort             :
SecondaryPeerSubnet            : 131.107.0.4/30
State                          : Enabled
VlanId                         : 200
```

To update Azure public peering configuration

You can update any part of the configuration using the following cmdlet

```
Set-AzureBGPPeering -AccessType Public -ServiceKey "*********************************" -PrimaryPeerSubnet
"131.107.0.0/30" -SecondaryPeerSubnet "131.107.0.4/30" -PeerAsn 1234 -VlanId 600 -SharedKey "A1B2C3D4"
```

The VLAN ID of the circuit is being updated from 200 to 600 in the above example.

To delete Azure public peering

You can remove your peering configuration by running the following cmdlet

```
Remove-AzureBGPPeering -AccessType Public -ServiceKey "*********************************"
```

Microsoft peering

This section provides instructions on how to create, get, update and delete the Microsoft peering configuration for an ExpressRoute circuit.

To create Microsoft peering

1. **Import the PowerShell module for ExpressRoute.**

   You must import the Azure and ExpressRoute modules into the PowerShell session in order to start using the ExpressRoute cmdlets. Run the following commands to import the Azure and ExpressRoute modules into the PowerShell session.

   ```
    Import-Module 'C:\Program Files (x86)\Microsoft
   SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1'
    Import-Module 'C:\Program Files (x86)\Microsoft
   SDKs\Azure\PowerShell\ServiceManagement\Azure\ExpressRoute\ExpressRoute.psd1'
   ```

2. **Create an ExpressRoute circuit**

   Follow the instructions to create an ExpressRoute circuit and have it provisioned by the connectivity provider. If your connectivity provider offers managed Layer 3 services, you can request your

connectivity provider to enable Azure private peering for you. In that case, you won't need to follow instructions listed in the next sections. However, if your connectivity provider does not manage routing for you, after creating your circuit, follow the instructions below.

3. **Check ExpressRoute circuit to ensure it is provisioned**

   You must first check to see if the ExpressRoute circuit is in Provisioned and Enabled state.

   ```
   PS C:\> Get-AzureDedicatedCircuit -ServiceKey "**********************************"

   Bandwidth                       : 200
   CircuitName                     : MyTestCircuit
   Location                        : Silicon Valley
   ServiceKey                      : ********************************
   ServiceProviderName             : equinix
   ServiceProviderProvisioningState : Provisioned
   Sku                             : Standard
   Status                          : Enabled
   ```

   Make sure that the circuit shows as Provisioned and Enabled. If it doesn't, work with your connectivity provider to get your circuit to the required state and status.

   ```
   ServiceProviderProvisioningState : Provisioned
   Status                          : Enabled
   ```

4. **Configure Microsoft peering for the circuit**

   Make sure that you have the following information before you proceed.

   - A /30 subnet for the primary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR.
   - A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID.
   - AS number for peering. You can use both 2-byte and 4-byte AS numbers.
   - Advertised prefixes: You must provide a list of all prefixes you plan to advertise over the BGP session. Only public IP address prefixes are accepted. You can send a comma separated list if you plan to send a set of prefixes. These prefixes must be registered to you in an RIR / IRR.
   - Customer ASN: If you are advertising prefixes that are not registered to the peering AS number, you can specify the AS number to which they are registered. **This is optional**.
   - Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
   - An MD5 hash, if you choose to use one. **This is optional.**

     You can run the following cmdlet to configure Microsoft pering for your circuit

     New-AzureBGPPeering -AccessType Microsoft -ServiceKey "**********************************" -PrimaryPeerSubnet "131.107.0.0/30" -SecondaryPeerSubnet "131.107.0.4/30" -VlanId 300 -PeerAsn 1234 -CustomerAsn 2245 -AdvertisedPublicPrefixes "123.0.0.0/30" -RoutingRegistryName "ARIN" -SharedKey "A1B2C3D4"

To view Microsoft peering details

You can get configuration details using the following cmdlet.

```
Get-AzureBGPPeering -AccessType Microsoft -ServiceKey "********************************"

AdvertisedPublicPrefixes       : 123.0.0.0/30
AdvertisedPublicPrefixesState  : Configured
AzureAsn                       : 12076
CustomerAutonomousSystemNumber : 2245
PeerAsn                        : 1234
PrimaryAzurePort               :
PrimaryPeerSubnet              : 10.0.0.0/30
RoutingRegistryName            : ARIN
SecondaryAzurePort             :
SecondaryPeerSubnet            : 10.0.0.4/30
State                          : Enabled
VlanId                         : 300
```

To update Microsoft peering configuration

You can update any part of the configuration using the following cmdlet.

```
    Set-AzureBGPPeering -AccessType Microsoft -ServiceKey "********************************" -
PrimaryPeerSubnet "131.107.0.0/30" -SecondaryPeerSubnet "131.107.0.4/30" -VlanId 300 -PeerAsn 1234 -
CustomerAsn 2245 -AdvertisedPublicPrefixes "123.0.0.0/30" -RoutingRegistryName "ARIN" -SharedKey "A1B2C3D4"
```

To delete Microsoft peering

You can remove your peering configuration by running the following cmdlet.

```
Remove-AzureBGPPeering -AccessType Microsoft -ServiceKey "********************************"
```

Next steps

Next, Link a VNet to an ExpressRoute circuit.

- For more information about workflows, see ExpressRoute workflows.
- For more information about circuit peering, see ExpressRoute circuits and routing domains.

Connect a virtual network to an ExpressRoute circuit

1/17/2017 • 2 min to read • Edit on GitHub

This article will help you link virtual networks (VNets) to Azure ExpressRoute circuits by using the Resource Manager deployment model and the Azure portal. Virtual networks can either be in the same subscription, or they can be part of another subscription.

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 prerequisites

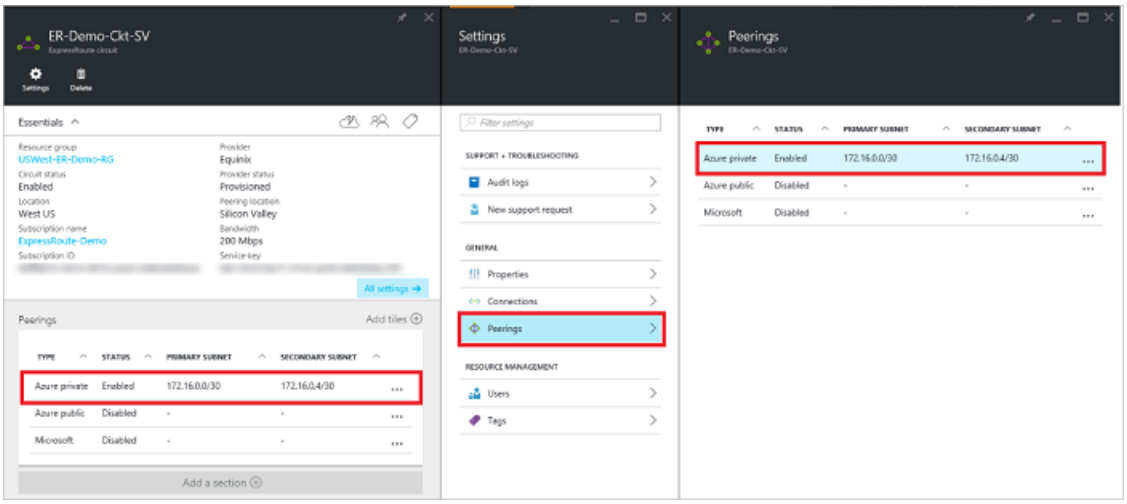
- Make sure that you have reviewed the prerequisites, routing requirements, and workflows before you begin configuration.
- You must have an active ExpressRoute circuit.
    - Follow the instructions to create an ExpressRoute circuit and have the circuit enabled by your connectivity provider.
    - Ensure that you have Azure private peering configured for your circuit. See the Configure routing article for routing instructions.
    - Ensure that Azure private peering is configured and the BGP peering between your network and Microsoft is up so that you can enable end-to-end connectivity.
    - Ensure that you have a virtual network and a virtual network gateway created and fully provisioned. Follow the instructions to create a VPN gateway (follow only steps 1-5).

You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit. You can link a virtual networks outside of the geopolitical region of the ExpressRoute circuit, or connect a larger number of virtual networks to your ExpressRoute circuit if you enabled the ExpressRoute premium add-on. Check the FAQ for more details on the premium add-on.

Connect a virtual network in the same subscription to a circuit

To create a connection

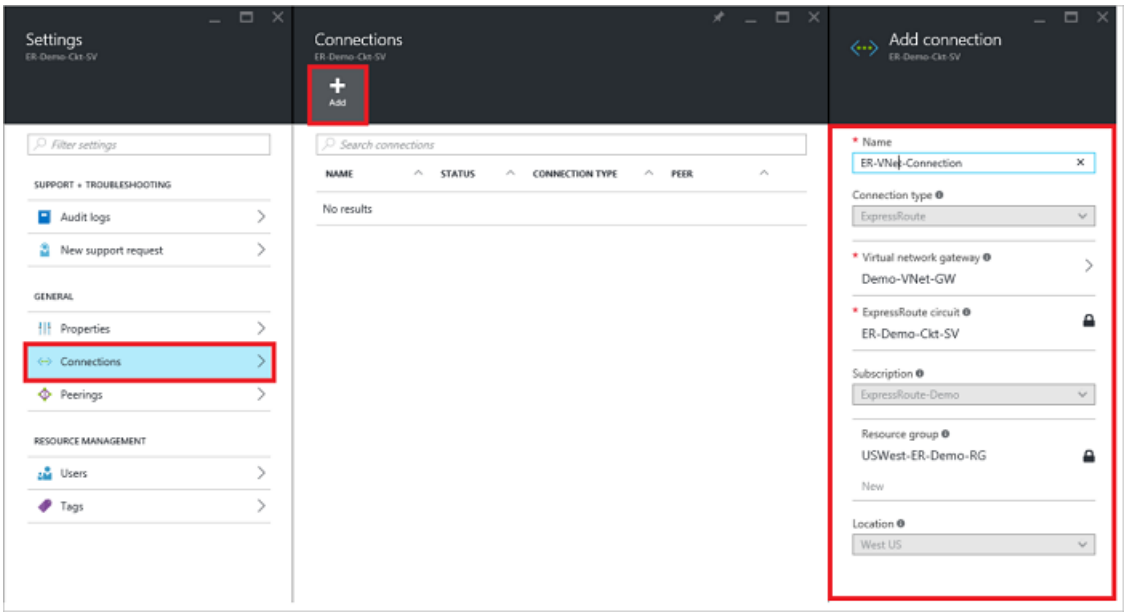
1. Ensure that your ExpressRoute circuit and Azure private peering have been configured successfully. Follow the instructions in Create an ExpressRoute circuit and Configure routing. Your ExpressRoute circuit should look like the following image.



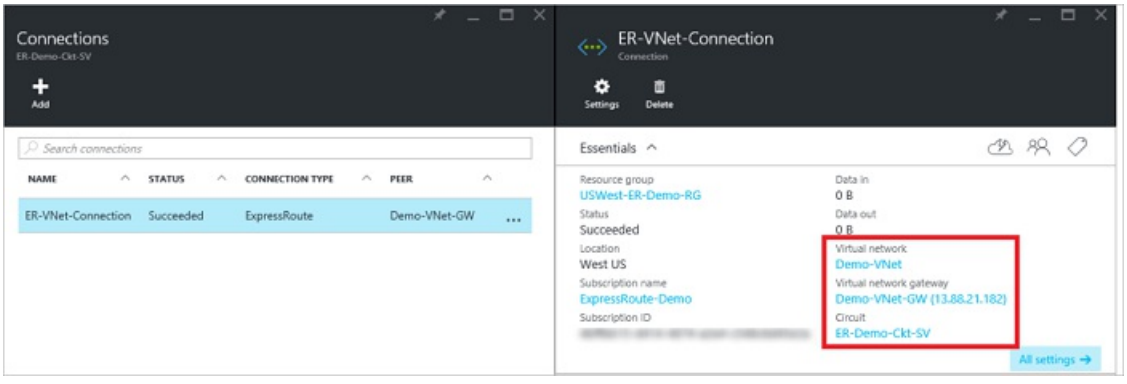
NOTE

BGP configuration information will not show up if the layer 3 provider configured your peerings. If your circuit is in a provisioned state, you should be able to create connections.

2. You can now start provisioning a connection to link your virtual network gateway to your ExpressRoute circuit. Click **Connection** > **Add** to open the **Add connection** blade, and then configure the values. See the following reference example.



3. After your connection has been successfully configured, your connection object will show the information for the connection.



To delete a connection

You can delete a connection by selecting the **Delete** icon on the blade for your connection.

Connect a virtual network in a different subscription to a circuit

At this time, you cannot connect virtual networks across subscriptions by using the Azure portal. However, you can use PowerShell to do this. See the PowerShell article for more information.

Next steps

For more information about ExpressRoute, see the ExpressRoute FAQ.

Connect a virtual network to an ExpressRoute circuit

1/17/2017 • 4 min to read • Edit on GitHub

This article will help you link virtual networks (VNets) to Azure ExpressRoute circuits by using the Resource Manager deployment model and PowerShell. Virtual networks can either be in the same subscription or part of another subscription.

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 prerequisites

- You need the latest version of the Azure PowerShell modules (at least version 1.0). See How to install and configure Azure PowerShell for more information about installing the PowerShell cmdlets.
- You need to review the prerequisites, routing requirements, and workflows before you begin configuration.
- You must have an active ExpressRoute circuit.
  - Follow the instructions to create an ExpressRoute circuit and have the circuit enabled by your connectivity provider.
  - Ensure that you have Azure private peering configured for your circuit. See the configure routing article for routing instructions.
  - Ensure that Azure private peering is configured and the BGP peering between your network and Microsoft is up so that you can enable end-to-end connectivity.
  - Ensure that you have a virtual network and a virtual network gateway created and fully provisioned. Follow the instructions to create a VPN gateway, but be sure to use `-GatewayType ExpressRoute`.

You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit.

You can link a virtual networks outside of the geopolitical region of the ExpressRoute circuit, or connect a larger number of virtual networks to your ExpressRoute circuit if you enabled the ExpressRoute premium add-on. Check the FAQ for more details on the premium add-on.

Connect a virtual network in the same subscription to a circuit

You can connect a virtual network gateway to an ExpressRoute circuit by using the following cmdlet. Make sure that the virtual network gateway is created and is ready for linking before you run the cmdlet:

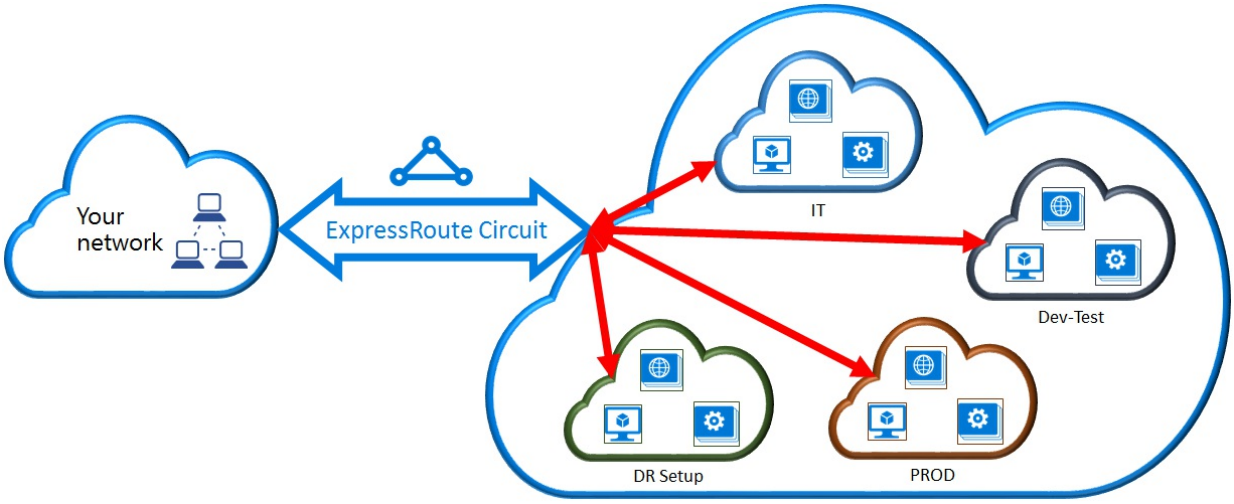
```
$circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$gw = Get-AzureRmVirtualNetworkGateway -Name "ExpressRouteGw" -ResourceGroupName "MyRG"
$connection = New-AzureRmVirtualNetworkGatewayConnection -Name "ERConnection" -ResourceGroupName "MyRG" -
Location "East US" -VirtualNetworkGateway1 $gw -PeerId $circuit.Id -ConnectionType ExpressRoute
```

Connect a virtual network in a different subscription to a circuit

You can share an ExpressRoute circuit across multiple subscriptions. The following figure shows a simple schematic of how sharing works for ExpressRoute circuits across multiple subscriptions.

Each of the smaller clouds within the large cloud is used to represent subscriptions that belong to different departments within an organization. Each of the departments within the organization can use their own subscription for deploying their services--but they can share a single ExpressRoute circuit to connect back to your on-premises network. A single department (in this example: IT) can own the ExpressRoute circuit. Other subscriptions within the organization can use the ExpressRoute circuit.

> **NOTE**
> Connectivity and bandwidth charges for the dedicated circuit will be applied to the ExpressRoute circuit owner. All virtual networks share the same bandwidth.



Administration

The *circuit owner* is an authorized power user of the ExpressRoute circuit resource. The circuit owner can create authorizations that can be redeemed by *circuit users*. *Circuit users* are owners of virtual network gateways (that are not within the same subscription as the ExpressRoute circuit). *Circuit users* can redeem authorizations (one authorization per virtual network).

The *circuit owner* has the power to modify and revoke authorizations at any time. Revoking an authorization results in all link connections being deleted from the subscription whose access was revoked.

Circuit owner operations

Creating an authorization

The circuit owner creates an authorization. This results in the creation of an authorization key that can be used by a circuit user to connect their virtual network gateways to the ExpressRoute circuit. An authorization is valid for only one connection.

The following cmdlet snippet shows how to create an authorization:

```
$circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
Add-AzureRmExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization1"
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $circuit

    $circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$auth1 = Get-AzureRmExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization1"
```

The response to this will contain the authorization key and status:

```
Name                  : MyAuthorization1
Id                    :
/subscriptions/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&/resourceGroups/ERCrossSubTestRG/providers/Microsoft.Netwo
rk/expressRouteCircuits/CrossSubTest/authorizations/MyAuthorization1
Etag                  : &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
AuthorizationKey      : ####################################
AuthorizationUseStatus : Available
ProvisioningState     : Succeeded
```

Reviewing authorizations

The circuit owner can review all authorizations that are issued on a particular circuit by running the following cmdlet:

```
$circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$authorizations = Get-AzureRmExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit
```

Adding authorizations

The circuit owner can add authorizations by using the following cmdlet:

```
$circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
Add-AzureRmExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization2"
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $circuit

$circuit = Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$authorizations = Get-AzureRmExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit
```

Deleting authorizations

The circuit owner can revoke/delete authorizations to the user by running the following cmdlet:

```
Remove-AzureRmExpressRouteCircuitAuthorization -Name "MyAuthorization2" -ExpressRouteCircuit $circuit
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $circuit
```

Circuit user operations

The circuit user needs the peer ID and an authorization key from the circuit owner. The authorization key is a GUID.

Peer ID is, can be checked from the following command.

```
Get-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
```

Redeeming connection authorizations

The circuit user can run the following cmdlet to redeem a link authorization:

```
$id =
"/subscriptions/******************************/resourceGroups/ERCrossSubTestRG/providers/Microsoft.Network/
expressRouteCircuits/MyCircuit"
$gw = Get-AzureRmVirtualNetworkGateway -Name "ExpressRouteGw" -ResourceGroupName "MyRG"
$connection = New-AzureRmVirtualNetworkGatewayConnection -Name "ERConnection" -ResourceGroupName
"RemoteResourceGroup" -Location "East US" -VirtualNetworkGateway1 $gw -PeerId $id -ConnectionType
ExpressRoute -AuthorizationKey "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"
```

Releasing connection authorizations

You can release an authorization by deleting the connection that links the ExpressRoute circuit to the virtual

network.

Next steps

For more information about ExpressRoute, see the ExpressRoute FAQ.

Connect a virtual network to an ExpressRoute circuit

1/17/2017 • 4 min to read • Edit on GitHub

This article will help you link virtual networks (VNets) to Azure ExpressRoute circuits by using the classic deployment model and PowerShell. Virtual networks can either be in the same subscription or can be part of another subscription.

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 prerequisites

1. You need the latest version of the Azure PowerShell modules. You can download the latest PowerShell modules from the PowerShell section of the Azure Downloads page. Follow the instructions in How to install and configure Azure PowerShell for step-by-step guidance on how to configure your computer to use the Azure PowerShell modules.
2. You need to review the prerequisites, routing requirements, and workflows before you begin configuration.
3. You must have an active ExpressRoute circuit.
   - Follow the instructions to create an ExpressRoute circuit and have your connectivity provider enable the circuit.
   - Ensure that you have Azure private peering configured for your circuit. See the Configure routing article for routing instructions.
   - Ensure that Azure private peering is configured and the BGP peering between your network and Microsoft is up so that you can enable end-to-end connectivity.
   - You must have a virtual network and a virtual network gateway created and fully provisioned. Follow the instructions to configure a virtual network for ExpressRoute.

You can link up to 10 virtual networks to an ExpressRoute circuit. All virtual networks must be in the same geopolitical region. You can link a larger number of virtual networks to your ExpressRoute circuit, or link virtual networks that are in other geopolitical regions if you enabled the ExpressRoute premium add-on. Check the FAQ for more details on the premium add-on.

Connect a virtual network in the same subscription to a circuit

You can link a virtual network to an ExpressRoute circuit by using the following cmdlet. Make sure that the virtual network gateway is created and is ready for linking before you run the cmdlet.

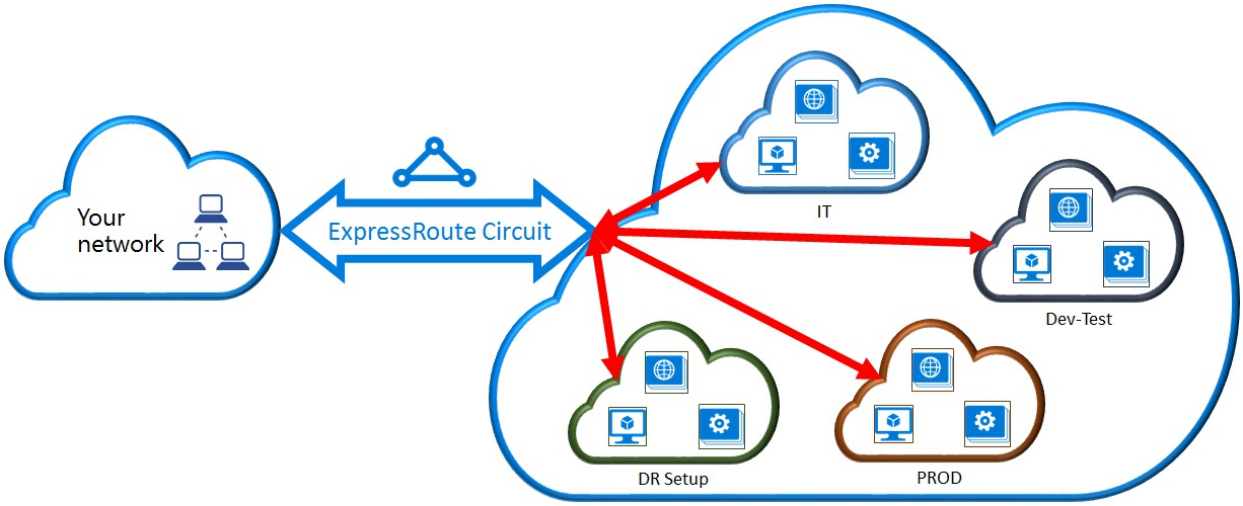
```
New-AzureDedicatedCircuitLink -ServiceKey "**************************" -VNetName "MyVNet"
Provisioned
```

Connect a virtual network in a different subscription to a circuit

You can share an ExpressRoute circuit across multiple subscriptions. The following figure shows a simple schematic of how sharing works for ExpressRoute circuits across multiple subscriptions.

Each of the smaller clouds within the large cloud is used to represent subscriptions that belong to different departments within an organization. Each of the departments within the organization can use their own subscription for deploying their services--but the departments can share a single ExpressRoute circuit to connect back to your on-premises network. A single department (in this example: IT) can own the ExpressRoute circuit. Other subscriptions within the organization can use the ExpressRoute circuit.

> **NOTE**
> Connectivity and bandwidth charges for the dedicated circuit will be applied to the ExpressRoute circuit owner. All virtual networks share the same bandwidth.



Administration

The *circuit owner* is the administrator/coadministrator of the subscription in which the ExpressRoute circuit is created. The circuit owner can authorize administrators/coadministrators of other subscriptions, referred to as *circuit users*, to use the dedicated circuit that they own. Circuit users who are authorized to use the organization's ExpressRoute circuit can link the virtual network in their subscription to the ExpressRoute circuit after they are authorized.

The circuit owner has the power to modify and revoke authorizations at any time. Revoking an authorization will result in all links being deleted from the subscription whose access was revoked.

Circuit owner operations

Creating an authorization

The circuit owner authorizes the administrators of other subscriptions to use the specified circuit. In the following example, the administrator of the circuit (Contoso IT) enables the administrator of another subscription (Dev-Test) to link up to two virtual networks to the circuit. The Contoso IT administrator enables this by specifying the Dev-Test Microsoft ID. The cmdlet doesn't send email to the specified Microsoft ID. The circuit owner needs to explicitly notify the other subscription owner that the authorization is complete.

```
New-AzureDedicatedCircuitLinkAuthorization -ServiceKey "**************************" -Description "Dev-Test
Links" -Limit 2 -MicrosoftIds 'devtest@contoso.com'

Description         : Dev-Test Links
Limit               : 2
LinkAuthorizationId : **********************************
MicrosoftIds        : devtest@contoso.com
Used                : 0
```

Reviewing authorizations

The circuit owner can review all authorizations that are issued on a particular circuit by running the following

cmdlet:

```
Get-AzureDedicatedCircuitLinkAuthorization -ServiceKey: "************************"

Description         : EngineeringTeam
Limit               : 3
LinkAuthorizationId : ###################################
MicrosoftIds        : engadmin@contoso.com
Used                : 1

Description         : MarketingTeam
Limit               : 1
LinkAuthorizationId : @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
MicrosoftIds        : marketingadmin@contoso.com
Used                : 0

Description         : Dev-Test Links
Limit               : 2
LinkAuthorizationId : &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
MicrosoftIds        : salesadmin@contoso.com
Used                : 2
```

Updating authorizations

The circuit owner can modify authorizations by using the following cmdlet:

```
Set-AzureDedicatedCircuitLinkAuthorization -ServiceKey "************************" -AuthorizationId
"&&&&&&&&&&&&&&&&&&&&&&&&&&&"-Limit 5

Description         : Dev-Test Links
Limit               : 5
LinkAuthorizationId : &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
MicrosoftIds        : devtest@contoso.com
Used                : 0
```

Deleting authorizations

The circuit owner can revoke/delete authorizations to the user by running the following cmdlet:

```
Remove-AzureDedicatedCircuitLinkAuthorization -ServiceKey "****************************" -AuthorizationId
"###############################"
```

Circuit user operations

Reviewing authorizations

The circuit user can review authorizations by using the following cmdlet:

```
Get-AzureAuthorizedDedicatedCircuit

Bandwidth                       : 200
CircuitName                     : ContosoIT
Location                        : Washington DC
MaximumAllowedLinks             : 2
ServiceKey                      : &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
ServiceProviderName             : equinix
ServiceProviderProvisioningState : Provisioned
Status                          : Enabled
UsedLinks                       : 0
```

Redeeming link authorizations

The circuit user can run the following cmdlet to redeem a link authorization:

```
New-AzureDedicatedCircuitLink –servicekey "&&&&&&&&&&&&&&&&&&&&&&&&&&&&" –VnetName 'SalesVNET1'

State VnetName
----- --------
Provisioned SalesVNET1
```

Next steps

For more information about ExpressRoute, see the ExpressRoute FAQ.

Configure a virtual network gateway for ExpressRoute using Resource Manager and PowerShell

1/17/2017 • 3 min to read • Edit on GitHub

This article will walk you through the steps to add, resize, and remove a virtual network (VNet) gateway for a pre-existing VNet. The steps for this configuration are specifically for VNets that were created using the **Resource Manager deployment model** and that will be be used in an ExpressRoute configuration.

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.

Before beginning

Verify that you have installed the Azure PowerShell cmdlets needed for this configuration (1.0.2 or later). If you haven't installed the cmdlets, you'll need to do so before beginning the configuration steps. For more information about installing Azure PowerShell, see How to install and configure Azure PowerShell.

The steps for this task use a VNet based on the values below. Additional settings and names are also outlined in this list. We don't use this list directly in any of the steps, although we do add variables based on the values in this list. You can copy the list to use as a reference, replacing the values with your own.

Configuration reference list:

- Virtual Network Name = "TestVNet"
- Virtual Network address space = 192.168.0.0/16
- Resource Group = "TestRG"
- Subnet1 Name = "FrontEnd"
- Subnet1 address space = "192.168.0.0/16"
- Gateway Subnet name: "GatewaySubnet" You must always name a gateway subnet *GatewaySubnet*.
- Gateway Subnet address space = "192.168.200.0/26"
- Region = "East US"
- Gateway Name = "GW"
- Gateway IP Name = "GWIP"
- Gateway IP configuration Name = "gwipconf"
- Type = "ExpressRoute" This type is required for an ExpressRoute configuration.
- Gateway Public IP Name = "gwpip"

Add a gateway

1. Connect to your Azure Subscription.

```
Login-AzureRmAccount
Get-AzureRmSubscription
Select-AzureRmSubscription -SubscriptionName "Name of subscription"
```

2. Declare your variables for this exercise. This example will use the use the variables in the sample below. Be sure to edit this to reflect the settings that you want to use.

```
$RG = "TestRG"
$Location = "East US"
$GWName = "GW"
$GWIPName = "GWIP"
$GWIPconfName = "gwipconf"
$VNetName = "TestVNet"
```

3. Store the virtual network object as a variable.

```
$vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG
```

4. Add a gateway subnet to your Virtual Network. The gateway subnet must be named "GatewaySubnet". You'll want to create a gateway that is /27 or larger (/26, /25, etc.).

```
Add-AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet -AddressPrefix
192.168.200.0/26
```

5. Set the configuration.

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

6. Store the gateway subnet as a variable.

```
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -VirtualNetwork $vnet
```

7. Request a public IP address. The IP address is requested before creating the gateway. You cannot specify the IP address that you want to use; it's dynamically allocated. You'll use this IP address in the next configuration section. The AllocationMethod must be Dynamic.

```
$pip = New-AzureRmPublicIpAddress -Name $GWIPName  -ResourceGroupName $RG -Location $Location -
AllocationMethod Dynamic
```

8. Create the configuration for your gateway. The gateway configuration defines the subnet and the public IP address to use. In this step, you are specifying the configuration that will be used when you create the gateway. This step does not actually create the gateway object. Use the sample below to create your gateway configuration.

```
$ipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress
$pip
```

9. Create the gateway. In this step, the **-GatewayType** is especially important. You must use the value **ExpressRoute**. Note that after running these cmdlets, the gateway can take 20 minutes or more to create.

```
New-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG -Location $Location -
IpConfigurations $ipconf -GatewayType Expressroute -GatewaySku Standard
```

Verify the gateway was created

Use the command below to verify that the gateway has been created.

```
Get-AzureRmVirtualNetworkGateway -ResourceGroupName $RG
```

Resize a gateway

There are a number of Gateway SKUs. You can use the following command to change the Gateway SKU at any time.

> **IMPORTANT**
> This command doesn't work for UltraPerformance gateway. To change your gateway to an UltraPerformance gateway, first remove the existing ExpressRoute gateway, and then create a new UltraPerformance gateway. To downgrade your gateway from an UltraPerformance gateway, first remove the UltraPerformance gateway, and then create a new gateway.

```
$gw = Get-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
Resize-AzureRmVirtualNetworkGateway -VirtualNetworkGateway $gw -GatewaySku HighPerformance
```

Remove a gateway

Use the command below to remove a gateway

```
Remove-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG
```

Next steps

After you have created the VNet gateway, you can link your VNet to an ExpressRoute circuit. See Link a Virtual Network to an ExpressRoute circuit.

Configure a virtual network gateway for ExpressRoute using the classic deployment model and PowerShell

1/17/2017 • 2 min to read • Edit on GitHub

This article will walk you through the steps to add, resize, and remove a virtual network (VNet) gateway for a pre-existing VNet. The steps for this configuration are specifically for VNets that were created using the **classic deployment model** and that will be be used in an ExpressRoute configuration.

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.

Before beginning

Verify that you have installed the Azure PowerShell cmdlets needed for this configuration (1.0.2 or later). If you haven't installed the cmdlets, you'll need to do so before beginning the configuration steps. For more information about installing Azure PowerShell, see How to install and configure Azure PowerShell.

You must create a VNet and a gateway subnet first, before working on the following tasks. See the article Configure a Virtual Network using the classic portal for more information.

Add a gateway

Use the command below to create a gateway. Be sure to substitute any values for your own.

```
New-AzureVirtualNetworkGateway -VNetName "MyAzureVNET" -GatewayName "ERGateway" -GatewayType Dedicated -
GatewaySKU  Standard
```

Verify the gateway was created

Use the command below to verify that the gateway has been created. This command also retrieves the gateway ID, which you need for other operations.

```
Get-AzureVirtualNetworkGateway
```

Resize a gateway

There are a number of Gateway SKUs. You can use the following command to change the Gateway SKU at any time.

IMPORTANT

This command doesn't work for UltraPerformance gateway. To change your gateway to an UltraPerformance gateway, first remove the existing ExpressRoute gateway, and then create a new UltraPerformance gateway. To downgrade your gateway from an UltraPerformance gateway, first remove the UltraPerformance gateway, and then create a new gateway.

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

Remove a gateway

Use the command below to remove a gateway

```
Remove-AzureVirtualNetworkGateway -GatewayId <Gateway ID>
```

Next steps

After you have created the VNet gateway, you can link your VNet to an ExpressRoute circuit. See Link a Virtual Network to an ExpressRoute circuit.

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 ExressRoute, 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.

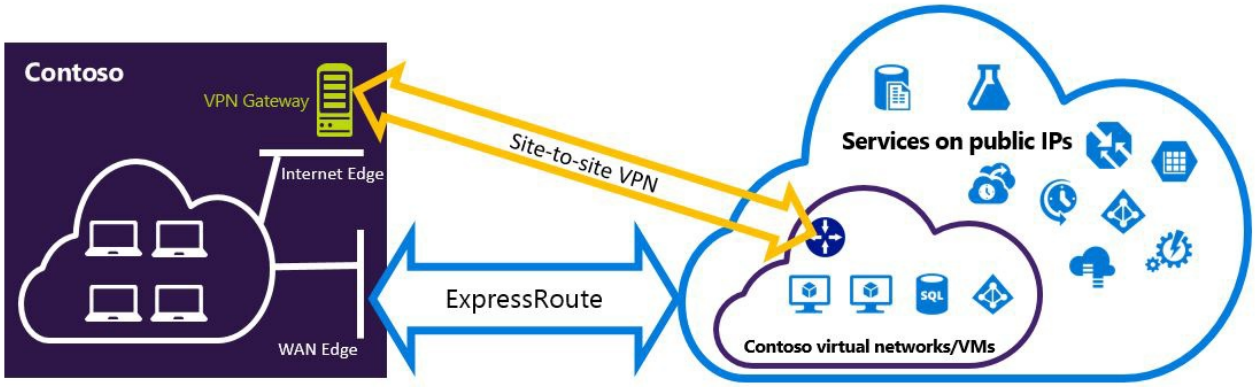
Configuration designs

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.

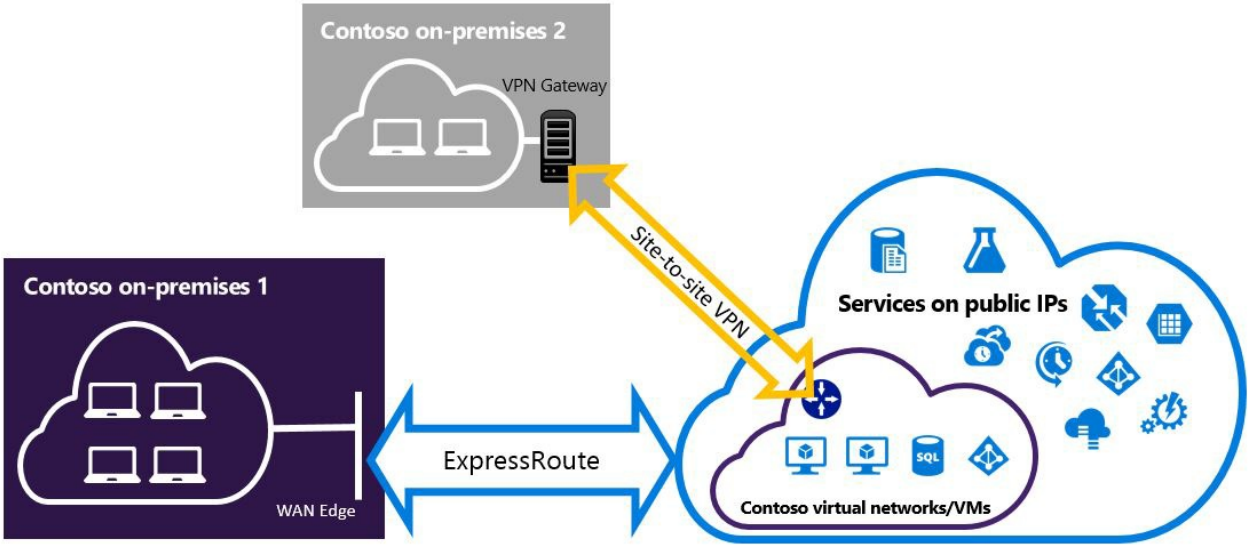
NOTE

While ExpressRoute circuit is prefered over Site-to-Site VPN when both routes are the same, Azure will use the longuest 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 a 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 coexsiting 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
```

8. Configure your local VPN device to connect to the new Azure VPN gateway. 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.

```
 $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 coexsiting 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 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.

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

3. Delete Gateway Subnet.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Remove-AzureRmVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet
```

4. Add a Gateway Subnet that is /27 or larger.

> **NOTE**
> If you don't have enough IP addresses left in your virtual network to increase the gateway subnet size, you need to add more IP address space.

```
$vnet = Get-AzureRmVirtualNetwork -Name <yourvnetname> -ResourceGroupName <yourresourcegroup>
Add-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet -AddressPrefix
"10.200.255.0/24"
```

Save the VNet configuration.

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

5. At this point, you'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 ExressRoute, 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.

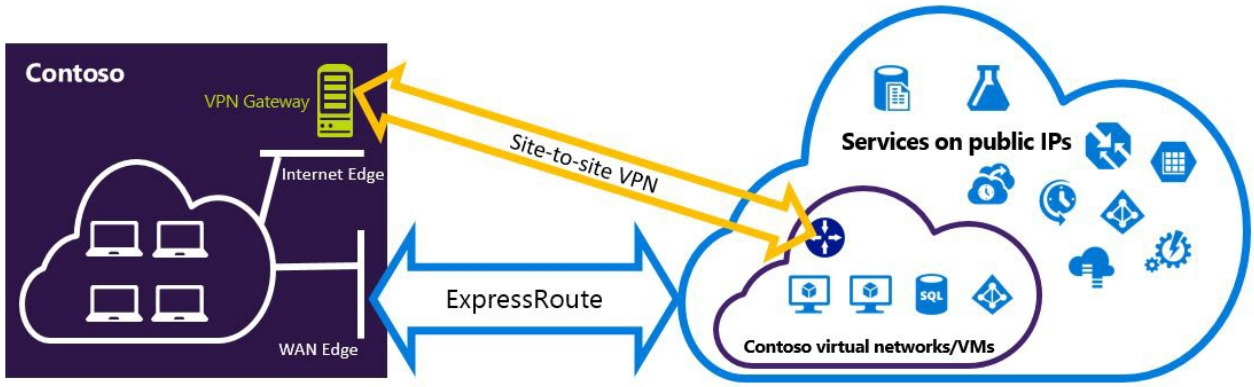
Limits and limitations

- **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

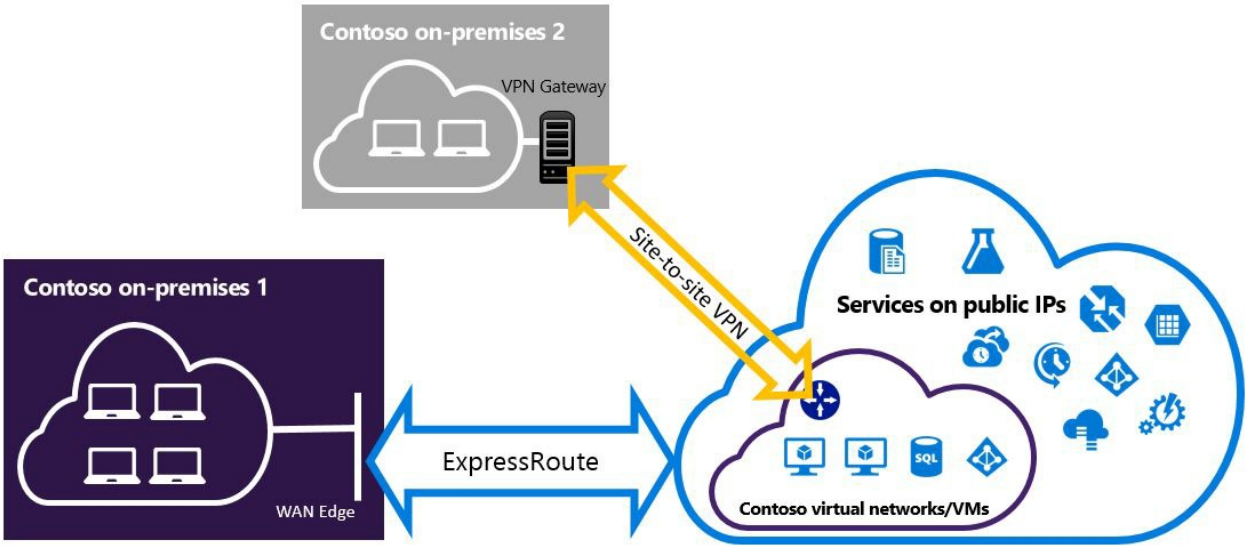
Configure a Site-to-Site VPN as a failover path for ExpressRoute

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 a 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 coexsiting 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="MyAzureVNET" Location="Central US">
     <AddressSpace>
       <AddressPrefix>10.17.159.192/26</AddressPrefix>
     </AddressSpace>
     <Subnets>
       <Subnet name="Subnet-1">
         <AddressPrefix>10.17.159.192/27</AddressPrefix>
       </Subnet>
       <Subnet name="GatewaySubnet">
         <AddressPrefix>10.17.159.224/27</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
```

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.

```
New-AzureDedicatedCircuitLink -ServiceKey <service-key> -VNetName MyAzureVNET
```

6. 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.x.y
DefaultSite          :
GatewaySKU           : HighPerformance
Location             :
VnetId               : 979aabcf-e47f-4136-ab9b-b4780c1e1bd5
SubnetId             :
EnableBgp            : False
OperationDescription : Get-AzureVirtualNetworkGateway
OperationId          : 42773656-85e1-a6b6-8705-35473f1e6f6a
OperationStatus      : Succeeded
```

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.

> **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.x.y
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 coexsiting 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

Move ExpressRoute circuits from the classic to the Resource Manager deployment model

1/17/2017 • 3 min to read • Edit on GitHub

Configuration prerequisites

- You need the latest version of the Azure PowerShell modules (at least version 1.0).
- Make sure that you have reviewed the prerequisites, routing requirements, and workflows before you begin configuration.
- Before preceding further, review information that is provided under Moving an ExpressRoute circuit from classic to Resource Manager. Ensure that you have fully understood the limits and limitations of what's possible.
- If you want to move an Azure ExpressRoute circuit from the classic deployment model to the Azure Resource Manager deployment model, you must have the circuit fully configured and operational in the classic deployment model.
- Ensure that you have a resource group that was created in the Resource Manager deployment model.

Move the ExpressRoute circuit to the Resource Manager deployment model

You must move an ExpressRoute circuit to the Resource Manager deployment model so that you can use it across both the classic and the Resource Manager deployment models. You can do this by running the following PowerShell commands.

Step 1: Gather circuit details from the classic deployment model

You need to gather information about your ExpressRoute circuit first.

Sign in to the Azure classic environment, and gather the service key. You can use the following PowerShell snippet to gather the information:

```
# Sign in to your Azure account
Add-AzureAccount

# Select the appropriate Azure subscription
Select-AzureSubscription "<Enter Subscription Name here>"

# Import the PowerShell modules for Azure and ExpressRoute
Import-Module 'C:\Program Files (x86)\Microsoft SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1'
Import-Module 'C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\ExpressRoute\ExpressRoute.psd1'

# Get the service keys of all your ExpressRoute circuits
Get-AzureDedicatedCircuit
```

Copy the **service key** of the circuit that you want to move over to the Resource Manager deployment model.

Step 2: Sign in to the Resource Manager environment, and create a new resource group

You can create a new resource group by using the following snippet:

```
# Sign in to your Azure Resource Manager environment
Login-AzureRmAccount

# Select the appropriate Azure subscription
Get-AzureRmSubscription -SubscriptionName "<Enter Subscription Name here>" | Select-AzureRmSubscription

#Create a new resource group if you don't already have one
New-AzureRmResourceGroup -Name "DemoRG" -Location "West US"
```

You can also use an existing resource group if you already have one.

Step 3: Move the ExpressRoute circuit to the Resource Manager deployment model

You are now ready to move over your ExpressRoute circuit from the classic to the Resource Manager deployment model. Review the information provided under Moving an ExpressRoute circuit from the classic to the Resource Manager deployment model before proceeding further.

You can do this by running the following snippet:

```
Move-AzureRmExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "DemoRG" -Location "West US" -ServiceKey "
<Service-key>"
```

> **NOTE**
> After the move has finished, the new name that is listed in the previous cmdlet will be used to address the resource. The circuit will essentially be renamed.

Enable an ExpressRoute circuit for both deployment models

You must move your ExpressRoute circuit to the Resource Manager deployment model before controlling access to the deployment model.

Run the following cmdlet to enable access to both deployment models:

```
# Get details of the ExpressRoute circuit
$ckt = Get-AzureRmExpressRouteCircuit -Name "DemoCkt" -ResourceGroupName "DemoRG"

#Set "Allow Classic Operations" to TRUE
$ckt.AllowClassicOperations = $true

# Update circuit
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

After this operation has finished successfully, you will be able to view the circuit in the classic deployment model.

Run the following to get the details of the ExpressRoute circuit:

```
get-azurededicatedcircuit
```

You must be able to see the service key listed. You can now manage links to the ExpressRoute circuit using your standard classic deployment model commands for classic VNets and your standard ARM commands for ARM VNETs. The following articles will walk you through how to manage links to the ExpressRoute circuit:

- Link your virtual network to your ExpressRoute circuit in the Resource Manager deployment model
- Link your virtual network to your ExpressRoute circuit in the classic deployment model

Disable the ExpressRoute circuit to the classic deployment model

Run the following cmdlet to disable access to the classic deployment model:

```
# Get details of the ExpressRoute circuit
$ckt = Get-AzureRmExpressRouteCircuit -Name "DemoCkt" -ResourceGroupName "DemoRG"

#Set "Allow Classic Operations" to FALSE
$ckt.AllowClassicOperations = $false

# Update circuit
Set-AzureRmExpressRouteCircuit -ExpressRouteCircuit $ckt
```

After this operation has finished successfully, you will not be able to view the circuit in the classic deployment model.

Next steps

After you create your circuit, make sure that you do the following:

- Create and modify routing for your ExpressRoute circuit
- Link your virtual network to your ExpressRoute circuit

Router configuration samples to set up and manage routing

1/17/2017 • 4 min to read • Edit on GitHub

This page provides interface and routing configuration samples for Cisco IOS-XE and Juniper MX series routers. These are intended to be samples for guidance only and must not be used as is. You can work with your vendor to come up with appropriate configurations for your network.

> **IMPORTANT**
>
> Samples in this page are intended to be purely for guidance. You must work with your vendor's sales / technical team and your networking team to come up with appropriate configurations to meet your needs. Microsoft will not support issues related to configurations listed in this page. You must contact your device vendor for support issues.

Router configuration samples below apply to all peerings. Review ExpressRoute peerings and ExpressRoute routing requirements for more details on routing.

Cisco IOS-XE based routers

The samples in this section apply for any router running the IOS-XE OS family.

1. Configuring interfaces and sub-interfaces

You will require a sub interface per peering in every router you connect to Microsoft. A sub interface can be identified with a VLAN ID or a stacked pair of VLAN IDs and an IP address.

Dot1Q interface definition

This sample provides the sub-interface definition for a sub-interface with a single VLAN ID. The VLAN ID is unique per peering. The last octet of your IPv4 address will always be an odd number.

```
interface GigabitEthernet<Interface_Number>.<Number>
 encapsulation dot1Q <VLAN_ID>
 ip address <IPv4_Address><Subnet_Mask>
```

QinQ interface definition

This sample provides the sub-interface definition for a sub-interface with a two VLAN IDs. The outer VLAN ID (s-tag), if used remains the same across all the peerings. The inner VLAN ID (c-tag) is unique per peering. The last octet of your IPv4 address will always be an odd number.

```
interface GigabitEthernet<Interface_Number>.<Number>
 encapsulation dot1Q <s-tag> seconddot1Q <c-tag>
 ip address <IPv4_Address><Subnet_Mask>
```

2. Setting up eBGP sessions

You must setup a BGP session with Microsoft for every peering. The sample below enables you to setup a BGP session with Microsoft. If the IPv4 address you used for your sub interface was a.b.c.d, the IP address of the BGP neighbor (Microsoft) will be a.b.c.d+1. The last octet of the BGP neighbor's IPv4 address will always be an even number.

```
router bgp <Customer_ASN>
 bgp log-neighbor-changes
 neighbor <IP#2_used_by_Azure> remote-as 12076
 !
 address-family ipv4
 neighbor <IP#2_used_by_Azure> activate
 exit-address-family
!
```

3. Setting up prefixes to be advertised over the BGP session

You can configure your router to advertise select prefixes to Microsoft. You can do so using the sample below.

```
router bgp <Customer_ASN>
 bgp log-neighbor-changes
 neighbor <IP#2_used_by_Azure> remote-as 12076
 !
 address-family ipv4
  network <Prefix_to_be_advertised> mask <Subnet_mask>
  neighbor <IP#2_used_by_Azure> activate
 exit-address-family
!
```

4. Route maps

You can use route-maps and prefix lists to filter prefixes propagated into your network. You can use the sample below to accomplish the task. Ensure that you have appropriate prefix lists setup.

```
router bgp <Customer_ASN>
 bgp log-neighbor-changes
 neighbor <IP#2_used_by_Azure> remote-as 12076
 !
 address-family ipv4
  network <Prefix_to_be_advertised> mask <Subnet_mask>
  neighbor <IP#2_used_by_Azure> activate
  neighbor <IP#2_used_by_Azure> route-map <MS_Prefixes_Inbound> in
 exit-address-family
!
route-map <MS_Prefixes_Inbound> permit 10
 match ip address prefix-list <MS_Prefixes>
!
```

Juniper MX series routers

The samples in this section apply for any Juniper MX series routers.

1. Configuring interfaces and sub-interfaces

Dot1Q interface definition

This sample provides the sub-interface definition for a sub-interface with a single VLAN ID. The VLAN ID is unique per peering. The last octet of your IPv4 address will always be an odd number.

```
interfaces {
    vlan-tagging;
    <Interface_Number> {
        unit <Number> {
            vlan-id <VLAN_ID>;
            family inet {
                address <IPv4_Address/Subnet_Mask>;
            }
        }
    }
}
```

QinQ interface definition

This sample provides the sub-interface definition for a sub-interface with a two VLAN IDs. The outer VLAN ID (s-tag), if used remains the same across all the peerings. The inner VLAN ID (c-tag) is unique per peering. The last octet of your IPv4 address will always be an odd number.

```
interfaces {
    <Interface_Number> {
        flexible-vlan-tagging;
        unit <Number> {
            vlan-tags outer <S-tag> inner <C-tag>;
            family inet {
                address <IPv4_Address/Subnet_Mask>;
            }
        }
    }
}
```

2. Setting up eBGP sessions

You must setup a BGP session with Microsoft for every peering. The sample below enables you to setup a BGP session with Microsoft. If the IPv4 address you used for your sub interface was a.b.c.d, the IP address of the BGP neighbor (Microsoft) will be a.b.c.d+1. The last octet of the BGP neighbor's IPv4 address will always be an even number.

```
routing-options {
    autonomous-system <Customer_ASN>;
}
}
protocols {
    bgp {
        group <Group_Name> {
            peer-as 12076;
            neighbor <IP#2_used_by_Azure>;
        }
    }
}
```

3. Setting up prefixes to be advertised over the BGP session

You can configure your router to advertise select prefixes to Microsoft. You can do so using the sample below.

```
policy-options {
    policy-statement <Policy_Name> {
        term 1 {
            from protocol OSPF;
    route-filter <Prefix_to_be_advertised/Subnet_Mask> exact;
            then {
                accept;
            }
        }
    }
}
protocols {
    bgp {
        group <Group_Name> {
            export <Policy_Name>
            peer-as 12076;
            neighbor <IP#2_used_by_Azure>;
        }
    }
}
```

4. Route maps

You can use route-maps and prefix lists to filter prefixes propagated into your network. You can use the sample below to accomplish the task. Ensure that you have appropriate prefix lists setup.

```
policy-options {
    prefix-list MS_Prefixes {
        <IP_Prefix_1/Subnet_Mask>;
        <IP_Prefix_2/Subnet_Mask>;
    }
    policy-statement <MS_Prefixes_Inbound> {
        term 1 {
            from {
    prefix-list MS_Prefixes;
            }
            then {
                accept;
            }
        }
    }
}
protocols {
    bgp {
        group <Group_Name> {
            export <Policy_Name>
            import <MS_Prefixes_Inbound>
            peer-as 12076;
            neighbor <IP#2_used_by_Azure>;
        }
    }
}
```

Next Steps

See the ExpressRoute FAQ for more details.

Router configuration samples to set up and manage NAT

1/17/2017 • 4 min to read • Edit on GitHub

This page provides NAT configuration samples for Cisco ASA and Juniper SRX series routers. These are intended to be samples for guidance only and must not be used as is. You can work with your vendor to come up with appropriate configurations for your network.

> **IMPORTANT**
> Samples in this page are intended to be purely for guidance. You must work with your vendor's sales / technical team and your networking team to come up with appropriate configurations to meet your needs. Microsoft will not support issues related to configurations listed in this page. You must contact your device vendor for support issues.

- Router configuration samples below apply to Azure Public and Microsoft peerings. You must not configure NAT for Azure private peering. Review ExpressRoute peerings and ExpressRoute NAT requirements for more details.
- You MUST use separate NAT IP pools for connectivity to the internet and ExpressRoute. Using the same NAT IP pool across the internet and ExpressRoute will result in asymmetric routing and loss of connectivity.

Cisco ASA firewalls

PAT configuration for traffic from customer network to Microsoft

```
object network MSFT-PAT
  range <SNAT-START-IP> <SNAT-END-IP>


object-group network MSFT-Range
  network-object <IP> <Subnet_Mask>

object-group network on-prem-range-1
  network-object <IP> <Subnet-Mask>

object-group network on-prem-range-2
  network-object <IP> <Subnet-Mask>

object-group network on-prem
  network-object object on-prem-range-1
  network-object object on-prem-range-2

nat (outside,inside) source dynamic on-prem pat-pool MSFT-PAT destination static MSFT-Range MSFT-Range
```

PAT configuration for traffic from Microsoft to customer network

Interfaces and Direction:

```
Source Interface (where the traffic enters the ASA): inside
Destination Interface (where the traffic exits the ASA): outside
```

Configuration:

NAT Pool:

```
object network outbound-PAT
    host <NAT-IP>
```

Target Server:

```
object network Customer-Network
    network-object <IP> <Subnet-Mask>
```

Object Group for Customer IP Addresses

```
object-group network MSFT-Network-1
    network-object <MSFT-IP> <Subnet-Mask>

object-group network MSFT-PAT-Networks
    network-object object MSFT-Network-1
```

NAT Commands:

```
nat (inside,outside) source dynamic MSFT-PAT-Networks pat-pool outbound-PAT destination static Customer-Network
Customer-Network
```

Juniper SRX series routers

1. Create redundant Ethernet interfaces for the cluster

```
interfaces {
    reth0 {
        description "To Internal Network";
        vlan-tagging;
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 100 {
            vlan-id 100;
            family inet {
                address <IP-Address/Subnet-mask>;
            }
        }
    }
    reth1 {
        description "To Microsoft via Edge Router";
        vlan-tagging;
        redundant-ether-options {
            redundancy-group 2;
        }
        unit 100 {
            description "To Microsoft via Edge Router";
            vlan-id 100;
            family inet {
                address <IP-Address/Subnet-mask>;
            }
        }
    }
}
```

2. Create two security zones

- Trust Zone for internal network and Untrust Zone for external network facing Edge Routers
- Assign appropriate interfaces to the zones
- Allow services on the interfaces

security { zones { security-zone Trust { host-inbound-traffic { system-services { ping; } protocols { bgp; } } interfaces { reth0.100; } } security-zone Untrust { host-inbound-traffic { system-services { ping; } protocols { bgp; } } interfaces { reth1.100; } } } }

3. Create security policies between zones

```
security {
    policies {
        from-zone Trust to-zone Untrust {
            policy allow-any {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone Untrust to-zone Trust {
            policy allow-any {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
}
```

4. Configure NAT policies

- Create two NAT pools. One will be used to NAT traffic outbound to Microsoft and other from Microsoft to the customer.
- Create rules to NAT the respective traffic

```
security {
    nat {
        source {
            pool SNAT-To-ExpressRoute {
                routing-instance {
                    External-ExpressRoute;
                }
                address {
                    <NAT-IP-address/Subnet-mask>;
                }
            }
            pool SNAT-From-ExpressRoute {
                routing-instance {
                    Internal;
                }
                address {
                    <NAT-IP-address/Subnet-mask>;
                }
            }
            rule-set Outbound_NAT {
                from routing-instance Internal;
                to routing-instance External-ExpressRoute;
                rule SNAT-Out {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            pool {
                                SNAT-To-ExpressRoute;
                            }
                        }
                    }
                }
            }
            rule-set Inbound-NAT {
                from routing-instance External-ExpressRoute;
                to routing-instance Internal;
                rule SNAT-In {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            pool {
                                SNAT-From-ExpressRoute;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

5. Configure BGP to advertise selective prefixes in each direction

Refer to samples in Routing configuration samples page.

6. Create policies

```
routing-options {
             autonomous-system <Customer-ASN>;
}
policy-options {
    prefix-list Microsoft-Prefixes {
        <IP-Address/Subnet-Mask;
```

```
        <IP-Address/Subnet-Mask;
    }
    prefix-list private-ranges {
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
        100.64.0.0/10;
    }
    policy-statement Advertise-NAT-Pools {
        from {
            protocol static;
            route-filter <NAT-Pool-Address/Subnet-mask> prefix-length-range /32-/32;
        }
        then accept;
    }
    policy-statement Accept-from-Microsoft {
        term 1 {
            from {
                instance External-ExpressRoute;
                prefix-list-filter Microsoft-Prefixes orlonger;
            }
            then accept;
        }
        term deny {
            then reject;
        }
    }
    policy-statement Accept-from-Internal {
        term no-private {
            from {
                instance Internal;
                prefix-list-filter private-ranges orlonger;
            }
            then reject;
        }
        term bgp {
            from {
                instance Internal;
                protocol bgp;
            }
            then accept;
        }
        term deny {
            then reject;
        }
    }
}
routing-instances {
    Internal {
        instance-type virtual-router;
        interface reth0.100;
        routing-options {
            static {
                route <NAT-Pool-IP-Address/Subnet-mask> discard;
            }
            instance-import Accept-from-Microsoft;
        }
        protocols {
            bgp {
                group customer {
                    export <Advertise-NAT-Pools>;
                    peer-as <Customer-ASN-1>;
                    neighbor <BGP-Neighbor-IP-Address>;
                }
            }
        }
    }
    External-ExpressRoute {
        instance-type virtual-router;
```

```
        interface reth1.100;
        routing-options {
            static {
                route <NAT-Pool-IP-Address/Subnet-mask> discard;
            }
            instance-import Accept-from-Internal;
        }
        protocols {
            bgp {
                group edge-router {
                    export <Advertise-NAT-Pools>;
                    peer-as <Customer-Public-ASN>;
                    neighbor <BGP-Neighbor-IP-Address>;
                }
            }
        }
    }
}
```

Next steps

See the ExpressRoute FAQ for more details.

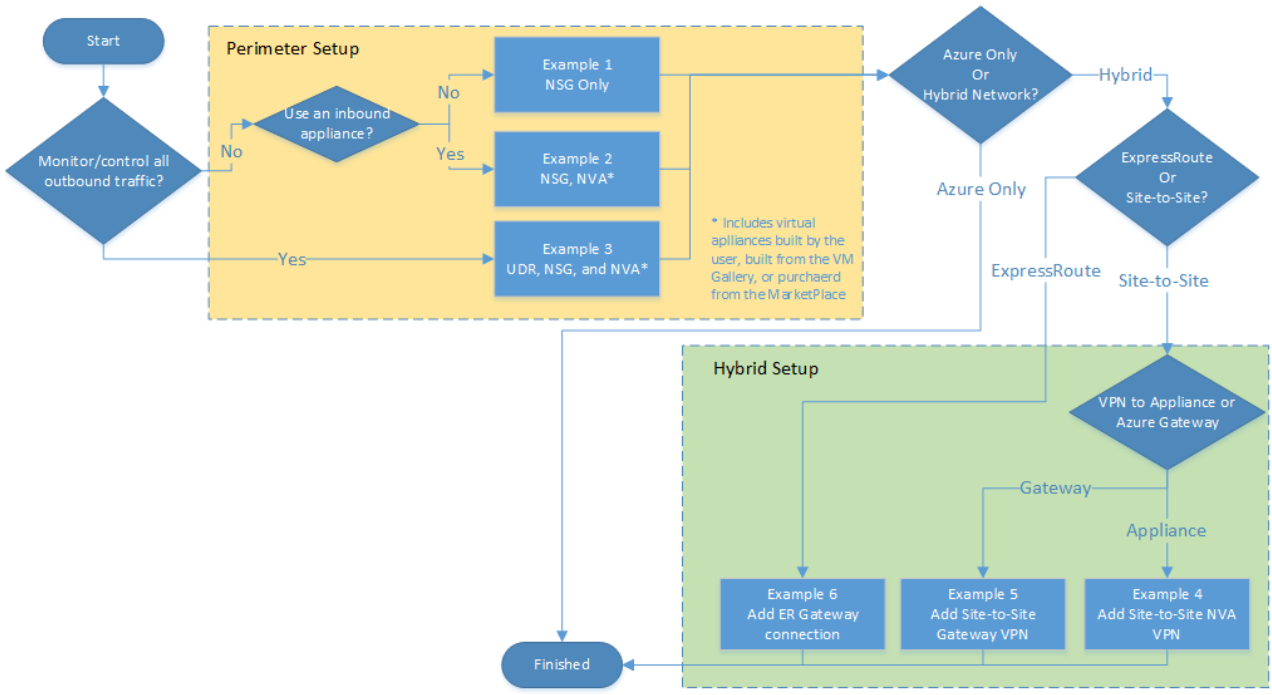
Microsoft cloud services and network security

1/17/2017 • 37 min to read • Edit on GitHub

Microsoft cloud services deliver hyper-scale services and infrastructure, enterprise-grade capabilities, and many choices for hybrid connectivity. Customers can choose to access these services either via the Internet or with Azure ExpressRoute, which provides private network connectivity. The Microsoft Azure platform allows customers to seamlessly extend their infrastructure into the cloud and build multi-tier architectures. Additionally, third parties can enable enhanced capabilities by offering security services and virtual appliances. This white paper provides an overview of security and architectural issues that customers should consider when using Microsoft cloud services accessed via ExpressRoute. It also covers creating more secure services in Azure virtual networks.

Fast start

The following logic chart can direct you to a specific example of the many security techniques available with the Azure platform. For quick reference, find the example that best fits your case. For expanded explanations, continue reading through the paper.



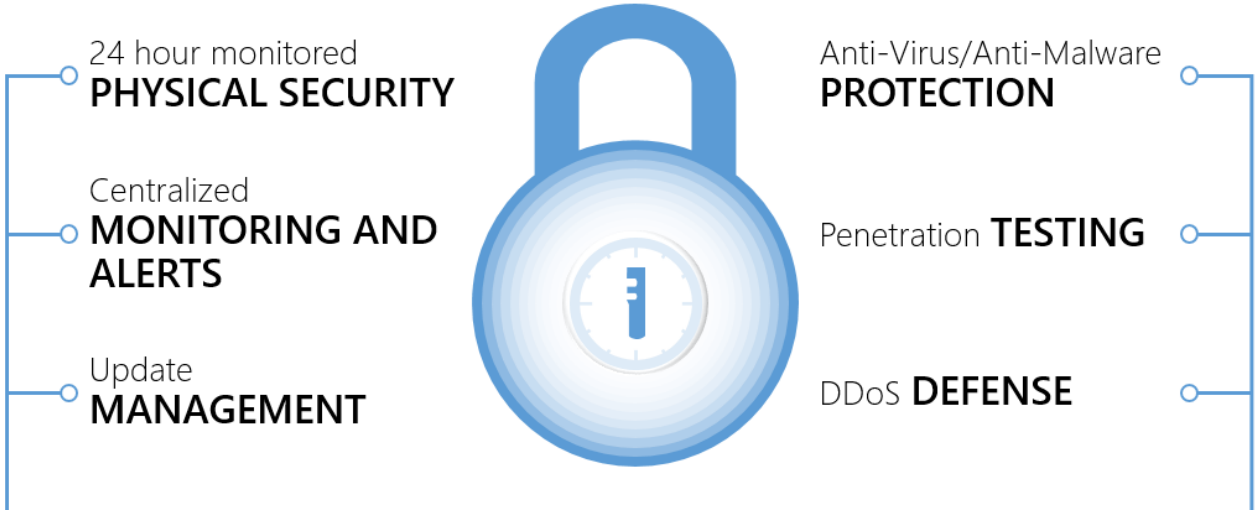
Example 1: Build a perimeter network (also known as DMZ, demilitarized zone, or screened subnet) to help protect applications with network security groups (NSGs).
Example 2: Build a perimeter network to help protect applications with a firewall and NSGs.
Example 3: Build a perimeter network to help protect networks with a firewall, user-defined route (UDR), and NSG.
Example 4: Add a hybrid connection with a site-to-site, virtual appliance virtual private network (VPN).
Example 5: Add a hybrid connection with a site-to-site, Azure VPN gateway.
Example 6: Add a hybrid connection with ExpressRoute.
Examples for adding connections between virtual networks, high availability, and service chaining will be added to this document over the next few months.

Microsoft compliance and infrastructure protection

To help organizations comply with national, regional, and industry-specific requirements governing the collection and use of individuals' data, Microsoft offers over 40 certifications and attestations. The most comprehensive set of any cloud service provider.

For more information, see the compliance information on the Microsoft Trust Center.

Microsoft has a comprehensive approach to protect cloud infrastructure needed to run hyper-scale global services. Microsoft cloud infrastructure includes hardware, software, networks, and administrative and operations staff, in addition to the physical data centers.

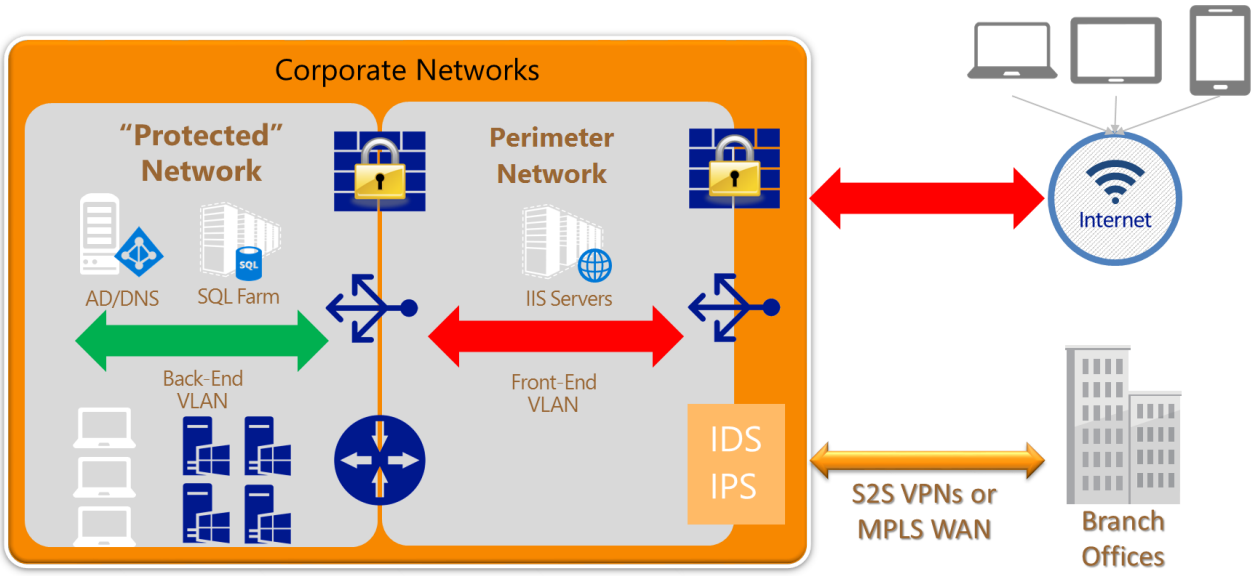


This approach provides a more secure foundation for customers to deploy their services in the Microsoft cloud. The next step is for customers to design and create a security architecture to protect these services.

Traditional security architectures and perimeter networks

Although Microsoft invests heavily in protecting the cloud infrastructure, customers must also protect their cloud services and resource groups. A multilayered approach to security provides the best defense. A perimeter network security zone protects internal network resources from an untrusted network. A perimeter network refers to the edges or parts of the network that sit between the Internet and the protected enterprise IT infrastructure.

In typical enterprise networks, the core infrastructure is heavily fortified at the perimeters, with multiple layers of security devices. The boundary of each layer consists of devices and policy enforcement points. Each layer can include a combination of the following network security devices: firewalls, Denial of Service (DoS) prevention, Intrusion Detection or Protection Systems (IDS/IPS), and VPN devices. Policy enforcement can take the form of firewall policies, access control lists (ACLs), or specific routing. The first line of defense in the network, directly accepting incoming traffic from the Internet, is a combination of these mechanisms to block attacks and harmful traffic while allowing legitimate requests further into the network. This traffic routes directly to resources in the perimeter network. That resource may then “talk” to resources deeper in the network, transiting the next boundary for validation first. The outermost layer is called the perimeter network because this part of the network is exposed to the Internet, usually with some form of protection on both sides. The following figure shows an example of a single subnet perimeter network in a corporate network, with two security boundaries.

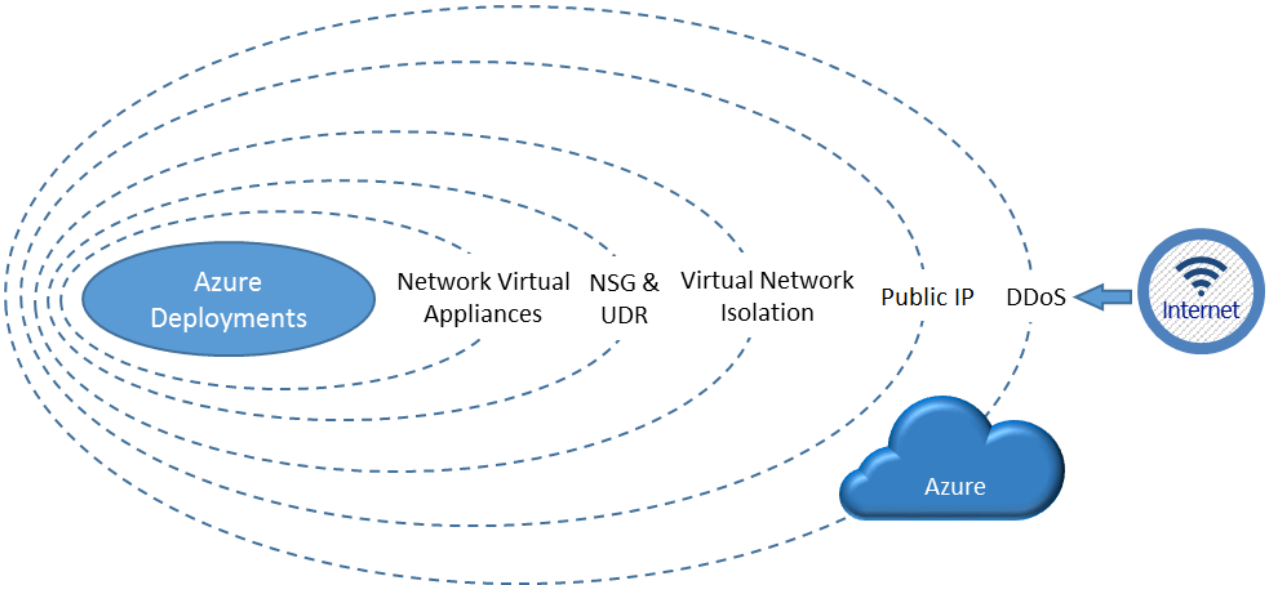


There are many architectures used to implement a perimeter network. These architectures can range from a simple load balancer to a multiple-subnet perimeter network with varied mechanisms at each boundary to block traffic and protect the deeper layers of the corporate network. How the perimeter network is built depends on the specific needs of the organization and its overall risk tolerance.

As customers move their workloads to public clouds, it is critical to support similar capabilities for perimeter network architecture in Azure to meet compliance and security requirements. This document provides guidelines on how customers can build a secure network environment in Azure. It focuses on the perimeter network, but also includes a comprehensive discussion of many aspects of network security. The following questions inform this discussion:

- How can a perimeter network in Azure be built?
- What are some of the Azure features available to build the perimeter network?
- How can back-end workloads be protected?
- How are Internet communications controlled to the workloads in Azure?
- How can the on-premises networks be protected from deployments in Azure?
- When should native Azure security features be used versus third-party appliances or services?

The following diagram shows various layers of security Azure provides to customers. These layers are both native in the Azure platform itself and customer-defined features:



Inbound from the Internet, Azure DDoS helps protect against large-scale attacks against Azure. The next layer is

customer-defined public IP addresses (endpoints), which are used to determine which traffic can pass through the cloud service to the virtual network. Native Azure virtual network isolation ensures complete isolation from all other networks and that traffic only flows through user configured paths and methods. These paths and methods are the next layer, where NSGs, UDR, and network virtual appliances can be used to create security boundaries to protect the application deployments in the protected network.

The next section provides an overview of Azure virtual networks. These virtual networks are created by customers, and are what their deployed workloads are connected to. Virtual networks are the basis of all the network security features required to establish a perimeter network to protect customer deployments in Azure.

Overview of Azure virtual networks

Before Internet traffic can get to the Azure virtual networks, there are two layers of security inherent to the Azure platform:

1. **DDoS protection**: DDoS protection is a layer of the Azure physical network that protects the Azure platform itself from large-scale Internet-based attacks. These attacks use multiple "bot" nodes in an attempt to overwhelm an Internet service. Azure has a robust DDoS protection mesh on all inbound, outbound, and cross-Azure region connectivity. This DDoS protection layer has no user configurable attributes and is not accessible to the customer. The DDoS protection layer protects Azure as a platform from large-scale attacks, it also monitors out-bound traffic and cross-Azure region traffic. Using network virtual appliances on the VNet, additional layers of resilience can be configured by the customer against a smaller scale attack that doesn't trip the platform level protection. An example of DDoS in action; if an internet facing IP address was attacked by a large-scale DDoS attack, Azure would detect the sources of the attacks and scrub the offending traffic before it reached its intended destination. In almost all cases, the attacked endpoint isn't affected by the attack. In the rare cases that an endpoint is affected, no traffic is affected to other endpoints, only the attacked endpoint. Thus other customers and services would see no impact from that attack. It's critical to note that Azure DDoS is only looking for large-scale attacks. It is possible that your specific service could be overwhelmed before the platform level protection thresholds are exceeded. For example, a web site on a single A0 IIS server, could be taken offline by a DDoS attack before Azure platform level DDoS protection registered a threat.
2. **Public IP Addresses**: Public IP addresses (enabled via service endpoints, Public IP addresses, Application Gateway, and other Azure features that present a public IP address to the internet routed to your resource) allow cloud services or resource groups to have public Internet IP addresses and ports exposed. The endpoint uses Network Address Translation (NAT) to route traffic to the internal address and port on the Azure virtual network. This path is the primary way for external traffic to pass into the virtual network. The Public IP addresses are configurable to determine which traffic is passed in, and how and where it's translated on to the virtual network.

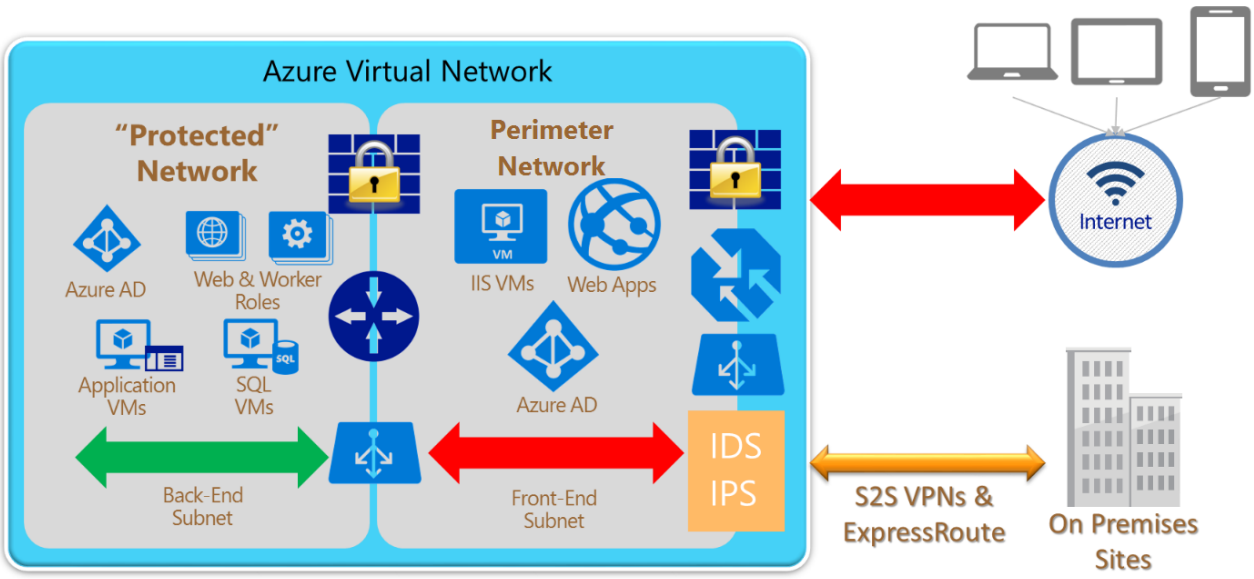
Once traffic reaches the virtual network, there are many features that come into play. Azure virtual networks are the foundation for customers to attach their workloads and where basic network-level security applies. It is a private network (a virtual network overlay) in Azure for customers with the following features and characteristics:

- **Traffic isolation**: A virtual network is the traffic isolation boundary on the Azure platform. Virtual machines (VMs) in one virtual network cannot communicate directly to VMs in a different virtual network, even if both virtual networks are created by the same customer. Isolation is a critical property that ensures customer VMs and communication remains private within a virtual network.

> **NOTE**
> Traffic isolation refers only to traffic *inbound* to the virtual network. By default outbound traffic from the VNet to the internet is allowed, but can be prevented if desired by NSGs.

- **Multi-tier topology**: Virtual networks allow customers to define multi-tier topology by allocating subnets and designating separate address spaces for different elements or "tiers" of their workloads. These logical groupings and topologies enable customers to define different access policy based on the workload types, and also control traffic flows between the tiers.
- **Cross-premises connectivity**: Customers can establish cross-premises connectivity between a virtual network and multiple on-premises sites or other virtual networks in Azure. To construct a connection, customers can use VNet Peering, Azure VPN Gateways, third-party network virtual appliances, or ExpressRoute. Azure supports site-to-site (S2S) VPNs using standard IPsec/IKE protocols and ExpressRoute private connectivity.
- **NSG** allows customers to create rules (ACLs) at the desired level of granularity: network interfaces, individual VMs, or virtual subnets. Customers can control access by permitting or denying communication between the workloads within a virtual network, from systems on customer's networks via cross-premises connectivity, or direct Internet communication.
- **UDR** and **IP Forwarding** allow customers to define the communication paths between different tiers within a virtual network. Customers can deploy a firewall, IDS/IPS, and other virtual appliances, and route network traffic through these security appliances for security boundary policy enforcement, auditing, and inspection.
- **Network virtual appliances** in the Azure Marketplace: Security appliances such as firewalls, load balancers, and IDS/IPS are available in the Azure Marketplace and the VM Image Gallery. Customers can deploy these appliances into their virtual networks, and specifically, at their security boundaries (including the perimeter network subnets) to complete a multi-tiered secure network environment.

With these features and capabilities, one example of how a perimeter network architecture could be constructed in Azure is the following diagram:



Perimeter network characteristics and requirements

The perimeter network is the front end of the network, directly interfacing communication from the Internet. The incoming packets should flow through the security appliances, such as the firewall, IDS, and IPS, before reaching the back-end servers. Internet-bound packets from the workloads can also flow through the security appliances in the perimeter network for policy enforcement, inspection, and auditing purposes, before leaving the network. Additionally, the perimeter network can host cross-premises VPN gateways between customer virtual networks and on-premises networks.

Perimeter network characteristics

Referencing the previous figure, some of the characteristics of a good perimeter network are as follows:

- Internet-facing:
  - The perimeter network subnet itself is Internet-facing, directly communicating with the Internet.

    - Public IP addresses, VIPs, and/or service endpoints pass Internet traffic to the front-end network and devices.
    - Inbound traffic from the Internet passes through security devices before other resources on the front-end network.
    - If outbound security is enabled, traffic passes through security devices, as the final step, before passing to the Internet.
- Protected network:
    - There is no direct path from the Internet to the core infrastructure.
    - Channels to the core infrastructure must traverse through security devices such as NSGs, firewalls, or VPN devices.
    - Other devices must not bridge Internet and the core infrastructure.
    - Security devices on both the Internet-facing and the protected network facing boundaries of the perimeter network (for example, the two firewall icons shown in the previous figure) may actually be a single virtual appliance with differentiated rules or interfaces for each boundary. For example, one physical device, logically separated, handling the load for both boundaries of the perimeter network.
- Other common practices and constraints:
    - Workloads must not store business critical information.
    - Access and updates to perimeter network configurations and deployments are limited to only authorized administrators.

Perimeter network requirements

To enable these characteristics, follow these guidelines on virtual network requirements to implement a successful perimeter network:

- **Subnet architecture:** Specify the virtual network such that an entire subnet is dedicated as the perimeter network, separated from other subnets in the same virtual network. This separation ensures the traffic between the perimeter network and other internal or private subnet tiers flows through a firewall or IDS/IPS virtual appliance. User-defined routes on the boundary subnets are required to forward this traffic to the virtual appliance.
- **NSG:** The perimeter network subnet itself should be open to allow communication with the Internet, but that does not mean customers should be bypassing NSGs. Follow common security practices to minimize the network surfaces exposed to the Internet. Lock down the remote address ranges allowed to access the deployments or the specific application protocols and ports that are open. There may be circumstances, however, in which a complete lock-down is not possible. For example, if customers have an external website in Azure, the perimeter network should allow the incoming web requests from any public IP addresses, but should only open the web application ports: TCP on port 80 and/or TCP on port 443.
- **Routing table:** The perimeter network subnet itself should be able to communicate to the Internet directly, but should not allow direct communication to and from the back end or on-premises networks without going through a firewall or security appliance.
- **Security appliance configuration:** To route and inspect packets between the perimeter network and the rest of the protected networks, the security appliances such as firewall, IDS, and IPS devices may be multi-homed. They may have separate NICs for the perimeter network and the back-end subnets. The NICs in the perimeter network communicate directly to and from the Internet, with the corresponding NSGs and the perimeter network routing table. The NICs connecting to the back-end subnets have more restricted NSGs and routing tables of the corresponding back-end subnets.
- **Security appliance functionality:** The security appliances deployed in the perimeter network typically perform the following functionality:
    - Firewall: Enforcing firewall rules or access control policies for the incoming requests.
    - Threat detection and prevention: Detecting and mitigating malicious attacks from the Internet.
    - Auditing and logging: Maintaining detailed logs for auditing and analysis.

- Reverse proxy: Redirecting the incoming requests to the corresponding back-end servers. This redirection involves mapping and translating the destination addresses on the front-end devices, typically firewalls, to the back-end server addresses.
- Forward proxy: Providing NAT and performing auditing for communication initiated from within the virtual network to the Internet.
- Router: Forwarding incoming and cross-subnet traffic inside the virtual network.
- VPN device: Acting as the cross-premises VPN gateways for cross-premises VPN connectivity between customer on-premises networks and Azure virtual networks.
- VPN server: Accepting VPN clients connecting to Azure virtual networks.

> **TIP**
> Keep the following two groups separate: the individuals authorized to access the perimeter network security gear and the individuals authorized as application development, deployment, or operations administrators. Keeping these groups separate allows for a segregation of duties and prevents a single person from bypassing both applications security and network security controls.

Questions to be asked when building network boundaries

In this section, unless specifically mentioned, the term "networks" refers to private Azure virtual networks created by a subscription administrator. The term doesn't refer to the underlying physical networks within Azure.

Also, Azure virtual networks are often used to extend traditional on-premises networks. It is possible to incorporate either site-to-site or ExpressRoute hybrid networking solutions with perimeter network architectures. This hybrid link is an important consideration in building network security boundaries.

The following three questions are critical to answer when you're building a network with a perimeter network and multiple security boundaries.

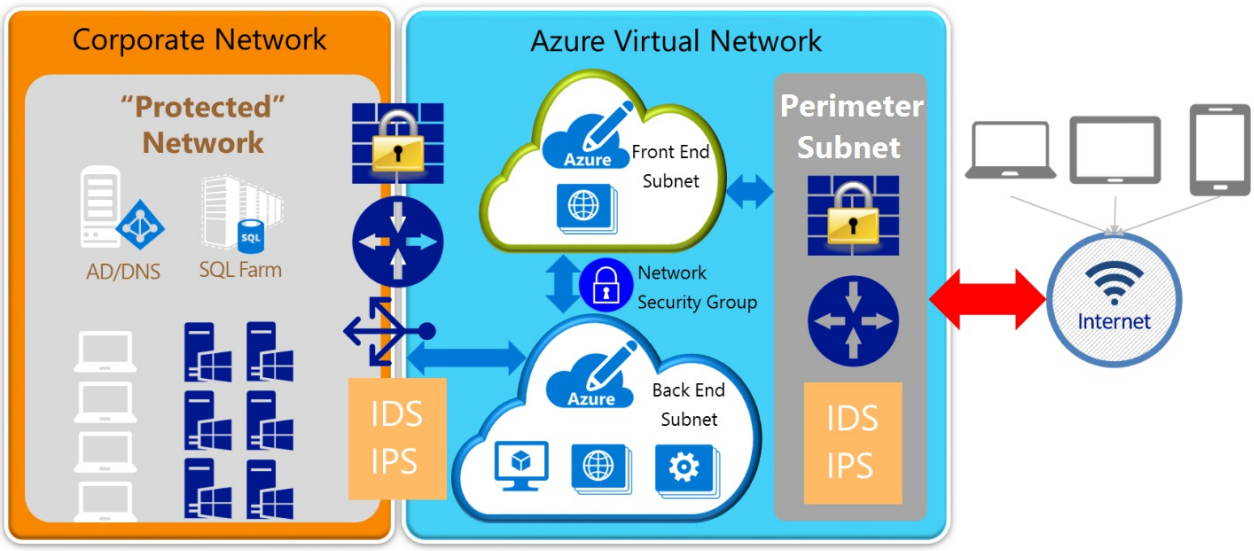
1) How many boundaries are needed?

The first decision point is to decide how many security boundaries are needed in a given scenario:

- A single boundary: One on the front-end perimeter network, between the virtual network and the Internet.
- Two boundaries: One on the Internet side of the perimeter network, and another between the perimeter network subnet and the back-end subnets in the Azure virtual networks.
- Three boundaries: One on the Internet side of the perimeter network, one between the perimeter network and back-end subnets, and one between the back-end subnets and the on-premises network.
- N boundaries: A variable number. Depending on security requirements, there is no limit to the number of security boundaries that can be applied in a given network.

The number and type of boundaries needed vary based on a company’s risk tolerance and the specific scenario being implemented. This decision is often made together with multiple groups within an organization, often including a risk and compliance team, a network and platform team, and an application development team. People with knowledge of security, the data involved, and the technologies being used should have a say in this decision to ensure the appropriate security stance for each implementation.

> **TIP**
> Use the smallest number of boundaries that satisfy the security requirements for a given situation. With more boundaries, operations and troubleshooting can be more difficult, as well as the management overhead involved with managing the multiple boundary policies over time. However, insufficient boundaries increase risk. Finding the balance is critical.



The preceding figure shows a high-level view of a three security boundary network. The boundaries are between the perimeter network and the Internet, the Azure front-end and back-end private subnets, and the Azure back-end subnet and the on-premises corporate network.

2) Where are the boundaries located?

Once the number of boundaries are decided, where to implement them is the next decision point. There are generally three choices:

- Using an Internet-based intermediary service (for example, a cloud-based Web application firewall, which is not discussed in this document)
- Using native features and/or network virtual appliances in Azure
- Using physical devices on the on-premises network

On purely Azure networks, the options are native Azure features (for example, Azure Load Balancers) or network virtual appliances from the rich partner ecosystem of Azure (for example, Check Point firewalls).

If a boundary is needed between Azure and an on-premises network, the security devices can reside on either side of the connection (or both sides). Thus a decision must be made on the location to place security gear.

In the previous figure, the Internet-to-perimeter network and the front-to-back-end boundaries are entirely contained within Azure, and must be either native Azure features or network virtual appliances. Security devices on the boundary between Azure (back-end subnet) and the corporate network could be either on the Azure side or the on-premises side, or even a combination of devices on both sides. There can be significant advantages and disadvantages to either option that must be seriously considered.

For example, using existing physical security gear on the on-premises network side has the advantage that no new gear is needed. It just needs reconfiguration. The disadvantage, however, is that all traffic must come back from Azure to the on-premises network to be seen by the security gear. Thus Azure-to-Azure traffic could incur significant latency, and affect application performance and user experience, if it was forced back to the on-premises network for security policy enforcement.

3) How are the boundaries implemented?

Each security boundary will likely have different capability requirements (for example, IDS and firewall rules on the Internet side of the perimeter network, but only ACLs between the perimeter network and back-end subnet). Deciding on which device (or how many devices) to use depends on the scenario and security requirements. In the following section, examples 1, 2, and 3 discuss some options that could be used. Reviewing the Azure native network features and the devices available in Azure from the partner ecosystem shows the myriad options available to solve virtually any scenario.

Another key implementation decision point is how to connect the on-premises network with Azure. Should you use the Azure virtual gateway or a network virtual appliance? These options are discussed in greater detail in the

following section (examples 4, 5, and 6).

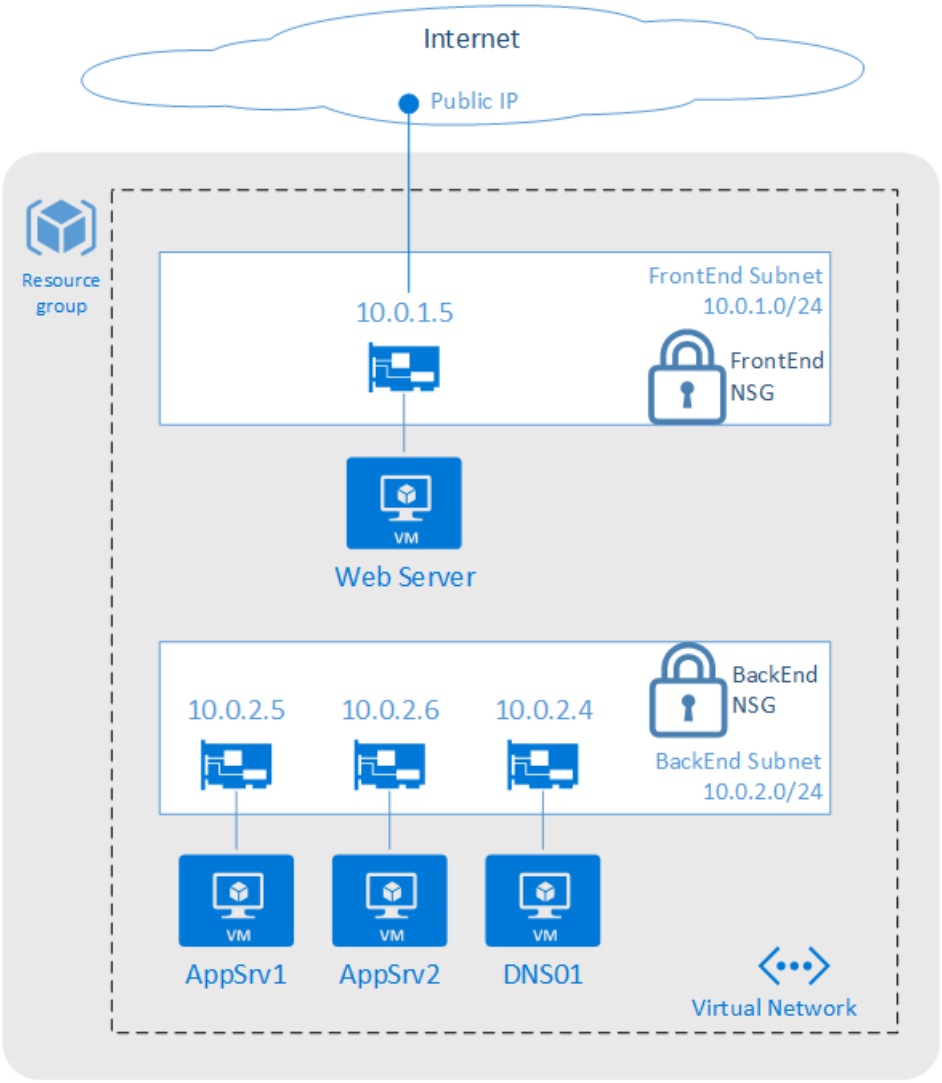
Additionally, traffic between virtual networks within Azure may be needed. These scenarios will be added in the future.

Once you know the answers to the previous questions, the Fast Start section can help identify which examples are most appropriate for a given scenario.

Examples: Building security boundaries with Azure virtual networks

Example 1 Build a perimeter network to help protect applications with NSGs

Back to Fast start | Detailed build instructions for this example



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with two subnets: "FrontEnd" and "BackEnd"
- A Network Security Group that is applied to both subnets
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")
- A public IP associated with the application web server

For scripts and an Azure Resource Manager template, see the detailed build instructions.

NSG description

In this example, an NSG group is built and then loaded with six rules.

> **TIP**
>
> Generally speaking, you should create your specific "Allow" rules first, followed by the more generic "Deny" rules. The given priority dictates which rules are evaluated first. Once traffic is found to apply to a specific rule, no further rules are evaluated. NSG rules can apply in either the inbound or outbound direction (from the perspective of the subnet).

Declaratively, the following rules are being built for inbound traffic:

1. Internal DNS traffic (port 53) is allowed.
2. RDP traffic (port 3389) from the Internet to any Virtual Machine is allowed.
3. HTTP traffic (port 80) from the Internet to web server (IIS01) is allowed.
4. Any traffic (all ports) from IIS01 to AppVM1 is allowed.
5. Any traffic (all ports) from the Internet to the entire virtual network (both subnets) is denied.
6. Any traffic (all ports) from the front-end subnet to the back-end subnet is denied.

With these rules bound to each subnet, if an HTTP request was inbound from the Internet to the web server, both rules 3 (allow) and 5 (deny) would apply. But because rule 3 has a higher priority, only it would apply, and rule 5 would not come into play. Thus the HTTP request would be allowed to the web server. If that same traffic was trying to reach the DNS01 server, rule 5 (deny) would be the first to apply, and the traffic would not be allowed to pass to the server. Rule 6 (deny) blocks the front-end subnet from talking to the back-end subnet (except for allowed traffic in rules 1 and 4). This rule-set protects the back-end network in case an attacker compromises the web application on the front end. The attacker would have limited access to the back-end "protected" network (only to resources exposed on the AppVM01 server).

There is a default outbound rule that allows traffic out to the Internet. For this example, we're allowing outbound traffic and not modifying any outbound rules. To lock down traffic in both directions, user-defined routing is required (see example 3).

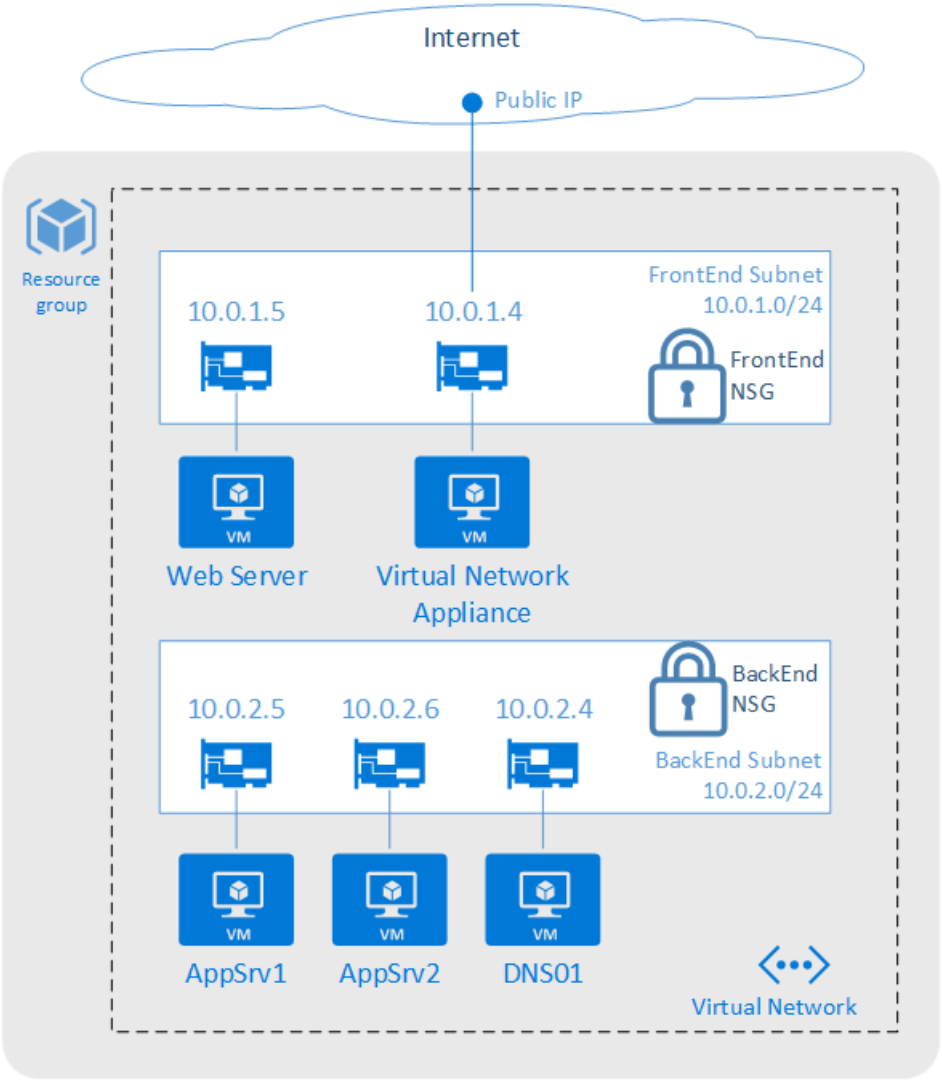
Conclusion

This example is a relatively simple and straightforward way of isolating the back-end subnet from inbound traffic. For more information, see the detailed build instructions. These instructions include:

- How to build this perimeter network with classic PowerShell scripts.
- How to build this perimeter network with an Azure Resource Manager template.
- Detailed descriptions of each NSG command.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 2 Build a perimeter network to help protect applications with a firewall and NSGs

Back to Fast start | Detailed build instructions for this example



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with two subnets: "FrontEnd" and "BackEnd"
- A Network Security Group that is applied to both subnets
- A network virtual appliance, in this case a firewall, connected to the front-end subnet
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")

For scripts and an Azure Resource Manager template, see the detailed build instructions.

NSG description

In this example, an NSG group is built and then loaded with six rules.

> **TIP**
>
> Generally speaking, you should create your specific "Allow" rules first, followed by the more generic "Deny" rules. The given priority dictates which rules are evaluated first. Once traffic is found to apply to a specific rule, no further rules are evaluated. NSG rules can apply in either the inbound or outbound direction (from the perspective of the subnet).

Declaratively, the following rules are being built for inbound traffic:

1. Internal DNS traffic (port 53) is allowed.

2. RDP traffic (port 3389) from the Internet to any Virtual Machine is allowed.
3. Any Internet traffic (all ports) to the network virtual appliance (firewall) is allowed.
4. Any traffic (all ports) from IIS01 to AppVM1 is allowed.
5. Any traffic (all ports) from the Internet to the entire virtual network (both subnets) is denied.
6. Any traffic (all ports) from the front-end subnet to the back-end subnet is denied.

With these rules bound to each subnet, if an HTTP request was inbound from the Internet to the firewall, both rules 3 (allow) and 5 (deny) would apply. But because rule 3 has a higher priority, only it would apply, and rule 5 would not come into play. Thus the HTTP request would be allowed to the firewall. If that same traffic was trying to reach the IIS01 server, even though it's on the front-end subnet, rule 5 (deny) would apply, and the traffic would not be allowed to pass to the server. Rule 6 (deny) blocks the front-end subnet from talking to the back-end subnet (except for allowed traffic in rules 1 and 4). This rule-set protects the back-end network in case an attacker compromises the web application on the front end. The attacker would have limited access to the back-end "protected" network (only to resources exposed on the AppVM01 server).

There is a default outbound rule that allows traffic out to the Internet. For this example, we're allowing outbound traffic and not modifying any outbound rules. To lock down traffic in both directions, user-defined routing is required (see example 3).

Firewall rule description

On the firewall, forwarding rules should be created. Since this example only routes Internet traffic in-bound to the firewall and then to the web server, only one forwarding network address translation (NAT) rule is needed.

The forwarding rule accepts any inbound source address that hits the firewall trying to reach HTTP (port 80 or 443 for HTTPS). It's sent out of the firewall's local interface and redirected to the web server with the IP Address of 10.0.1.5.

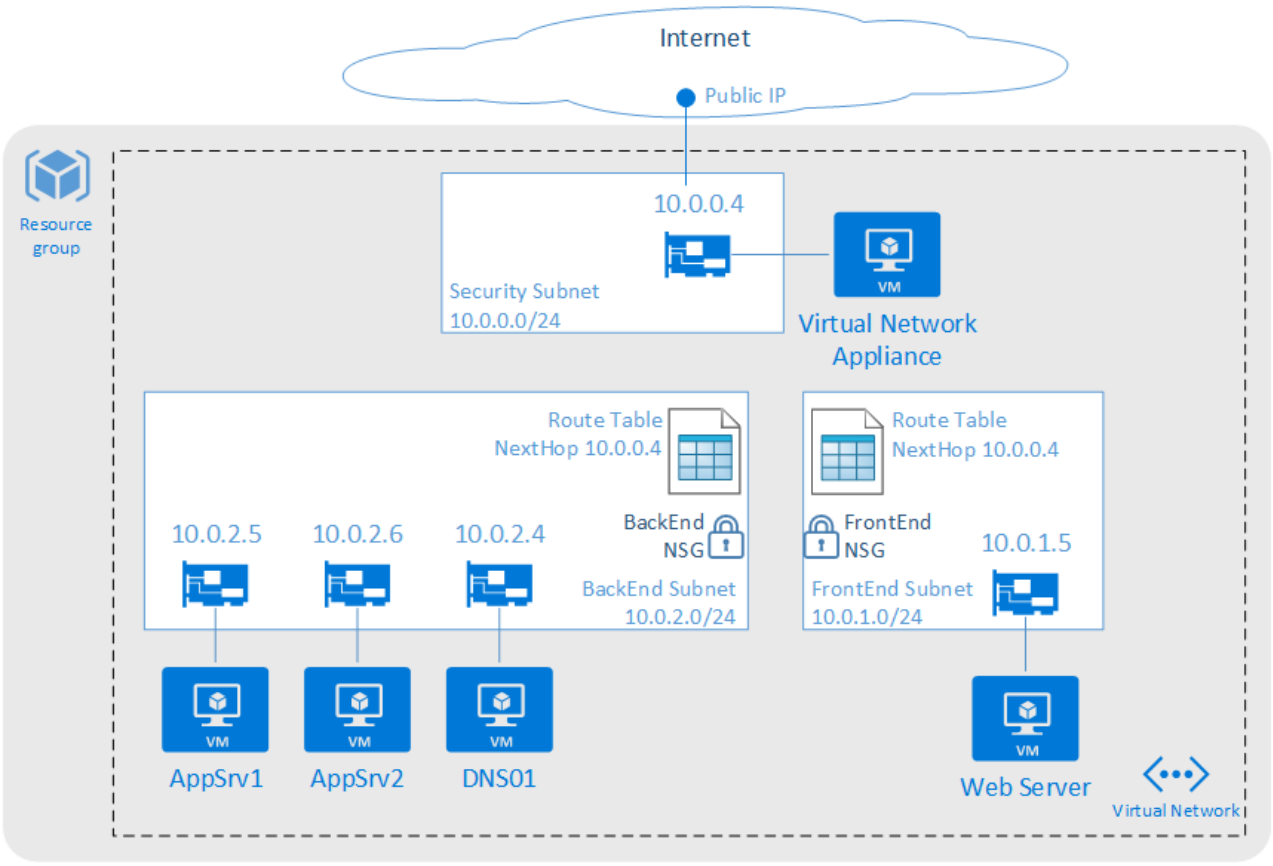
Conclusion

This example is a relatively straightforward way of protecting your application with a firewall and isolating the back-end subnet from inbound traffic. For more information, see the detailed build instructions. These instructions include:

- How to build this perimeter network with classic PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed descriptions of each NSG command and firewall rule.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 3 Build a perimeter network to help protect networks with a firewall and UDR and NSG

Back to Fast start | Detailed build instructions for this example



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with three subnets: "SecNet", "FrontEnd", and "BackEnd"
- A network virtual appliance, in this case a firewall, connected to the SecNet subnet
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")

For scripts and an Azure Resource Manager template, see the detailed build instructions.

UDR description

By default, the following system routes are defined as:

```
Effective routes :
 Address Prefix    Next hop type    Next hop IP address Status   Source
 --------------    -------------    ------------------- ------   ------
 {10.0.0.0/16}     VNETLocal                            Active   Default
 {0.0.0.0/0}       Internet                             Active   Default
 {10.0.0.0/8}      Null                                 Active   Default
 {100.64.0.0/10}   Null                                 Active   Default
 {172.16.0.0/12}   Null                                 Active   Default
 {192.168.0.0/16}  Null                                 Active   Default
```

The VNETLocal is always one or more defined address prefixes that make up the virtual network for that specific network (that is, it changes from virtual network to virtual network, depending on how each specific virtual network is defined). The remaining system routes are static and default as indicated in the table.

In this example, two routing tables are created, one each for the front-end and back-end subnets. Each table is loaded with static routes appropriate for the given subnet. In this example, each table has three routes that direct all traffic (0.0.0.0/0) through the firewall (Next hop = Virtual Appliance IP address):

1. Local subnet traffic with no Next Hop defined to allow local subnet traffic to bypass the firewall.
2. Virtual network traffic with a Next Hop defined as firewall. This next hop overrides the default rule that allows local virtual network traffic to route directly.
3. All remaining traffic (0/0) with a Next Hop defined as the firewall.

> **TIP**
>
> Not having the local subnet entry in the UDR breaks local subnet communications.
>
> - In our example, 10.0.1.0/24 pointing to VNETLocal is critical! Without it, packet leaving the Web Server (10.0.1.4) destined to another local server (for example) 10.0.1.25 will fail as they will be sent to the NVA. The NVA will send it to the subnet, and the subnet will resend it to the NVA in an infinite loop.
> - The chances of a routing loop are typically higher on appliances with multiple NICs that are connected to separate subnets, which is often of traditional, on-premises appliances.

Once the routing tables are created, they must be bound to their subnets. The front-end subnet routing table, once created and bound to the subnet, would look like this output:

```
Effective routes :
 Address Prefix    Next hop type    Next hop IP address Status   Source
 --------------    -------------    ------------------- ------   ------
 {10.0.1.0/24}     VNETLocal                            Active
 {10.0.0.0/16}     VirtualAppliance 10.0.0.4            Active
 {0.0.0.0/0}       VirtualAppliance 10.0.0.4            Active
```

> **NOTE**
>
> UDR can now be applied to the gateway subnet on which the ExpressRoute circuit is connected.
>
> Examples of how to enable your perimeter network with ExpressRoute or site-to-site networking are shown in examples 3 and 4.

IP Forwarding description

IP Forwarding is a companion feature to UDR. IP Forwarding is a setting on a virtual appliance that allows it to receive traffic not specifically addressed to the appliance, and then forward that traffic to its ultimate destination.

For example, if AppVM01 makes a request to the DNS01 server, UDR would route this traffic to the firewall. With IP Forwarding enabled, the traffic for the DNS01 destination (10.0.2.4) is accepted by the appliance (10.0.0.4) and then forwarded to its ultimate destination (10.0.2.4). Without IP Forwarding enabled on the firewall, traffic would not be accepted by the appliance even though the route table has the firewall as the next hop. To use a virtual appliance, it's critical to remember to enable IP Forwarding along with UDR.

NSG description

In this example, an NSG group is built and then loaded with a single rule. This group is then bound only to the front-end and back-end subnets (not the SecNet). Declaratively the following rule is being built:

- Any traffic (all ports) from the Internet to the entire virtual network (all subnets) is denied.

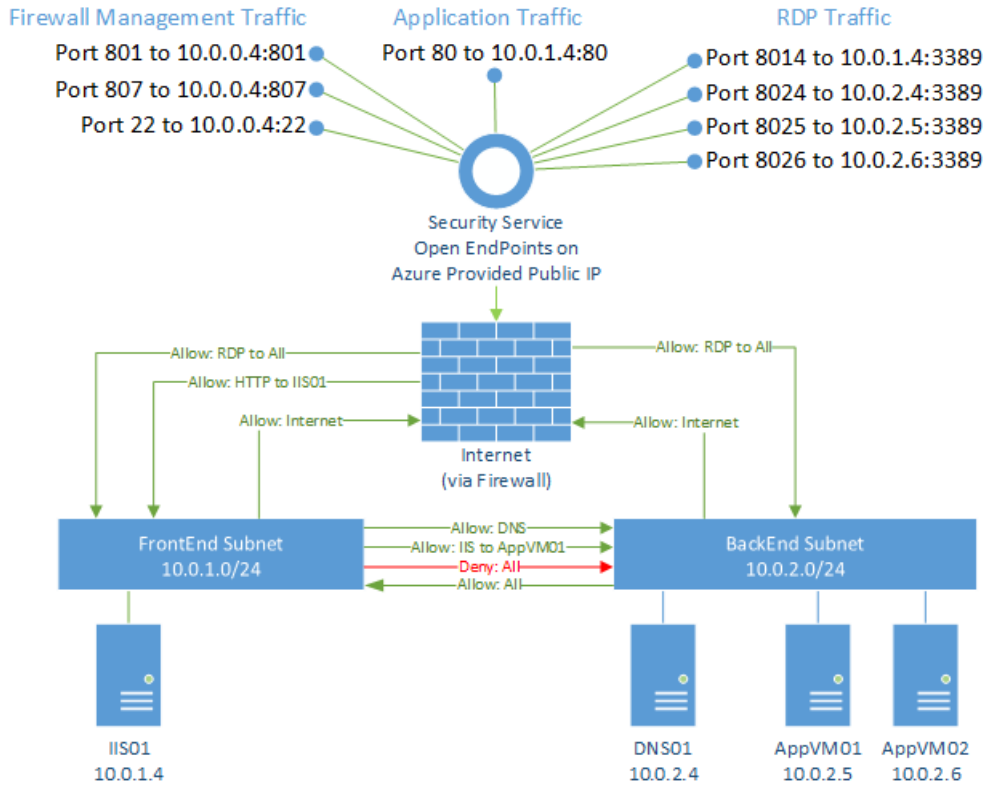
Although NSGs are used in this example, its main purpose is as a secondary layer of defense against manual misconfiguration. The goal is to block all inbound traffic from the Internet to either the front-end or back-end subnets. Traffic should only flow through the SecNet subnet to the firewall (and then, if appropriate, on to the front-end or back-end subnets). Plus, with the UDR rules in place, any traffic that did make it into the front-end or back-end subnets would be directed out to the firewall (thanks to UDR). The firewall would see this traffic as an asymmetric flow and would drop the outbound traffic. Thus there are three layers of security protecting the subnets:

- No Public IP addresses on any FrontEnd or BackEnd NICs.
- NSGs denying traffic from the Internet.
- The firewall dropping asymmetric traffic.

One interesting point regarding the NSG in this example is that it contains only one rule, which is to deny Internet traffic to the entire virtual network, including the Security subnet. However, since the NSG is only bound to the front-end and back-end subnets, the rule isn't processed on traffic inbound to the Security subnet. As a result, traffic flows to the Security subnet.

Firewall rules

On the firewall, forwarding rules should be created. Since the firewall is blocking or forwarding all inbound, outbound, and intra-virtual network traffic, many firewall rules are needed. Also, all inbound traffic hits the Security Service public IP address (on different ports), to be processed by the firewall. A best practice is to diagram the logical flows before setting up the subnets and firewall rules, to avoid rework later. The following figure is a logical view of the firewall rules for this example:



> **NOTE**
>
> Based on the Network Virtual Appliance used, the management ports vary. In this example, a Barracuda NextGen Firewall is referenced, which uses ports 22, 801, and 807. Consult the appliance vendor documentation to find the exact ports used for management of the device being used.

Firewall rules description

In the preceding logical diagram, the security subnet is not shown because the firewall is the only resource on that subnet. The diagram is showing the firewall rules and how they logically allow or deny traffic flows, not the actual routed path. Also, the external ports selected for the RDP traffic are higher ranged ports (8014 – 8026) and were selected to loosely align with the last two octets of the local IP address for easier readability (for example, local server address 10.0.1.4 is associated with external port 8014). Any higher non-conflicting ports, however, could be used.

For this example, we need seven types of rules:

- External rules (for inbound traffic):

1. Firewall management rule: This App Redirect rule allows traffic to pass to the management ports of the network virtual appliance.
2. RDP rules (for each Windows server): These four rules (one for each server) allow management of the individual servers via RDP. The four RDP rules could also be collapsed into one rule, depending on the capabilities of the network virtual appliance being used.
3. Application traffic rules: There are two of these rules, the first for the front-end web traffic, and the second for the back-end traffic (for example, web server to data tier). The configuration of these rules depends on the network architecture (where your servers are placed) and traffic flows (which direction the traffic flows, and which ports are used).
    - The first rule allows the actual application traffic to reach the application server. While the other rules allow for security and management, application traffic rules are what allow external users or services to access the applications. For this example, there is a single web server on port 80. Thus a single firewall application rule redirects inbound traffic to the external IP, to the web servers internal IP address. The redirected traffic session would be translated via NAT to the internal server.
    - The second rule is the back-end rule to allow the web server to talk to the AppVM01 server (but not AppVM02) via any port.

- Internal rules (for intra-virtual network traffic)
    1. Outbound to Internet rule: This rule allows traffic from any network to pass to the selected networks. This rule is usually a default rule already on the firewall, but in a disabled state. This rule should be enabled for this example.
    2. DNS rule: This rule allows only DNS (port 53) traffic to pass to the DNS server. For this environment, most traffic from the front end to the back end is blocked. This rule specifically allows DNS from any local subnet.
    3. Subnet to subnet rule: This rule is to allow any server on the back-end subnet to connect to any server on the front-end subnet (but not the reverse).
- Fail-safe rule (for traffic that doesn't meet any of the previous):
    1. Deny all traffic rule: This deny rule should always be the final rule (in terms of priority), and as such if a traffic flow fails to match any of the preceding rules it is dropped by this rule. This rule is a default rule and usually in-place and active. No modifications are usually needed to this rule.

> **TIP**
> On the second application traffic rule, to simplify this example, any port is allowed. In a real scenario, the most specific port and address ranges should be used to reduce the attack surface of this rule.

Once the previous rules are created, it's important to review the priority of each rule to ensure traffic is allowed or denied as desired. For this example, the rules are in priority order.

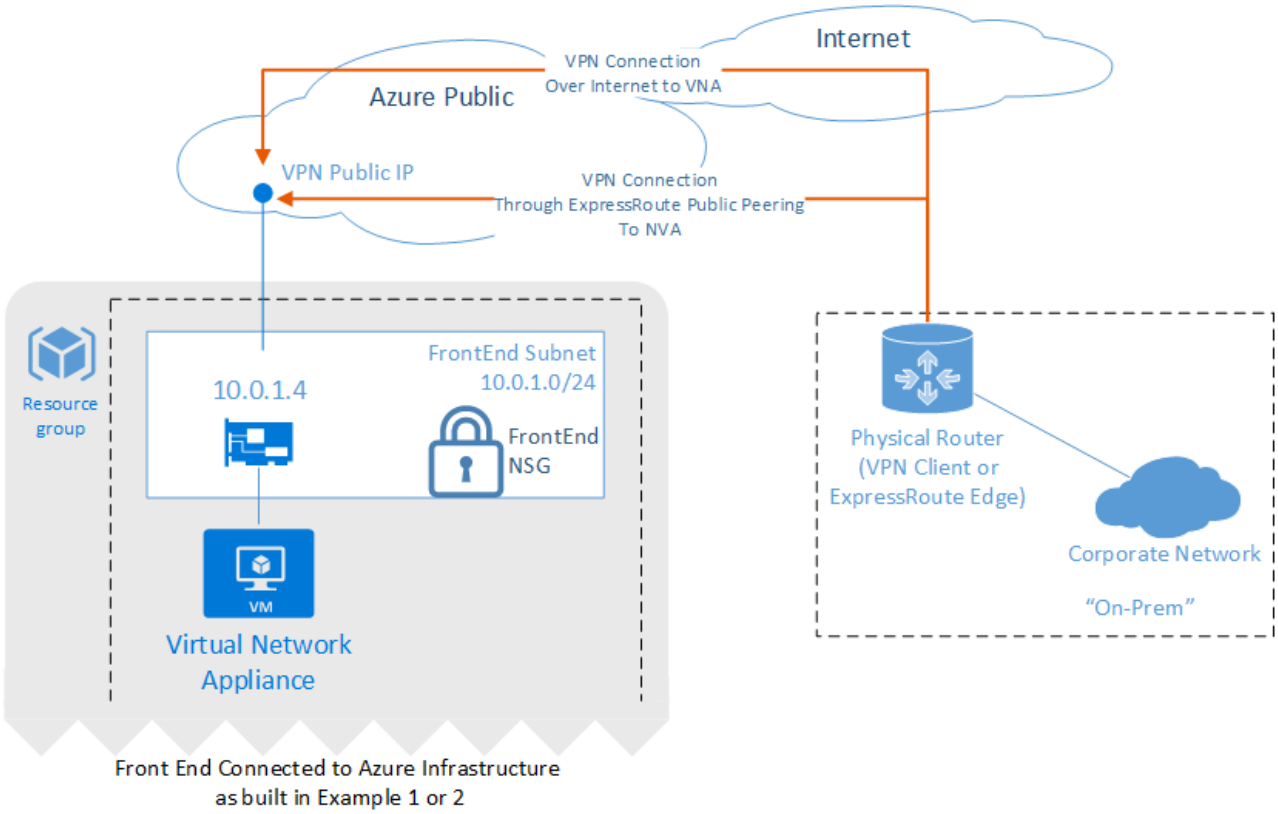
Conclusion

This example is a more complex but complete way of protecting and isolating the network than the previous examples. (Example 2 protects just the application, and Example 1 just isolates subnets). This design allows for monitoring traffic in both directions, and protects not just the inbound application server but enforces network security policy for all servers on this network. Also, depending on the appliance used, full traffic auditing and awareness can be achieved. For more information, see the detailed build instructions. These instructions include:

- How to build this example perimeter network with classic PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed descriptions of each UDR, NSG command, and firewall rule.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 4 Add a hybrid connection with a site-to-site, virtual appliance VPN

Back to Fast start | Detailed build instructions available soon



Environment description

Hybrid networking using a network virtual appliance (NVA) can be added to any of the perimeter network types described in examples 1, 2, or 3.

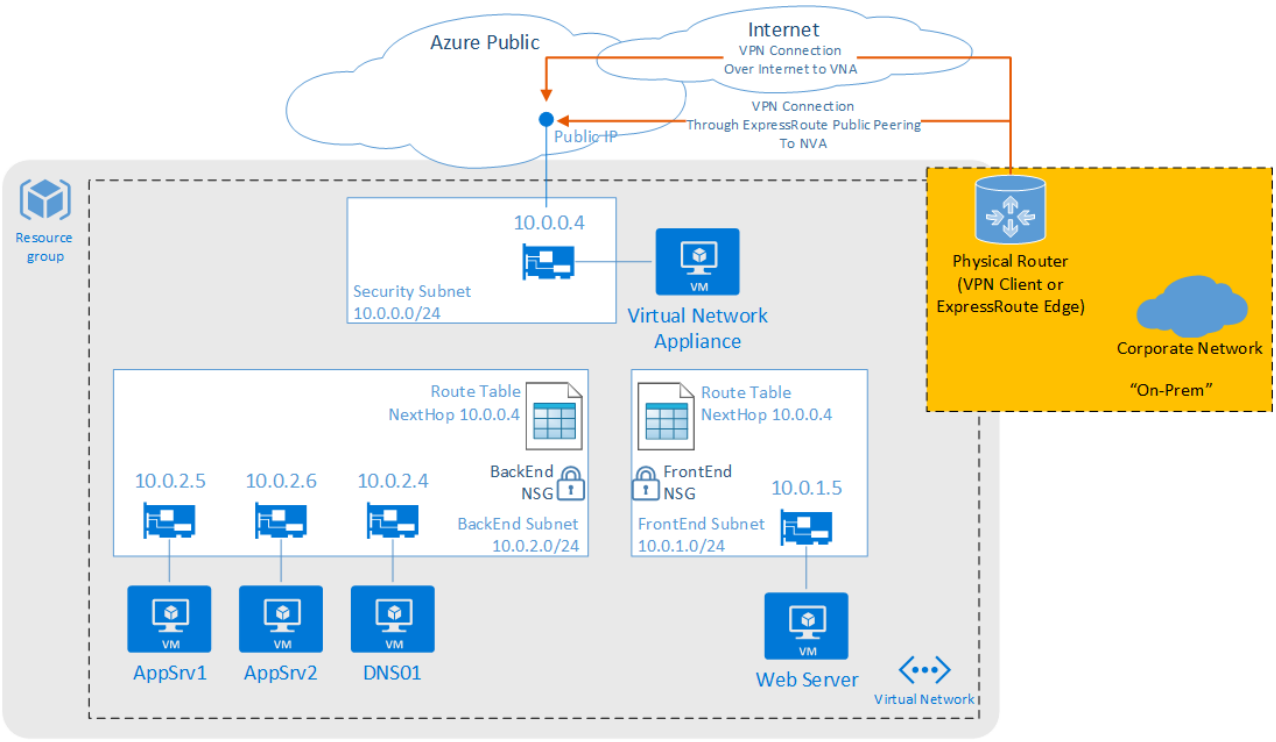
As shown in the previous figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an NVA.

> **NOTE**
>
> If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This static route should route to the NVA VPN IP address out your corporate Internet and not via the ExpressRoute connection. The NAT required on the ExpressRoute Azure Public Peering option can break the VPN session.

Once the VPN is in place, the NVA becomes the central hub for all networks and subnets. The firewall forwarding rules determine which traffic flows are allowed, are translated via NAT, are redirected, or are dropped (even for traffic flows between the on-premises network and Azure).

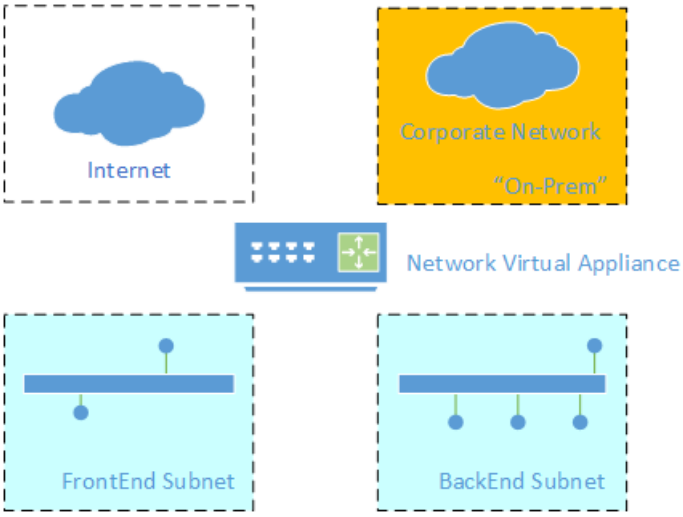
Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 3, and then adding a site-to-site VPN hybrid network connection, produces the following design:



The on-premises router, or any other network device that is compatible with your NVA for VPN, would be the VPN client. This physical device would be responsible for initiating and maintaining the VPN connection with your NVA.

Logically to the NVA, the network looks like four separate "security zones" with the rules on the NVA being the primary director of traffic between these zones:



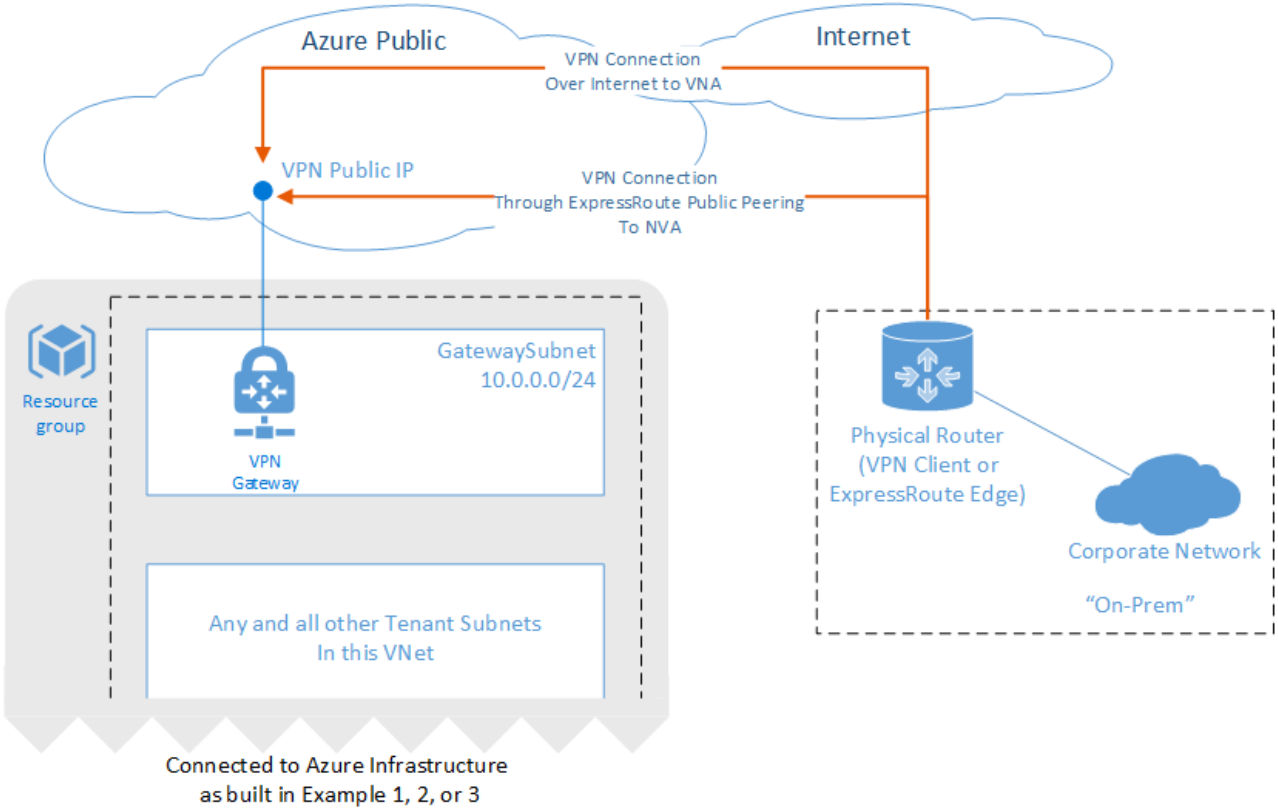
Conclusion

The addition of a site-to-site VPN hybrid network connection to an Azure virtual network can extend the on-premises network into Azure in a secure manner. In using a VPN connection, your traffic is encrypted and routes via the Internet. The NVA in this example provides a central location to enforce and manage the security policy. For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

Example 5 Add a hybrid connection with a site-to-site, Azure VPN gateway

Back to Fast start | Detailed build instructions available soon



Environment description

Hybrid networking using an Azure VPN gateway can be added to either perimeter network type described in examples 1 or 2.

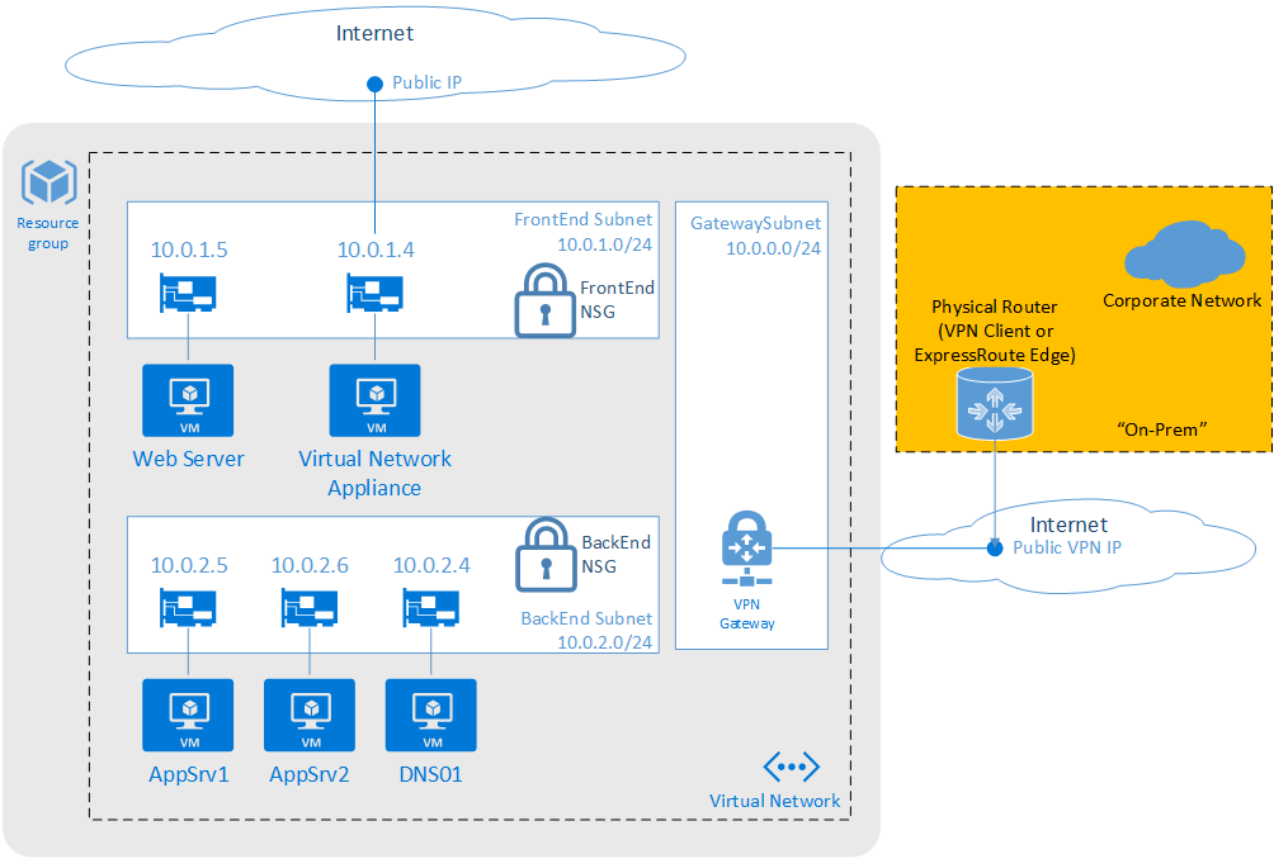
As shown in the preceding figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an Azure VPN gateway.

> **NOTE**
>
> If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This static route should route to the NVA VPN IP address out your corporate Internet and not via the ExpressRoute WAN. The NAT required on the ExpressRoute Azure Public Peering option can break the VPN session.

The following figure shows the two network edges in this example. On the first edge, the NVA and NSGs control traffic flows for intra-Azure networks and between Azure and the Internet. The second edge is the Azure VPN gateway, which is a separate and isolated network edge between on-premises and Azure.

Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 1, and then adding a site-to-site VPN hybrid network connection, produces the following design:



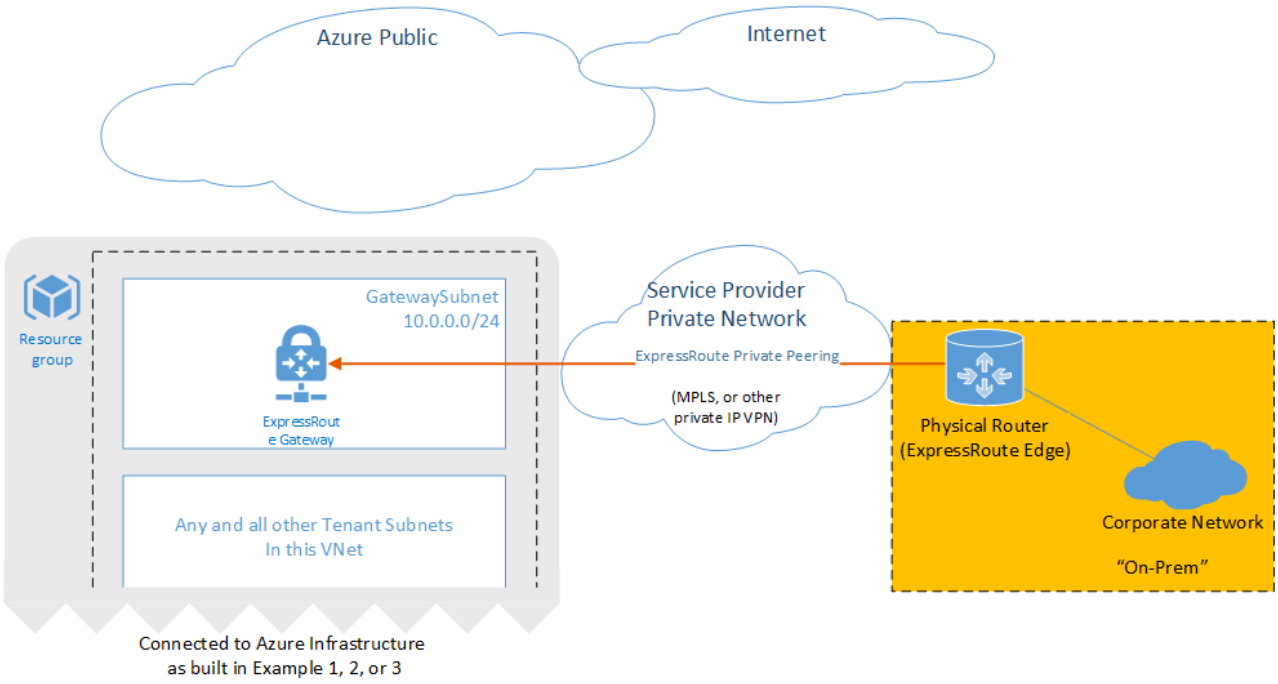
Conclusion

The addition of a site-to-site VPN hybrid network connection to an Azure virtual network can extend the on-premises network into Azure in a secure manner. Using the native Azure VPN gateway, your traffic is IPSec encrypted and routes via the Internet. Also, using the Azure VPN gateway can provide a lower-cost option (no additional licensing cost as with third-party NVAs). This option is most economical in example 1, where no NVA is used. For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

Example 6 Add a hybrid connection with ExpressRoute

Back to Fast start | Detailed build instructions available soon



Environment description

Hybrid networking using an ExpressRoute private peering connection can be added to either perimeter network type described in examples 1 or 2.

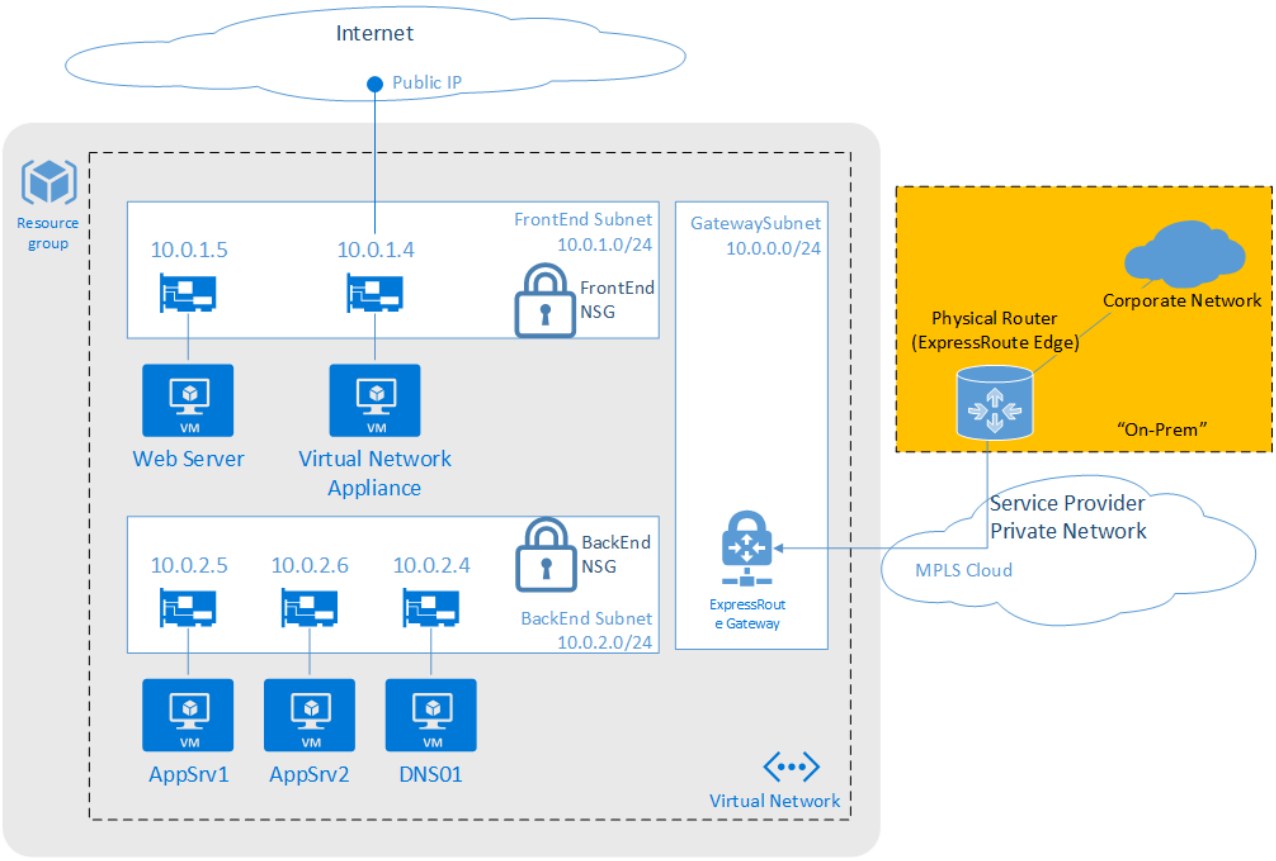
As shown in the preceding figure, ExpressRoute private peering provides a direct connection between your on-premises network and the Azure virtual network. Traffic transits only the service provider network and the Microsoft Azure network, never touching the Internet.

> **TIP**
> Using ExpressRoute keeps corporate network traffic off the Internet. It also allows for service level agreements from your ExpressRoute provider. The Azure Gateway can pass up to 10 Gbps with ExpressRoute, whereas with site-to-site VPNs, the Azure Gateway maximum throughput is 200 Mbps.

As seen in the following diagram, with this option the environment now has two network edges. The NVA and NSG control traffic flows for intra-Azure networks and between Azure and the Internet, while the gateway is a separate and isolated network edge between on-premises and Azure.

Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 1, and then adding an ExpressRoute hybrid network connection, produces the following design:



Conclusion

The addition of an ExpressRoute Private Peering network connection can extend the on-premises network into Azure in a secure, lower latency, higher performing manner. Also, using the native Azure Gateway, as in this example, provides a lower-cost option (no additional licensing as with third-party NVAs). For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

References

Helpful websites and documentation

- Access Azure with Azure Resource Manager:
- Accessing Azure with PowerShell: https://docs.microsoft.com/powershell/azureps-cmdlets-docs/
- Virtual networking documentation: https://docs.microsoft.com/azure/virtual-network/
- Network security group documentation: https://docs.microsoft.com/azure/virtual-network/virtual-networks-nsg
- User-defined routing documentation: https://docs.microsoft.com/azure/virtual-network/virtual-networks-udr-overview
- Azure virtual gateways: https://docs.microsoft.com/azure/vpn-gateway/
- Site-to-Site VPNs: https://docs.microsoft.com/azure/vpn-gateway/vpn-gateway-create-site-to-site-rm-powershell
- ExpressRoute documentation (be sure to check out the "Getting Started" and "How To" sections): https://docs.microsoft.com/azure/expressroute/

Asymmetric routing with multiple network paths

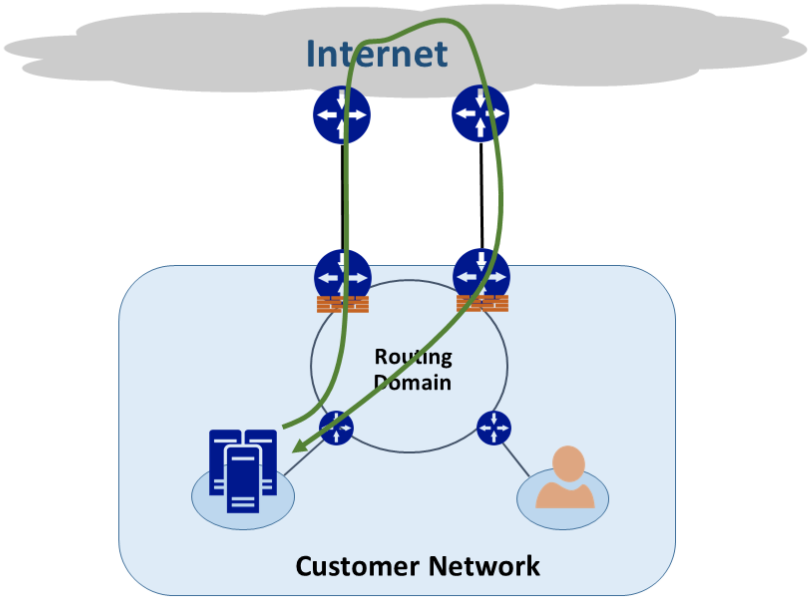
1/17/2017 • 6 min to read • Edit on GitHub

This article explains how forward and return network traffic might take different routes when multiple paths are available between network source and destination.

It's important to understand two concepts to understand asymmetric routing. One is the effect of multiple network paths. The other is how devices, like a firewall, keep state. These types of devices are called stateful devices. A combination of these two factors creates scenarios in which network traffic is dropped by a stateful device because the stateful device didn't detect that traffic originated with the device itself.

Multiple network paths

When an enterprise network has only one link to the Internet through their Internet service provider, all traffic to and from the Internet travels the same path. Often, companies purchase multiple circuits, as redundant paths, to improve network uptime. When this happens, it's possible that traffic that goes outside of the network, to the Internet, goes through one link, and the return traffic goes through a different link. This is commonly known as asymmetric routing. In asymmetric routing, reverse network traffic takes a different path from the original flow.



Although it primarily occurs on the Internet, asymmetric routing also applies to other combinations of multiple paths. It applies, for example, both to an Internet path and a private path that go to the same destination, and to multiple private paths that go to the same destination.

Each router along the way, from source to destination, computes the best path to reach a destination. The router's determination of best possible path is based on two main factors:

- Routing between external networks is based on a routing protocol, Border Gateway Protocol (BGP). BGP takes advertisements from neighbors and runs them through a series of steps to determine the best path to the intended destination. It stores the best path in its routing table.
- The length of a subnet mask associated with a route influences routing paths. If a router receives multiple advertisements for the same IP address but with different subnet masks, the router prefers the advertisement with a longer subnet mask because it's considered a more specific route.

Stateful devices

Routers look at the IP header of a packet for routing purposes. Some devices look even deeper inside the packet. Typically, these devices look at Layer4 (Transmission Control Protocol, or TCP; or User Datagram Protocol, or UDP), or even Layer7 (Application Layer) headers. These kinds of devices are either security devices or bandwidth-optimization devices.

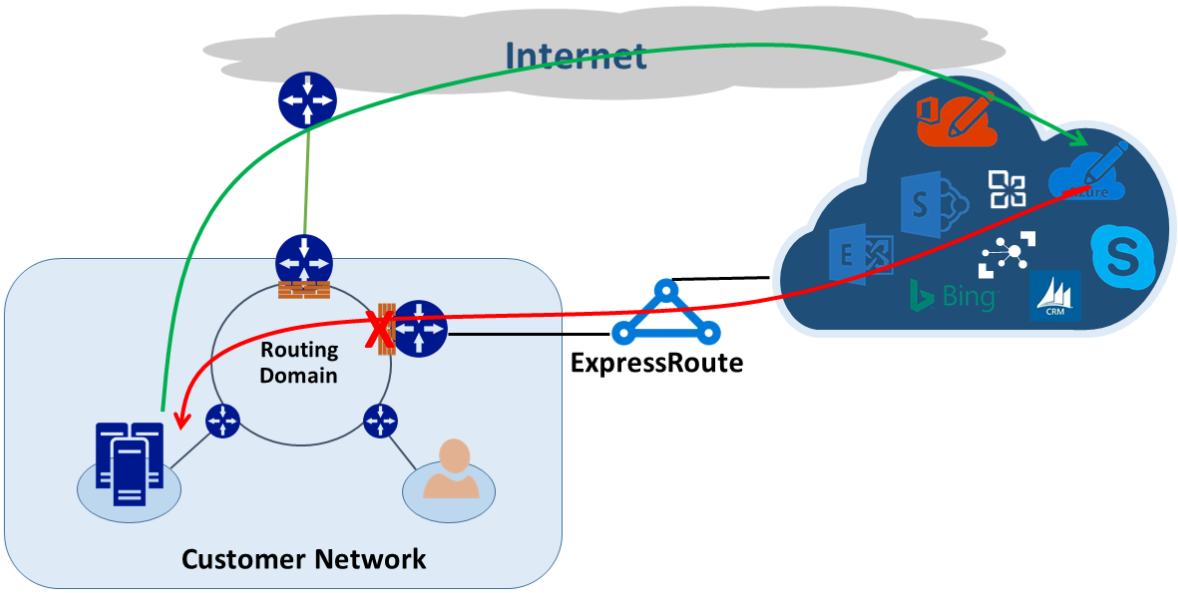
A firewall is a common example of a stateful device. A firewall allows or denies a packet to pass through its interfaces based on various fields such as protocol, TCP/UDP port, and URL headers. This level of packet inspection puts a heavy processing load on the device. To improve performance, the firewall inspects the first packet of a flow. If it allows the packet to proceed, it keeps the flow information in its state table. All subsequent packets related to this flow are allowed based on the initial determination. A packet that is part of an existing flow might arrive at the firewall. If the firewall has no prior state information about it, the firewall drops the packet.

Asymmetric routing with ExpressRoute

When you connect to Microsoft through Azure ExpressRoute, your network changes like this:

- You have multiple links to Microsoft. One link is your existing Internet connection, and the other is via ExpressRoute. Some traffic to Microsoft might go through the Internet but come back via ExpressRoute, or vice versa.
- You receive more specific IP addresses via ExpressRoute. So, for traffic from your network to Microsoft for services offered via ExpressRoute, routers always prefer ExpressRoute.

To understand the effect these two changes have on a network, let's consider some scenarios. As an example, you have only one circuit to the Internet and you consume all Microsoft services via the Internet. The traffic from your network to Microsoft and back traverses the same Internet link and passes through the firewall. The firewall records the flow as it sees the first packet and return packets are allowed because the flow exists in the state table.



Then, you turn on ExpressRoute and consume services offered by Microsoft over ExpressRoute. All other services from Microsoft are consumed over the Internet. You deploy a separate firewall at your edge that is connected to ExpressRoute. Microsoft advertises more specific prefixes to your network over ExpressRoute for specific services. Your routing infrastructure chooses ExpressRoute as the preferred path for those prefixes. If you are not advertising your public IP addresses to Microsoft over ExpressRoute, Microsoft communicates with your public IP addresses via

the Internet. Forward traffic from your network to Microsoft uses ExpressRoute, and reverse traffic from Microsoft uses the Internet. When the firewall at the edge sees a response packet for a flow that it does not find in the state table, it drops the return traffic.

If you choose to use the same network address translation (NAT) pool for ExpressRoute and for the Internet, you'll see similar issues with the clients in your network on private IP addresses. Requests for services like Windows Update go via the Internet because IP addresses for these services are not advertised via ExpressRoute. However, the return traffic comes back via ExpressRoute. If Microsoft receives an IP address with the same subnet mask from the Internet and ExpressRoute, it prefers ExpressRoute over the Internet. If a firewall or another stateful device that is on your network edge and facing ExpressRoute has no prior information about the flow, it drops the packets that belong to that flow.

Asymmetric routing solutions

You have two main options to solve the problem of asymmetric routing. One is through routing, and the other is by using source-based NAT (SNAT).

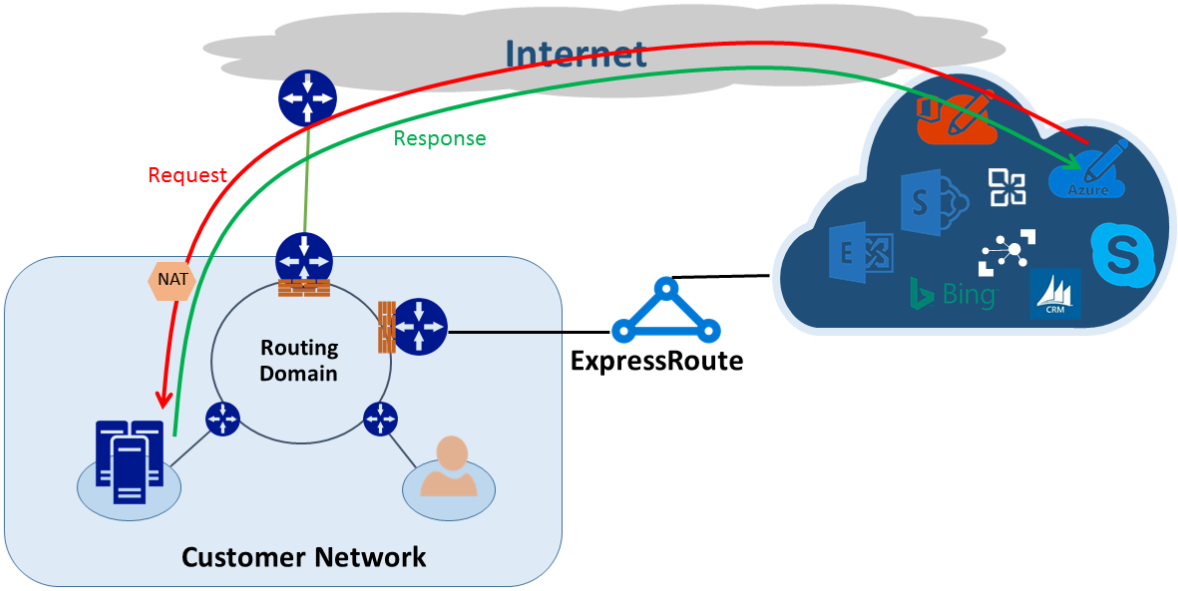
Routing

Ensure that your public IP addresses are advertised to appropriate wide area network (WAN) links. For example, if you want to use the Internet for authentication traffic and ExpressRoute for your mail traffic, you should not advertise your Active Directory Federation Services (AD FS) public IP addresses over ExpressRoute. Similarly, be sure not to expose an on-premises AD FS server to IP addresses that the router receives over ExpressRoute. Routes received over ExpressRoute are more specific so they make ExpressRoute the preferred path for authentication traffic to Microsoft. This causes asymmetric routing.

If you want to use ExpressRoute for authentication, make sure that you are advertising AD FS public IP addresses over ExpressRoute without NAT. This way, traffic that originates from Microsoft and goes to an on-premises AD FS server goes over ExpressRoute. Return traffic from customer to Microsoft uses ExpressRoute because it's the preferred route over the Internet.

Source-based NAT

Another way of solving asymmetric routing issues is by using SNAT. For example, you have not advertised the public IP address of an on-premises Simple Mail Transfer Protocol (SMTP) server over ExpressRoute because you intend to use the Internet for this type of communication. A request that originates with Microsoft and then goes to your on-premises SMTP server traverses the Internet. You SNAT the incoming request to an internal IP address. Reverse traffic from the SMTP server goes to the edge firewall (which you use for NAT) instead of through ExpressRoute. The return traffic goes back via the Internet.



Asymmetric routing detection

Traceroute is the best way to make sure that your network traffic is traversing the expected path. If you expect traffic from your on-premises SMTP server to Microsoft to take the Internet path, the expected traceroute is from the SMTP server to Office 365. The result validates that traffic is indeed leaving your network toward the Internet and not toward ExpressRoute.

Verifying ExpressRoute Connectivity

1/17/2017 • 13 min to read • Edit on GitHub

ExpressRoute, which extends an on-premises network into the Microsoft cloud over a dedicated private connection that is facilitated by a connectivity provider, involves the following three distinct network zones:

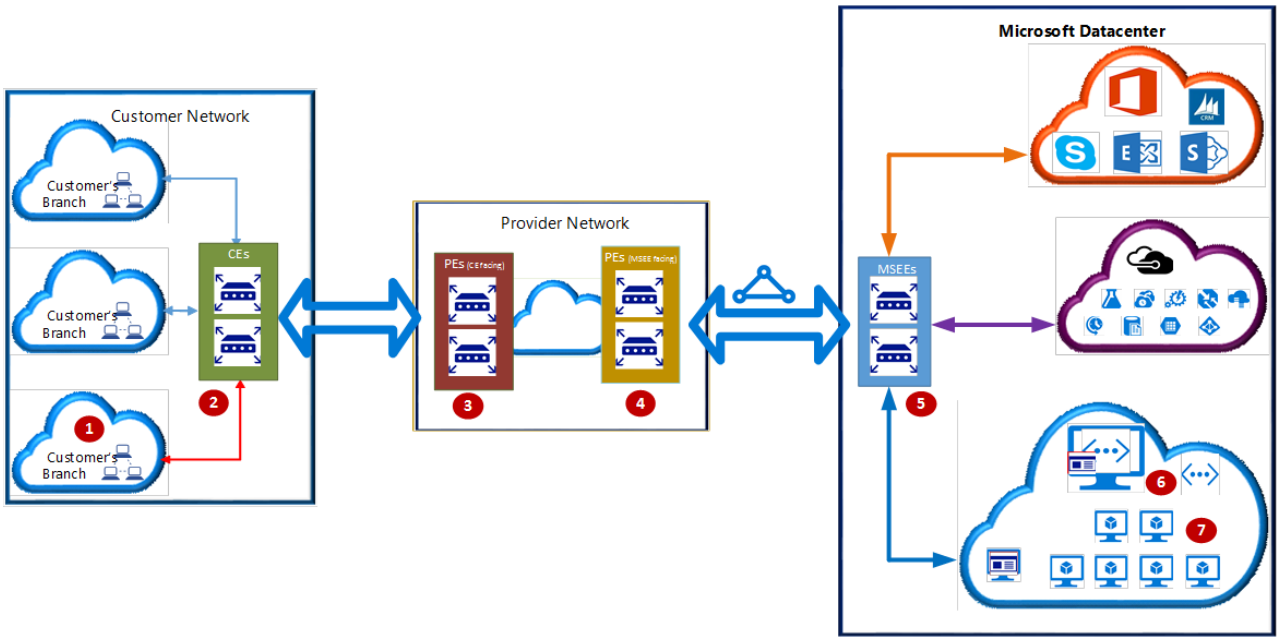
- Customer Network
- Provider Network
- Microsoft Datacenter

The purpose of this document is to help user to identify where (or even if) a connectivity issue exists and within which zone, thereby to seek help from appropriate team to resolve the issue. If Microsoft support is needed to resolve an issue, open a support ticket with Microsoft Support.

> **IMPORTANT**
> This document is intended to help diagnosing and fixing simple issues. It is not intended to be a replacement for Microsoft support. Open a support ticket with Microsoft Support if you are unable to solve the problem using the guidance provided.

Overview

The following diagram shows the logical connectivity of a customer network to Microsoft network using ExpressRoute.



In the preceding diagram, the numbers indicate key network points. The network points are referenced often through this article by their associated number.

Depending on the ExpressRoute connectivity model (Cloud Exchange Co-location, Point-to-Point Ethernet Connection, or Any-to-any (IPVPN)) the network points 3 and 4 may be switches (Layer 2 devices). The key network points illustrated are as follows:

1. Customer compute device (for example, a server or PC)
2. CEs: Customer edge routers
3. PEs (CE facing): Provider edge routers/switches that are facing customer edge routers

4. PEs (MSEE facing: Provider edge routers/switches that are facing MSEEs
5. MSEEs: Microsoft Enterprise Edge (MSEE) ExpressRoute routers
6. Virtual Network (VNet) Gateway
7. Compute device on the Azure VNet

If the Cloud Exchange Co-location or Point-to-Point Ethernet Connection connectivity models are used, the customer edge router (2) would establish BGP peering with MSEEs (5). Network points 3 and 4 would still exist but be somewhat transparent as Layer 2 devices.

If the Any-to-any (IPVPN) connectivity model is used, the PEs (MSEE facing) (4) would establish BGP peering with MSEEs (5). Routes would then propagate back to the customer network via the IPVPN service provider network.

> **NOTE**
> For ExpressRoute high availability, Microsoft requires a redundant pair of BGP sessions between MSEEs (5) and MSEE-PRs (4). A redundant pair of network paths is also encouraged between customer network and MSEE-PRs. However, in Any-to-any (IPVPN) connection model, a single CE device (2) may be connected to one or more PEs (3).

To validate an ExpressRoute circuit, the following steps are covered (with the network point indicated by the associated number):

1. Validate circuit provisioning and state (5)
2. Validate at least one ExpressRoute peering is configured (5)
3. Validate ARP between Microsoft and the service provider (link between 4 and 5)
4. Validate BGP and routes on the MSEE (BGP between 4 to 5, and 5 to 6 if a VNet is connected)
5. Check the Traffic Statistics (Traffic passing through 5)

More validations and checks will be added in the future, check back monthly!

Validate circuit provisioning and state

Regardless of the connectivity model, an ExpressRoute circuit has to be created and thus a service key generated for circuit provisioning. Provisioning an ExpressRoute circuit establishes a redundant Layer 2 connections between MSEE-PRs (4) and MSEEs (5). For more information on how to create, modify, provision, and verify an ExpressRoute circuit, see the article Create and modify an ExpressRoute circuit.

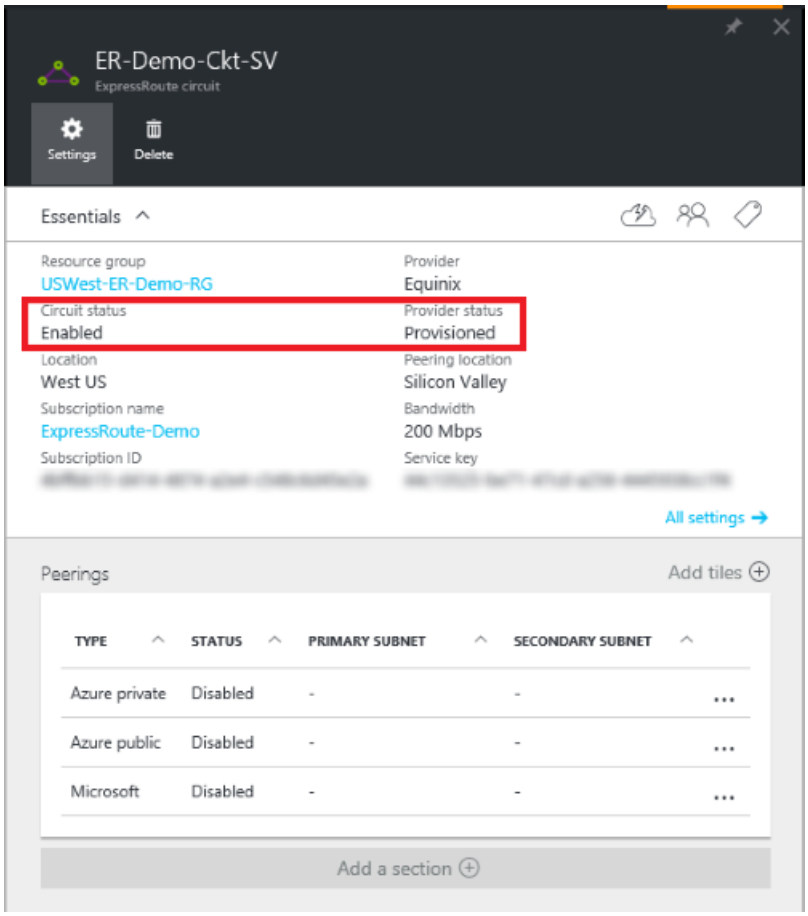
> **TIP**
> A service key uniquely identifies an ExpressRoute circuit. This key is required for most of the powershell commands mentioned in this document. Also, should you need assistance from Microsoft or from an ExpressRoute partner to troubleshoot an ExpressRoute issue, provide the service key to readily identify the circuit.

Verification via the Azure portal

In the Azure portal, the status of an ExpressRoute circuit can be checked by selecting 

on the left-side-bar menu and then selecting the ExpressRoute circuit. Selecting an ExpressRoute circuit listed under "All resources" opens the ExpressRoute circuit blade. In the

section of the blade, the ExpressRoute essentials are listed as shown in the following screen shot:



In the ExpressRoute Essentials, *Circuit status* indicates the status of the circuit on the Microsoft side. *Provider status* indicates if the circuit has been *Provisioned/Not provisioned* on the service-provider side.

For an ExpressRoute circuit to be operational, the *Circuit status* must be *Enabled* and the *Provider status* must be *Provisioned*.

> **NOTE**
> If the *Circuit status* is not enabled, contact Microsoft Support. If the *Provider status* is not provisioned, contact your service provider.

Verification via PowerShell

To list all the ExpressRoute circuits in a Resource Group, use the following command:

```
Get-AzureRmExpressRouteCircuit -ResourceGroupName "Test-ER-RG"
```

> **TIP**
> You can get your resource group name through the Azure portal. See the previous subsection of this document and note that the resource group name is listed in the example screen shot.

To select a particular ExpressRoute circuit in a Resource Group, use the following command:

```
Get-AzureRmExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
```

A sample response is:

```
Name                              : Test-ER-Ckt
ResourceGroupName                 : Test-ER-RG
Location                          : westus2
Id                                : /subscriptions/**************************/resourceGroups/Test-ER-
RG/providers/**********/expressRouteCircuits/Test-ER-Ckt
Etag                              : W/"################################"
ProvisioningState                 : Succeeded
Sku                               : {
                                      "Name": "Standard_UnlimitedData",
                                      "Tier": "Standard",
                                      "Family": "UnlimitedData"
                                    }
CircuitProvisioningState          : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes              :
ServiceProviderProperties         : {
                                      "ServiceProviderName": "****",
                                      "PeeringLocation": "******",
                                      "BandwidthInMbps": 100
                                    }
ServiceKey                        : ************************************
Peerings                          : []
Authorizations                    : []
```

To confirm if an ExpressRoute circuit is operational, pay particular attention to the following fields:

```
CircuitProvisioningState          : Enabled
ServiceProviderProvisioningState : Provisioned
```

> **NOTE**
>
> If the *CircuitProvisioningState* is not enabled, contact Microsoft Support. If the *ServiceProviderProvisioningState* is not provisioned, contact your service provider.

Verification via PowerShell (Classic)

To list all the ExpressRoute circuits under a subscription, use the following command:

```
Get-AzureDedicatedCircuit
```

To select a particular ExpressRoute circuit, use the following command:

```
Get-AzureDedicatedCircuit -ServiceKey ************************************
```

A sample response is:

```
andwidth                          : 100
BillingType                       : UnlimitedData
CircuitName                       : Test-ER-Ckt
Location                          : westus2
ServiceKey                        : ************************************
ServiceProviderName               : ****
ServiceProviderProvisioningState : Provisioned
Sku                               : Standard
Status                            : Enabled
```

To confirm if an ExpressRoute circuit is operational, pay particular attention to the following fields:
ServiceProviderProvisioningState : Provisioned Status : Enabled

> **NOTE**
>
> If the *Status* is not enabled, contact Microsoft Support. If the *ServiceProviderProvisioningState* is not provisioned, contact your service provider.

Validate Peering Configuration

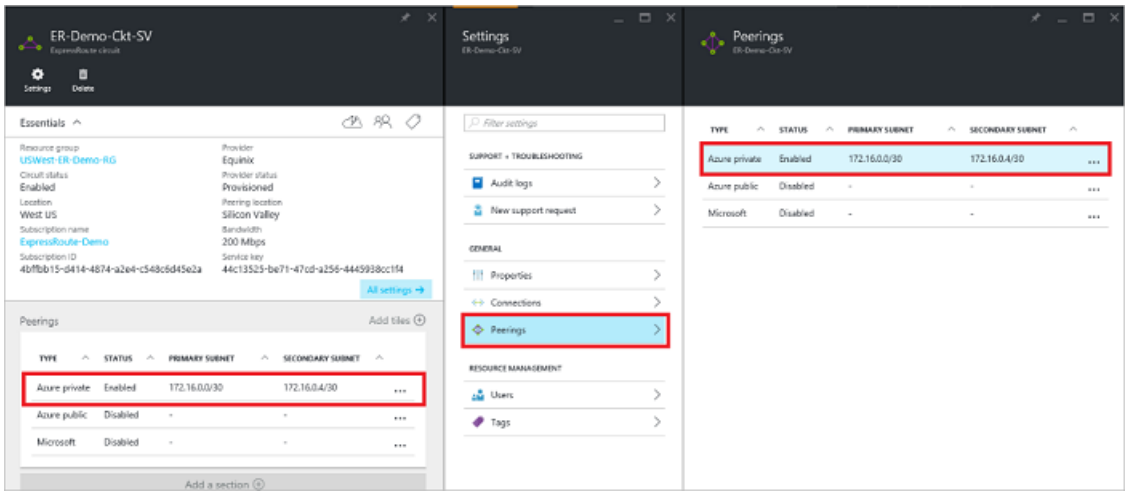
After the service provider has completed the provisioning the ExpressRoute circuit, a routing configuration can be created over the ExpressRoute circuit between MSEE-PRs (4) and MSEEs (5). Each ExpressRoute circuit can have one, two, or three routing contexts enabled: Azure private peering (traffic to private virtual networks in Azure), Azure public peering (traffic to public IP addresses in Azure), and Microsoft peering (traffic to Office 365 and CRM Online). For more information on how to create and modify routing configuration, see the article Create and modify routing for an ExpressRoute circuit.

Verification via the Azure portal

> **IMPORTANT**
>
> There is a known bug in the Azure portal wherein ExpressRoute peerings are *NOT* shown in the portal if configured by the service provider. Adding ExpressRoute peerings via the portal or PowerShell *overwrites the service provider settings*. This action breaks the routing on the ExpressRoute circuit and requires the support of the service provider to restore the settings and reestablish normal routing. Only modify the ExpressRoute peerings if it is certain that the service provider is providing layer 2 services only!

> **NOTE**
>
> If layer 3 is provided by the service provider and the peerings are blank in the portal, PowerShell can be used to see the service provider configured settings.

In the Azure portal, status of an ExpressRoute circuit can be checked by selecting All resources on the left-side-bar menu and then selecting the ExpressRoute circuit. Selecting an ExpressRoute circuit listed under "All resources" would open the ExpressRoute circuit blade. In the Overview section of the blade, the ExpressRoute essentials would be listed as shown in the following screen shot:



In the preceding example, as noted Azure private peering routing context is enabled, whereas Azure public and Microsoft peering routing contexts are not enabled. A successfully enabled peering context would also have the primary and secondary point-to-point (required for BGP) subnets listed. The /30 subnets are used for the interface IP address of the MSEEs and MSEE-PRs.

> **NOTE**
>
> If a peering is not enabled, check if the primary and secondary subnets assigned match the configuration on MSEE-PRs. If not, to change the configuration on MSEE routers, refer to Create and modify routing for an ExpressRoute circuit

Verification via PowerShell

To get the Azure private peering configuration details, use the following commands:

```
$ckt = Get-AzureRmExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -Circuit $ckt
```

A sample response, for a successfully configured private peering, is:

```
Name                       : AzurePrivatePeering
Id                         : /subscriptions/**************************/resourceGroups/Test-ER-
RG/providers/***********/expressRouteCircuits/Test-ER-Ckt/peerings/AzurePrivatePeering
Etag                       : W/"################################"
PeeringType                : AzurePrivatePeering
AzureASN                   : 12076
PeerASN                    : ####
PrimaryPeerAddressPrefix   : 172.16.0.0/30
SecondaryPeerAddressPrefix : 172.16.0.4/30
PrimaryAzurePort           :
SecondaryAzurePort         :
SharedKey                  :
VlanId                     : 300
MicrosoftPeeringConfig     : null
ProvisioningState          : Succeeded
```

A successfully enabled peering context would have the primary and secondary address prefixes listed. The /30 subnets are used for the interface IP address of the MSEEs and MSEE-PRs.

To get the Azure public peering configuration details, use the following commands:

```
$ckt = Get-AzureRmExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -Circuit $ckt
```

To get the Microsoft peering configuration details, use the following commands:

```
$ckt = Get-AzureRmExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzureRmExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -Circuit $ckt
```

If a peering is not configured, there would be an error message. A sample response, when the stated peering (Azure Public peering in this example) is not configured within the circuit:

```
Get-AzureRmExpressRouteCircuitPeeringConfig : Sequence contains no matching element
At line:1 char:1
    + Get-AzureRmExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering ...
    + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        + CategoryInfo          : CloseError: (:) [Get-AzureRmExpr...itPeeringConfig], InvalidOperationException
        + FullyQualifiedErrorId :
Microsoft.Azure.Commands.Network.GetAzureExpressRouteCircuitPeeringConfigCommand
```

> **IMPORTANT**
>
> If layer 3 peerings were set by the service provider, setting the ExpressRoute peerings via the portal or PowerShell overwrites the service provider settings. Resetting the provider side peering settings requires the support of the service provider. Only modify the ExpressRoute peerings if it is certain that the service provider is providing layer 2 services only!

> **NOTE**
>
> If a peering is not enabled, check if the primary and secondary subnets assigned match the configuration on the linked MSEE-PR. Also check if the correct *VlandId*, *AzureASN*, and *PeerASN* are used on MSEEs and if these values maps to the ones used on the linked MSEE-PR. If MD5 hashing is chosen, the shared key should be same on MSEE and MSEE-PR pair. To change the configuration on the MSEE routers, refer to Create and modify routing for an ExpressRoute circuit.

Verification via PowerShell (Classic)

To get the Azure private peering configuration details, use the following command:

```
Get-AzureBGPPeering -AccessType Private -ServiceKey "*********************************"
```

A sample response, for a successfully configured private peering is:

```
AdvertisedPublicPrefixes       :
AdvertisedPublicPrefixesState  : Configured
AzureAsn                       : 12076
CustomerAutonomousSystemNumber :
PeerAsn                        : ####
PrimaryAzurePort               :
PrimaryPeerSubnet              : 10.0.0.0/30
RoutingRegistryName            :
SecondaryAzurePort             :
SecondaryPeerSubnet            : 10.0.0.4/30
State                          : Enabled
VlanId                         : 100
```

A successfully, enabled peering context would have the primary and secondary peer subnets listed. The /30 subnets are used for the interface IP address of the MSEEs and MSEE-PRs.

To get the Azure public peering configuration details, use the following commands:

```
Get-AzureBGPPeering -AccessType Public -ServiceKey "*********************************"
```

To get the Microsoft peering configuration details, use the following commands:

```
Get-AzureBGPPeering -AccessType Microsoft -ServiceKey "*********************************"
```

> **IMPORTANT**
>
> If layer 3 peerings were set by the service provider, setting the ExpressRoute peerings via the portal or PowerShell overwrites the service provider settings. Resetting the provider side peering settings requires the support of the service provider. Only modify the ExpressRoute peerings if it is certain that the service provider is providing layer 2 services only!

NOTE

If a peering is not enabled, check if the primary and secondary peer subnets assigned match the configuration on the linked MSEE-PR. Also check if the correct *VlanId*, *AzureAsn*, and *PeerAsn* are used on MSEEs and if these values maps to the ones used on the linked MSEE-PR. To change the configuration on the MSEE routers, refer to Create and modify routing for an ExpressRoute circuit.

Validate ARP between Microsoft and the service provider

This section uses PowerShell (Classic) commands. If you have been using PowerShell Azure Resource Manager commands, ensure that you have admin/co-admin access to the subscription via Azure classic portal

NOTE

To get ARP, both the Azure portal and Azure Resource Manager PowerShell commands can be used. If errors are encountered with the Azure Resource Manager PowerShell commands, classic PowerShell commands should work as Classic PowerShell commands also work with Azure Resource Manager ExpressRoute circuits.

To get the ARP table from the primary MSEE router for the private peering, use the following command:

```
Get-AzureDedicatedCircuitPeeringArpInfo -AccessType Private -Path Primary -ServiceKey
"********************************"
```

An example response for the command, in the successful scenario:

```
ARP Info:

            Age           Interface           IpAddress          MacAddress
            113             On-Prem       10.0.0.1            e8ed.f335.4ca9
              0           Microsoft       10.0.0.2           7c0e.ce85.4fc9
```

Similarly, you can check the ARP table from the MSEE in the *Primary/Secondary* path, for *Private/Public/Microsoft* peerings.

The following example shows the response of the command for a peering does not exist.

```
ARP Info:
```

NOTE

If the ARP table does not have IP addresses of the interfaces mapped to MAC addresses, review the following information:

1. If the first IP address of the /30 subnet assigned for the link between the MSEE-PR and MSEE is used on the interface of MSEE-PR. Azure always uses the second IP address for MSEEs.
2. Verify if the customer (C-Tag) and service (S-Tag) VLAN tags match both on MSEE-PR and MSEE pair.

Validate BGP and routes on the MSEE

This section uses PowerShell (Classic) commands. If you have been using PowerShell Azure Resource Manager commands, ensure that you have admin/co-admin access to the subscription via Azure classic portal

NOTE

To get BGP information, both the Azure portal and Azure Resource Manager PowerShell commands can be used. If errors are encountered with the Azure Resource Manager PowerShell commands, classic PowerShell commands should work as classic PowerShell commands also work with Azure Resource Manager ExpressRoute circuits.

To get the routing table (BGP neighbor) summary for a particular routing context, use the following command:

```
Get-AzureDedicatedCircuitPeeringRouteTableSummary -AccessType Private -Path Primary -ServiceKey
"**********************************"
```

An example response is:

```
Route Table Summary:

        Neighbor                V               AS            UpDown         StatePfxRcd
        10.0.0.1                4             ####              8w4d                  50
```

As shown in the preceding example, the command is useful to determine for how long the routing context has been established. It also indicates number of route prefixes advertised by the peering router.

NOTE

If the state is in Active or Idle, check if the primary and secondary peer subnets assigned match the configuration on the linked MSEE-PR. Also check if the correct *VlanId*, *AzureAsn*, and *PeerAsn* are used on MSEEs and if these values maps to the ones used on the linked MSEE-PR. If MD5 hashing is chosen, the shared key should be same on MSEE and MSEE-PR pair. To change the configuration on the MSEE routers, refer to Create and modify routing for an ExpressRoute circuit.

NOTE

If certain destinations are not reachable over a particular peering, check the route table of the MSEEs belonging to the particular peering context. If a matching prefix (could be NATed IP) is present in the routing table, then check if there are firewalls/NSG/ACLs on the path and if they permit the traffic.

To get the full routing table from MSEE on the *Primary* path for the particular *Private* routing context, use the following command:

```
Get-AzureDedicatedCircuitPeeringRouteTableInfo -AccessType Private -Path Primary -ServiceKey
"**********************************"
```

An example successful outcome for the command is:

```
Route Table Info:

         Network             NextHop             LocPrf             Weight               Path
     10.1.0.0/16           10.0.0.1                                    0    #### ##### #####
     10.2.0.0/16           10.0.0.1                                    0    #### ##### #####
...
```

Similarly, you can check the routing table from the MSEE in the *Primary/Secondary* path, for *Private/Public/Microsoft* a peering context.

The following example shows the response of the command for a peering does not exist:

```
Route Table Info:
```

Check the Traffic Statistics

To get the combined primary and secondary path traffic statistics--bytes in and out--of a peering context, use the following command:

```
Get-AzureDedicatedCircuitStats -ServiceKey 97f85950-01dd-4d30-a73c-bf683b3a6e5c -AccessType Private
```

A sample output of the command is:

```
PrimaryBytesIn PrimaryBytesOut SecondaryBytesIn SecondaryBytesOut
-------------- --------------- ---------------- ------------------
     240780020       239863857        240565035         239628474
```

A sample output of the command for a non-existent peering is:

```
Get-AzureDedicatedCircuitStats : ResourceNotFound: Can not find any subinterface for peering type 'Public' for
circuit '97f85950-01dd-4d30-a73c-bf683b3a6e5c' .
At line:1 char:1
+ Get-AzureDedicatedCircuitStats -ServiceKey 97f85950-01dd-4d30-a73c-bf ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : CloseError: (:) [Get-AzureDedicatedCircuitStats], CloudException
    + FullyQualifiedErrorId :
Microsoft.WindowsAzure.Commands.ExpressRoute.GetAzureDedicatedCircuitPeeringStatsCommand
```

Next Steps

For more information or help, check out the following links:

- Microsoft Support
- Create and modify an ExpressRoute circuit
- Create and modify routing for an ExpressRoute circuit

ExpressRoute Troubleshooting guide - Getting ARP tables in the Resource Manager deployment model

1/17/2017 • 5 min to read • Edit on GitHub

This article walks you through the steps to learn the ARP tables for your ExpressRoute circuit.

> **IMPORTANT**
> This document is intended to help you diagnose and fix simple issues. It is not intended to be a replacement for Microsoft support. You must open a support ticket with Microsoft support if you are unable to solve the problem using the guidance described below.

Address Resolution Protocol (ARP) and ARP tables

Address Resolution Protocol (ARP) is a layer 2 protocol defined in RFC 826. ARP is used to map the Ethernet address (MAC address) with an ip address.

The ARP table provides a mapping of the ipv4 address and MAC address for a particular peering. The ARP table for an ExpressRoute circuit peering provides the following information for each interface (primary and secondary)

1. Mapping of on-premises router interface ip address to the MAC address
2. Mapping of ExpressRoute router interface ip address to the MAC address
3. Age of the mapping

ARP tables can help validate layer 2 configuration and troubleshooting basic layer 2 connectivity issues.

Example ARP table:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           10.0.0.1 ffff.eeee.dddd
  0 Microsoft         10.0.0.2 aaaa.bbbb.cccc
```

The following section provides information on how you can view the ARP tables seen by the ExpressRoute edge routers.

Prerequisites for learning ARP tables

Ensure that you have the following before you progress further

- A Valid ExpressRoute circuit configured with at least one peering. The circuit must be fully configured by the connectivity provider. You (or your connectivity provider) must have configured at least one of the peerings (Azure private, Azure public and Microsoft) on this circuit.
- IP address ranges used for configuring the peerings (Azure private, Azure public and Microsoft). Review the ip address assignment examples in the ExpressRoute routing requirements page to get an understanding of how ip addresses are mapped to interfaces on your side and on the ExpressRoute side. You can get information on the peering configuration by reviewing the ExpressRoute peering configuration page.
- Information from your networking team / connectivity provider on the MAC addresses of interfaces used with these IP addresses.
- You must have the latest PowerShell module for Azure (version 1.50 or newer).

Getting the ARP tables for your ExpressRoute circuit

This section provides instructions on how you can view the ARP tables per peering using PowerShell. You or your connectivity provider must have configured the peering before progressing further. Each circuit has two paths (primary and secondary). You can check the ARP table for each path independently.

ARP tables for Azure private peering

The following cmdlet provides the ARP tables for Azure private peering

```
    # Required Variables
    $RG = "<Your Resource Group Name Here>"
    $Name = "<Your ExpressRoute Circuit Name Here>"

    # ARP table for Azure private peering - Primary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePrivatePeering -DevicePath Primary

    # ARP table for Azure private peering - Secodary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePrivatePeering -DevicePath Secondary
```

Sample output is shown below for one of the paths

```
    Age InterfaceProperty IpAddress  MacAddress
    --- ----------------- ---------  ----------
     10 On-Prem           10.0.0.1 ffff.eeee.dddd
      0 Microsoft         10.0.0.2 aaaa.bbbb.cccc
```

ARP tables for Azure public peering

The following cmdlet provides the ARP tables for Azure public peering

```
    # Required Variables
    $RG = "<Your Resource Group Name Here>"
    $Name = "<Your ExpressRoute Circuit Name Here>"

    # ARP table for Azure public peering - Primary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePublicPeering -DevicePath Primary

    # ARP table for Azure public peering - Secodary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePublicPeering -DevicePath Secondary
```

Sample output is shown below for one of the paths

```
    Age InterfaceProperty IpAddress  MacAddress
    --- ----------------- ---------  ----------
     10 On-Prem           64.0.0.1 ffff.eeee.dddd
      0 Microsoft         64.0.0.2 aaaa.bbbb.cccc
```

ARP tables for Microsoft peering

The following cmdlet provides the ARP tables for Microsoft peering

```
    # Required Variables
    $RG = "<Your Resource Group Name Here>"
    $Name = "<Your ExpressRoute Circuit Name Here>"

    # ARP table for Microsoft peering - Primary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
MicrosoftPeering -DevicePath Primary

    # ARP table for Microsoft peering - Secodary path
    Get-AzureRmExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
MicrosoftPeering -DevicePath Secondary
```

Sample output is shown below for one of the paths

```
    Age InterfaceProperty IpAddress  MacAddress
    --- ----------------- ---------  ----------
     10 On-Prem           65.0.0.1 ffff.eeee.dddd
      0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

How to use this information

The ARP table of a peering can be used to determine validate layer 2 configuration and connectivity. This section provides an overview of how ARP tables will look under different scenarios.

ARP table when a circuit is in operational state (expected state)

- The ARP table will have an entry for the on-premises side with a valid IP address and MAC address and a similar entry for the Microsoft side.
- The last octet of the on-premises ip address will always be an odd number.
- The last octet of the Microsoft ip address will always be an even number.
- The same MAC address will appear on the Microsoft side for all 3 peerings (primary / secondary).

```
 Age InterfaceProperty IpAddress  MacAddress
 --- ----------------- ---------  ----------
  10 On-Prem           65.0.0.1 ffff.eeee.dddd
   0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

ARP table when on-premises / connectivity provider side has problems

- Only one entry will appear in the ARP table. This will show the mapping between the MAC address and IP address used in the Microsoft side.

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
  0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

> **NOTE**
> Open a support request with your connectivity provider to debug such issues.

ARP table when Microsoft side has problems

- You will not see an ARP table shown for a peering if there are issues on the Microsoft side.
- Open a support ticket with Microsoft support. Specify that you have an issue with layer 2 connectivity.

Next Steps

- Validate Layer 3 configurations for your ExpressRoute circuit
  - Get route summary to determine the state of BGP sessions
  - Get route table to determine which prefixes are advertised across ExpressRoute
- Validate data transfer by reviewing bytes in / out
- Open a support ticket with Microsoft support if you are still experiencing issues.

ExpressRoute troubleshooting guide: Getting ARP tables in the classic deployment model

1/17/2017 • 4 min to read • Edit on GitHub

This article walks you through the steps for getting the Address Resolution Protocol (ARP) tables for your Azure ExpressRoute circuit.

> **IMPORTANT**
> This document is intended to help you diagnose and fix simple issues. It is not intended to be a replacement for Microsoft support. If you can't solve the problem by using the following guidance, open a support request with Microsoft Azure Help+support.

Address Resolution Protocol (ARP) and ARP tables

ARP is a Layer 2 protocol that's defined in RFC 826. ARP is used to map an Ethernet address (MAC address) to an IP address.

An ARP table provides a mapping of the IPv4 address and MAC address for a particular peering. The ARP table for an ExpressRoute circuit peering provides the following information for each interface (primary and secondary):

1. Mapping of an on-premises router interface IP address to a MAC address
2. Mapping of an ExpressRoute router interface IP address to a MAC address
3. The age of the mapping

ARP tables can help with validating Layer 2 configuration and with troubleshooting basic Layer 2 connectivity issues.

Following is an example of an ARP table:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           10.0.0.1 ffff.eeee.dddd
  0 Microsoft         10.0.0.2 aaaa.bbbb.cccc
```

The following section provides information about how to view the ARP tables that are seen by the ExpressRoute edge routers.

Prerequisites for using ARP tables

Ensure that you have the following before you continue:

- A valid ExpressRoute circuit that's configured with at least one peering. The circuit must be fully configured by the connectivity provider. You (or your connectivity provider) must configure at least one of the peerings (Azure private, Azure public, or Microsoft) on this circuit.
- IP address ranges that are used for configuring the peerings (Azure private, Azure public, and Microsoft). Review the IP address assignment examples in the ExpressRoute routing requirements page to get an understanding of how IP addresses are mapped to interfaces on your aise and on the ExpressRoute side. You can get information about the peering configuration by reviewing the ExpressRoute peering configuration page.
- Information from your networking team or connectivity provider about the MAC addresses of the interfaces

that are used with these IP addresses.

- The latest Windows PowerShell module for Azure (version 1.50 or later).

ARP tables for your ExpressRoute circuit

This section provides instructions about how to view the ARP tables for each type of peering by using PowerShell. Before you continue, either you or your connectivity provider needs to configure the peering. Each circuit has two paths (primary and secondary). You can check the ARP table for each path independently.

ARP tables for Azure private peering

The following cmdlet provides the ARP tables for Azure private peering:

```
# Required variables
$ckt = "<your Service Key here>

# ARP table for Azure private peering--primary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Private -Path Primary

# ARP table for Azure private peering--secondary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Private -Path Secondary
```

Following is sample output for one of the paths:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           10.0.0.1 ffff.eeee.dddd
  0 Microsoft         10.0.0.2 aaaa.bbbb.cccc
```

ARP tables for Azure public peering:

The following cmdlet provides the ARP tables for Azure public peering:

```
# Required variables
$ckt = "<your Service Key here>

# ARP table for Azure public peering--primary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Public -Path Primary

# ARP table for Azure public peering--secondary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Public -Path Secondary
```

Following is sample output for one of the paths:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           10.0.0.1 ffff.eeee.dddd
  0 Microsoft         10.0.0.2 aaaa.bbbb.cccc
```

Following is sample output for one of the paths:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           64.0.0.1 ffff.eeee.dddd
  0 Microsoft         64.0.0.2 aaaa.bbbb.cccc
```

ARP tables for Microsoft peering

The following cmdlet provides the ARP tables for Microsoft peering:

```
# ARP table for Microsoft peering--primary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Microsoft -Path Primary

# ARP table for Microsoft peering--secondary path
Get-AzureDedicatedCircuitPeeringArpInfo -ServiceKey $ckt -AccessType Microsoft -Path Secondary
```

Sample output is shown below for one of the paths:

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           65.0.0.1 ffff.eeee.dddd
  0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

How to use this information

The ARP table of a peering can be used to validate Layer 2 configuration and connectivity. This section provides an overview of how ARP tables look in different scenarios.

ARP table when a circuit is in an operational (expected) state

- The ARP table has an entry for the on-premises side with a valid IP and MAC address, and a similar entry for the Microsoft side.
- The last octet of the on-premises IP address is always an odd number.
- The last octet of the Microsoft IP address is always an even number.
- The same MAC address appears on the Microsoft side for all three peerings (primary/secondary).

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
 10 On-Prem           65.0.0.1 ffff.eeee.dddd
  0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

ARP table when it's on-premises or when the connectivity-provider side has problems

Only one entry appears in the ARP table. It shows the mapping between the MAC address and the IP address that's used on the Microsoft side.

```
Age InterfaceProperty IpAddress  MacAddress
--- ----------------- ---------  ----------
  0 Microsoft         65.0.0.2 aaaa.bbbb.cccc
```

> **NOTE**
> If you experience an issue like this, open a support request with your connectivity provider to resolve it.

ARP table when the Microsoft side has problems

- You will not see an ARP table shown for a peering if there are issues on the Microsoft side.
- Open a support request with Microsoft Azure Help+support. Specify that you have an issue with Layer 2 connectivity.

Next steps

- Validate Layer 3 configurations for your ExpressRoute circuit:
  - Get a route summary to determine the state of BGP sessions.
  - Get a route table to determine which prefixes are advertised across ExpressRoute.
- Validate data transfer by reviewing bytes in and out.

- Open a support request with Microsoft Azure Help+support if you are still experiencing issues.

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

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

[2]This 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]

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

[2]This 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.

[3]These 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 combnation 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

[1]Virtual 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.

[2]Input 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]

[1]With 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

[1]Contact 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

[1]This 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.

[3]Azure 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]

[1]Refer 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

[1]Each 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]

[1]Apps and storage quotas are per App Service plan unless noted otherwise.
[2]The 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.
[3]Dedicated instances can be of different sizes. See App Service Pricing for more details.
[4]Premium 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.
[5]The 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
[6]These resources are constrained by physical resources on the dedicated instances (the instance size and the number of instances).
[7]If you scale an app in the Basic tier to two instances, you have 350 concurrent connections for each of the two instances.
[8]Premium tier allows backup intervals down up to every 5 minutes when using App Service Environments, and 50 times per day otherwise.
[9]Run 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.
[10]SLA 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.	Maximum message size: 256KB (Standard tier) / 1MB (Premium tier). **Note** Due to system overhead, this limit is usually slightly less. Maximum header size: 64KB Maximum number of header properties in property bag: **byte/int.MaxValue** Maximum size of property in property bag: No explicit limit. Limited by maximum header size.
Message property size for a queue/topic/subscription entity	System-wide	Static	A **SerializationException** exception is generated.	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).
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	A single user can only be associated with a maximum of 20 Azure Active Directory directories. Examples of possible combinations: • 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	• 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	• String type extensions can have maximum of 256 characters. • Binary type extensions are limited to 256 bytes. • 100 extension values (across ALL types and ALL applications) can be written to any single object. • Only "User", "Group", "TenantDetail", "Device", "Application" and "ServicePrincipal" entities can be extended with "String" type or "Binary" type single-valued attributes. • Schema extensions are available only in Graph API-version 1.21-preview. The application must be granted write access to register an extension.
Applications	A maximum of 10 users can be owners of a single application.
Groups	• 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	• 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	• 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	• 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	• 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

[1]When 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

- [1]The 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]

[1]API 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

[1]This 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