

Contents

[ExpressRoute Documentation](#)

[Overview](#)

[What is ExpressRoute?](#)

[Tutorials](#)

[Create and modify a circuit](#)

[Create and modify peering configuration](#)

[Link a virtual network to an ExpressRoute circuit](#)

[Configure route filters for Microsoft peering](#)

[Concepts](#)

[Connectivity models](#)

[Circuits and peering](#)

[Locations and partners](#)

[Providers by location](#)

[Locations by provider](#)

[About virtual network gateways](#)

[Prerequisites](#)

[Workflows](#)

[Routing requirements](#)

[BFD over ExpressRoute](#)

[QoS requirements](#)

[About moving circuits from classic to Resource Manager](#)

[About ExpressRoute FastPath](#)

[About ExpressRoute Direct](#)

[About ExpressRoute Global Reach](#)

[About ExpressRoute encryption](#)

[Connect Azure to public cloud](#)

[Backend Connectivity Interoperability](#)

[Preface and Test Setup](#)

[Test Setup Configuration](#)

[Control Plane Analysis](#)

[Data Plane Analysis](#)

[Cross-network connectivity](#)

[Built-in security controls](#)

[Designing for High Availability](#)

[Designing for Disaster Recovery](#)

[Private Peering](#)

[Using VPN as a backup](#)

[How-to guides](#)

[Create and modify a circuit](#)

[Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Azure Resource Manager template](#)

[Create and modify peering configuration](#)

[Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Link a virtual network to an ExpressRoute circuit](#)

[Azure portal](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Encrypt traffic over circuits](#)

[Configure a site-to-site VPN over Microsoft peering](#)

[Configure IPsec transport mode for Windows hosts](#)

[Configure ExpressRoute Global Reach](#)

[Azure PowerShell](#)

[Azure CLI](#)

[Configure a virtual network gateway for ExpressRoute](#)

[Azure portal](#)

[Azure PowerShell](#)

[Configure ExpressRoute and site-to-site coexisting connections](#)

Configure ExpressRoute Direct

Azure PowerShell

Azure CLI

Configure MACsec for ExpressRoute Direct ports

Configure route filters for Microsoft peering

Azure portal

Azure PowerShell

Azure CLI

Enable and disable peerings

Move from public peering to Microsoft peering

Move a circuit from classic to Resource Manager

Migrate associated virtual networks from classic to Resource Manager

Create a zone-redundant VNet gateway in Azure Availability Zones

Configure a router for ExpressRoute

Configure a router

Router configuration samples for NAT

Monitoring

ExpressRoute monitoring, metrics, and alerts

Configure Network Performance Monitor for ExpressRoute

Classic and legacy articles

Modify a circuit

Create and modify peering configuration

Link a virtual network to an ExpressRoute circuit

Configure ExpressRoute and S2S coexisting connections

Add a gateway to a VNet

Create and modify public peering

Best Practices

Best practices for network security and cloud services

Optimize routing

Asymmetric routing

NAT for ExpressRoute

Troubleshoot

- [Verifying ExpressRoute connectivity](#)
- [Resolving network performance issues](#)
- [Reset a failed circuit](#)
- [Getting ARP tables](#)
- [Getting ARP tables \(Classic\)](#)
- ## Reference

 - [Azure PowerShell](#)
 - [Azure CLI](#)
 - [REST](#)
 - [REST \(classic\)](#)
 - [Resource Manager template](#)
- ## Resources

 - [ExpressRoute FAQ](#)
 - [Azure Roadmap](#)
 - [Case Studies](#)
 - [Networking Blog](#)
 - [Pricing](#)
 - [Pricing calculator](#)
 - [Service updates](#)
 - [SLA](#)
 - [Subscription and Service Limits](#)
 - [ExpressRoute for Cloud Solution Providers \(CSP\)](#)
 - [CrossConnections API development for partners](#)
- ## Videos

 - [Connect a virtual network gateway to a circuit](#)
 - [Create a virtual network for ExpressRoute](#)
 - [Create a virtual network gateway for ExpressRoute](#)
 - [Create an ExpressRoute circuit](#)
 - [Evolve your network infrastructure for connectivity](#)
 - [How to set up private peering for your circuit](#)
 - [Hybrid partnerships: Enabling on-premises scenarios](#)
 - [Set up Microsoft peering for your circuit](#)

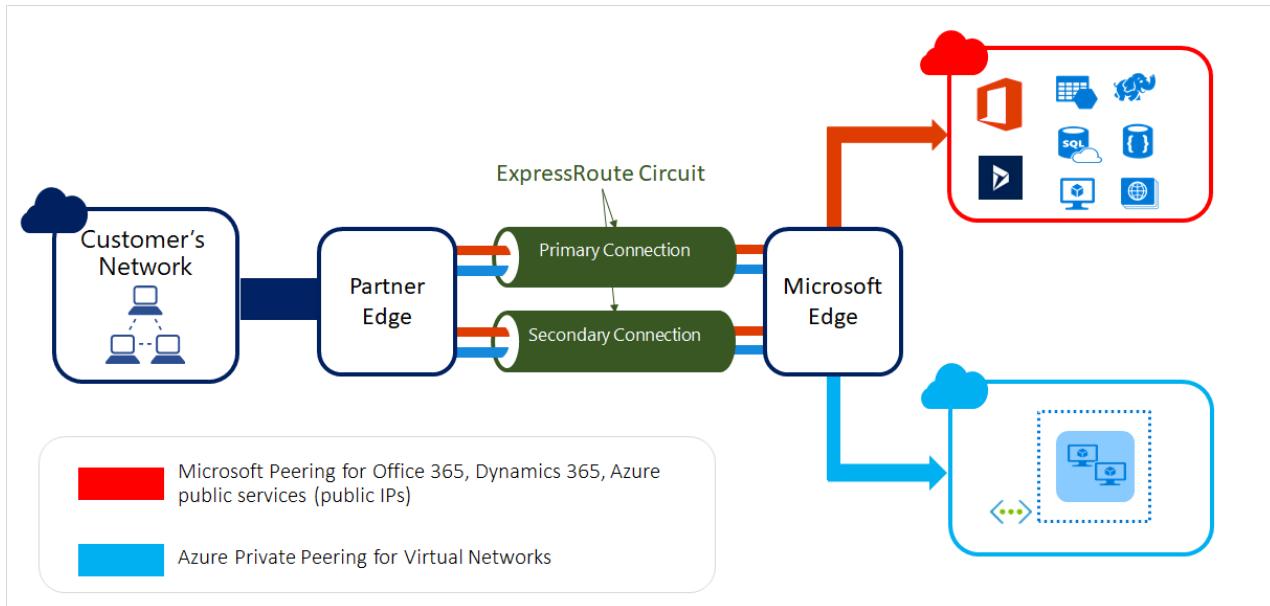
[Set up public peering for your circuit](#)

ExpressRoute overview

1/8/2020 • 5 minutes to read • [Edit Online](#)

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure and Office 365.

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, consistent latencies, and higher security than typical connections over the Internet. For information on how to connect your network to Microsoft using ExpressRoute, see [ExpressRoute connectivity models](#).



Key benefits

- 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 the ExpressRoute premium add-on.
- Dynamic routing between your network and Microsoft via BGP.
- Built-in redundancy in every peering location for higher reliability.
- Connection uptime [SLA](#).
- QoS support for Skype for Business.

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

Features

Layer 3 connectivity

Microsoft uses BGP, an industry standard dynamic routing protocol, 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) at an [ExpressRoute Location](#) from the connectivity provider/your network edge. Microsoft requires dual BGP connection from the connectivity provider/your network edge – 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 connections enable access to the following services:

- Microsoft Azure services
- Microsoft Office 365 services

NOTE

Office 365 was created to be accessed securely and reliably via the Internet. Because of this, we recommend ExpressRoute for specific scenarios. For information about using ExpressRoute to access Office 365, visit [Azure ExpressRoute for Office 365](#).

For a detailed list of services supported over ExpressRoute, visit the [ExpressRoute FAQ](#) page.

Connectivity to all regions within a geopolitical region

You can connect to Microsoft in one of our [peering locations](#) and access regions within the geopolitical region.

For example, if you connect to Microsoft in Amsterdam through ExpressRoute, you'll have access to all Microsoft cloud services hosted in Northern and Western Europe. For an overview of the geopolitical regions, the associated Microsoft cloud regions, and corresponding ExpressRoute peering locations, see the [ExpressRoute partners and peering locations](#) article.

Global connectivity with ExpressRoute Premium

You can enable [ExpressRoute Premium](#) to extend connectivity across geopolitical boundaries. For example, if you connect 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 North and West Europe regions.

Local connectivity with ExpressRoute Local

You can transfer data cost-effectively by enabling the [Local SKU](#) if you can bring your data to an ExpressRoute location near your desired Azure region. With Local, Data transfer is included in the ExpressRoute port charge.

Across on-premises connectivity with ExpressRoute Global Reach

You can enable ExpressRoute Global Reach to exchange data across your on-premises sites by connecting your ExpressRoute circuits. For example, if you have a private data center in California connected to ExpressRoute in Silicon Valley, and another private data center in Texas connected to ExpressRoute in Dallas, with ExpressRoute Global Reach, you can connect your private data centers together through two ExpressRoute circuits. Your cross-data-center traffic will traverse through Microsoft's network.

For more information, see [ExpressRoute Global Reach](#).

Rich connectivity partner ecosystem

ExpressRoute has a constantly growing ecosystem of connectivity providers and systems integrator partners. For the latest information, refer to [ExpressRoute partners and peering locations](#).

Connectivity to national clouds

Microsoft operates isolated cloud environments for special geopolitical regions and customer segments. Refer to the [ExpressRoute partners and peering locations](#) page for a list of national clouds and providers.

ExpressRoute Direct

ExpressRoute Direct provides customers the opportunity to connect directly into Microsoft's global network at peering locations strategically distributed across the world. ExpressRoute Direct provides dual 100Gbps connectivity, which supports Active/Active connectivity at scale.

Key features that ExpressRoute Direct provides include, but are not limited to:

- Massive Data Ingestion into services like Storage and Cosmos DB
- Physical isolation for industries that are regulated and require dedicated and isolated connectivity, such as: Banking, Government, and Retail
- Granular control of circuit distribution based on business unit

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

Bandwidth options

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

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

Dynamic scaling of bandwidth

You can increase the ExpressRoute circuit bandwidth (on a best effort basis) without having to tear down your connections. For more information see [Modifying an ExpressRoute circuit](#).

Flexible billing models

You can pick a billing model that works best for you. Choose between the billing models listed below. For more information, see [ExpressRoute FAQ](#).

- **Unlimited data.** Billing is based on a monthly fee; all inbound and outbound data transfer is included free of charge.
- **Metered data.** Billing is 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.** ExpressRoute premium is an add-on to 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.

FAQ

For frequently asked questions about ExpressRoute, see [ExpressRoute FAQ](#).

Next steps

- Learn about [ExpressRoute connectivity models](#).
- 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 and modify an ExpressRoute circuit](#)
 - [Create and modify peering for an ExpressRoute circuit](#)
 - [Connect a virtual network to an ExpressRoute circuit](#)
- Learn about some of the other Azure key [networking capabilities](#).

Tutorial: Create and modify an ExpressRoute circuit

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

This article helps you create an ExpressRoute circuit using the Azure portal and the Azure Resource Manager deployment model. You can also check the status, update, delete, or deprovision a circuit.

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.

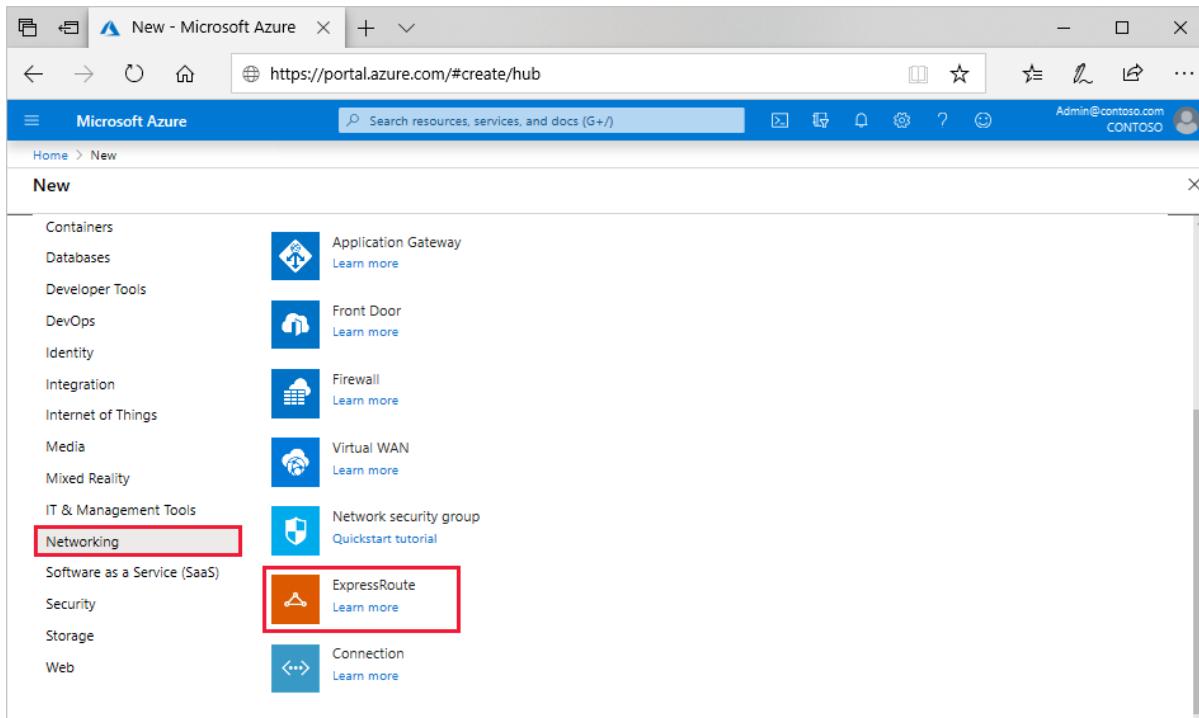
2. Create a new ExpressRoute circuit

IMPORTANT

Your ExpressRoute circuit is 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.

You can create an ExpressRoute circuit by selecting the option to create a new resource.

1. On the Azure portal menu or from the **Home** page, select **Create a resource**. Select **Networking > ExpressRoute**, as shown in the following image:



2. After you click **ExpressRoute**, you'll see the **Create ExpressRoute circuit** page. When you're filling in the values on this page, make sure that you specify the correct SKU tier (Standard, or Premium) and data metering billing model (Unlimited or Metered).

Create ExpressRoute circuit

Create new or import from classic i

Create new Import

* Circuit name
TestCkt ✓

* Provider i
Equinix

* Peering location i
Seattle

* Bandwidth i
100Mbps

* SKU i
 Standard Premium

* Billing model i
 Unlimited Metered

Allow classic operations i

* Subscription
Windows Azure Internal Consumption

* Resource group
 Create new Use existing
DemoRG ✓

* Location
West US

- **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**.

IMPORTANT

You can't change the type from **Unlimited** to **Metered**.

- **Peering Location** is the physical location where you are peering with Microsoft.

IMPORTANT

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.

A screenshot of the Azure portal's 'All resources' blade. At the top, there are buttons for 'Add', 'Columns', and 'Refresh'. Below that is a search bar with 'Filter items...' and a dropdown set to 'ExpressRoute-Demo'. The main table has columns for NAME, RESOURCE GROUP, LOCATION, and SUBSCRIPTION. One row is visible, showing 'ER-Demo-Okt-SV' under 'NAME', 'USWest-ER-Demo-RG' under 'RESOURCE GROUP', 'West US' under 'LOCATION', and 'ExpressRoute-Demo' under 'SUBSCRIPTION'.

View the properties

You can view the properties of the circuit by selecting it. On the **Overview** page for your circuit, the service key appears in the service key field. You must copy the service key for your circuit and pass it down to the service provider to complete the provisioning process. The circuit service key is specific to your circuit.

A screenshot of the Azure portal showing the 'TestCkt' ExpressRoute circuit details. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Connections, Authorizations, Peerings, and Properties. The main pane shows the 'Overview' tab selected. It includes a note to 'Initiate the provisioning process with your service provider.' Below is a table with the following data:

Resource group (change)	Provider
DemoRG	Equinix
Circuit status	Provider status
Enabled	Not provisioned
Location	Peering location
West US	Seattle
Subscription name (change)	Bandwidth
Windows Azure Internal Consumption	100 Mbps
Subscription ID	Service key

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

On this page, **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 is in the following state:

Provider status: Not provisioned

Circuit status: Enabled

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

The circuit changes 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 configures and manages 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 and viewing the Overview page.

Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity. You can modify the bandwidth, SKU, billing model and allow classic operations on the **Configuration** page. For information on limits and limitations, see the [ExpressRoute FAQ](#).

You can perform the following tasks with no downtime:

- Enable or disable an ExpressRoute Premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit, provided there is capacity available on the port.

IMPORTANT

Downgrading the bandwidth of a circuit is not supported.

- Change the metering plan from *Metered Data* to *Unlimited Data*.

IMPORTANT

Changing the metering plan from Unlimited Data to Metered Data is not supported.

- You can enable and disable *Allow Classic Operations*.

IMPORTANT

You may have to recreate the ExpressRoute circuit if there is inadequate capacity on the existing port. You cannot upgrade the circuit if there is no additional capacity available at that location.

Although you can seamlessly upgrade the bandwidth, 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.

Disabling the Premium add-on operation can fail if you're using resources that are greater than what is permitted for the standard circuit.

To modify an ExpressRoute circuit, click **Configuration**.

The screenshot shows the Azure portal interface for managing an ExpressRoute circuit. The left sidebar has a search bar and a list of navigation items:

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- SETTINGS** (selected)
- Configuration** (highlighted with a red box)
- Connections
- Authorizations
- Peerings
- Properties
- Locks
- Automation script
- SUPPORT + TROUBLESHOOTING
- New support request

The main content area displays configuration settings:

- Bandwidth**: 100 Mbps
- SKU**: Standard (highlighted with a red box) to Premium
- Billing model**: Unlimited (highlighted with a red box) to Metered
- Allow classic operations**: Disabled (highlighted with a red box) to Enabled

Deprovisioning and deleting an ExpressRoute circuit

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

- 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 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 delete the circuit. This stops billing for the circuit.

Next steps

After you create your circuit, continue with the following next steps:

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Create and modify peering for an ExpressRoute circuit

12/17/2019 • 7 minutes to read • [Edit Online](#)

This article helps you create and manage routing configuration for an Azure Resource Manager (ARM) ExpressRoute circuit, using the Azure portal. You can also check the status, update, or delete and deprovision peerings for an ExpressRoute circuit. If you want to use a different method to work with your circuit, select an article from the following list:

You can configure private peering and Microsoft peering for an ExpressRoute circuit (Azure public peering is deprecated for new circuits). Peerings can be configured in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time. For more information about routing domains and peerings, see [ExpressRoute routing domains](#). For information about public peering, see [ExpressRoute public peering](#).

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. In order to configure peering(s), the ExpressRoute circuit must be in a provisioned and enabled state.
- If you plan to use a shared key/MD5 hash, be sure to use this on both sides of the tunnel and limit the number of alphanumeric characters to a maximum of 25. Special characters are not supported.

These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider that offers managed Layer 3 services (typically an IPVPN, like MPLS), your connectivity provider configures and manages 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. Check with your service provider before configuring BGP peerings.

Microsoft peering

This section helps you create, get, update, and delete the Microsoft peering configuration for an ExpressRoute circuit.

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through the Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit. For more information, see [Configure a route filter for Microsoft peering](#).

To create Microsoft peering

1. Configure the ExpressRoute circuit. Check the **Provider status** to ensure that the circuit is fully provisioned by the connectivity provider before continuing further.

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

Circuit - Provider status: Not provisioned

Setting	Value
Resource group (change)	: USWest-ER-Demo-RG
Circuit status	: Enabled
Location	: West US 2
Subscription (change)	: ExpressRoute-Lab
Subscription ID	: 4bffb15-d414-4874-a2e4
Tags (change)	: Click here to add tags

Peering					
Type	Status	Primary Subnet	Secondary Subnet	Last Modified By	
Azure private	Not provisioned	-	-	-	...
Azure public	Not provisioned	-	-	-	...
Microsoft	Not provisioned	-	-	-	...

Circuit - Provider status: Provisioned

Setting	Value
Resource group (change)	: USWest-ER-Demo-RG
Circuit status	: Enabled
Location	: West US 2
Subscription (change)	: ExpressRoute-Lab
Subscription ID	: 4bffb15-d414-4874-a2e4
Tags (change)	: Click here to add tags

Peering					
Type	Status	Primary Subnet	Secondary Subnet	Last Modified By	
Azure private	Not provisioned	-	-	-	...
Azure public	Not provisioned	-	-	-	...
Microsoft	Not provisioned	-	-	-	...

- 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. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID. For both Primary and Secondary links you must use 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. If you plan to send a set of prefixes, you can send a comma-separated list. These prefixes must be registered to you in an RIR / IRR.
- Optional** - 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.
- Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
- Optional** - An MD5 hash if you choose to use one.

- You can select the peering you wish to configure, as shown in the following example. Select the Microsoft peering row.

Home > ER-Demo-Ckt

ER-Demo-Ckt
ExpressRoute circuit

Search (Ctrl+)

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

Settings

- Configuration
- Connections
- Authorizations
- Peering
- Properties

Move Delete Refresh

Resource group (change) : USWest-ER-Demo-RG
Circuit status : Enabled
Location : West US 2
Subscription (changed) : ExpressRoute-Lab
Subscription ID : 4bfbfb15-d414-4874
Tags (change) : Click here to add tags

Provider : Equinix
Provider status : Provisioned
Peering location : Seattle
Bandwidth : 200 Mbps
Service key : 74b7c74d-4b76-4a23

Peerings

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	-
Azure public	Not provisioned	-	-	-
Microsoft	Not provisioned	-	-	-

4. Configure Microsoft peering. **Save** the configuration once you have specified all parameters. The following image shows an example configuration:

Home > ER-Demo-Ckt > Microsoft peering

Microsoft peering

ER-Demo-Ckt

Save **Discard** **Delete**

To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. Learn More.

* Peer ASN [?](#)
394749

* Primary subnet [?](#)
64.191.192.240/30

* Secondary subnet [?](#)
64.191.192.244/30

* VLAN ID [?](#)
152

* Advertised public prefixes [?](#)
64.191.192.224/28 Status: Not configured

Customer ASN [?](#)
394749

Routing registry name [?](#)
ARIN

Shared key

IMPORTANT

Microsoft verifies if the specified 'Advertised public prefixes' and 'Peer ASN' (or 'Customer ASN') are assigned to you in the Internet Routing Registry. If you are getting the public prefixes from another entity and if the assignment is not recorded with the routing registry, the automatic validation will not complete and will require manual validation. If the automatic validation fails, you will see the message 'Validation needed'.

If you see the message 'Validation needed', collect the document(s) that show the public prefixes are assigned to your organization by the entity that is listed as the owner of the prefixes in the routing registry and submit these documents for manual validation by opening a support ticket as shown below.

If your circuit gets to a 'Validation needed' state, 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 in the following

example:

The screenshot shows the 'Help + support - New support request' page. On the left, there's a sidebar with 'Overview', 'Support', 'New support request' (which is selected), 'All support requests', 'Support Plans', 'Service Health', 'Advisor', and 'Get started with Azure'. The main area has tabs for 'Basics', 'Solutions', 'Details', and 'Review + create'. Under the 'Basics' tab, there's a description: 'Create a new support request to get assistance with billing, subscription, technical or quota management issues. Complete the Basics tab by selecting the options that best describe your problem. Providing detailed, accurate information can help to solve your issues faster.' Below this are several input fields with validation stars: 'Issue type' (set to 'Technical'), 'Subscription' (set to 'ExpressRoute-Lab (4bfffbb15-d414-4874-a2e4-c548c6d45e...)'), 'Service' (radio button selected for 'My services'), 'Resource' (set to 'ExpressRoute'), 'Problem type' (set to 'Configuration and Setup'), 'Problem subtype' (set to 'ExpressRoute Peerings'), and 'Subject' (containing 'Proof of ownership for public prefixes.').

5. After the configuration has been accepted successfully, you'll see something similar to the following image:

The screenshot shows the 'Microsoft peering' configuration page for 'ER-Demo-Ckt'. It includes a 'Save', 'Discard', and 'Delete' button. A note says: 'To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. Learn More.' Below are configuration fields with validation stars: 'Peer ASN' (set to '394749'), 'Primary subnet' (set to '64.191.192.240/30'), 'Secondary subnet' (set to '64.191.192.244/30'), 'VLAN ID' (set to '152'), 'Advertised public prefixes' (set to '64.191.192.224/28'), 'Customer ASN' (set to '394749'), 'Routing registry name' (set to 'ARIN'), and 'Shared key' (empty). A red box highlights the 'Status: Configured' status message next to the advertised public prefixes field.

To view Microsoft peering details

You can view the properties of Microsoft peering by selecting the row for the peering.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt

ER-Demo-Ckt

ExpressRoute circuit

Search (Ctrl+F)

Move Delete Refresh

Resource group : USWest-ER-Demo-RG

Circuit status : Enabled

Location : West US 2

Subscription : ExpressRoute-Lab

Subscription ID : 4bfbb15-d414-4874

Tags : Click here to add tags

Provider : Equinix

Provider status : Provisioned

Peering location : Seattle

Bandwidth : 200 Mbps

Service key : 74b7c74d-4b76-4a23

Peerings

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY	...
Azure private	Not provisioned	-	-	-	...
Azure public	Not provisioned	-	-	-	...
Microsoft	Provisioned	64.191.192.240/30	64.191.192.244/30	Customer	...

To update Microsoft peering configuration

You can select the row for the peering that you want to modify, then modify the peering properties and save your modifications.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Microsoft peering

Microsoft peering

ER-Demo-Ckt

Save Discard Delete

To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. [Learn More](#)

* Peer ASN [?](#)
394749

* Primary subnet [?](#)
64.191.192.240/30

* Secondary subnet [?](#)
64.191.192.244/30

* VLAN ID [?](#)
152

* Advertised public prefixes [?](#)
64.191.192.224/28
Status: Configured

Customer ASN [?](#)
394749

Routing registry name [?](#)
APNIC

- None
- ARIN
- APNIC
- AFRINIC
- LACNIC
- RIPENCC
- RADB
- ALTDB

To delete Microsoft peering

You can remove your peering configuration by clicking the delete icon, as shown in the following image:

Microsoft peering

ER-Demo-Ckt

 Save Discard Delete

i To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. [Learn More.](#)

* Peer ASN [i](#)

394749

* Primary subnet [i](#)

64.191.192.240/30

* Secondary subnet [i](#)

64.191.192.244/30

* VLAN ID [i](#)

152

* Advertised public prefixes [i](#)

64.191.192.224/28

Status: Configured

Customer ASN [i](#)

394749

Routing registry name [i](#)

ARIN

▼

Shared key

[Get ARP records](#)[Get route table](#)[Get route table summary](#)

Azure private peering

This section helps you create, get, update, and delete the Azure private peering configuration for an ExpressRoute circuit.

To create Azure private peering

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

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

Circuit - Provider status: Not provisioned

The screenshot shows the Azure portal interface for an ExpressRoute circuit named 'ER-Demo-Ckt'. In the main pane, under the 'Provider' section, the status 'Provider status : Not provisioned' is highlighted with a red box. Other details shown include Resource group (USWest-ER-Demo-RG), Circuit status (Enabled), Location (West US 2), Subscription (ExpressRoute-Lab), Subscription ID (4bfbb15-d414-4874), and Tags (Click here to add tags). Below this, the 'Peerings' table lists three entries: Azure private (Not provisioned), Azure public (Not provisioned), and Microsoft (Not provisioned).

Circuit - Provider status: Provisioned

This screenshot is identical to the one above, but the 'Provider status : Not provisioned' entry is now replaced by 'Provider status : Provisioned', which is also highlighted with a red box. All other circuit details remain the same.

- 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. The subnet must not be part of any address space reserved for virtual networks. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A /30 subnet for the secondary link. The subnet must not be part of any address space reserved for virtual networks. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A valid VLAN ID to establish this peering. Ensure that no other peering in the circuit uses the same VLAN ID. For both Primary and Secondary links you must use 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 except for the number from 65515 to 65520, inclusively.
- You must advertise the routes from your on-premises Edge router to Azure via BGP when you set up the private peering.
- Optional** - An MD5 hash if you choose to use one.

- Select the Azure private peering row, as shown in the following example:

This screenshot shows the same Azure portal interface as the previous ones, but the 'Azure private' row in the 'Peerings' table is now highlighted with a red box. The rest of the circuit configuration and provider status are identical to the previous examples.

- Configure private peering. **Save** the configuration once you have specified all parameters.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

Save **Discard** **Delete**

* Peer ASN [?](#)
394749 ✓

* Primary subnet [?](#)
172.16.0.0/30 ✓

* Secondary subnet [?](#)
172.16.0.4/30 ✓

* VLAN ID [?](#)
154 ✓

Shared key

5. After the configuration has been accepted successfully, you see something similar to the following example:

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

Save **Discard** **Delete**

* Peer ASN [?](#)
394749

* Primary subnet [?](#)
172.16.0.0/30

* Secondary subnet [?](#)
172.16.0.4/30

* VLAN ID [?](#)
154

Shared key

[Get ARP records](#)
[Get route table](#)
[Get route table summary](#)

To view Azure private peering details

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

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY	
Azure private	Provisioned	172.16.0.0/30	172.16.0.4/30	Customer	...
Azure public	Not provisioned	-	-	-	...
Microsoft	Provisioned	64.191.192.240/30	64.191.192.244/30	Customer	...

To update Azure private peering configuration

You can select the row for peering and modify the peering properties. After updating, save your changes.

Private peering
ER-Demo-Ckt

Save Discard Delete

* Peer ASN

* Primary subnet

* Secondary subnet

* VLAN ID

Shared key

[Get ARP records](#)

[Get route table](#)

[Get route table summary](#)

To delete Azure private peering

You can remove your peering configuration by selecting the delete icon, as shown in the following image:

WARNING

You must ensure that all virtual networks and ExpressRoute Global Reach connections are removed before running this example.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

* Peer ASN [?](#)

394749

* Primary subnet [?](#)

172.16.0.0/30

* Secondary subnet [?](#)

172.16.0.4/30

* VLAN ID [?](#)

154 

Shared key

[Get ARP records](#)

[Get route table](#)

[Get route table summary](#)

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](#).

Connect a virtual network to an ExpressRoute circuit using the portal

11/13/2019 • 4 minutes to read • [Edit Online](#)

This article helps you create a connection to link a virtual network to an Azure ExpressRoute circuit using the Azure portal. The virtual networks that you connect to your Azure ExpressRoute circuit can either be in the same subscription, or they can be part of another subscription.

Before you begin

- Review the [prerequisites](#), [routing requirements](#), and [workflows](#) before you begin configuration.
- You must have an active ExpressRoute circuit.
 - Follow the instructions to [create an ExpressRoute circuit](#) and have the circuit enabled by your connectivity provider.
 - Ensure that you have Azure private peering configured for your circuit. See the [Create and modify peering for an ExpressRoute circuit](#) article for peering and routing instructions.
 - Ensure that Azure private peering 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 virtual network gateway for ExpressRoute](#). A virtual network gateway for ExpressRoute uses the GatewayType 'ExpressRoute', not VPN.
- You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit.
- A single VNet can be linked to up to four ExpressRoute circuits. Use the process below to create a new connection object for each ExpressRoute circuit you are connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.
- You can link a virtual network 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.
- You can [view a video](#) before beginning to better understand the steps.

Connect a VNet to a circuit - same subscription

NOTE

BGP configuration information will not 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.

To create a connection

1. Ensure that your ExpressRoute circuit and Azure private peering have been configured successfully. Follow the instructions in [Create an ExpressRoute circuit](#) and [Create and modify peering for an ExpressRoute circuit](#). Your ExpressRoute circuit should look like the following image:

The figure shows three windows side-by-side:

- Left Window (Settings):** Shows basic circuit information like provider (Equinix), status (Enabled), location (West US), and bandwidth (200 Mbps). It also lists peerings.
- Middle Window (Peerings):** A table of peering configurations. One row is highlighted with a red box: "Azure private" (Type), "Enabled" (Status), "172.16.0.0/30" (Primary Subnet), and "172.16.0.4/30" (Secondary Subnet).
- Right Window (Connections):** A list of connections. The "Connections" section is highlighted with a red box. An "Add" button is also highlighted with a red box.

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** page, and then configure the values.

The figure shows three windows illustrating the connection creation process:

- Left Window (Settings):** Shows the "Connections" section highlighted with a red box. An "Add" button is highlighted with a red box.
- Middle Window (Connections):** An empty list of connections with the heading "No results".
- Right Window (Add connection):** A configuration form for a new connection. Fields include:
 - Name:** ER-VNet-Connection
 - Connection type:** ExpressRoute
 - Virtual network gateway:** Demo-VNet-GW
 - ExpressRoute circuit:** ER-Demo-Ckt-SV
 - Subscription:** ExpressRoute-Demo
 - Resource group:** USWest-ER-Demo-RG
 - Location:** West US

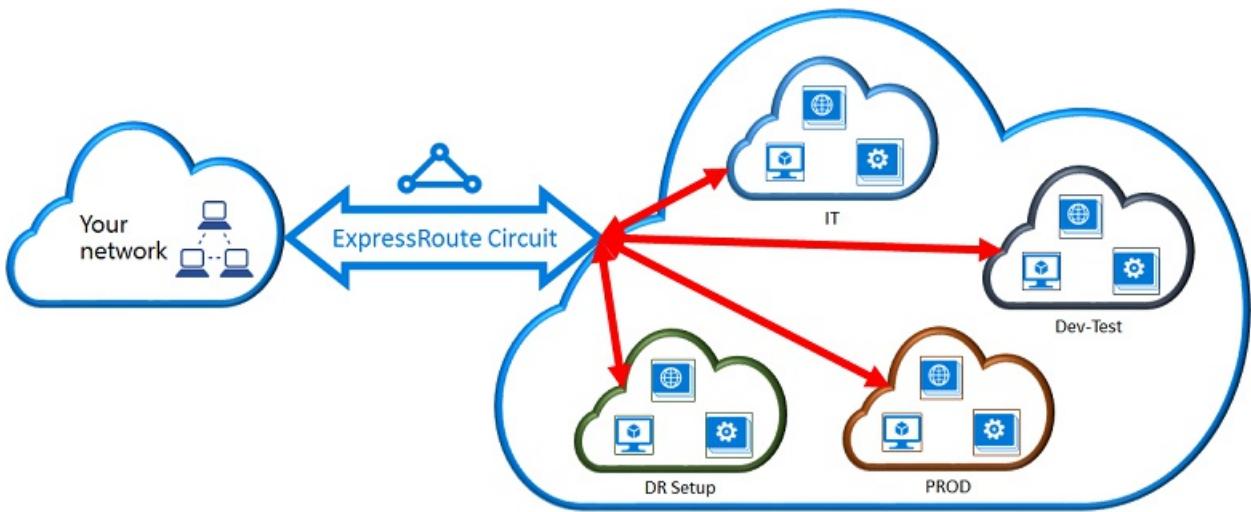
3. After your connection has been successfully configured, your connection object will show the information for the connection.

The figure shows two windows related to the connection configuration:

- Left Window (Connections):** Shows the "Connections" list with one entry: "ER-VNet-Connection" (Status: Succeeded, Connection Type: ExpressRoute, Peer: Demo-VNet-GW). The "Add" button is highlighted with a red box.
- Right Window (ER-VNet-Connection):** Details for the "ER-VNet-Connection" object. Key fields shown in the "Essentials" section include:
 - Resource group: USWest-ER-Demo-RG
 - Status: Succeeded
 - Location: West US
 - Subscription name: ExpressRoute-Demo
 - Circuit: ER-Demo-Ckt-SV
 - Virtual network: Demo-VNet
 - Virtual network gateway: Demo-VNet-GW (13.88.21.182)

Connect a VNet to a circuit - different subscription

You can share an ExpressRoute circuit across multiple subscriptions. The figure below 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 and authorizations associated to the circuit, including subscriptions linked to other Azure Active Directory tenants and Enterprise Agreement enrollments.

NOTE

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

Administration - About circuit owners and circuit users

The 'circuit owner' is an authorized Power User of the ExpressRoute circuit resource. The circuit owner can create authorizations that can be redeemed by 'circuit users'. Circuit users are owners of virtual network gateways that 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

To create a connection 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.

NOTE

Each connection requires a separate authorization.

1. In the ExpressRoute page, Click **Authorizations** and then type a **name** for the authorization and click **Save**.

The screenshot shows the 'Authorizations' blade for a specific circuit. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, and Connections. The 'Authorizations' section is highlighted with a red box. At the top right, there are 'Save', 'Discard', and 'Refresh' buttons. Below them is a note about redeeming authorizations. The main area shows a table with one row, where the 'NAME' column contains 'EDemo' and the 'PROVISIONING STATUS' column shows 'Succeeded'. The 'AUTHORIZATION KEY' column contains a long string of characters.

- Once the configuration is saved, copy the **Resource ID** and the **Authorization Key**.

This screenshot shows the same 'Authorizations' blade after saving. The 'Resource ID' and 'Authorization Key' columns are highlighted with red boxes to indicate the values that should be copied.

To delete a connection authorization

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

Circuit user operations

The circuit user needs the resource ID and an authorization key from the circuit owner.

To redeem a connection authorization

- Click the **+New** button.

The screenshot shows the Microsoft Azure homepage. The 'New' button, which is used to create new resources, is highlighted with a red box. Other visible options include 'All resources', 'Resource groups', 'App Services', 'SQL databases', 'SQL data warehouses', 'NoSQL (DocumentDB)', 'Virtual machines', and 'Load balancers'.

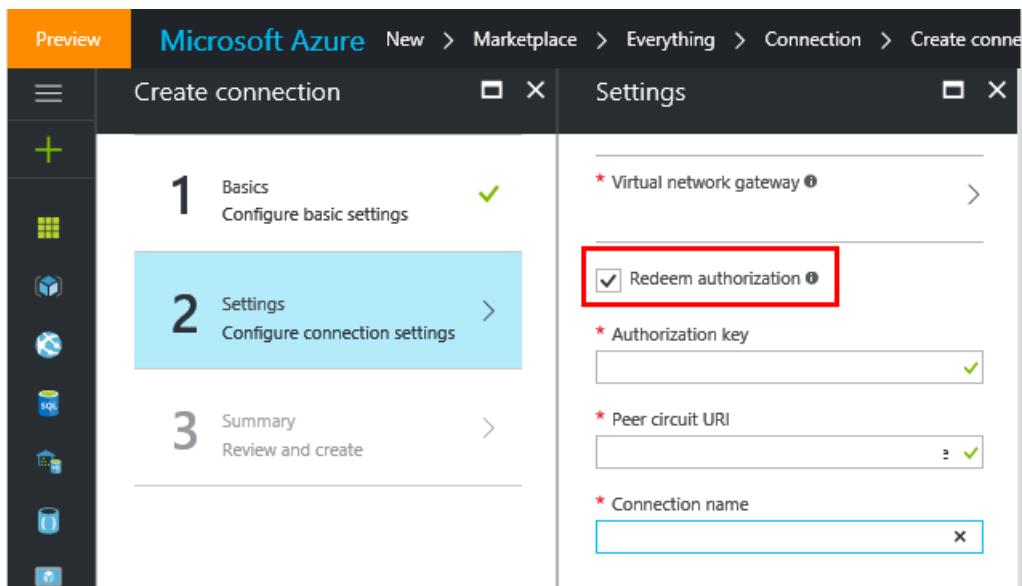
- Search for "**Connection**" in the Marketplace, select it, and click **Create**.

3. Make sure the **Connection type** is set to "ExpressRoute".

4. Fill in the details, then click **OK** in the Basics page.

5. In the **Settings** page, Select the **Virtual network gateway** and check the **Redeem authorization** check box.

6. Enter the **Authorization key** and the **Peer circuit URI** and give the connection a name. Click **OK**. The **Peer Circuit URI** is the Resource ID of the ExpressRoute circuit (which you can find under the Properties Setting pane of the ExpressRoute Circuit).



7. Review the information in the **Summary** page and click **OK**.

To release a connection authorization

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

Delete a connection to unlink a VNet

You can delete a connection and unlink your VNet to an ExpressRoute circuit by selecting the **Delete** icon on the page for your connection.

Next steps

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

Configure route filters for Microsoft peering: Azure portal

11/13/2019 • 5 minutes to read • [Edit Online](#)

Route filters are a way to consume a subset of supported services through Microsoft peering. The steps in this article help you configure and manage route filters for ExpressRoute circuits.

Office 365 services such as Exchange Online, SharePoint Online, and Skype for Business, and Azure services such as storage and SQL DB are accessible through Microsoft peering. When Microsoft peering is configured in an ExpressRoute circuit, all prefixes related to these services are advertised through the BGP sessions that are established. A BGP community value is attached to every prefix to identify the service that is offered through the prefix. For a list of the BGP community values and the services they map to, see [BGP communities](#).

If you require connectivity to all services, a large number of prefixes are advertised through BGP. This significantly increases the size of the route tables maintained by routers within your network. If you plan to consume only a subset of services offered through Microsoft peering, you can reduce the size of your route tables in two ways. You can:

- Filter out unwanted prefixes by applying route filters on BGP communities. This is a standard networking practice and is used commonly within many networks.
- Define route filters and apply them to your ExpressRoute circuit. A route filter is a new resource that lets you select the list of services you plan to consume through Microsoft peering. ExpressRoute routers only send the list of prefixes that belong to the services identified in the route filter.

About route filters

When Microsoft peering is configured on your ExpressRoute circuit, the Microsoft edge routers establish a pair of BGP sessions with the edge routers (yours or your connectivity provider's). No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter.

A route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. It is essentially a list of all the BGP community values you want to allow. Once a route filter resource is defined and attached to an ExpressRoute circuit, all prefixes that map to the BGP community values are advertised to your network.

To be able to attach route filters with Office 365 services on them, you must have authorization to consume Office 365 services through ExpressRoute. If you are not authorized to consume Office 365 services through ExpressRoute, the operation to attach route filters fails. For more information about the authorization process, see [Azure ExpressRoute for Office 365](#).

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit.

Workflow

To be able to successfully connect to services through Microsoft peering, you must complete the following configuration steps:

- You must have an active ExpressRoute circuit that has Microsoft peering provisioned. You can use the

following instructions to accomplish these tasks:

- [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.
- [Create Microsoft peering](#) if you manage the BGP session directly. Or, have your connectivity provider provision Microsoft peering for your circuit.
- You must create and configure a route filter.
 - Identify the services you wish to consume through Microsoft peering
 - Identify the list of BGP community values associated with the services
 - Create a rule to allow the prefix list matching the BGP community values
- You must attach the route filter to the ExpressRoute circuit.

Before you begin

Before you begin configuration, make sure you meet the following criteria:

- Review the [prerequisites](#) 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 before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state.
- You must have an active Microsoft peering. Follow instructions at [Create and modifying peering configuration](#)

Step 1: Get a list of prefixes and BGP community values

1. Get a list of BGP community values

BGP community values associated with services accessible through Microsoft peering is available in the [ExpressRoute routing requirements](#) page.

2. Make a list of the values that you want to use

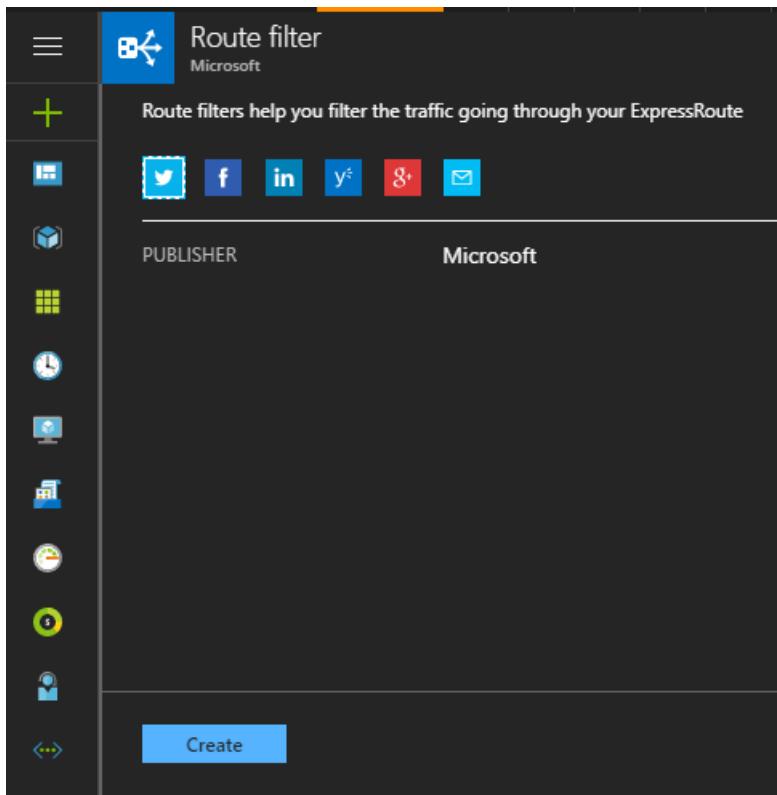
Make a list of [BGP community values](#) you want to use in the route filter.

Step 2: Create a route filter and a filter rule

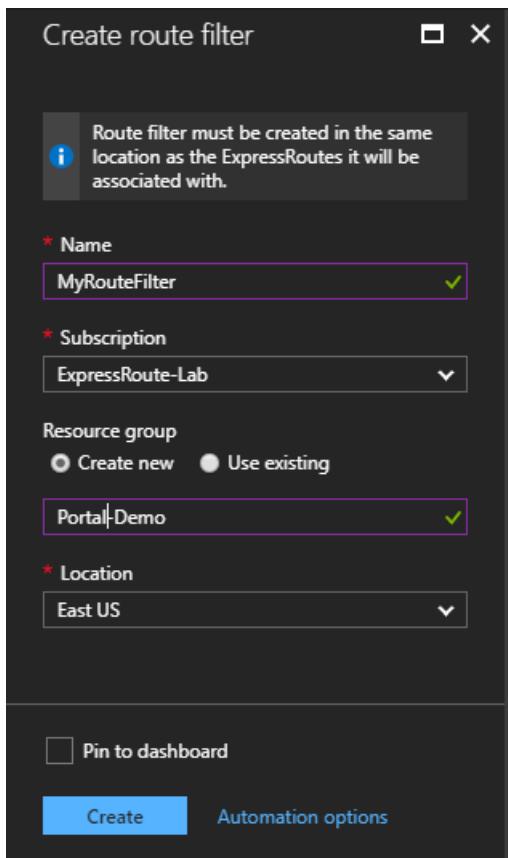
A route filter can have only one rule, and the rule must be of type 'Allow'. This rule can have a list of BGP community values associated with it.

1. Create a route filter

You can create a route filter by selecting the option to create a new resource. Click **Create a resource** > **Networking** > **RouteFilter**, as shown in the following image:



You must place the route filter in a resource group.



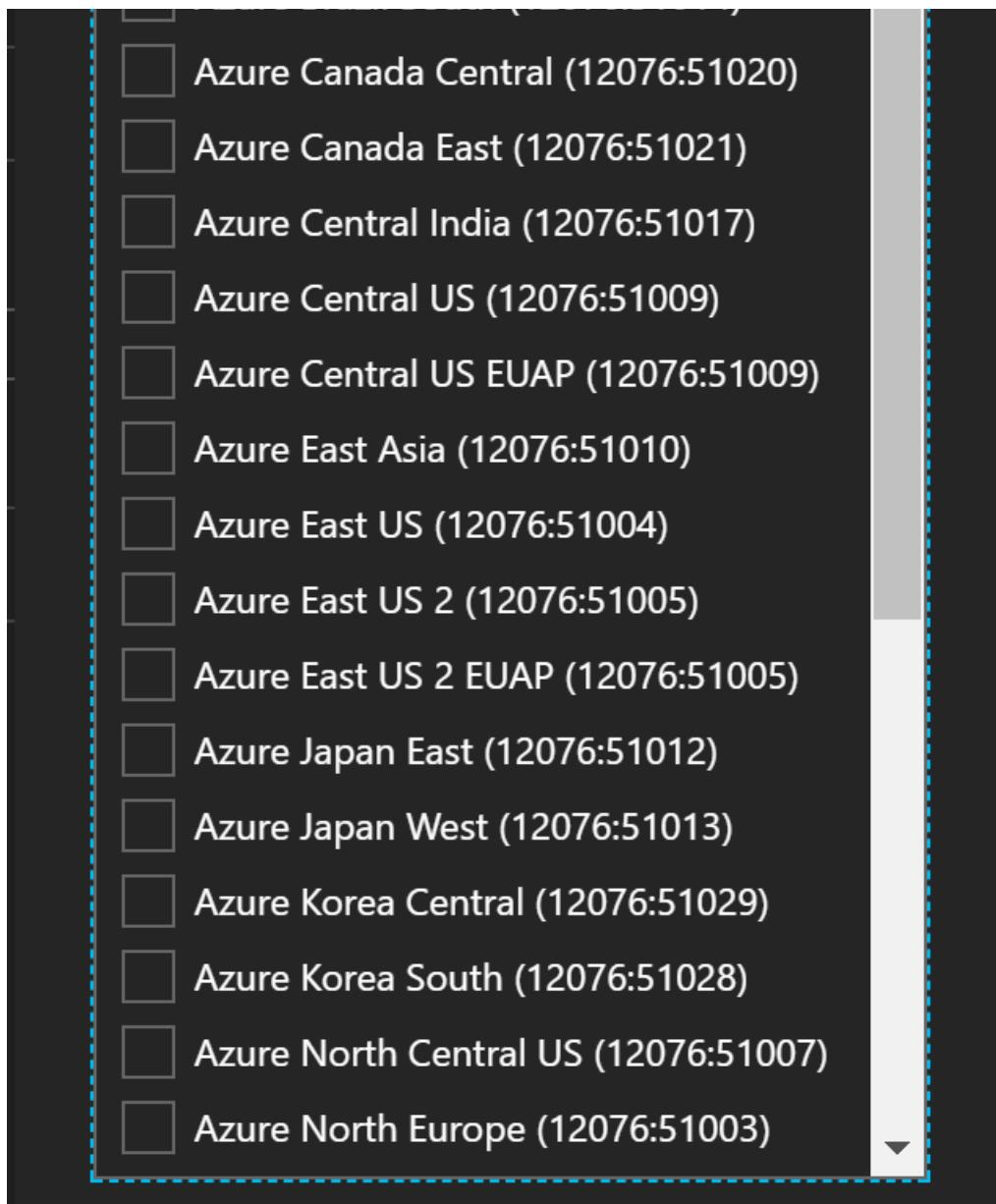
2. Create a filter rule

You can add and update rules by selecting the manage rule tab for your route filter.

The screenshot shows the Azure portal interface for a resource named 'MyRouteFilter'. On the left, there's a sidebar with navigation links like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Circuits', 'Locks', and 'Automation script'. The main content area has tabs for 'Essentials', 'Allowed service communities', and 'Circuits'. The 'Manage rule' button is highlighted with a red box at the top of the 'Essentials' tab. Below it, there are sections for 'Resource group', 'Status', 'Location', 'Subscription', and 'Communities filtered'. Under 'Allowed service communities', there's a table with columns 'NAME' and 'VALUE', showing 'No data'. Under 'Circuits', there's a table with columns 'NAME', 'CIRCUIT STATUS', 'PROVIDER STATUS', and 'PROVIDER', also showing 'No data'.

You can select the services you want to connect to from the drop-down list and save the rule when done.

The screenshot shows the 'Manage rule' dialog box. At the top, there's a logo, the title 'Manage rule', and a subtitle 'AllowSPO'. Below that are 'Save' and 'Discard' buttons. The main area starts with a required field 'Rule name' containing 'Rule1'. Below it is a section 'Allowed service communities' with a heading '2 selected'. A dropdown menu lists several services with checkboxes: 'Select all' (unchecked), 'Exchange (12076:5010)' (unchecked), 'Other Office 365 Services (12076:5100)' (checked), 'SharePoint Online (12076:5020)' (checked), 'Skype For Business (12076:5030)' (unchecked), 'CRM Online (12076:5040)' (unchecked), 'Azure Australia East (12076:51015)' (unchecked), 'Azure Australia Southeast (12076:510...)' (unchecked), and 'Azure Brazil South (12076:51014)' (unchecked).



Step 3: Attach the route filter to an ExpressRoute circuit

You can attach the route filter to a circuit by selecting the "add Circuit" button and selecting the ExpressRoute circuit from the drop-down list.

NAME	CIRCUIT STATUS	PROVIDER STATUS	PROVIDER
No data			

If the connectivity provider configures peering for your ExpressRoute circuit refresh the circuit from the

ExpressRoute circuit blade before you select the "add Circuit" button.

The screenshot shows the 'NTT_SMC_Test' ExpressRoute circuit blade. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Connections, Authorizations, Peerings, Properties, Locks, Automation script, and New support request. The main area has a 'Refresh' button highlighted with a red box. The 'Essentials' section displays resource group (NTT_SMC_Test), provider (NTT SmartConnect), and various status details like Enabled, Peering location (Osaka), and Bandwidth (50 Mbps). A table lists circuits by Type (Azure private, Azure public, Microsoft) and Status (Provisioned, Not provisioned). The Microsoft row is also highlighted with a red box.

Common tasks

To get the properties of a route filter

You can view properties of a route filter when you open the resource in the portal.

The screenshot shows the 'TestRouteFilter' Route filter blade. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, and New support request. The main area has a 'Manage rule' button highlighted with a red box. The 'Essentials' section displays resource group (USWest-ER-Demo-RG), status (Succeeded), and communities associated (5 communities, 1 circuits). The 'Allowed service communities' table lists various services with their corresponding BGP community values. The 'Circuits' table shows one circuit named 'ER-Demo-Ckt-SV' with its status (Enabled) and provider (Equinix).

To update the properties of a route filter

You can update the list of BGP community values attached to a circuit by selecting the "Manage rule" button.

MyRouteFilter
Route filter

Search (Ctrl+)

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Circuits

Locks

Automation script

New support request

Move Delete Manage rule Add circuit

Resource group [change](#) Portal-Demo

Status: Succeeded

Location: East US

Subscription [change](#) ExpressRoute-Lab

Subscription ID: 4bfffbb15-d414-4874-a2e4-c548c6d45e2a

Communities filtered: 0 communities

Circuits associated: 0 circuits

Allowed service communities

Search communities

NAME	VALUE
No data	

Circuits

Search circuits

NAME	CIRCUIT STATUS	PROVIDER STATUS	PROVIDER
No data			

Manage rule

AllowSPO

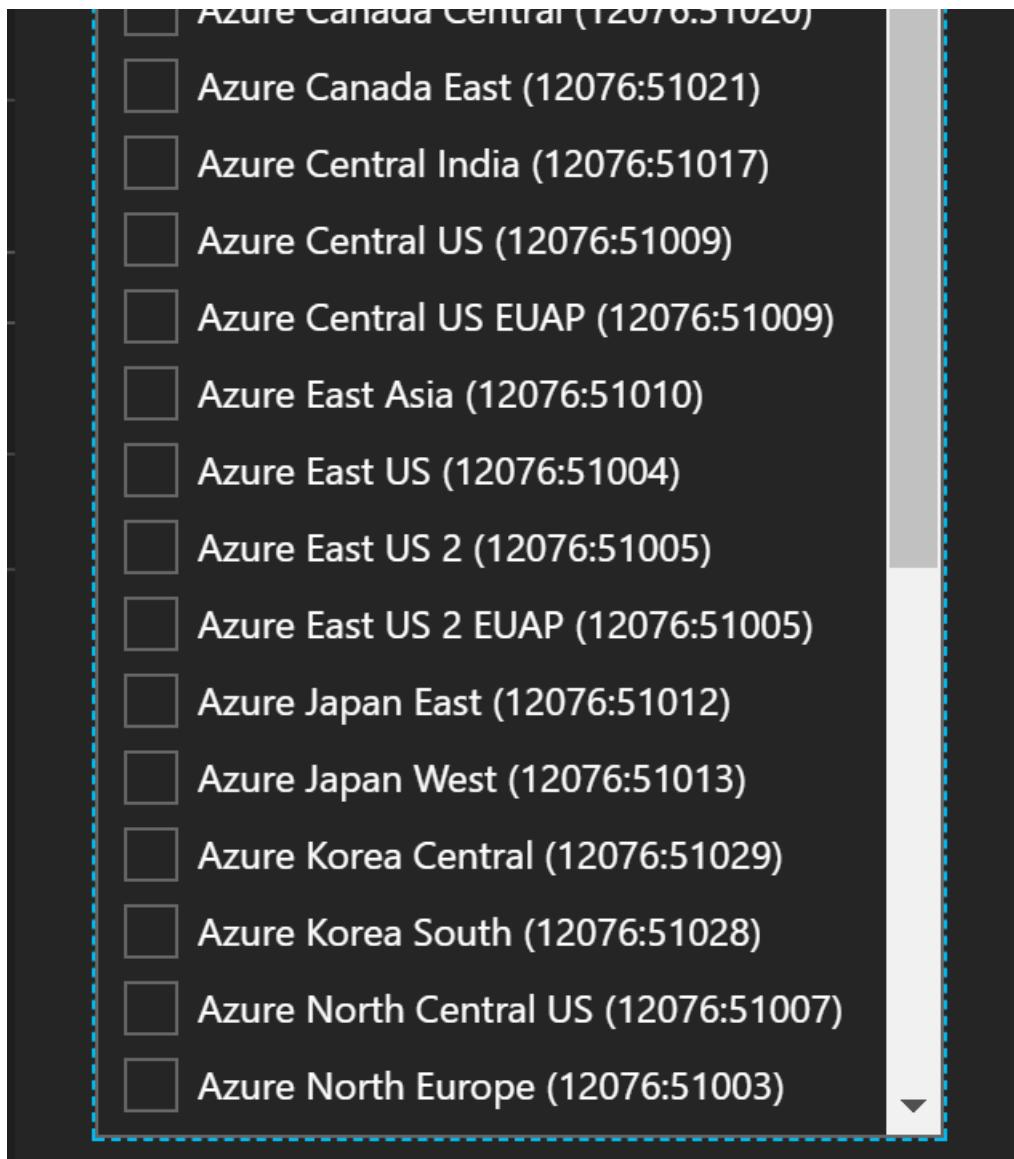
Save Discard

* Rule name

* Allowed service communities

2 selected

- Select all
- Exchange (12076:5010)
- Other Office 365 Services (12076:5100)
- SharePoint Online (12076:5020)
- Skype For Business (12076:5030)
- CRM Online (12076:5040)
- Azure Australia East (12076:51015)
- Azure Australia Southeast (12076:510...)
- Azure Brazil South (12076:51014)
- Azure Canada Central (12076:51020)



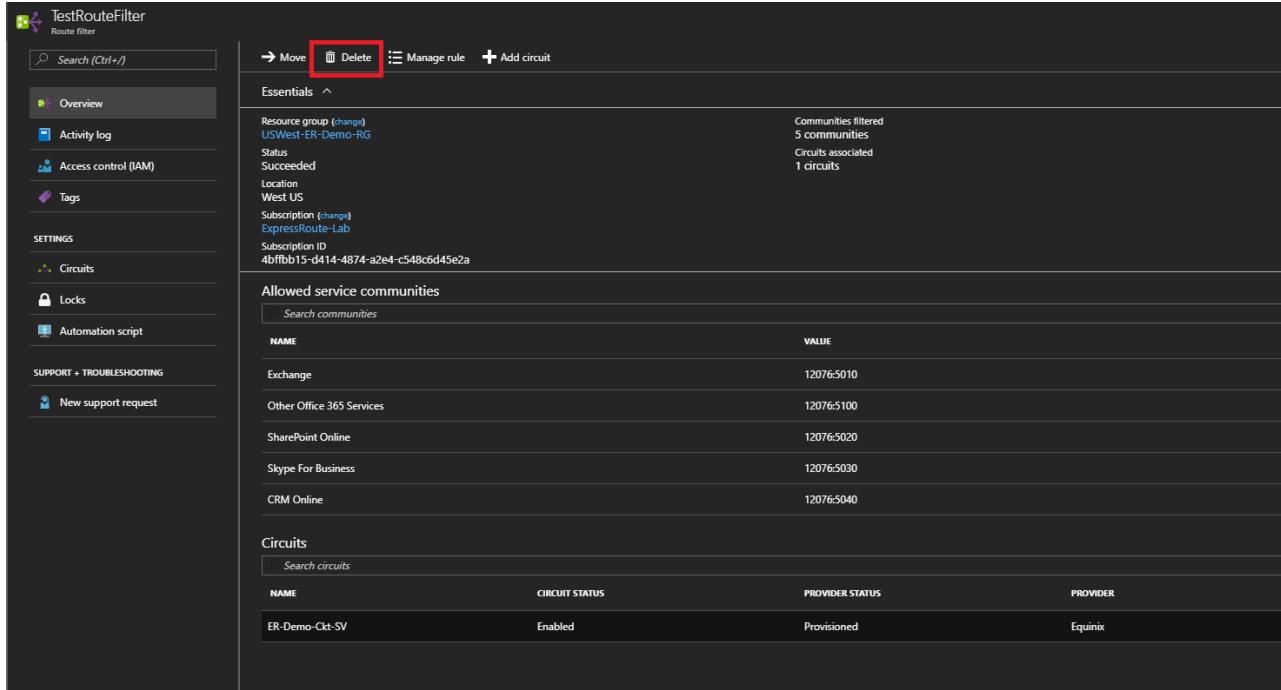
To detach a route filter from an ExpressRoute circuit

To detach a circuit from the route filter, right-click on the circuit and click on "disassociate".

The screenshot shows the Azure portal interface for managing a route filter named 'TestRouteFilter'. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, New support request, and Support + Troubleshooting. The main content area displays the route filter settings. In the 'Circuits' section, there is one entry: 'ER-Demo-Ckt-SV' with 'Status' set to 'Enabled'. Below the circuit table, there is a 'Dissociate' button highlighted with a red box.

To delete a route filter

You can delete a route filter by selecting the delete button.



The screenshot shows the Azure portal interface for managing a route filter named 'TestRouteFilter'. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, and New support request. The main content area displays the route filter's properties: Resource group (USWest-ER-Demo-RG), Status (Succeeded), Location (West US), Subscription (ExpressRoute-Lab), and Subscription ID (4bfbb15-d414-4874-a2e4-c548c6d45e2a). It also lists 'Allowed service communities' (Exchange, Other Office 365 Services, SharePoint Online, Skype For Business, CRM Online) and associated values. A 'Circuits' section shows one circuit named 'ER-Demo-Ckt-SV' with status 'Enabled', provider status 'Provisioned', and provider 'Equinix'. The top navigation bar features Move, Delete (highlighted with a red box), Manage rule, and Add circuit buttons.

Next Steps

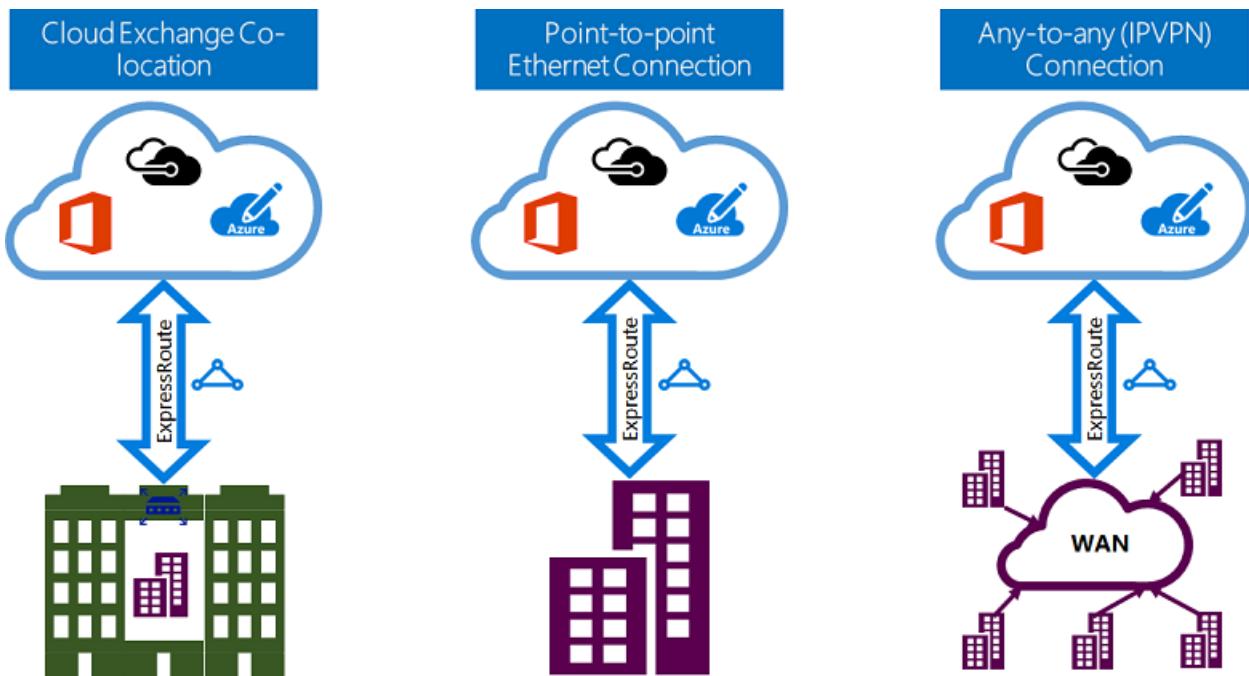
- For more information about ExpressRoute, see the [ExpressRoute FAQ](#).
- For information about router configuration samples, see [Router configuration samples to set up and manage routing](#).

ExpressRoute connectivity models

11/14/2019 • 2 minutes to read • [Edit Online](#)

You can create a connection between your on-premises network and the Microsoft cloud in three different ways, [CloudExchange Co-location](#), [Point-to-point Ethernet Connection](#), and [Any-to-any \(IPVPN\) Connection](#).

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.



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.

Next steps

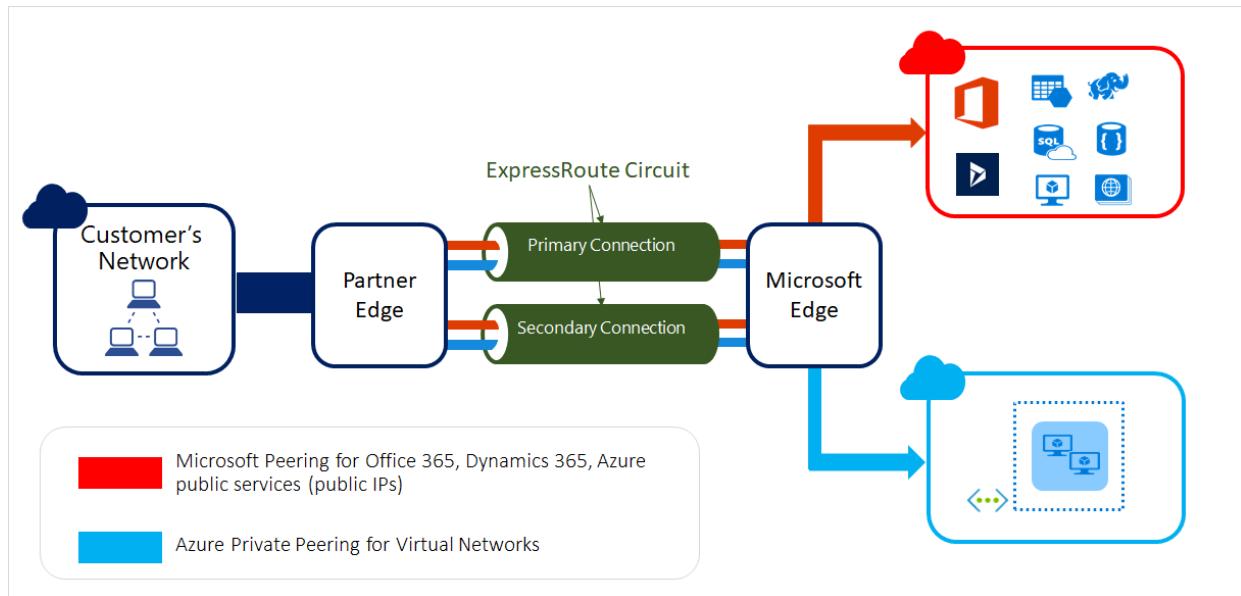
- Learn about ExpressRoute connections and routing domains. See [ExpressRoute circuits and routing domains](#).

- Learn about ExpressRoute features. See the [ExpressRoute Technical Overview](#)
- 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 circuits and peering

1/3/2020 • 4 minutes to read • [Edit Online](#)

ExpressRoute circuits connect your on-premises infrastructure to Microsoft through a connectivity provider. This article helps you understand ExpressRoute circuits and routing domains/peering. The following figure shows a logical representation of connectivity between your WAN and Microsoft.



IMPORTANT

Azure public peering has been deprecated and is not available for new ExpressRoute circuits. New circuits support Microsoft peering and private peering.

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 your 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.

New ExpressRoute circuits can include two independent peerings: Private peering and Microsoft peering. Whereas existing ExpressRoute circuits may contain three 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 ($1 \leq N \leq 3$) mapping between an ExpressRoute circuit and routing domains. An ExpressRoute circuit can have any one, two, or all three peerings enabled per ExpressRoute circuit.

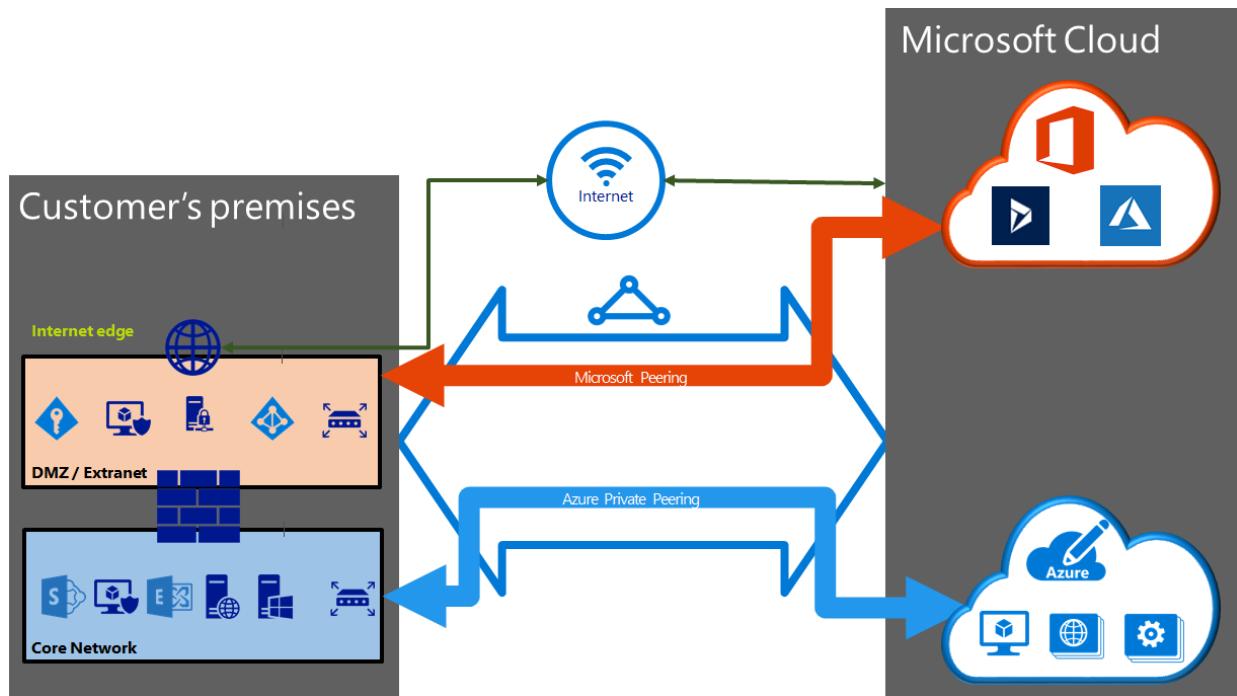
Each circuit has a fixed bandwidth (50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 10 Gbps) and is mapped to a connectivity provider and a peering location. The bandwidth you select is shared across all circuit peerings.

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 peering

An ExpressRoute circuit has multiple routing domains/peerings associated with it: Azure public, Azure private, and Microsoft. Each peering 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.



Azure 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.

Microsoft peering

Office 365 was created to be accessed securely and reliably via the Internet. Because of this, we recommend ExpressRoute for specific scenarios. For information about using ExpressRoute to access Office 365, visit [Azure ExpressRoute for Office 365](#).

Connectivity to Microsoft online services (Office 365 and Azure PaaS services) occurs through 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. For more information, see the [ExpressRoute prerequisites](#) page.

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.

Peering comparison

The following table compares the three peerings:

	PRIVATE PEERING	MICROSOFT PEERING	PUBLIC PEERING (DEPRECATED FOR NEW CIRCUITS)
Max. # prefixes supported per peering	4000 by default, 10,000 with ExpressRoute Premium	200	200
IP address ranges supported	Any valid IP address within your WAN.	Public IP addresses owned by you or your connectivity provider.	Public IP 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.
IP protocols supported	IPv4	IPv4, IPv6	IPv4
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 may enable one or more of the routing domains as part of your 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.

Each peering requires separate 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 are responsible for configuring and managing routing. You can learn more by reviewing the [workflows](#) for setting up ExpressRoute.

ExpressRoute health

ExpressRoute circuits may be monitored for availability, connectivity to VNets and bandwidth utilization using [Network Performance Monitor](#) (NPM).

NPM monitors the health of Azure private peering and Microsoft peering. Check out our [post](#) for more information.

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 and manage ExpressRoute circuits](#)
 - [Configure routing \(peering\) for ExpressRoute circuits](#)

ExpressRoute partners and peering locations

2/20/2020 • 11 minutes to read • [Edit Online](#)

The tables in this article provide information on ExpressRoute geographical coverage and locations, ExpressRoute connectivity providers, and ExpressRoute System Integrators (SIs).

NOTE

Azure regions and ExpressRoute locations are two distinct and different concepts, understanding the difference between the two is critical to exploring Azure hybrid networking connectivity.

Azure regions

Azure regions are global datacenters where Azure compute, networking and storage resources are located. When creating an Azure resource, a customer needs to select a resource location. The resource location determines which Azure datacenter (or availability zone) the resource is created in.

ExpressRoute locations

ExpressRoute locations (sometimes referred to as peering locations or meet-me-locations) are co-location facilities where Microsoft Enterprise edge (MSEE) devices are located. ExpressRoute locations are the entry point to Microsoft's network – and are globally distributed, providing customers the opportunity to connect to Microsoft's network around the world. These locations are where ExpressRoute partners and ExpressRoute Direct customers issue cross connections to Microsoft's network. In general, the ExpressRoute location does not need to match the Azure region. For example, a customer can create an ExpressRoute circuit with the resource location *East US*, in the *Seattle* Peering location.

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
Australia Government	Australia Central, Australia Central 2	Canberra, Canberra2
Europe	France Central, France South, Germany North, Germany West Central, North Europe, Norway East, Norway West, Switzerland North, Switzerland West, UK West, UK South, West Europe	Amsterdam, Amsterdam2, Copenhagen, Dublin, Frankfurt, Geneva, London, London2, Marseille, Milan, Munich, Newport(Wales), Oslo, Paris, Stavanger, Stockholm, Zurich, Munich
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, Denver, Las Vegas, Los Angeles, Miami, New York, San Antonio, Seattle, Silicon Valley, Silicon Valley2, Washington DC, Washington DC2, Montreal, Quebec City, Toronto

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
Asia	East Asia, Southeast Asia	Bangkok, Hong Kong, Hong Kong2, Jakarta, Kuala Lumpur, Singapore, Singapore2, Taipei
India	India West, India Central, India South	Chennai, Chennai2, Mumbai, Mumbai2
Japan	Japan West, Japan East	Osaka, Tokyo, Tokyo2
Oceania	Australia Southeast, Australia East	Auckland, Melbourne, Perth, Sydney, Sydney2
South Korea	Korea Central, Korea South	Busan, Seoul
UAE	UAE Central, UAE North	Dubai, Dubai2
South Africa	South Africa West, South Africa North	Cape Town, Johannesburg
South America	Brazil South	Sao Paulo

Azure 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 Arizona, US Gov Iowa, US Gov Texas, US Gov Virginia, US DoD Central, US DoD East	Chicago, Dallas, New York, Phoenix, San Antonio, Seattle, Silicon Valley, Washington DC
China East	China East, China East2	Shanghai, Shanghai2
China North	China North, China North2	Beijing, Beijing2
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.

ExpressRoute connectivity providers

The following table shows connectivity locations and the service providers for each location. If you want to view service providers and the locations for which they can provide service, see [Locations by service provider](#).

- **Local Azure Regions** are the ones that [ExpressRoute Local](#) at each peering location can access. **n/a** indicates that ExpressRoute Local is not available at that peering location.
- **Zone** refers to [pricing](#).

Global commercial Azure

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Amsterdam	Equinix AM5	1	West Europe	10G, 100G	Aryaka Networks, AT&T NetBond, British Telecom, Colt, Equinix, euNetworks, GÉANT, InterCloud, Interxion, KPN, IX Reach, Level 3 Communications, Megaport, NTT Communications, Orange, Tata Communications, Telefonica, Telenor, Telia Carrier, Verizon, Zayo
Amsterdam2	Interxion AMS8	1	West Europe	10G, 100G	CenturyLink Cloud Connect, Colt, DE-CIX, euNetworks, Interxion, Orange, Vodafone
Atlanta	Equinix AT2	1	n/a	n/a	Equinix, Megaport
Auckland	Vocus Group NZ Albany	2	n/a	10G	Devoli, Kordia, Megaport, Spark NZ, Vocus Group NZ
Bangkok	AIS	2	n/a	10G	AIS
Busan	LG CNS	2	Korea South	n/a	LG CNS
Canberra	CDC	1	Australia Central	10G, 100G	CDC
Canberra2	CDC	1	Australia Central 2	10G, 100G	CDC
Cape Town	Teraco CT1	3	South Africa West	10G	BCX, Internet Solutions - Cloud Connect, Liquid Telecom, Teraco
Chennai	Tata Communications	2	South India	10G	Global CloudXchange (GCX), SIFY, Tata Communications
Chennai2	Airtel	2	South India	n/a	Airtel

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Chicago	Equinix CH1	1	North Central US	10G, 100G	Aryaka Networks, AT&T NetBond, CenturyLink Cloud Connect, Cologix, Colt, Comcast, Coresite, Equinix, InterCloud, Internet2, Level 3 Communications, Megaport, PacketFabric, PCCW Global Limited, Sprint, Telia Carrier, Verizon, Zayo
Copenhagen	Interxion CPH1	1	n/a	10G	Interxion
Dallas	Equinix DA3	1	n/a	10G, 100G	Aryaka Networks, AT&T NetBond, Cologix, Equinix, Internet2, Level 3 Communications, Megaport, Neutrona Networks, Telmex Uninet, Telia Carrier, Transtelco, Verizon, Zayo
Denver	CoreSite DE1	1	West Central US	n/a	CoreSite, Megaport, Zayo
Dubai	PCCS	3	UAE North	n/a	Etisalat UAE
Dubai2	du datamena	3	UAE North	n/a	du datamena, Megaport, Orange, Orixcom
Dublin	Equinix DB3	1	North Europe	10G, 100G	Colt, eir, Equinix, euNetworks, Interxion, Megaport
Frankfurt	Interxion FRA11	1	Germany West Central	10G, 100G	Colt, DE-CIX, GEANT, Interxion, Megaport, Orange, Telia Carrier
Geneva	Equinix GV2	1	Switzerland West	10G, 100G	Equinix, Megaport

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Hong Kong	Equinix HK1	2	East Asia	n/a	Aryaka Networks, British Telecom, CenturyLink Cloud Connect, Chief Telecom, China Telecom Global, Equinix, InterCloud, Megaport, NTT Communications, Orange, PCCW Global Limited, Tata Communications, Telia Carrier, Verizon
Hong Kong2	MEGA-i	2	n/a	10G	
Jakarta	Telkom Indonesia	4	n/a	10G	
Johannesburg	Teraco JB1	3	South Africa North	10G	British Telecom, Internet Solutions - Cloud Connect, Liquid Telecom, Orange, Teraco
Kuala Lumpur	TIME dotCom Menara AIMS	2	n/a	n/a	TIME dotCom
Las Vegas	Switch LV	1	n/a	n/a	CenturyLink Cloud Connect, Megaport
London	Equinix LD5	1	UK South	10G, 100G	AT&T NetBond, British Telecom, Colt, Equinix, euNetworks, InterCloud, Internet Solutions - Cloud Connect, Interxion, Jisc, Level 3 Communications, Megaport, MTN, NTT Communications, Orange, PCCW Global Limited, Tata Communications, Telehouse - KDDI, Telenor, Telia Carrier, Verizon, Vodafone, Zayo

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
London2	Telehouse North Two	1	UK South	10G, 100G	CenturyLink Cloud Connect, Colt, IX Reach, Equinix, Megaport, Telehouse - KDDI
Los Angeles	CoreSite LA1	1	n/a	n/a	CoreSite, Equinix, Megaport, Neutrona Networks, NTT, Transtelco, Zayo
Marseille	Interxion MRS1	1	France South	n/a	DE-CIX, GEANT, Interxion, Jaguar Network
Melbourne	NextDC M1	2	Australia Southeast	10G, 100G	AARNet, Devoli, Equinix, Megaport, NEXTDC, Optus, Telstra Corporation, TPG Telecom
Miami	Equinix MI1	1	n/a	10G	C3ntro, Equinix, Megaport, Neutrona Networks
Milan	IRIDEOS	1	n/a	10G	
Montreal	Cologix MTL3	1	n/a	10G, 100G	Bell Canada, Cologix, Megaport, Telus, Zayo
Mumbai	Tata Communications	2	West India	n/a	Global CloudXchange (GCX), Reliance Jio, Sify, Tata Communications, Verizon
Mumbai2	Airtel	2	West India	n/a	Airtel, Sify, Vodafone Idea
Munich	EdgeConneX	1	n/a	10G, 100G	
New York	Equinix NY9	1	n/a	n/a	CenturyLink Cloud Connect, Colt, Coresite, Equinix, InterCloud, Megaport, Packet, Zayo

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Newport(Wales)	Next Generation Data	1	UK West	n/a	British Telecom, Colt, Level 3 Communications, Next Generation Data
Osaka	Equinix OS1	2	Japan West	10G, 100G	Colt, Equinix, Internet Initiative Japan Inc. - IIJ, Megaport, NTT Communications, NTT SmartConnect, Softbank
Oslo	DigiPlex Ulven	1	Norway East	10G, 100G	Megaport, Telenor, Telia Carrier
Paris	Interxion PAR5	1	France Central	10G, 100G	CenturyLink Cloud Connect, Colt, Equinix, Intercloud, Interxion, Orange, Telia Carrier, Zayo
Perth	NextDC P1	2	n/a	10G	Megaport, NextDC
Quebec City	Vantage	1	Canada East	n/a	Bell Canada, Megaport
San Antonio	CyrusOne SA1	1	South Central US	10G, 100G	CenturyLink Cloud Connect, Megaport
Sao Paulo	Equinix SP2	3	Brazil South	n/a	Aryaka Networks, Ascenty Data Centers, British Telecom, Equinix, Level 3 Communications, Neutrona Networks, Orange, Tata Communications, Telefonica, UOLDIVEO
Seattle	Equinix SE2	1	West US 2	10G, 100G	Aryaka Networks, Equinix, Level 3 Communications, Megaport, Telus, Zayo

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Seoul	KINX Gasan IDC	2	Korea Central	10G, 100G	KINX, KT, LG CNS, Sejong Telecom
Silicon Valley	Equinix SV1	1	West US	10G, 100G	Aryaka Networks, AT&T NetBond, British Telecom, CenturyLink Cloud Connect, Colt, Comcast, Coresite, Equinix, InterCloud, Internet2, IX Reach, Packet, PacketFabric, Level 3 Communications, Megaport, Orange, Sprint, Tata Communications, Telia Carrier, Verizon, Zayo
Silicon Valley2	Coresite SV7	1	West US	10G, 100G	Colt, Coresite
Singapore	Equinix SG1	2	Southeast Asia	10G, 100G	Aryaka Networks, AT&T NetBond, British Telecom, China Mobile International, Epsilon Global Communications, Equinix, InterCloud, Level 3 Communications, Megaport, NTT Communications, Orange, SingTel, Tata Communications, Telstra Corporation, Verizon, Vodafone
Singapore2	Global Switch Tai Seng	2	Southeast Asia	10G, 100G	China Unicom Global, Colt, Epsilon Global Communications, Megaport, SingTel
Stavanger	Green Mountain DC1	1	Norway West	10G, 100G	

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Stockholm	Equinix SK1	1	n/a	10G	Equinix, Telia Carrier
Sydney	Equinix SY2	2	Australia East	10G, 100G	AARNet, AT&T NetBond, British Telecom, Devoli, Equinix, Kordia, Megaport, NEXTDC, NTT Communications, Optus, Orange, Spark NZ, Telstra Corporation, TPG Telecom, Verizon, Vocus Group NZ
Sydney2	NextDC S1	2	Australia East	10G, 100G	Megaport, NextDC
Taipei	Chief Telecom	2	n/a	10G	Chief Telecom, FarEasTone
Tokyo	Equinix TY4	2	Japan East	10G, 100G	Aryaka Networks, AT&T NetBond, BBIX, British Telecom, CenturyLink Cloud Connect, Colt, Equinix, Internet Initiative Japan Inc. - IIJ, Megaport, NTT Communications, NTT EAST, Orange, Softbank, Verizon
Tokyo2	At Tokyo	2	Japan East	10G, 100G	
Toronto	Cologix TOR1	1	Canada Central	10G, 100G	AT&T NetBond, Bell Canada, CenturyLink Cloud Connect, Cologix, Equinix, IX Reach, Megaport, Telus, Verizon, Zayo

LOCATION	ADDRESS	ZONE	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Washington DC	Equinix DC2	1	East US, East US 2	10G, 100G	Aryaka Networks, AT&T NetBond, British Telecom, CenturyLink Cloud Connect, Cologix, Colt, Comcast, Coresite, Equinix, Internet2, InterCloud, IX Reach, Level 3 Communications, Megaport, Neutrona Networks, NTT Communications, Orange, PacketFabric, SES, Sprint, Tata Communications, Telia Carrier, Verizon, Zayo
Washington DC2	Coresite Reston	1	East US, East US 2	10G, 100G	CenturyLink Cloud Connect, Coresite, Intelsat, Viasat, Zayo
Zurich	Interxion ZUR2	1	Switzerland North	10G, 100G	Intercloud, Interxion, Megaport, Swisscom

+ denotes coming soon

National cloud environments

Azure national clouds are isolated from each other and from global commercial Azure. ExpressRoute for one Azure cloud can't connect to the Azure regions in the others.

US Government cloud

LOCATION	ADDRESS	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Chicago	Equinix CH1	n/a	10G, 100G	AT&T NetBond, Equinix, Level 3 Communications, Verizon
Dallas	Equinix DA3	n/a	10G, 100G	Equinix, Megaport, Verizon
New York	Equinix NY5	n/a	10G, 100G	Equinix, CenturyLink Cloud Connect, Verizon

LOCATION	ADDRESS	LOCAL AZURE REGIONS	ER DIRECT	SERVICE PROVIDERS
Phoenix	CyrusOne Chandler	US Gov Arizona	n/a	AT&T NetBond, CenturyLink Cloud Connect, Megaport
San Antonio	CyrusOne SA2	US Gov Texas	n/a	CenturyLink Cloud Connect, Megaport
Silicon Valley	Equinix SV4	n/a	10G, 100G	AT&T, Equinix, Level 3 Communications, Verizon
Seattle	Equinix SE2	n/a	n/a	Equinix, Megaport
Washington DC	Equinix DC2	US DoD East, US Gov Virginia	10G, 100G	AT&T NetBond, CenturyLink Cloud Connect, Equinix, Level 3 Communications, Megaport, Verizon

China

LOCATION	SERVICE PROVIDERS
Beijing	China Telecom
Beijing2	China Telecom, China Unicom, GDS
Shanghai	China Telecom
Shanghai2	China Telecom, GDS

To learn more, see [ExpressRoute in China](#)

Germany

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

Connectivity through Exchange providers

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](#)
- [PacketFabric](#)
- [Teraco](#)
- 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 through satellite operators

If you are remote and don't have fiber connectivity or you want to explore other connectivity options you can check the following satellite operators.

- Intelsat
- [SES](#)
- [Viasat](#)

Connectivity through additional service providers

LOCATION	EXCHANGE	CONNECTIVITY PROVIDERS
Amsterdam	Equinix, Interxion, Level 3 Communications	BICS, CloudXpress, Eurofiber, Fastweb S.p.A, Gulf Bridge International, Kalaam Telecom Bahrain B.S.C, MainOne, Nianet, POST Telecom Luxembourg, Proximus, TDC Erhverv, Telecom Italia Sparkle, Telekom Deutschland GmbH, Telia
Atlanta	Equinix	Crown Castle
Cape Town	Teraco	MTN
Chicago	Equinix	Crown Castle, Spectrum Enterprise, Windstream
Dallas	Equinix, Megaport	Axtel, C3ntro Telecom, Cox Business, Crown Castle, Data Foundry, Spectrum Enterprise, Transtelco
Frankfurt	Interxion	BICS, Cinia, Equinix, Nianet, QSC AG, Telekom Deutschland GmbH
Hamburg	Equinix	Cinia
Hong Kong SAR	Equinix	Chief, Macroview Telecom
Johannesburg	Teraco	MTN

LOCATION	EXCHANGE	CONNECTIVITY PROVIDERS
London	BICS, Equinix, euNetworks	Bezeq International Ltd., CoreAzure, Epsilon Telecommunications Limited, Exponential E, HSO, NexGen Networks, Proximus, Tamares Telecom, Zain
Los Angeles	Equinix	Crown Castle, Spectrum Enterprise, Transtelco
Madrid	Level3	Zertia
Montreal	Cologix, Equinix	Airgate Technologies, Inc. Aptum Technologies, Rogers, Zirro
New York	Equinix, Megaport	Altice Business, Crown Castle, Spectrum Enterprise, Webair
Paris	Equinix	Proximus
Quebec City	Megaport	Fibrenoire
Sao Paula	Equinix	Venha Pra Nuvem
Seattle	Equinix	Alaska Communications
Silicon Valley	Coresite, Equinix	Cox Business, Spectrum Enterprise, Windstream, X2nsat Inc.
Singapore	Equinix	1CLOUDSTAR, BICS, CMC Telecom, Epsilon Telecommunications Limited, LGA Telecom, United Information Highway (UIH)
Slough	Equinix	HSO
Sydney	Megaport	Macquarie Telecom Group
Tokyo	Equinix	ARTERIA Networks Corporation, BroadBand Tower, Inc.
Toronto	Equinix, Megaport	Airgate Technologies Inc., Beanfield Metroconnect, Aptum Technologies, IVedha Inc, Rogers, Thinktel, Zirro
Washington DC	Equinix	Altice Business, BICS, Cox Business, Crown Castle, Gtt Communications Inc, Epsilon Telecommunications Limited, Masergy, Windstream

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	Ensysit, IT Consultancy, MOQdigital, Vigilant.IT
Europe	Avanade Inc., Altogee, Bright Skies GmbH, Inframont, MSG Services, New Signature, Nelite, Orange Networks, sol-tec
North America	Avanade Inc., Equinix Professional Services, FlexManage, Lightstream, Perficient, Presidio
South America	Avanade Inc., Venha Pra Nuvem

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

2/16/2020 • 11 minutes to read • [Edit Online](#)

The tables in this article provide information on ExpressRoute geographical coverage and locations, ExpressRoute connectivity providers, and ExpressRoute System Integrators (SIs).

NOTE

Azure regions and ExpressRoute locations are two distinct and different concepts, understanding the difference between the two is critical to exploring Azure hybrid networking connectivity.

Azure regions

Azure regions are global datacenters where Azure compute, networking and storage resources are located. When creating an Azure resource, a customer needs to select a resource location. The resource location determines which Azure datacenter (or availability zone) the resource is created in.

ExpressRoute locations

ExpressRoute locations (sometimes referred to as peering locations or meet-me-locations) are co-location facilities where Microsoft Enterprise Edge (MSEE) devices are located. ExpressRoute locations are the entry point to Microsoft's network – and are globally distributed, providing customers the opportunity to connect to Microsoft's network around the world. These locations are where ExpressRoute partners and ExpressRoute Direct customers issue cross connections to Microsoft's network. In general, the ExpressRoute location does not need to match the Azure region. For example, a customer can create an ExpressRoute circuit with the resource location *East US*, in the *Seattle* Peering location.

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
Australia Government	Australia Central, Australia Central 2	Canberra, Canberra2
Europe	France Central, France South, Germany North, Germany West Central, North Europe, Norway East, Norway West, Switzerland North, Switzerland West, UK West, UK South, West Europe	Amsterdam, Amsterdam2, Copenhagen, Dublin, Frankfurt, Geneva, London, London2, Marseille, Milan, Munich, Newport(Wales), Oslo, Paris, Stavanger, Stockholm, Zurich
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, Denver, Las Vegas, Los Angeles, Miami, New York, San Antonio, Seattle, Silicon Valley, Silicon Valley2, Washington DC, Washington DC2, Montreal, Quebec City, Toronto

GEOPOLITICAL REGION	AZURE REGIONS	EXPRESSROUTE LOCATIONS
Asia	East Asia, Southeast Asia	Bangkok, Hong Kong, Hong Kong2, Jakarta, Kuala Lumpur, Singapore, Singapore2, Taipei
India	India West, India Central, India South	Chennai, Chennai2, Mumbai, Mumbai2
Japan	Japan West, Japan East	Osaka, Tokyo, Tokyo2
Oceania	Australia Southeast, Australia East	Auckland, Melbourne, Perth, Sydney, Sydney2
South Korea	Korea Central, Korea South	Busan, Seoul
UAE	UAE Central, UAE North	Dubai, Dubai2
South Africa	South Africa West, South Africa North	Cape Town, Johannesburg
South America	Brazil South	Sao Paulo

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 Arizona, US Gov Iowa, US Gov Texas, US Gov Virginia, US DoD Central, US DoD East	Chicago, Dallas, New York, Phoenix, San Antonio, Seattle, Silicon Valley, Washington DC
China East	China East, China East2	Shanghai, Shanghai2
China North	China North, China North2	Beijing, Beijing2
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.

ExpressRoute connectivity providers

The following table shows locations by service provider. If you want to view available providers by location, see [Service providers by location](#).

Global commercial Azure

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
AARNet	Supported	Supported	Melbourne, Sydney
Airtel	Supported	Supported	Chennai2, Mumbai2

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
AIS	Supported	Supported	Bangkok
Aryaka Networks	Supported	Supported	Amsterdam, Chicago, Dallas, Hong Kong SAR, Sao Paulo, Seattle, Silicon Valley, Singapore, Tokyo, Washington DC
Ascenty Data Centers	Supported	Supported	Sao Paulo
AT&T NetBond	Supported	Supported	Amsterdam, Chicago, Dallas, London, Silicon Valley, Singapore, Sydney, Tokyo, Toronto, Washington DC
BBIX	Supported	Supported	Tokyo
BCX	Supported	Supported	Cape Town
Bell Canada	Supported	Supported	Montreal, Toronto, Quebec City
British Telecom	Supported	Supported	Amsterdam, Hong Kong SAR, Johannesburg, London, Newport(Wales), Sao Paulo, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
C3ntro	Supported	Supported	Miami
CDC	Supported	Supported	Canberra, Canberra2
CenturyLink Cloud Connect	Supported	Supported	Amsterdam2, Chicago, Hong Kong, Las Vegas, London2, New York, Paris, San Antonio, Silicon Valley, Tokyo, Toronto, Washington DC, Washington DC2
Chief Telecom	Supported	Supported	Hong Kong, Taipei
China Mobile International	Supported	Supported	Singapore
China Telecom Global	Supported	Supported	Hong Kong
China Unicom Global	Supported	Supported	Singapore2
Cologix	Supported	Supported	Chicago, Dallas, Montreal, Toronto, Washington DC

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Colt	Supported	Supported	Amsterdam, Amsterdam2, Chicago, Dublin, Frankfurt, London, London2, Newport, New York, Osaka, Paris, Silicon Valley, Silicon Valley2, Singapore2, Tokyo, Washington DC
Comcast	Supported	Supported	Chicago, Silicon Valley, Washington DC
CoreSite	Supported	Supported	Chicago, Denver, Los Angeles, New York, Silicon Valley, Silicon Valley2, Washington DC, Washington DC2
DE-CIX	Supported	Supported	Amsterdam2, Frankfurt, Marseille
Devoli	Supported	Supported	Auckland, Melbourne, Sydney
du datamena	Supported	Supported	Dubai2
eir	Supported	Supported	Dublin
Epsilon Global Communications	Supported	Supported	Singapore, Singapore2
Equinix	Supported	Supported	Amsterdam, Atlanta, Chicago, Dallas, Dublin, Frankfurt, Geneva, Hong Kong SAR, London, London2, Los Angeles, Melbourne, Miami, New York, Osaka, Paris, Sao Paulo, Seattle, Silicon Valley, Singapore, Stockholm, Sydney, Tokyo, Toronto, Washington DC
Etisalat UAE	Supported	Supported	Dubai
euNetworks	Supported	Supported	Amsterdam, Amsterdam2, Dublin, London
FarEasTone	Supported	Supported	Taipei
GÉANT	Supported	Supported	Amsterdam, Frankfurt, Marseille
Global Cloud Xchange (GCX)	Supported	Supported	Chennai, Mumbai

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Intelsat	Supported	Supported	Washington DC2
InterCloud	Supported	Supported	Amsterdam, Chicago, Hong Kong, London, New York, Paris, Silicon Valley, Singapore, Washington DC, Zurich
Internet2	Supported	Supported	Chicago, Dallas, Silicon Valley, Washington DC
Internet Initiative Japan Inc. - IIJ	Supported	Supported	Osaka, Tokyo
Internet Solutions - Cloud Connect	Supported	Supported	Cape Town, Johannesburg, London
Interxion	Supported	Supported	Amsterdam, Amsterdam2, Copenhagen, Dublin, Frankfurt, London, Marseille, Paris, Zurich
IX Reach	Supported	Supported	Amsterdam, London2, Silicon Valley, Toronto, Washington DC
Jaguar Network	Supported	Supported	Marseille
Jisc	Supported	Supported	London
KINX	Supported	Supported	Seoul
Kordia	Supported	Supported	Auckland, Sydney
KPN	Supported	Supported	Amsterdam
KT	Supported	Supported	Seoul
Level 3 Communications	Supported	Supported	Amsterdam, Chicago, Dallas, London, Newport (Wales), Sao Paulo, Seattle, Silicon Valley, Singapore, Washington DC
LG CNS	Supported	Supported	Busan, Seoul
Liquid Telecom	Supported	Supported	Cape Town, Johannesburg

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Megaport	Supported	Supported	Amsterdam, Atlanta, Auckland, Chicago, Dallas, Denver, Dubai2, Dublin, Frankfurt, Geneva, Hong Kong SAR, Las Vegas, London, London2, Los Angeles, Melbourne, Miami, Montreal, New York, Oslo, Perth, Quebec City, San Antonio, Seattle, Silicon Valley, Singapore, Singapore2, Sydney, Sydney2, Tokyo, Toronto, Washington DC, Zurich
MTN	Supported	Supported	London
Neutrona Networks	Supported	Supported	Dallas, Los Angeles, Miami, Sao Paulo, Washington DC
Next Generation Data	Supported	Supported	Newport(Wales)
NEXTDC	Supported	Supported	Melbourne, Perth, Sydney, Sydney2
NTT Communications	Supported	Supported	Amsterdam, Hong Kong SAR, London, Los Angeles, Osaka, Singapore, Sydney, Tokyo, Washington DC
NTT EAST	Supported	Supported	Tokyo
NTT SmartConnect	Supported	Supported	Osaka
Optus	Supported	Supported	Melbourne, Sydney
Orange	Supported	Supported	Amsterdam, Amsterdam2, Dubai2, Frankfurt, Hong Kong SAR, Johannesburg, London, Paris, Sao Paulo, Silicon Valley, Singapore, Sydney, Tokyo, Washington DC
Orixcom	Supported	Supported	Dubai2
PacketFabric	Supported	Supported	Chicago, Silicon Valley, Washington DC
PCCW Global Limited	Supported	Supported	Chicago, Hong Kong SAR, London
Sejong Telecom	Supported	Supported	Seoul
SES	Supported	Supported	Washington DC

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
SIFY	Supported	Supported	Chennai, Mumbai2
SingTel	Supported	Supported	Singapore, Singapore2
Softbank	Supported	Supported	Osaka, Tokyo
Spark NZ	Supported	Supported	Auckland, Sydney
Sprint	Supported	Supported	Chicago, Silicon Valley, Washington DC
Swisscom	Supported	Supported	Zurich
Tata Communications	Supported	Supported	Amsterdam, Chennai, Hong Kong SAR, London, Mumbai, Sao Paulo, Silicon Valley, Singapore, Washington DC
Telefonica	Supported	Supported	Amsterdam, Sao Paulo
Telehouse - KDDI	Supported	Supported	London, London2
Telenor	Supported	Supported	Amsterdam, London, Oslo
Telia Carrier	Supported	Supported	Amsterdam, Chicago, Dallas, Frankfurt, Hong Kong, London, Oslo, Paris, Silicon Valley, Stockholm, Washington DC
Telmex Uninet	Supported	Supported	Dallas
Telstra Corporation	Supported	Supported	Melbourne, Singapore, Sydney
Telus	Supported	Supported	Montreal, Seattle, Toronto
Teraco	Supported	Supported	Cape Town, Johannesburg
TIME dotCom	Supported	Supported	Kuala Lumpur
Transtelco	Supported	Supported	Dallas, Los Angeles
UOLDIVEO	Supported	Supported	Sao Paulo
Verizon	Supported	Supported	Amsterdam, Chicago, Dallas, Hong Kong SAR, London, Mumbai, Silicon Valley, Singapore, Sydney, Tokyo, Toronto, Washington DC
Viasat	Supported	Supported	Washington DC2

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
Vocus Group NZ	Supported	Supported	Auckland, Sydney
Vodafone	Supported	Supported	Amsterdam2, London, Singapore
Vodafone Idea	Supported	Supported	Mumbai, Mumbai2
Zayo	Supported	Supported	Amsterdam, Chicago, Dallas, Denver, London, Los Angeles, Montreal, New York, Paris, Seattle, Silicon Valley, Toronto, Washington DC, Washington DC2

+ denotes coming soon

National cloud environment

Azure national clouds are isolated from each other and from global commercial Azure. ExpressRoute for one Azure cloud can't connect to the Azure regions in the others.

US Government cloud

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
AT&T NetBond	Supported	Supported	Chicago, Phoenix, Silicon Valley, Washington DC
CenturyLink Cloud Connect	Supported	Supported	New York, Phoenix, San Antonio, Washington DC
Equinix	Supported	Supported	Chicago, Dallas, New York, Seattle, Silicon Valley, Washington DC
Level 3 Communications	Supported	Supported	Chicago, Silicon Valley, Washington DC
Megaport	Supported	Supported	Chicago, Dallas, San Antonio, Seattle, Washington DC
Verizon	Supported	Supported	Chicago, Dallas, New York, Silicon Valley, Washington DC

China

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
China Telecom	Supported	Not Supported	Beijing, Beijing2, Shanghai, Shanghai2
China Unicom	Supported	Not Supported	Beijing2

SERVICE PROVIDER	MICROSOFT AZURE	OFFICE 365	LOCATIONS
GDS	Supported	Not Supported	Beijing2, Shanghai2

To learn more, see [ExpressRoute in China](#).

Germany

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

Connectivity through Exchange providers

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](#)
 - [IX Reach](#)
 - [Megaport](#)
 - [NextDC](#)
 - [PacketFabric](#)
 - [Teraco](#)
- 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 through satellite operators

If you are remote and don't have fiber connectivity or you want to explore other connectivity options you can check the following satellite operators.

- Intelsat

- SES
- Viasat

Connectivity through additional service providers

CONNECTIVITY PROVIDER	EXCHANGE	LOCATIONS
1CLOUDSTAR	Equinix	Singapore
Airgate Technologies, Inc.	Equinix, Cologix	Toronto, Montreal
Alaska Communications	Equinix	Seattle
Altice Business	Equinix	New York, Washington DC
Arteria Networks Corporation	Equinix	Tokyo
Axtel	Equinix	Dallas
Beanfield Metroconnect	Megaport	Toronto
Bezeq International Ltd.	euNetworks	London
BICS	Equinix	Amsterdam, Frankfurt, London, Singapore, Washington DC
BroadBand Tower, Inc.	Equinix	Tokyo
C3ntro Telecom	Equinix, Megaport	Dallas
Chief	Equinix	Hong Kong SAR
Cinia	Equinix, Megaport	Frankfurt, Hamburg
CloudXpress	Equinix	Amsterdam
CMC Telecom	Equinix	Singapore
Aptum Technologies	Equinix	Montreal, Toronto
CoreAzure	Equinix	London
Cox Business	Equinix	Dallas, Silicon Valley, Washington DC
Crown Castle	Equinix	Atlanta, Chicago, Dallas, Los Angeles, New York, Washington DC
Data Foundry	Megaport	Dallas
Epsilon Telecommunications Limited	Equinix	London, Singapore, Washington DC
Eurofiber	Equinix	Amsterdam

CONNECTIVITY PROVIDER	EXCHANGE	LOCATIONS
Exponential E	Equinix	London
Fastweb S.p.A	Equinix	Amsterdam
Fibrenoire	Megaport	Quebec City
Gtt Communications Inc	Equinix	Washington DC
Gulf Bridge International	Equinix	Amsterdam
HSO	Equinix	London, Slough
IVedha Inc	Equinix	Toronto
Kaalam Telecom Bahrain B.S.C	Level 3 Communications	Amsterdam
LGA Telecom	Equinix	Singapore
Macroview Telecom	Equinix	Hong Kong SAR
Macquarie Telecom Group	Megaport	Sydney
MainOne	Equinix	Amsterdam
Masergy	Equinix	Washington DC
MTN	Teraco	Cape Town, Johannesburg
NexGen Networks	Interxion	London
Nianet	Equinix	Amsterdam, Frankfurt
POST Telecom Luxembourg	Equinix	Amsterdam
Proximus	Equinix	Amsterdam, Dublin, London, Paris
QSC AG	Interxion	Frankfurt
Rogers	Cologix, Equinix	Montreal, Toronto
Spectrum Enterprise	Equinix	Chicago, Dallas, Los Angeles, New York, Silicon Valley
Tamares Telecom	Equinix	London
TDC Erhverv	Equinix	Amsterdam
Telecom Italia Sparkle	Equinix	Amsterdam
Telekom Deutschland GmbH	Interxion	Amsterdam, Frankfurt

CONNECTIVITY PROVIDER	EXCHANGE	LOCATIONS
Telia	Equinix	Amsterdam
ThinkTel	Equinix	Toronto
United Information Highway (UIH)	Equinix	Singapore
Venha Pra Nuvenm	Equinix	Sao Paulo
Webair	Megaport	New York
Windstream	Equinix	Chicago, Silicon Valley, Washington DC
X2nsat Inc.	Coresite	Silicon Valley, Silicon Valley 2
Zain	Equinix	London
Zertia	Level 3	Madrid
Zirro	Cologix, Equinix	Montreal, Toronto

Connectivity through datacenter providers

PROVIDER	EXCHANGE
CyrusOne	Megaport, PacketFabric
Cyxtera	Megaport, PacketFabric
Databank	Megaport
DataFoundry	Megaport
Digital Realty	IX Reach, Megaport PacketFabric
EdgeConnex	Megaport, PacketFabric
Flexential	IX Reach, Megaport, PacketFabric
QTS Data Centers	Megaport, PacketFabric
Stream Data Centers	Megaport
RagingWire Data Centers	IX Reach, Megaport, PacketFabric
vXchnge	IX Reach, Megaport
T5 Datacenters	IX Reach

Connectivity through National Research and Education Networks

(NREN)

PROVIDER
AARNET
DeIC, through GÉANT
GARR, through GÉANT
GÉANT
HEAnet, through GÉANT
Internet2
JISC
RedIRIS, through GÉANT
SINET
Surfnet, through GÉANT

- If your connectivity provider is not listed here, please check to see if they are connected to any of the ExpressRoute Exchange Partners listed above.

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
Altogee	Europe
Avanade Inc.	Asia, Europe, North America, South America
Bright Skies GmbH	Europe
Ensyst	Asia
Equinix Professional Services	North America
FlexManage	North America
Lightstream	North America
The IT Consultancy Group	Australia
MOQdigital	Australia

SYSTEM INTEGRATOR	CONTINENT
MSG Services	Europe (Germany)
Nelite	Europe
New Signature	Europe
OneAs1a	Asia
Orange Networks	Europe
Perficient	North America
Presidio	North America
sol-tec	Europe
Venha Pra Nuvem	South America
Vigilant.IT	Australia

Next steps

- For more information about ExpressRoute, see the [ExpressRoute FAQ](#).
- Ensure that all prerequisites are met. See [ExpressRoute prerequisites](#).

About ExpressRoute virtual network gateways

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

To connect your Azure virtual network and your on-premises network via ExpressRoute, you must create a virtual network gateway first. A virtual network gateway serves two purposes: exchange IP routes between the networks and route network traffic. This article explains gateway types, gateway SKUs, and estimated performance by SKU. This article also explains ExpressRoute [FastPath](#), a feature that enables the network traffic from your on-premises network to bypass the virtual network gateway to improve performance.

Gateway types

When you create a virtual network gateway, you need to specify several settings. One of the required settings, '-GatewayType', specifies whether the gateway is used for ExpressRoute, or VPN traffic. The two gateway types are:

- **Vpn** - To send encrypted traffic across the public Internet, you use the gateway type 'Vpn'. This is also referred to as a VPN gateway. Site-to-Site, Point-to-Site, and VNet-to-VNet connections all use a VPN gateway.
- **ExpressRoute** - To send network traffic on a private connection, you use the gateway type 'ExpressRoute'. This is also referred to as an ExpressRoute gateway and is the type of gateway used when configuring ExpressRoute.

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.

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-AzVirtualNetworkGateway' 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. Recreating a gateway incurs downtime.

Estimated performances by gateway SKU

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

	MEGABITS PER SECOND	PACKETS PER SECOND	CONNECTIONS PER SECOND	VPN GATEWAY AND EXPRESSROUTE COEXIST	FASTPATH
Basic SKU (deprecated)	500	Unknown	Unknown	No	No
Standard SKU/ErGw1AZ	1,000	100,000	7,000	Yes	No
High Performance SKU/ErGw2AZ	2,000	250,000	14,000	Yes	No
Ultra Performance SKU/ErGw3AZ	10,000	1,000,000	28,000	Yes	Yes

IMPORTANT

Application performance depends on multiple factors, such as the end-to-end latency, and the number of traffic flows the application opens. The numbers in the table represent the upper limit that the application can theoretically achieve in an ideal environment.

Gateway subnet

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

NOTE

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

When you create the gateway subnet, you specify the number of IP addresses that the subnet contains. The IP addresses in the gateway subnet are allocated to the gateway VMs and gateway services. Some configurations require more IP addresses than others.

When you are planning your gateway subnet size, refer to the documentation for the configuration that you are planning to create. For example, the ExpressRoute/VPN Gateway coexist configuration requires a larger gateway subnet than most other configurations. Additionally, you may want to make sure your gateway subnet contains enough IP addresses to accommodate possible future additional configurations. While you can create a gateway subnet as small as /29, we recommend that you create a gateway subnet of /27 or larger (/27, /26 etc.) if you have the available address space to do so. This will accommodate most configurations.

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

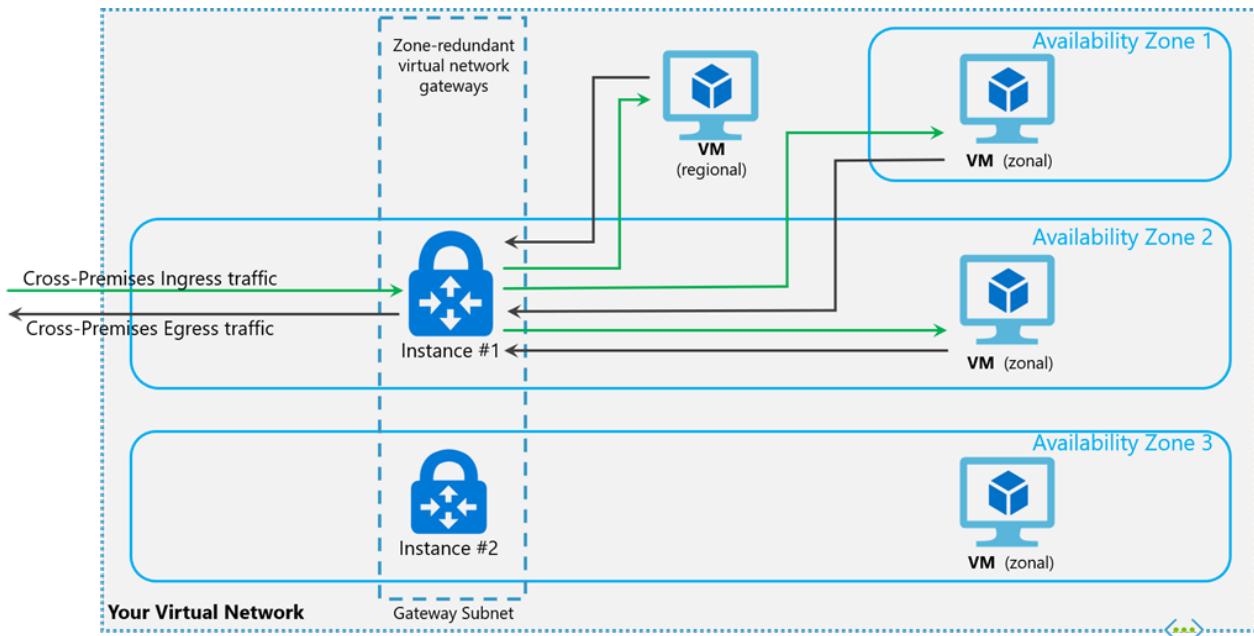
```
Add-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.0.3.0/27
```

IMPORTANT

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

Zone-redundant gateway SKUs

You can also deploy ExpressRoute gateways in Azure Availability Zones. This physically and logically separates them into different Availability Zones, protecting your on-premises network connectivity to Azure from zone-level failures.



Zone-redundant gateways use specific new gateway SKUs for ExpressRoute gateway.

- ErGw1AZ
- ErGw2AZ
- ErGw3AZ

The new gateway SKUs also support other deployment options to best match your needs. When creating a virtual network gateway using the new gateway SKUs, you also have the option to deploy the gateway in a specific zone. This is referred to as a zonal gateway. When you deploy a zonal gateway, all the instances of the gateway are deployed in the same Availability Zone.

FastPath

ExpressRoute virtual network gateway is designed to exchange network routes and route network traffic. FastPath is designed to improve the data path performance between your on-premises network and your virtual network. When enabled, FastPath sends network traffic directly to virtual machines in the virtual network, bypassing the gateway.

For more information about FastPath, including limitations and requirements, see [About FastPath](#).

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

For more information about available connection configurations, see [ExpressRoute Overview](#).

For more information about creating ExpressRoute gateways, see [Create a virtual network gateway for ExpressRoute](#).

For more information about configuring zone-redundant gateways, see [Create a zone-redundant virtual network gateway](#).

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

ExpressRoute prerequisites & checklist

11/14/2019 • 2 minutes to read • [Edit Online](#)

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

Office 365 was created to be accessed securely and reliably via the Internet. Because of this, we recommend ExpressRoute for specific scenarios. For information about using ExpressRoute to access Office 365, visit [Azure ExpressRoute for Office 365](#).

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.
- 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

- **Redundancy at each peering location:** Microsoft requires redundant BGP sessions to be set up between Microsoft's routers and the peering routers on each ExpressRoute circuit (even when you have just [one physical connection to a cloud exchange](#)).
- **Redundancy for Disaster Recovery:** Microsoft strongly recommends you set up at least two ExpressRoute circuits in different peering locations to avoid a single point of failure.
- **Routing:** depending on how you connect to the Microsoft Cloud, you or your provider needs to set up and manage the BGP sessions for [routing domains](#). Some Ethernet connectivity providers or cloud exchange providers 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 needs 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](#)

- High availability and failover with ExpressRoute
- 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

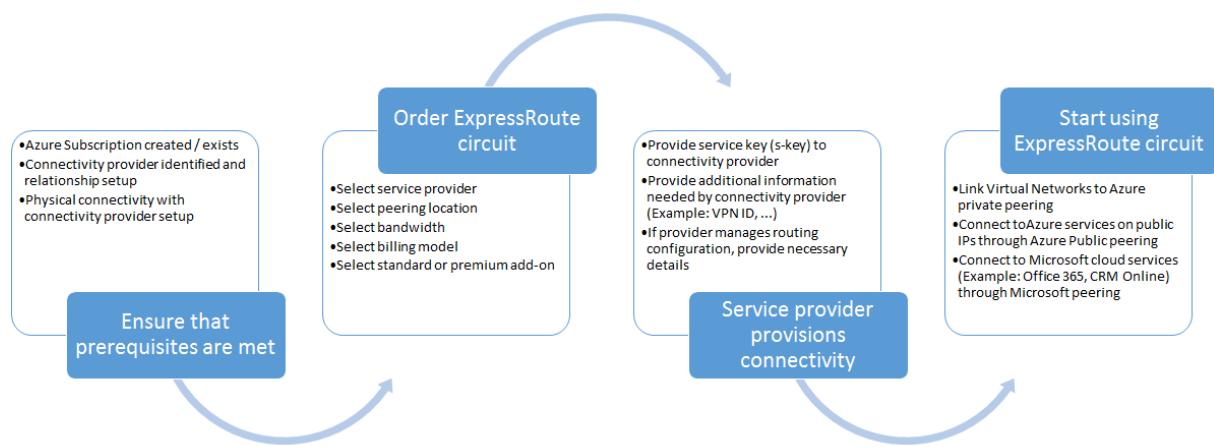
Next steps

- 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 workflows for circuit provisioning and circuit states

1/10/2020 • 3 minutes to read • [Edit Online](#)

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

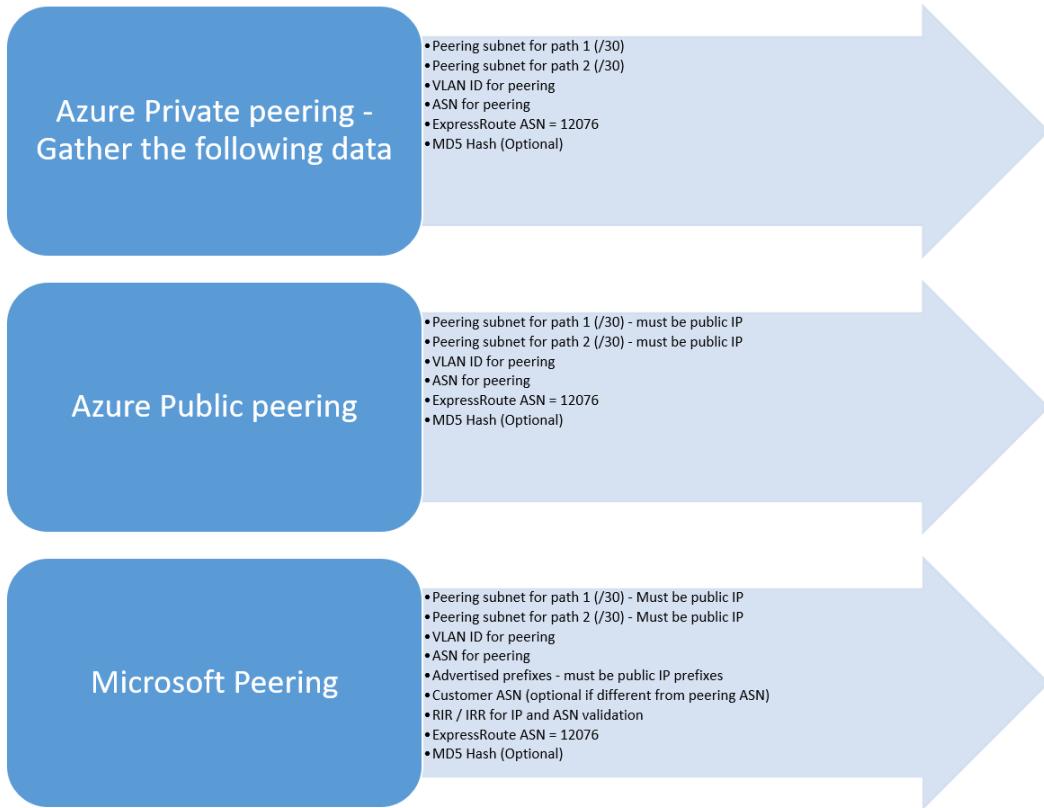


The following figure and corresponding steps outline the tasks to provision an ExpressRoute circuit end-to-end.

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 configuration, they will configure routing for your circuit. If your connectivity provider only offers Layer 2 services, you must configure routing per the guidelines described in the [routing requirements](#) and [routing configuration](#) pages.
 - Enable Azure private peering - Enable this peering to connect to VMs / cloud services deployed within virtual networks.
 - Enable Microsoft peering - Enable this to access Microsoft online services, such as Office 365. All Azure PaaS services are accessible through Microsoft peering.

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 a Provisioned state in order configure peering.

Possible states of an ExpressRoute circuit

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

At creation time

The ExpressRoute circuit will report the following states at resource creation.

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

When connectivity provider is in the process of provisioning the circuit

The ExpressRoute circuit will report the following states while the connectivity provider is working to provision the circuit.

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

When connectivity provider has completed the provisioning process

The ExpressRoute circuit will report the following states once the connectivity provider has successfully provisioned the circuit.

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

When connectivity provider is deprovisioning the circuit

If the ExpressRoute circuit needs to be deprovisioned, the circuit will report the following states once 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

A circuit cannot be deleted when the ServiceProviderProvisioningState is Provisioning or Provisioned. The connectivity provider needs to deprovision the circuit before it can be deleted. Microsoft will continue to bill the circuit until the ExpressRoute circuit resource is deleted in Azure.

Routing session configuration state

The BGP provisioning state reports if the BGP session has been enabled on the Microsoft edge. The state must be enabled to use private or Microsoft 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 the *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 need to 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 routing requirements

2/5/2020 • 12 minutes to read • [Edit Online](#)

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 (for example, 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 is split into two /30 subnets.
 - The first /30 subnet is used for the primary link and the second /30 subnet is 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 uses 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 is split into two /30 subnets. In the following example, notice how the a.b.c.d/29 subnet is used:

- a.b.c.d/29 is split to a.b.c.d/30 and a.b.c.d+4/30 and passed down to Microsoft through the provisioning APIs.
 - You 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 MSEEE.
 - You 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 MSEEE.

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 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.

- The IPs listed in the portal for Advertised Public Prefixes for Microsoft Peering will create ACLs for the Microsoft core routers to allow inbound traffic from these IPs.
- You must use a unique /29 (IPv4) or /125 (IPv6) subnet or two /30 (IPv4) or /126 (IPv6) 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 is split into two /30 subnets.
- The first /30 subnet is 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 uses the second IP address of the /30 subnet to set up a BGP session.
- If a /125 subnet is used, it is split into two /126 subnets.
- The first /126 subnet is used for the primary link and the second /126 subnet will be used for the secondary link.
- For each of the /126 subnets, you must use the first IP address of the /126 subnet on your router. Microsoft uses the second IP address of the /126 subnet to set up a BGP session.
- You must set up both BGP sessions for our [availability SLA](#) to be valid.

IP addresses used for Azure public peering

NOTE

Azure public peering is not available for new circuits.

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 is split into two /30 subnets.
 - The first /30 subnet is used for the primary link and the second /30 subnet is 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 uses 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.

Microsoft peering

The Microsoft peering path lets you connect to Microsoft cloud services. The list of services includes Office

365 services, such as Exchange Online, SharePoint Online, Skype for Business, and Microsoft Teams. 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 following registries:

- [ARIN](#)
- [APNIC](#)
- [AFRINIC](#)
- [LACNIC](#)
- [RIPENCC](#)
- [RADB](#)
- [ALTDB](#)

If your prefixes and AS number are not assigned to you in the preceding registries, you need to open a support case for manual validation of your prefixes and ASN. Support requires documentation, such as a Letter of Authorization, that proves you are allowed to use the resources.

A Private AS Number is allowed with Microsoft Peering, but will also require manual validation. In addition, we remove private AS numbers in the AS PATH for the received prefixes. As a result, you can't append private AS numbers in the AS PATH to [influence routing for Microsoft Peering](#).

IMPORTANT

Do not advertise the same public IP route to the public Internet and over ExpressRoute. To reduce the risk of incorrect configuration causing asymmetric routing, we strongly recommend that the [NAT IP addresses](#) advertised to Microsoft over ExpressRoute be from a range that is not advertised to the internet at all. If this is not possible to achieve, it is essential to ensure you advertise a more specific range over ExpressRoute than the one on the Internet connection. Besides the public route for NAT, you can also advertise over ExpressRoute the Public IP addresses used by the servers in your on-premises network that communicate with Office 365 endpoints within Microsoft.

Public peering (deprecated - not available for new circuits)

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 [ExpressRoute 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.

IMPORTANT

All Azure PaaS services are accessible through Microsoft peering.

A Private AS Number is allowed with public peering.

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 uses 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 to 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 is 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 Microsoft peering and public peering paths. This means you will have multiple paths from your network into Microsoft. This can potentially cause suboptimal routing decisions to be made within your network. As a result, you may experience suboptimal 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	REGIONAL BGP COMMUNITY	STORAGE BGP COMMUNITY	SQL BGP COMMUNITY	COSMOS DB BGP COMMUNITY
North America				
East US	12076:51004	12076:52004	12076:53004	12076:54004
East US 2	12076:51005	12076:52005	12076:53005	12076:54005
West US	12076:51006	12076:52006	12076:53006	12076:54006
West US 2	12076:51026	12076:52026	12076:53026	12076:54026
West Central US	12076:51027	12076:52027	12076:53027	12076:54027
North Central US	12076:51007	12076:52007	12076:53007	12076:54007
South Central US	12076:51008	12076:52008	12076:53008	12076:54008
Central US	12076:51009	12076:52009	12076:53009	12076:54009
Canada Central	12076:51020	12076:52020	12076:53020	12076:54020
Canada East	12076:51021	12076:52021	12076:53021	12076:54021
South America				
Brazil South	12076:51014	12076:52014	12076:53014	12076:54014
Europe				
North Europe	12076:51003	12076:52003	12076:53003	12076:54003
West Europe	12076:51002	12076:52002	12076:53002	12076:54002
UK South	12076:51024	12076:52024	12076:53024	12076:54024
UK West	12076:51025	12076:52025	12076:53025	12076:54025
France Central	12076:51030	12076:52030	12076:53030	12076:54030
France South	12076:51031	12076:52031	12076:53031	12076:54031
Switzerland North	12076:51038	12076:52038	12076:53038	12076:54038
Switzerland West	12076:51039	12076:52039	12076:53039	12076:54039

MICROSOFT AZURE REGION	REGIONAL BGP COMMUNITY	STORAGE BGP COMMUNITY	SQL BGP COMMUNITY	COSMOS DB BGP COMMUNITY
Germany North	12076:51040	12076:52040	12076:53040	12076:54040
Germany West Central	12076:51041	12076:52041	12076:53041	12076:54041
Norway East	12076:51042	12076:52042	12076:53042	12076:54042
Norway West	12076:51043	12076:52043	12076:53043	12076:54043
Asia Pacific				
East Asia	12076:51010	12076:52010	12076:53010	12076:54010
Southeast Asia	12076:51011	12076:52011	12076:53011	12076:54011
Japan				
Japan East	12076:51012	12076:52012	12076:53012	12076:54012
Japan West	12076:51013	12076:52013	12076:53013	12076:54013
Australia				
Australia East	12076:51015	12076:52015	12076:53015	12076:54015
Australia Southeast	12076:51016	12076:52016	12076:53016	12076:54016
Australia Government				
Australia Central	12076:51032	12076:52032	12076:53032	12076:54032
Australia Central 2	12076:51033	12076:52033	12076:53033	12076:54033
India				
India South	12076:51019	12076:52019	12076:53019	12076:54019
India West	12076:51018	12076:52018	12076:53018	12076:54018
India Central	12076:51017	12076:52017	12076:53017	12076:54017
Korea				
Korea South	12076:51028	12076:52028	12076:53028	12076:54028
Korea Central	12076:51029	12076:52029	12076:53029	12076:54029
South Africa				

MICROSOFT AZURE REGION	REGIONAL BGP COMMUNITY	STORAGE BGP COMMUNITY	SQL BGP COMMUNITY	COSMOS DB BGP COMMUNITY
South Africa North	12076:51034	12076:52034	12076:53034	12076:54034
South Africa West	12076:51035	12076:52035	12076:53035	12076:54035
UAE				
UAE North	12076:51036	12076:52036	12076:53036	12076:54036
UAE Central	12076:51037	12076:52037	12076:53037	12076:54037

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

IMPORTANT

Global prefixes are tagged with an appropriate community value.

Service to BGP community value

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. You can run the 'Get-AzBgpServiceCommunity' cmdlet for a full list of the latest values.

SERVICE	BGP COMMUNITY VALUE
Exchange Online**	12076:5010
SharePoint Online**	12076:5020
Skype For Business Online**	12076:5030
CRM Online***	12076:5040
Azure Global Services*	12076:5050
Azure Active Directory	12076:5060
Other Office 365 Online services**	12076:5100

*Azure Global Services includes only Azure DevOps at this time.

** Authorization required from Microsoft, refer [Configure route filters for Microsoft Peering](#)

*** CRM Online supports Dynamics v8.2 and below. For higher versions, select the regional community for your Dynamics deployments.

NOTE

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

BGP Community support in National Clouds

NATIONAL CLOUDS AZURE REGION	BGP COMMUNITY VALUE
US Government	
US Gov Arizona	12076:51106
US Gov Iowa	12076:51109
US Gov Virginia	12076:51105
US Gov Texas	12076:51108
US DoD Central	12076:51209
US DoD East	12076:51205
SERVICE IN NATIONAL CLOUDS	BGP COMMUNITY VALUE
US Government	
Exchange Online	12076:5110
SharePoint Online	12076:5120
Skype For Business Online	12076:5130
Other Office 365 Online services	12076:5200

Next steps

- Configure your ExpressRoute connection.
 - [Create and modify a circuit](#)
 - [Create and modify peering configuration](#)
 - [Link a VNet to an ExpressRoute circuit](#)

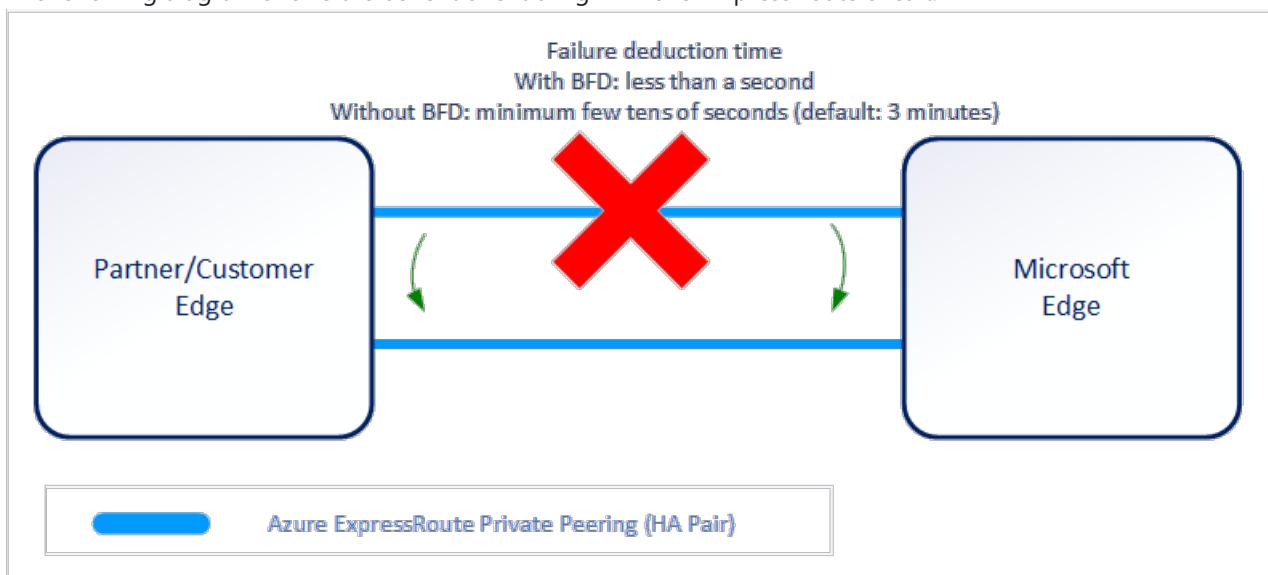
Configure BFD over ExpressRoute

11/14/2019 • 2 minutes to read • [Edit Online](#)

ExpressRoute supports Bidirectional Forwarding Detection (BFD) both over private and Microsoft peering. By enabling BFD over ExpressRoute, you can expedite link failure detection between Microsoft Enterprise edge (MSEE) devices and the routers on which you terminate the ExpressRoute circuit (CE/PE). You can terminate ExpressRoute over Customer Edge routing devices or Partner Edge routing devices (if you went with managed Layer 3 connection service). This document walks you through the need for BFD, and how to enable BFD over ExpressRoute.

Need for BFD

The following diagram shows the benefit of enabling BFD over ExpressRoute circuit:



You can enable ExpressRoute circuit either by Layer 2 connections or managed Layer 3 connections. In either case, if there are one or more Layer-2 devices in the ExpressRoute connection path, responsibility of detecting any link failures in the path lies with the overlying BGP.

On the MSEE devices, BGP keepalive and hold-time are typically configured as 60 and 180 seconds respectively. Therefore, following a link failure it would take up to three minutes to detect any link failure and switch traffic to alternate connection.

You can control the BGP timers by configuring lower BGP keepalive and hold-time on the customer edge peering device. If the BGP timers are mismatched between the two peering devices, the BGP session between the peers would use the lower timer value. The BGP keepalive can be set as low as three seconds, and the hold-time in the order of tens of seconds. However, setting BGP timers aggressively is less preferable because the protocol is process intensive.

In this scenario, BFD can help. BFD provides low-overhead link failure detection in a subsecond time interval.

Enabling BFD

BFD is configured by default under all the newly created ExpressRoute private peering interfaces on the MSEEs. Therefore, to enable BFD, you need to just configure BFD on your CEs/PEs (both on your primary and secondary devices). Configuring BFD is two-step process: you need to configure the BFD on the interface and then link it to the BGP session.

An example CE/PE (using Cisco IOS XE) configuration is shown below.

```
interface TenGigabitEthernet2/0/0.150
  description private peering to Azure
  encapsulation dot1Q 15 second-dot1q 150
  ip vrf forwarding 15
  ip address 192.168.15.17 255.255.255.252
  bfd interval 300 min_rx 300 multiplier 3

router bgp 65020
  address-family ipv4 vrf 15
    network 10.1.15.0 mask 255.255.255.128
    neighbor 192.168.15.18 remote-as 12076
    neighbor 192.168.15.18 fall-over bfd
    neighbor 192.168.15.18 activate
    neighbor 192.168.15.18 soft-reconfiguration inbound
  exit-address-family
```

NOTE

To enable BFD under an already existing private peering; you need to reset the peering. See [Reset ExpressRoute peerings](#)

BFD Timer Negotiation

Between BFD peers, the slower of the two peers determine the transmission rate. MSEEs BFD transmission/receive intervals are set to 300 milliseconds. In certain scenarios, the interval may be set at a higher value of 750 milliseconds. By configuring higher values, you can force these intervals to be longer; but, not shorter.

NOTE

If you have configured Geo-redundant ExpressRoute circuits or use Site-to-Site IPSec VPN connectivity as backup; enabling BFD would help failover quicker following an ExpressRoute connectivity failure.

Next Steps

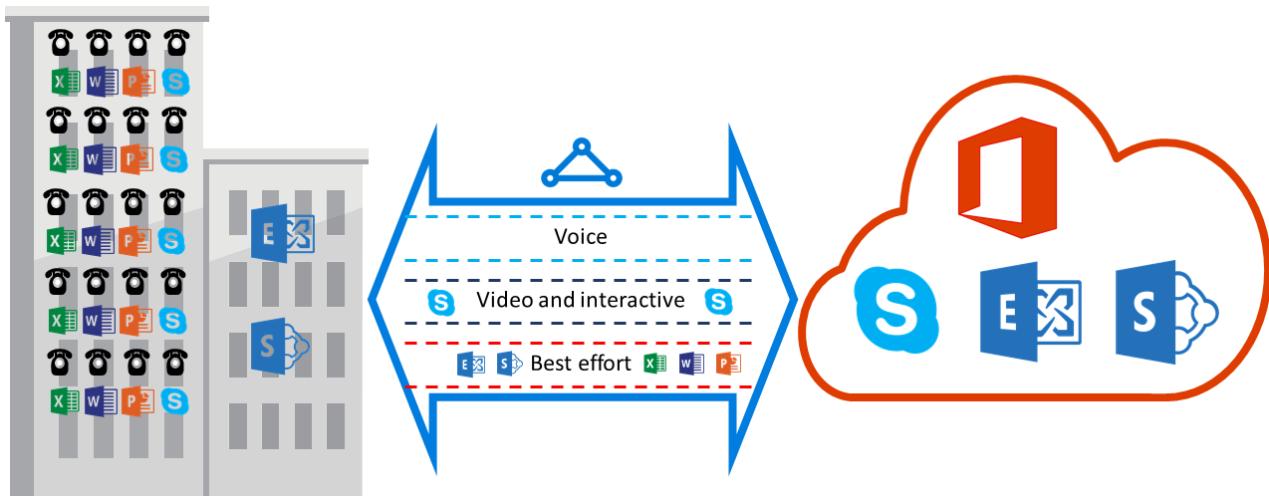
For more information or help, check out the following links:

- [Create and modify an ExpressRoute circuit](#)
- [Create and modify routing for an ExpressRoute circuit](#)

ExpressRoute QoS requirements

11/14/2019 • 2 minutes to read • [Edit Online](#)

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 Microsoft Teams and Skype for Business. Refer to [Managing QoS for Skype for Business](#) for more information.

TRAFFIC CLASS	TREATMENT (DSCP MARKING)	MICROSOFT TEAMS AND SKYPE FOR BUSINESS WORKLOADS
Voice	EF (46)	Skype / Microsoft Teams / Lync voice
Interactive	AF41 (34)	Video, VBSS
	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

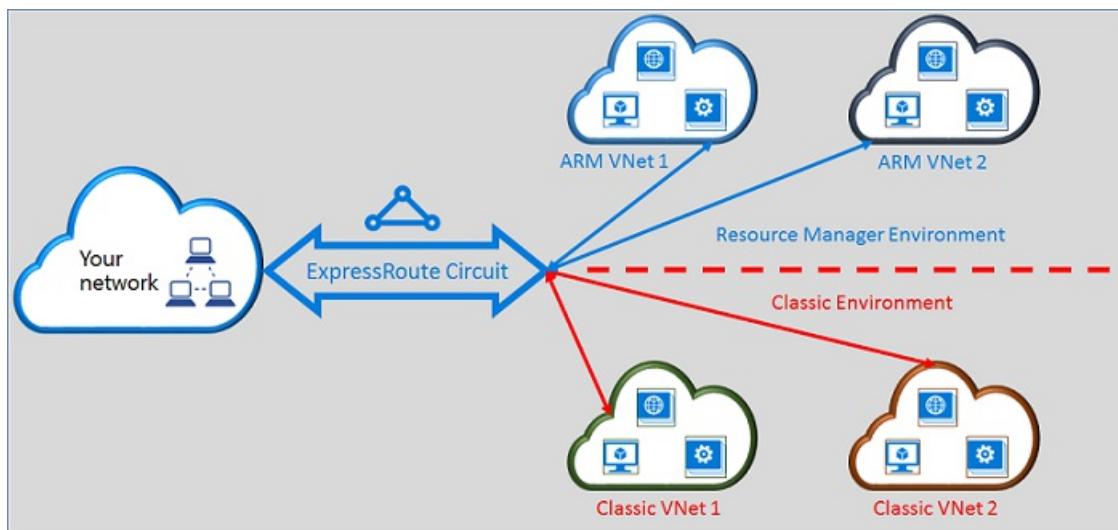
- Refer to the requirements for [Routing](#) and [NAT](#).
- See the following links to configure your ExpressRoute connection.
 - [Create an ExpressRoute circuit](#)
 - [Configure routing](#)
 - [Link a VNet to an ExpressRoute circuit](#)

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

1/3/2020 • 6 minutes to read • [Edit Online](#)

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.

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

However, when **allowClassicOperations** is set to TRUE, you cannot perform the following classic operations:

- 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.
- After you move an ExpressRoute circuit from the classic model to the Azure Resource Manager model, you can [migrate the virtual networks linked to the ExpressRoute circuit](#).

What's not supported

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

- 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

- Migrate the virtual networks linked to the ExpressRoute circuit from the classic model to the Azure Resource Manager model
- 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 ExpressRoute FastPath

12/16/2019 • 2 minutes to read • [Edit Online](#)

ExpressRoute virtual network gateway is designed to exchange network routes and route network traffic. FastPath is designed to improve the data path performance between your on-premises network and your virtual network. When enabled, FastPath sends network traffic directly to virtual machines in the virtual network, bypassing the gateway.

Requirements

Circuits

FastPath is available on all ExpressRoute circuits.

Gateways

FastPath still requires a virtual network gateway to be created to exchange routes between virtual network and on-premises network. For more information about virtual network gateways and ExpressRoute, including performance information and gateway SKUs, see [ExpressRoute virtual network gateways](#).

To configure FastPath, the virtual network gateway must be either:

- Ultra Performance
- ErGw3AZ

Estimated performances by gateway SKU

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

	MEGABITS PER SECOND	PACKETS PER SECOND	CONNECTIONS PER SECOND	VPN GATEWAY AND EXPRESSROUTE COEXIST	FASTPATH
Basic SKU (deprecated)	500	Unknown	Unknown	No	No
Standard SKU/ErGw1AZ	1,000	100,000	7,000	Yes	No
High Performance SKU/ErGw2AZ	2,000	250,000	14,000	Yes	No
Ultra Performance SKU/ErGw3AZ	10,000	1,000,000	28,000	Yes	Yes

IMPORTANT

Application performance depends on multiple factors, such as the end-to-end latency, and the number of traffic flows the application opens. The numbers in the table represent the upper limit that the application can theoretically achieve in an ideal environment.

Supported features

While FastPath supports most configurations, it does not support the following features:

- UDR on the gateway subnet: If you apply a UDR to the gateway subnet of your virtual network, the network traffic from your on-premises network will continue to be sent to the virtual network gateway.
- VNet Peering: If you have other virtual networks peered with the one that is connected to ExpressRoute, the network traffic from your on-premises network to the other virtual networks (i.e. the so-called "Spoke" VNets) will continue to be sent to the virtual network gateway. The workaround is to connect all the virtual networks to the ExpressRoute circuit directly.
- Basic Load Balancer: If you deploy a Basic internal load balancer in your virtual network or the Azure PaaS service you deploy in your virtual network uses a Basic internal load balancer, the network traffic from your on-premises network to the virtual IPs hosted on the Basic load balancer will be sent to the virtual network gateway. The solution is to upgrade the Basic load balancer to a [Standard load balancer](#).
- Private Link: If you connect to a [private endpoint](#) in your virtual network from your on-premises network, the connection will go through the virtual network gateway.

Next steps

To enable FastPath, see [Link a virtual network to ExpressRoute](#).

About ExpressRoute Direct

11/14/2019 • 3 minutes to read • [Edit Online](#)

ExpressRoute Direct gives you the ability to connect directly into Microsoft's global network at peering locations strategically distributed across the world. ExpressRoute Direct provides dual 100 Gbps or 10 Gbps connectivity, which supports Active/Active connectivity at scale.

Key features that ExpressRoute Direct provides include, but aren't limited to:

- Massive Data Ingestion into services like Storage and Cosmos DB
- Physical isolation for industries that are regulated and require dedicated and isolated connectivity like: Banking, Government, and Retail
- Granular control of circuit distribution based on business unit

Onboard to ExpressRoute Direct

Before using ExpressRoute Direct, you must first enroll your subscription. To enroll, send an Email to ExpressRouteDirect@microsoft.com with your subscription ID, including the following details:

- Scenarios you're looking to accomplish with **ExpressRoute Direct**
- Location preferences - see [Partners and peering locations](#) for a complete list of all locations
- Timeline for implementation
- Any other questions

ExpressRoute using a service provider and ExpressRoute Direct

EXPRESSROUTE USING A SERVICE PROVIDER	EXPRESSROUTE DIRECT
Utilizes service providers to enable fast onboarding and connectivity into existing infrastructure	Requires 100 Gbps/10 Gbps infrastructure and full management of all layers
Integrates with hundreds of providers including Ethernet and MPLS	Direct/Dedicated capacity for regulated industries and massive data ingestion
Circuits SKUs from 50 Mbps to 10 Gbps	<p>Customer may select a combination of the following circuit SKUs on 100 Gbps ExpressRoute Direct:</p> <ul style="list-style-type: none">• 5 Gbps• 10 Gbps• 40 Gbps• 100 Gbps <p>Customer may select a combination of the following circuit SKUs on 10 Gbps ExpressRoute Direct:</p> <ul style="list-style-type: none">• 1 Gbps• 2 Gbps• 5 Gbps• 10 Gbps
Optimized for single tenant	Optimized for single tenant with multiple business units and multiple work environments

ExpressRoute Direct circuits

Microsoft Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, and Office 365.

Each peering location has access to Microsoft's global network and can access any region in a geopolitical zone by default and can access all global regions with a premium circuit.

The functionality in most scenarios is equivalent to circuits that utilize an ExpressRoute service provider to operate. To support further granularity and new capabilities offered using ExpressRoute Direct, there are certain key capabilities that exist on ExpressRoute Direct Circuits.

Circuit SKUs

ExpressRoute Direct supports massive data ingestion scenarios into Azure storage and other big data services. ExpressRoute circuits on 100 Gbps ExpressRoute Direct now also support **40 Gbps** and **100 Gbps** circuit SKUs. The physical port pairs are **100 or 10 Gbps** only and can have multiple virtual circuits. Circuit sizes:

100 Gbps ExpressRoute Direct	10 Gbps ExpressRoute Direct
Subscribed Bandwidth: 200 Gbps	Subscribed Bandwidth: 20 Gbps
<ul style="list-style-type: none">• 5 Gbps• 10 Gbps• 40 Gbps• 100 Gbps	<ul style="list-style-type: none">• 1 Gbps• 2 Gbps• 5 Gbps• 10 Gbps

Technical Requirements

- Microsoft Enterprise Edge Router (MSEE) Interfaces:
 - Dual 10 or 100 Gigabit Ethernet ports only across router pair
 - Single Mode LR Fiber connectivity
 - IPv4 and IPv6
 - IP MTU 1500 bytes
- Switch/Router Layer 2/Layer 3 Connectivity:
 - Must support 1 802.1Q (Dot1Q) tag or two Tag 802.1Q (QinQ) tag encapsulation
 - Ethertype = 0x8100
 - Must add the outer VLAN tag (STAG) based on the VLAN ID specified by Microsoft - *applicable only on QinQ*
 - Must support multiple BGP sessions (VLANs) per port and device
 - IPv4 and IPv6 connectivity. *For IPv6 no additional sub-interface will be created. IPv6 address will be added to existing sub-interface.*
 - Optional: [Bidirectional Forwarding Detection \(BFD\)](#) support, which is configured by default on all Private Peerings on ExpressRoute circuits

VLAN Tagging

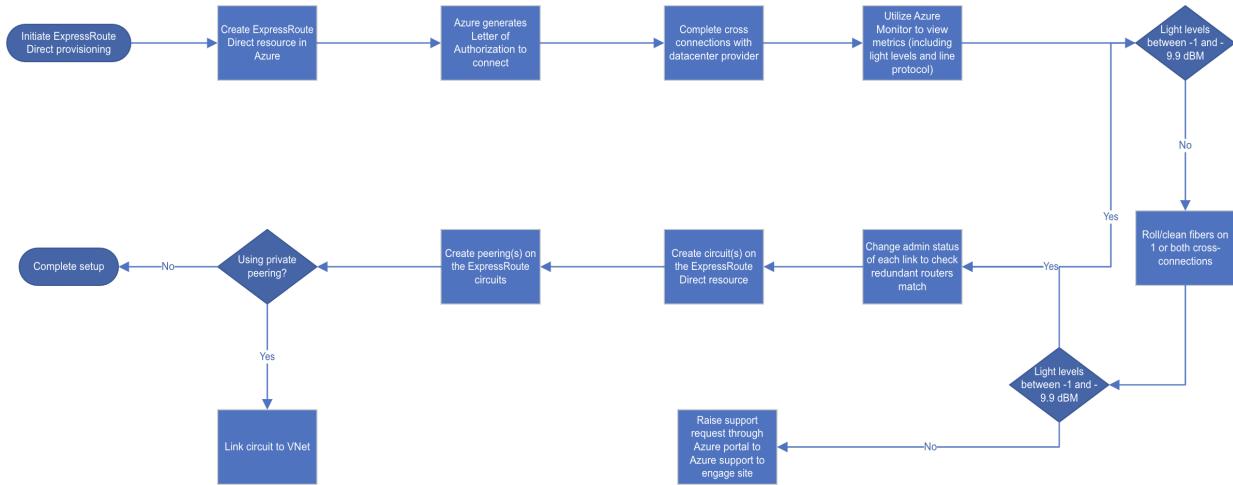
ExpressRoute Direct supports both QinQ and Dot1Q VLAN tagging.

- **QinQ VLAN Tagging** allows for isolated routing domains on a per ExpressRoute circuit basis. Azure

dynamically allocates an S-Tag at circuit creation and cannot be changed. Each peering on the circuit (Private and Microsoft) will utilize a unique C-Tag as the VLAN. The C-Tag is not required to be unique across circuits on the ExpressRoute Direct ports.

- **Dot1Q VLAN Tagging** allows for a single tagged VLAN on a per ExpressRoute Direct port pair basis. A C-Tag used on a peering must be unique across all circuits and peerings on the ExpressRoute Direct port pair.

Workflow



SLA

ExpressRoute Direct provides the same enterprise-grade SLA with Active/Active redundant connections into the Microsoft Global Network. ExpressRoute infrastructure is redundant and connectivity into the Microsoft Global Network is redundant and diverse and scales accordingly with customer requirements.

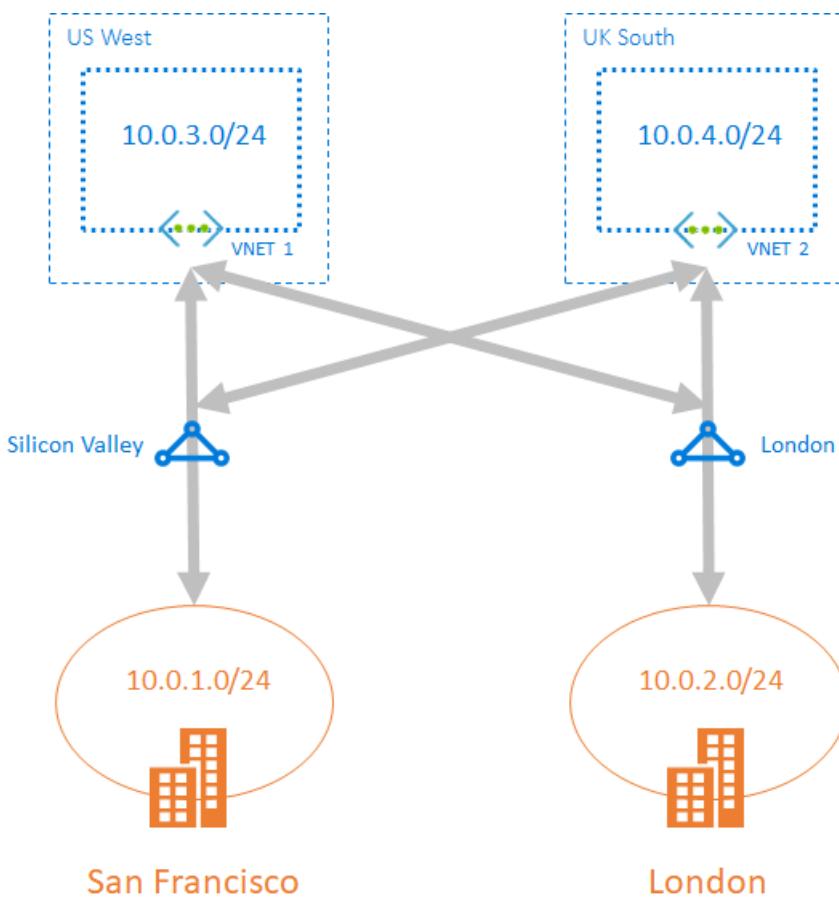
Next steps

[Configure ExpressRoute Direct](#)

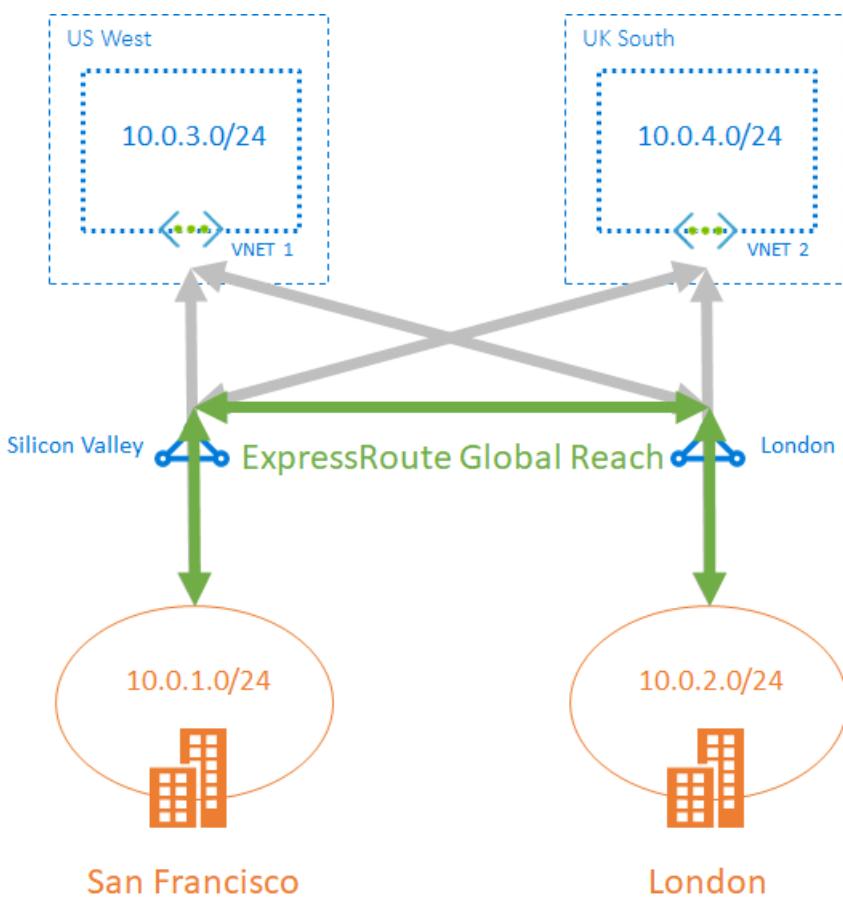
ExpressRoute Global Reach

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

ExpressRoute is a private and resilient way to connect your on-premises networks to Microsoft Cloud. You can access many Microsoft cloud services such as Azure, and Office 365 from your private data center or your corporate network. For example, you may have a branch office in San Francisco with an ExpressRoute circuit in Silicon Valley and another branch office in London with an ExpressRoute circuit in the same city. Both branch offices can have high speed connectivity to Azure resources in US West and UK South. However, the branch offices cannot exchange data directly with each other. In other words, 10.0.1.0/24 can send data to 10.0.3.0/24 and 10.0.4.0/24, but NOT to 10.0.2.0/24.

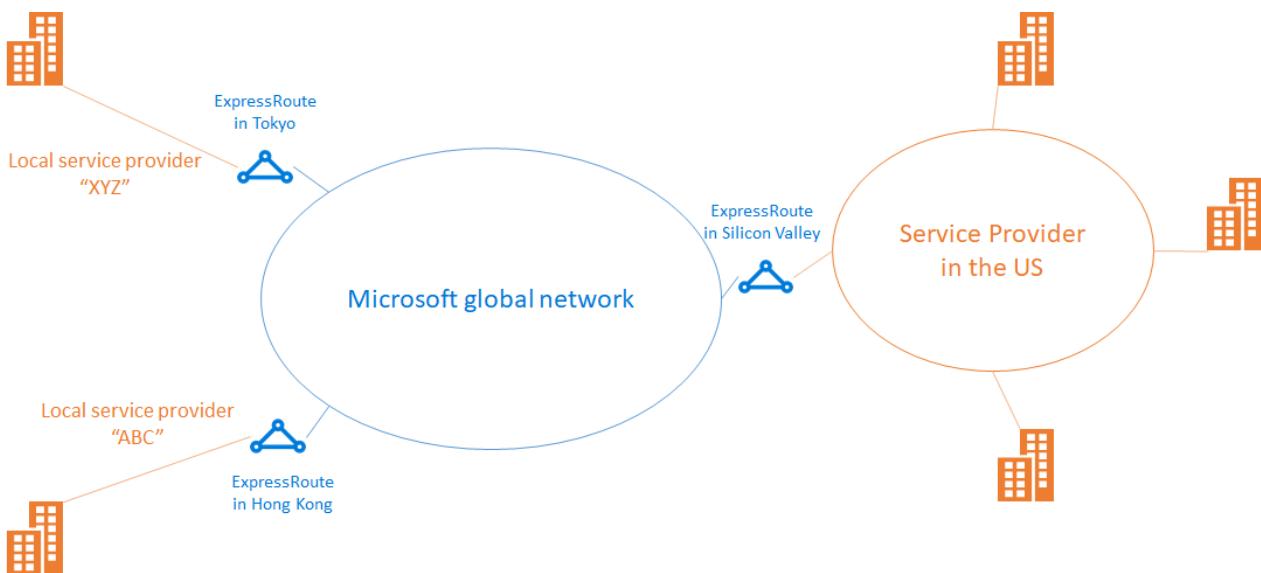


With **ExpressRoute Global Reach**, you can link ExpressRoute circuits together to make a private network between your on-premises networks. In the above example, with the addition of ExpressRoute Global Reach, your San Francisco office (10.0.1.0/24) can directly exchange data with your London office (10.0.2.0/24) through the existing ExpressRoute circuits and via Microsoft's global network.



Use case

ExpressRoute Global Reach is designed to complement your service provider's WAN implementation and connect your branch offices across the world. For example, if your service provider primarily operates in the United States and has linked all of your branches in the U.S., but the service provider doesn't operate in Japan and Hong Kong, with ExpressRoute Global Reach you can work with a local service provider and Microsoft will connect your branches there to the ones in the U.S. using ExpressRoute and our global network.



Availability

ExpressRoute Global Reach currently is supported in the following places.

- Australia
- Canada

- France
- Germany
- Hong Kong SAR
- Ireland
- Japan
- Korea
- Netherlands
- New Zealand
- Singapore
- Switzerland
- United Kingdom
- United States

Your ExpressRoute circuits must be created at the [ExpressRoute peering locations](#) in the above countries or region. To enable ExpressRoute Global Reach between [different geopolitical regions](#), your circuits must be Premium SKU.

Next steps

1. [Learn more about ExpressRoute Global Reach](#)
2. [How to enable ExpressRoute Global Reach](#)
3. [Link ExpressRoute circuit to Azure virtual network](#)

ExpressRoute encryption

12/19/2019 • 3 minutes to read • [Edit Online](#)

ExpressRoute supports a couple of encryption technologies to ensure confidentiality and integrity of the data traversing between your network and Microsoft's network.

Point-to-point encryption by MACsec FAQ

MACsec is an [IEEE standard](#). It encrypts data at the Media Access control (MAC) level or Network Layer 2. You can use MACsec to encrypt the physical links between your network devices and Microsoft's network devices when you connect to Microsoft via [ExpressRoute Direct](#). MACsec is disabled on ExpressRoute Direct ports by default. You bring your own MACsec key for encryption and store it in [Azure Key Vault](#). You decide when to rotate the key. See other FAQs below.

Can I enable MACsec on my ExpressRoute circuit provisioned by an ExpressRoute provider?

No. MACsec encrypts all traffic on a physical link with a key owned by one entity (i.e. customer). Therefore, it's available on ExpressRoute Direct only.

Can I encrypt some of the ExpressRoute circuits on my ExpressRoute Direct ports and leave other circuits on the same ports unencrypted?

No. Once MACsec is enabled all network control traffic, for example, the BGP data traffic, and customer data traffic are encrypted.

When I enable/disable MACsec or update MACsec key will my on-premises network lose connectivity to Microsoft over ExpressRoute?

Yes. For the MACsec configuration, we support the pre-shared key mode only. It means you need to update the key on both your devices and on Microsoft's (via our API). This change is not atomic, so you'll lose connectivity when there's a key mismatch between the two sides. We strongly recommend that you schedule a maintenance window for the configuration change. To minimize the downtime, we suggest you update the configuration on one link of ExpressRoute Direct at a time after you switch your network traffic to the other link.

Will traffic continue to flow if there's a mismatch in MACsec key between my devices and Microsoft's?

No. If MACsec is configured and a key mismatch occurs, you lose connectivity to Microsoft. In other words, we won't fall back to an unencrypted connection, exposing your data.

Will enabling MACsec on ExpressRoute Direct degrade network performance?

MACsec encryption and decryption occurs in hardware on the routers we use. There's no performance impact on our side. However, you should check with the network vendor for the devices you use and see if MACsec has any performance implication.

which cipher suites are supported for encryption?

We support AES128 and AES256.

End-to-end encryption by IPsec FAQ

IPsec is an [IETF standard](#). It encrypts data at the Internet Protocol (IP) level or Network Layer 3. You can use IPsec to encrypt an end-to-end connection between your on-premises network and your virtual network (VNET) on Azure. See other FAQs below.

Can I enable IPsec in addition to MACsec on my ExpressRoute Direct ports?

Yes. MACsec secures the physical connections between you and Microsoft. IPsec secures the end-to-end

connection between you and your virtual networks on Azure. You can enable them independently.

Can I use Azure VPN gateway to set up the IPsec tunnel between my on-premises network and my Azure virtual network?

Yes. You can set up this IPsec tunnel over Microsoft Peering of your ExpressRoute circuit. Follow our [configuration guide](#).

Can I use Azure VPN gateway to set up the IPsec tunnel over Azure Private Peering?

If you adopt Azure Virtual WAN you can follow [these steps](#) to encrypt the end-to-end connection. If you have regular Azure VNET you can deploy a third-party VPN gateway in your VNET and establish an IPsec tunnel between it and your on-premises VPN gateway.

What is the throughput I will get after enabling IPsec on my ExpressRoute connection?

If Azure VPN gateway is used, check the [performance numbers here](#). If a third-party VPN gateway is used, check with the vendor for the performance numbers.

Next steps

See [Configure MACsec](#) for more information about the MACsec configuration.

See [Configure IPsec](#) for more information about the IPsec configuration.

Connecting Azure with public clouds

11/8/2019 • 3 minutes to read • [Edit Online](#)

Many enterprises are pursuing a multi-cloud strategy because of business and technical goals. These include cost, flexibility, feature availability, redundancy, data sovereignty etc. This strategy helps them leverage best of both clouds.

This approach also poses challenges for the enterprise in terms of network and application architecture. Some of these challenges are latency and data throughput. To address these challenges customers are looking to connect to multiple clouds directly. Some service providers provide a solution to connect multiple cloud providers for the customers. In other cases, customer can deploy their own router to connect multiple public clouds.

Connectivity via ExpressRoute

ExpressRoute lets customers extend their on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, customers can establish connections to Microsoft cloud services.

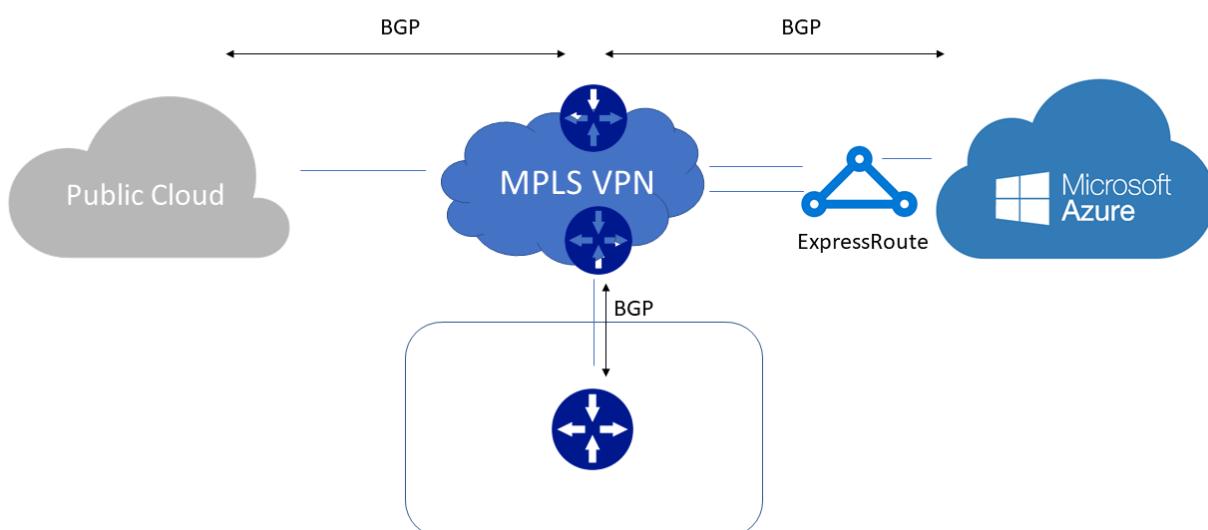
There are three ways to connect via ExpressRoute.

1. Layer3 provider
2. Layer2 provider
3. Direct connection

Layer3 Provider

Layer3 providers are commonly known as IP VPN or MPLS VPN providers. Customers leverage these providers for multipoint connectivity between their data centers, branches and the cloud. Customers connect to the L3 provider via BGP or via static default route. Service provider advertises routes between the customer sites, datacenters and public cloud.

When connecting through Layer3 provider, Microsoft will advertise customer VNET routes to the service provider over BGP. The provider can have two different implementations.



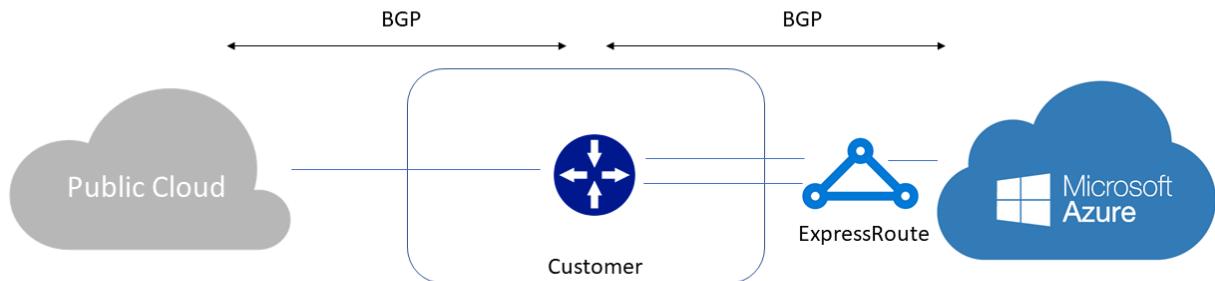
Provider may be landing each cloud provider in a separate VRF, if traffic from all the cloud providers will reach at customer router. If customer is running BGP with service provider, then these routes will be re-advertised to other cloud providers by default.

If service provider is landing all the cloud providers in the same VRF, then routes will be advertised to other cloud providers from the service provider directly. This is assuming standard BGP operation where eBGP routes are advertised to other eBGP neighbors by default.

Each public cloud has different prefix limit so while distributing the routes service provider should take caution in distributing the routes.

Layer2 Provider and Direct connection

Although physical connectivity in both models is different, but at layer3 BGP is established directly between MSEE and the customer router. For ExpressRoute Direct customer connects to MSEE directly. While in case of Layer2, service provider extends VLAN from customer premises to the cloud. Customers run BGP on top of layer2 network to connect their DCs to the cloud.



In both cases, customer will have point-to-point connections to each of the public clouds. Customer will establish separate BGP connection to each public cloud. Routes received by one cloud provider will be advertised to other cloud provider by default. Each cloud provider has different prefix limit so while advertising the routes customer should take care of these limits. Customer can use usual BGP knobs with Microsoft while advertising routes from other public clouds.

Direct connection with ExpressRoute

Customers can choose to connect ExpressRoute directly to the cloud provider's direct connectivity offering. Two cloud providers will be connected back to back and BGP will be established directly between their routers. This type of connection is available with Oracle today.

Site-to-site VPN

Customers can leverage Internet to connect their instances in Azure with other public clouds. Almost all the cloud providers offer site-to-site VPN capabilities. However, there could be incompatibilities because of lack of certain variants. For example, some cloud providers only support IKEv1 so there is a VPN termination endpoint required in that cloud. For those cloud providers supporting IKEv2 a direct tunnel can be established between VPN gateways at both cloud providers.

Site-to-site VPN is not considered a high throughput and low latency solution. However, it can be used as a backup to physical connectivity.

Next steps

See [ExpressRoute FAQ](#) for any further questions on ExpressRoute and virtual network connectivity.

See [Set up direct connection between Azure and Oracle Cloud](#) for connectivity between Azure and Oracle

Interoperability in Azure back-end connectivity features: Test setup

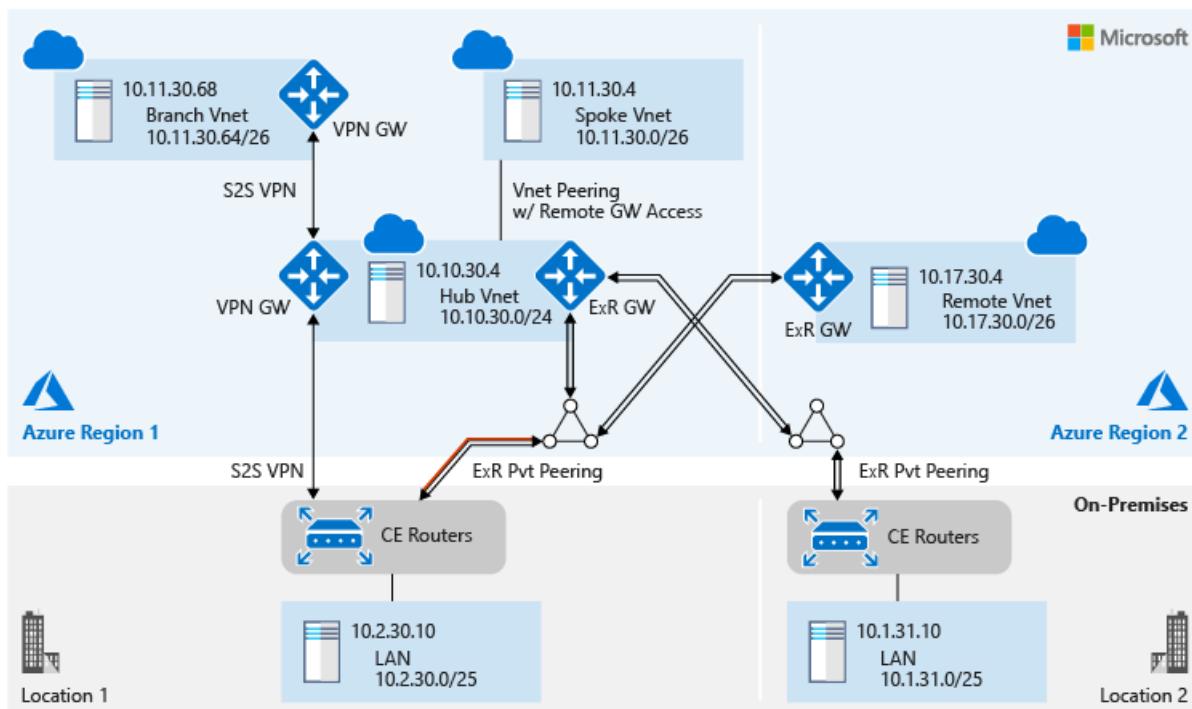
12/5/2019 • 4 minutes to read • [Edit Online](#)

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

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

Test setup

The following figure illustrates the test setup:



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

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

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

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

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

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

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

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

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

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

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

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

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

Branch VNet connectivity by using site-to-site VPN

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

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

Next steps

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

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

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

See the [ExpressRoute FAQ](#) to:

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

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

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

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

Spoke VNet connectivity by using VNet peering

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

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

General Information:

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

Peer details:

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

Configuration:

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

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

VNet01-Spoke01-Peering

VNet01

Save Discard Delete

Name
VNet01-Spoke01-Peering

Peering status
Connected

Provisioning state
Succeeded

Peer details

Address space
10.11.30.0/26

Virtual network
Spoke01-VNet

Configuration

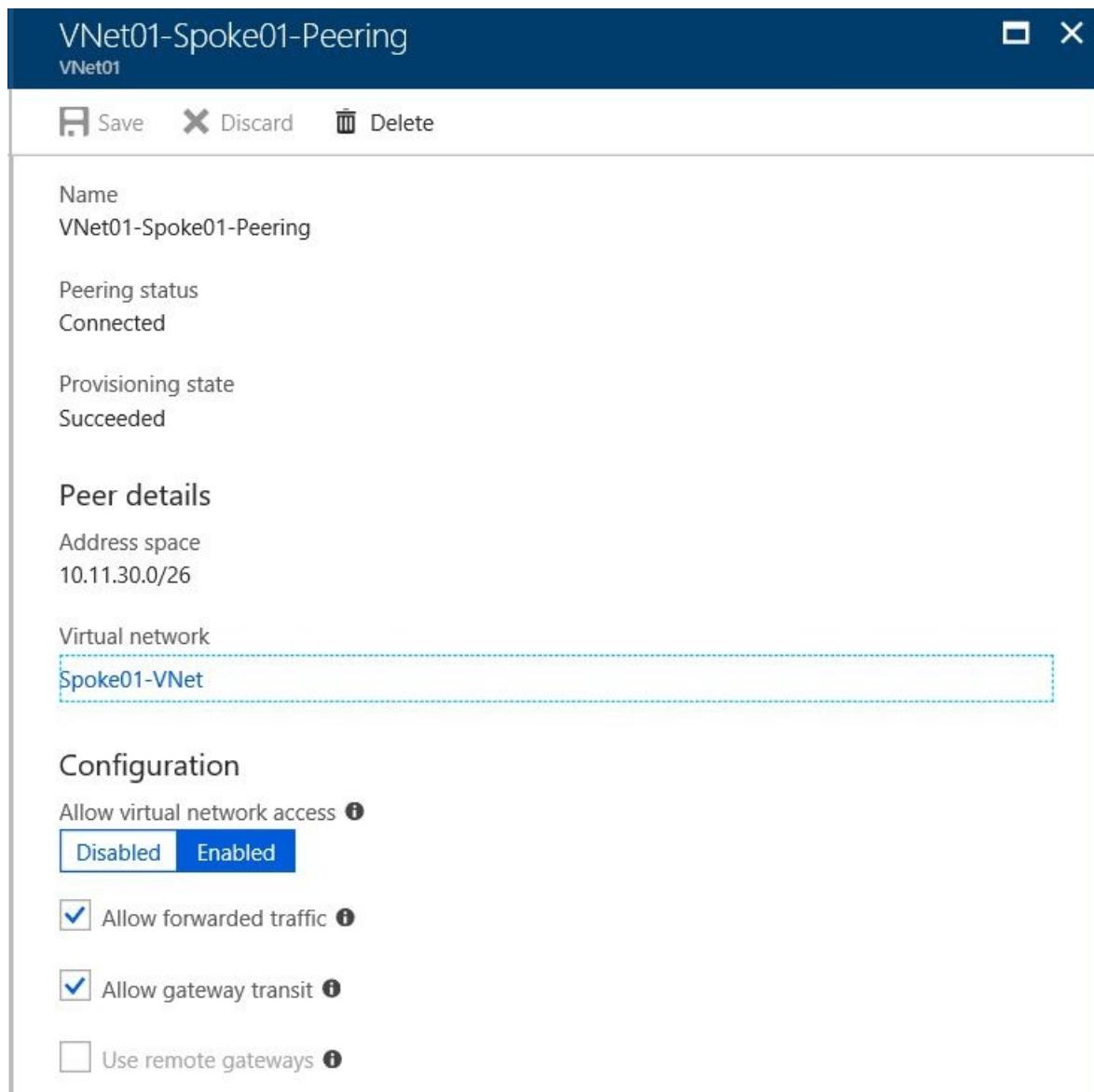
Allow virtual network access ⓘ

Disabled Enabled

Allow forwarded traffic ⓘ

Allow gateway transit ⓘ

Use remote gateways ⓘ



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

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

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

ExpressRoute 1 configuration details

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

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

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

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

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

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

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

Site-to-site VPN configuration details

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

```

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

```

On-premises Location 2 connectivity by using ExpressRoute

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

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

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

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

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

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

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

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

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

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

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

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

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

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

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

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

Branch VNet connectivity by using site-to-site VPN

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

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

Next steps

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

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

See the [ExpressRoute FAQ](#) to:

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

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

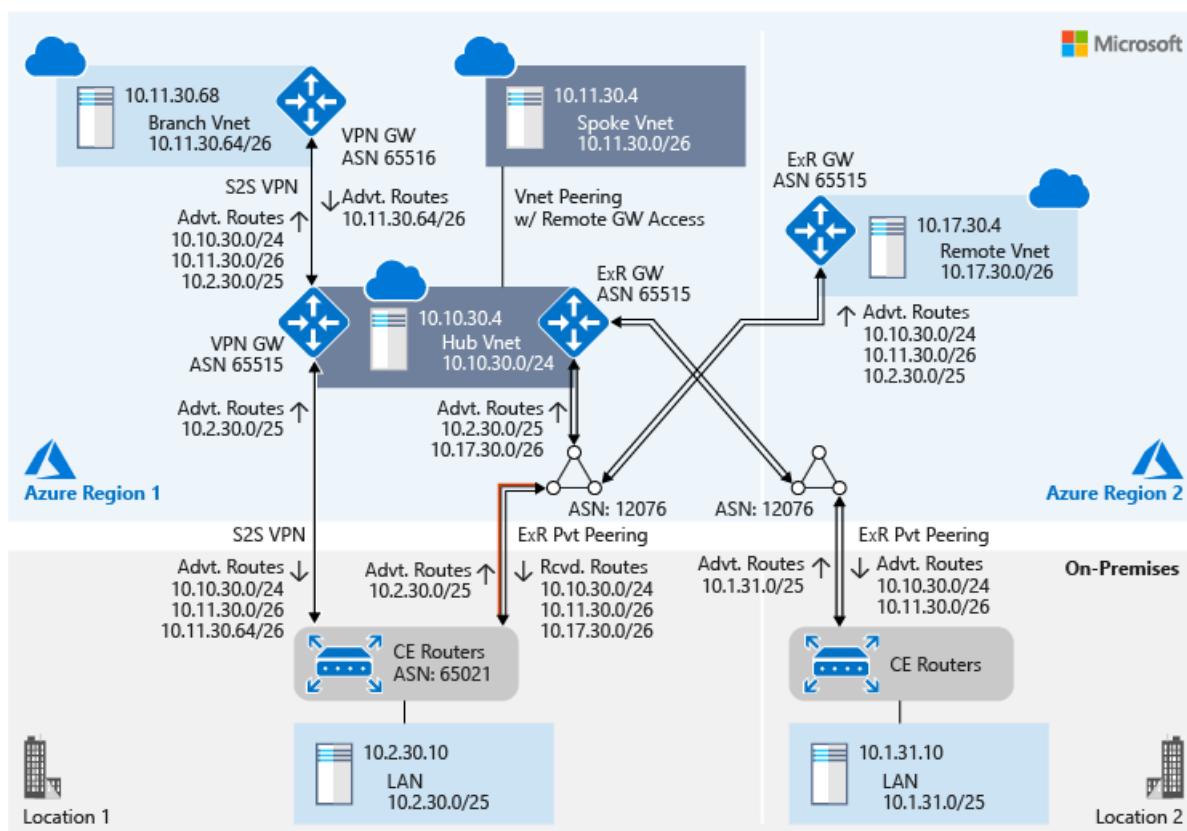
12/5/2019 • 4 minutes to read • [Edit Online](#)

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

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

Hub and spoke VNet perspective

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



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

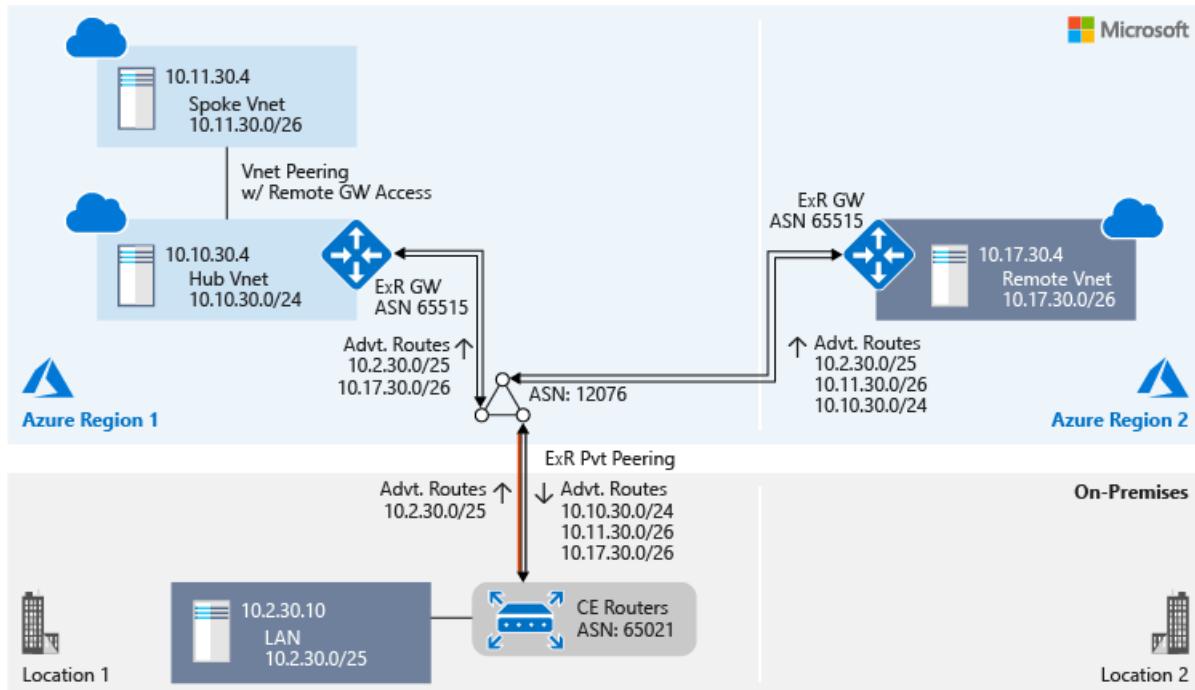
The following figure shows a sample ExpressRoute route table:

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

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

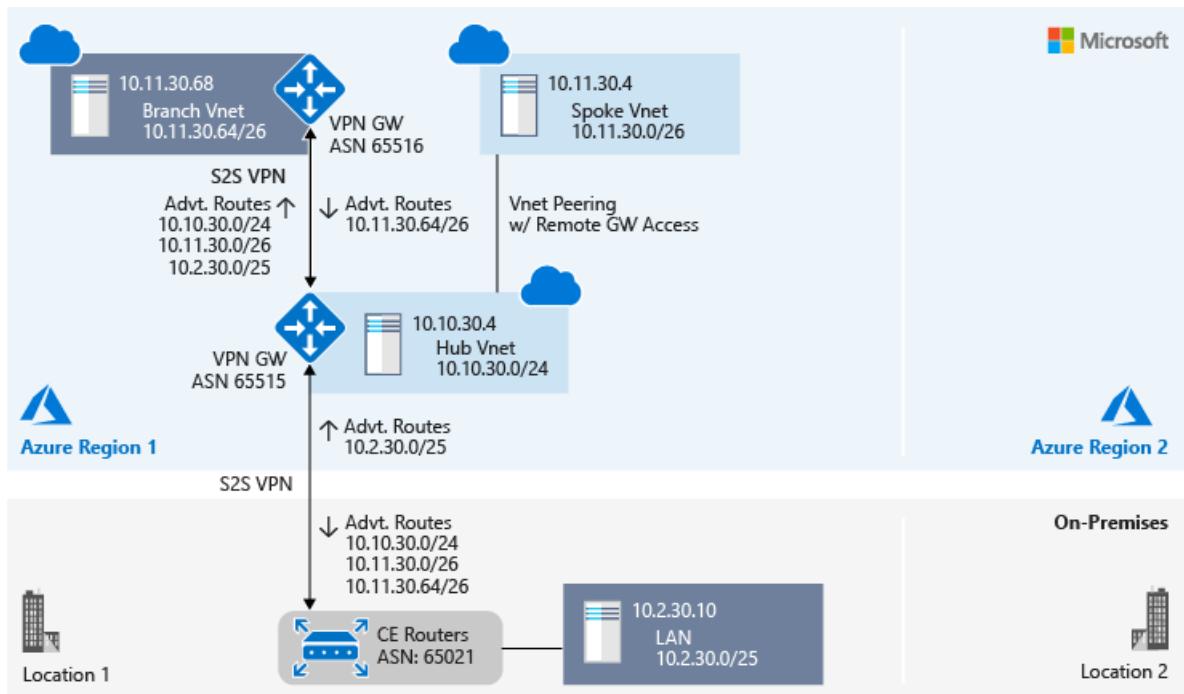
On-premises Location 1 and the remote VNet perspective via ExpressRoute 1

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



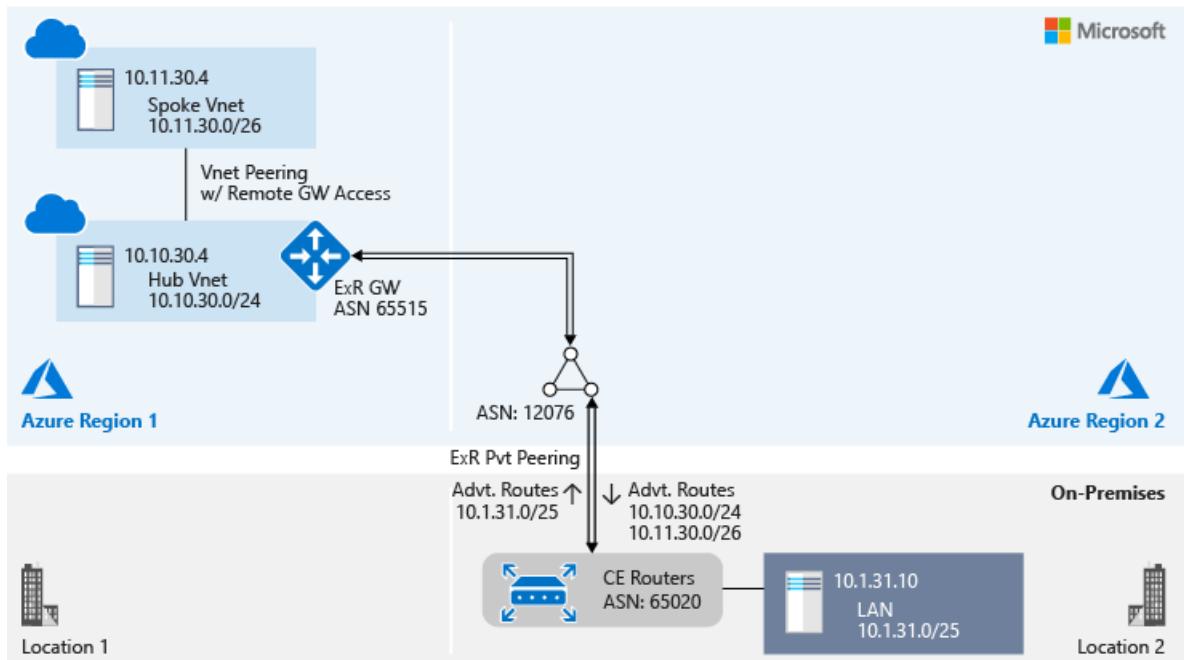
On-premises Location 1 and the branch VNet perspective via a site-to-site VPN

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



On-premises Location 2 perspective

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



ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

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

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

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

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

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

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

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

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

Branch VNet connectivity by using site-to-site VPN

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

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

Next steps

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

See the [ExpressRoute FAQ](#) to:

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

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

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

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

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

Data path from the hub VNet

Path to the spoke VNet

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

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

Tracing route to 10.11.30.4 over a maximum of 30 hops

 1      2 ms      1 ms      1 ms  10.11.30.4

Trace complete.
```

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



Path to the branch VNet

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

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

Tracing route to 10.11.30.68 over a maximum of 30 hops

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

Trace complete.
```

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

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



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

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

Average Latency in milliseconds

3

Minimum Latency in milliseconds

2

Maximum Latency in milliseconds

6

Probes Sent

66

Probes Failed

0

Path to on-premises Location 1

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

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

Tracing route to 10.2.30.10 over a maximum of 30 hops

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

Trace complete.
```

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

Path to on-premises Location 2

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

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

Tracing route to 10.1.31.10 over a maximum of 30 hops

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

Trace complete.
```

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

Path to the remote VNet

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

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

Tracing route to 10.17.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

Data path from the spoke VNet

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

Path to the hub VNet

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

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

Tracing route to 10.10.30.4 over a maximum of 30 hops

 1  <1 ms   <1 ms   <1 ms  10.10.30.4

Trace complete.
```

Path to the branch VNet

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

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

Tracing route to 10.11.30.68 over a maximum of 30 hops

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

Trace complete.
```

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

Path to on-premises Location 1

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

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

Tracing route to 10.2.30.10 over a maximum of 30 hops

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

Trace complete.
```

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

Path to on-premises Location 2

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

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

Tracing route to 10.1.31.10 over a maximum of 30 hops

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

Trace complete.
```

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

Path to the remote VNet

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

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

Tracing route to 10.17.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

Data path from the branch VNet

Path to the hub VNet

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

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

Tracing route to 10.10.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

Path to the spoke VNet

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

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

Tracing route to 10.11.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

Path to on-premises Location 1

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

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

Tracing route to 10.2.30.10 over a maximum of 30 hops

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

Trace complete.
```

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

Path to on-premises Location 2 and the remote VNet

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

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

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

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

C:\Users\rb>ping 10.17.30.4

Pinging 10.17.30.4 with 32 bytes of data:

Request timed out.
...
Request timed out.

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

Data path from on-premises Location 1

Path to the hub VNet

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

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

Tracing route to 10.10.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

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

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



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

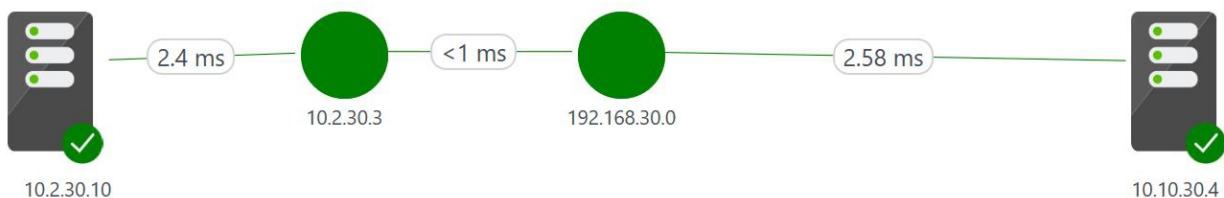
```
C:\Users\rb>tracert 10.10.30.4

Tracing route to 10.10.30.4 over a maximum of 30 hops

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

Trace complete.
```

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



Path to the spoke VNet

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

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

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

Tracing route to 10.11.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

Path to the branch VNet

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

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

Tracing route to 10.11.30.68 over a maximum of 30 hops

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

Trace complete.
```

Path to on-premises Location 2

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

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

Pinging 10.1.31.10 with 32 bytes of data:

Request timed out.
...
Request timed out.

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

Path to the remote VNet

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

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

Tracing route to 10.17.30.4 over a maximum of 30 hops

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

Trace complete.
```

Data path from on-premises Location 2

Path to the hub VNet

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

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

Tracing route to 10.10.30.4 over a maximum of 30 hops

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

Trace complete.
```

Path to the spoke VNet

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

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

Tracing route to 10.11.30.4 over a maximum of 30 hops

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

Trace complete.
```

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

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

Data path from the remote VNet

Path to the hub VNet

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

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

Tracing route to 10.10.30.4 over a maximum of 30 hops

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

Trace complete.
```

Path to the spoke VNet

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

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

Tracing route to 10.11.30.4 over a maximum of 30 hops

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

Trace complete.
```

Path to the branch VNet and on-premises Location 2

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

Path to on-premises Location 1

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

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

Tracing route to 10.2.30.10 over a maximum of 30 hops

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

Trace complete.
```

ExpressRoute and site-to-site VPN connectivity in tandem

Site-to-site VPN over ExpressRoute

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

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

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

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

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

Extend back-end connectivity to spoke VNets and branch locations

Spoke VNet connectivity by using VNet peering

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

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

Branch VNet connectivity by using site-to-site VPN

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

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

Next steps

See the [ExpressRoute FAQ](#) to:

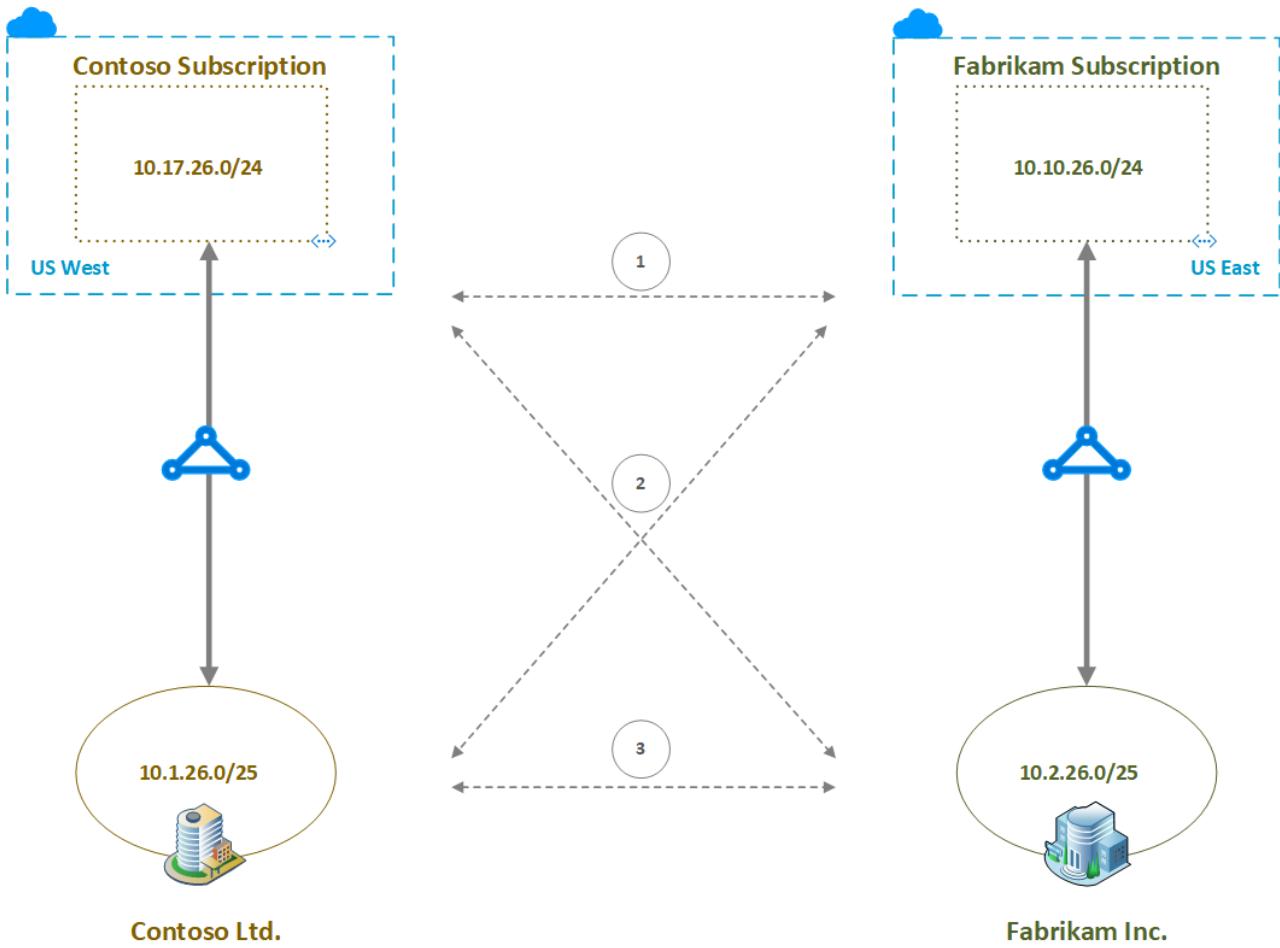
- Learn how many ExpressRoute circuits you can connect to an ExpressRoute gateway.
- Learn how many ExpressRoute gateways you can connect to an ExpressRoute circuit.
- Learn about other scale limits of ExpressRoute.

Cross-network connectivity

1/3/2020 • 4 minutes to read • [Edit Online](#)

Fabrikam Inc. has a large physical presence and Azure deployment in East US. Fabrikam has back-end connectivity between its on-premises and Azure deployments via ExpressRoute. Similarly, Contoso Ltd. has a presence and Azure deployment in West US. Contoso has back-end connectivity between its on-premises and Azure deployments via ExpressRoute.

Fabrikam Inc. acquires Contoso Ltd. Following the merger, Fabrikam wants to interconnect the networks. The following figure illustrates the scenario:



The dashed arrows in the middle of the above figure indicate the desired network interconnections. Specifically, there are three types cross connections desired: 1) Fabrikam and Contoso VNets cross connect, 2) Cross regional on-premises and VNets cross connects (that is, connecting Fabrikam on-premises network to Contoso VNet and connecting Contoso on-premises network to Fabrikam VNet), and 3) Fabrikam and Contoso on-premises network cross connect.

The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., before the merger.

Route table (Primary)

AzurePrivatePeering - Contoso-ER



[Download](#) [Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.1.26.0/25	192.168.26.17		0	65020
10.17.26.0/24	10.17.26.140		0	65515
	10.17.26.141		0	65515

The following table shows the effective routes of a VM in the Contoso subscription, before the merger. Per the table, the VM on the VNet is aware of the VNet address space and the Contoso on-premises network, apart from the default ones.

[Download](#) [Refresh](#)

i Showing only top 200 records, click Download above to see all.

Scope Network interface (Contoso-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.17.26.0/24	Virtual network	-
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., before the merger.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER



[Download](#) [Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.2.26.0/25	192.168.26.17		0	65021
10.10.26.0/24	10.10.26.140		0	65515
	10.10.26.141		0	65515

The following table shows the effective routes of a VM in the Fabrikam subscription, before the merger. Per the table, the VM on the VNet is aware of the VNet address space and the Fabrikam on-premises network, apart from the default ones.

[Download](#) [Refresh](#)

Showing only top 200 records, click Download above to see all.

Scope

Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-

In this article, let's go through step by step and discuss how to achieve the desired cross connections using the following Azure network features:

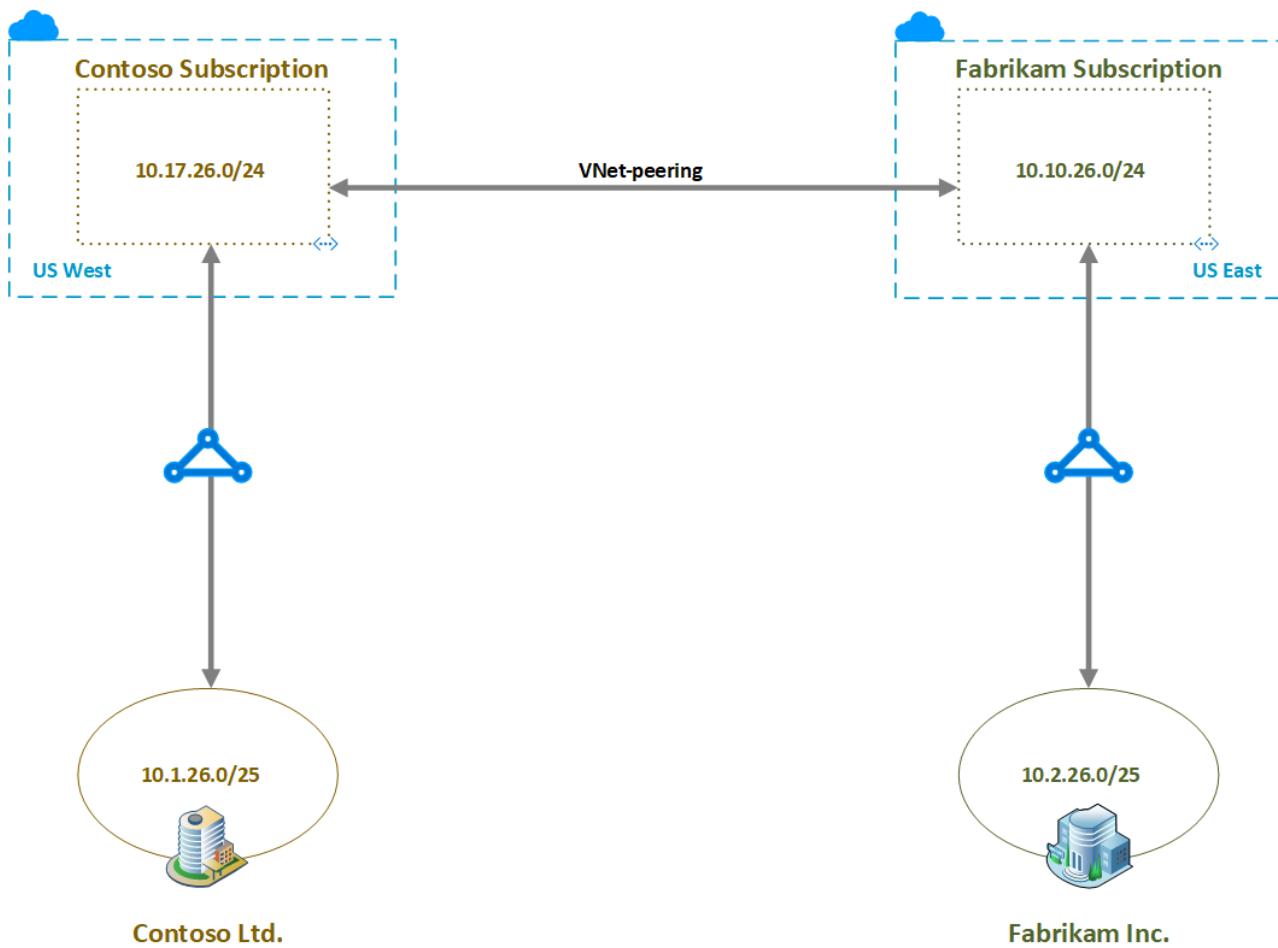
- [Virtual network peering](#)
- [Virtual network ExpressRoute connection](#)
- [Global Reach](#)

Cross connecting VNets

Virtual network peering (VNet peering) provides the most optimal and the best network performance when connecting two virtual networks. VNet peering supports peering two VNets both within the same Azure region (commonly called VNet peering) and in two different Azure regions (commonly called Global VNet peering).

Let's configure Global VNet peering between the VNets in Contoso and Fabrikam Azure subscriptions. For how to create the virtual network peering between two the virtual networks, see [Create a virtual network peering](#) article.

The following picture shows the network architecture after configuring Global VNet peering.



The following table shows the routes known to the Contoso subscription VM. Pay attention to the last entry of the table. This entry is the result of cross connecting the virtual networks.

Effective routes					
Source	State	Address prefixes	Next hop type	Next hop type IP address	
Default	Active	10.17.26.0/24	Virtual network	-	
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53	
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52	
Default	Active	0.0.0.0/0	Internet	-	
Default	Active	10.0.0.0/8	None	-	
Default	Active	100.64.0.0/10	None	-	
Default	Active	192.168.0.0/16	None	-	
Default	Active	10.10.26.0/24	VNetGlobalPeering	-	

The following table shows the routes known to the Fabrikam subscription VM. Pay attention to the last entry of the table. This entry is the result of cross connecting the virtual networks.

[Download](#) [Refresh](#)

Showing only top 200 records, click Download above to see all.

Scope

Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.17.26.0/24	VNetGlobalPeering	-

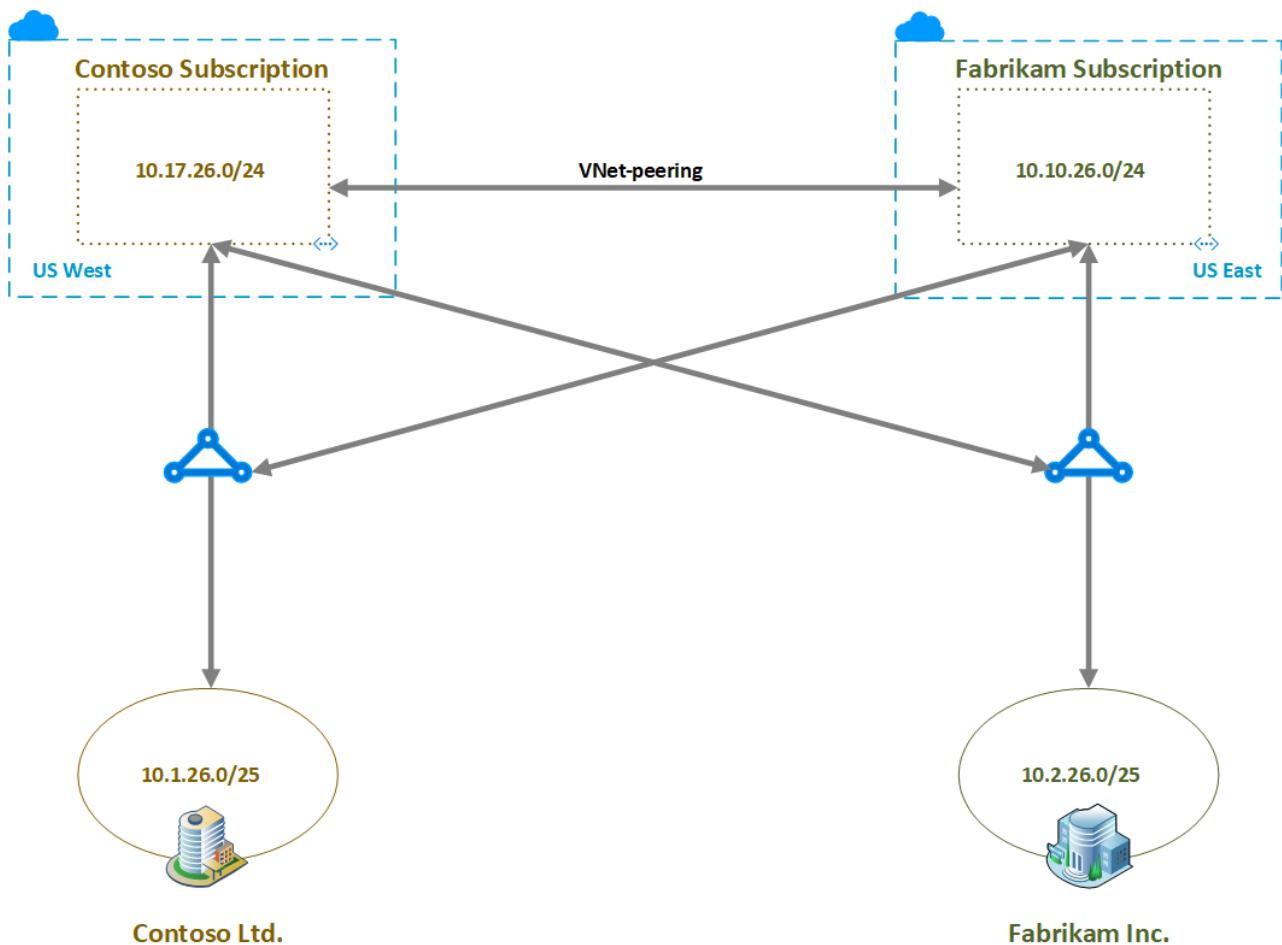
VNet peering directly links two virtual networks (see there are no next hop for *VNetGlobalPeering* entry in the above two tables)

Cross connecting VNets to the on-premises networks

We can connect an ExpressRoute circuit to multiple virtual networks. See [Subscription and service limits](#) for the maximum number of virtual networks that can be connected to an ExpressRoute circuit.

Let's connect Fabrikam ExpressRoute circuit to Contoso subscription VNet and similarly Contoso ExpressRoute circuit to Fabrikam subscription VNet to enable cross connectivity between virtual networks and the on-premises networks. To connect a virtual network to an ExpressRoute circuit in a different subscription, we need to create and use an authorization. See the article: [Connect a virtual network to an ExpressRoute circuit](#).

The following picture shows the network architecture after configuring the ExpressRoute cross connectivity to the virtual networks.



The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., after cross connecting virtual networks to the on-premises networks via ExpressRoute. See that the route table has routes belonging to both the virtual networks.

Route table (Primary)						
AzurePrivatePeering - Contoso-ER						
NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH		
10.1.26.0/25	192.168.26.17		0	65020		
10.10.26.0/24	10.10.26.141		0	65515		
	10.10.26.140		0	65515		
10.17.26.0/24	10.17.26.140		0	65515		
	10.17.26.141		0	65515		

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., after cross connecting virtual networks to the on-premises networks via ExpressRoute. See that the route table has routes belonging to both the virtual networks.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER

[Download](#)[Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH
10.2.26.0/25	192.168.26.17	0	65021	
10.10.26.0/24	10.10.26.140	0	65515	
	10.10.26.141	0	65515	
10.17.26.0/24	10.17.26.140	0	65515	
	10.17.26.141	0	65515	

The following table shows the routes known to the Contoso subscription VM. Pay attention to *Virtual network gateway* entries of the table. The VM sees routes for both the on-premises networks.

[Download](#)[Refresh](#)

i Showing only top 200 records, click Download above to see all.

Scope

Network interface (Contoso-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.17.26.0/24	Virtual network	-
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.10.26.0/24	VNetGlobalPeering	-

The following table shows the routes known to the Fabrikam subscription VM. Pay attention to *Virtual network gateway* entries of the table. The VM sees routes for both the on-premises networks.

[Download](#) [Refresh](#)

Showing only top 200 records, click Download above to see all.

Scope

Network interface (Fabrikam-VM01-nic)

Effective routes

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE	NEXT HOP TYPE IP ADDRESS
Default	Active	10.10.26.0/24	Virtual network	-
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.53
Virtual network gateway	Active	10.1.26.0/25	Virtual network gateway	10.3.129.52
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.24
Virtual network gateway	Active	10.2.26.0/25	Virtual network gateway	10.3.129.25
Default	Active	0.0.0.0/0	Internet	-
Default	Active	10.0.0.0/8	None	-
Default	Active	100.64.0.0/10	None	-
Default	Active	192.168.0.0/16	None	-
Default	Active	10.17.26.0/24	VNetGlobalPeering	-

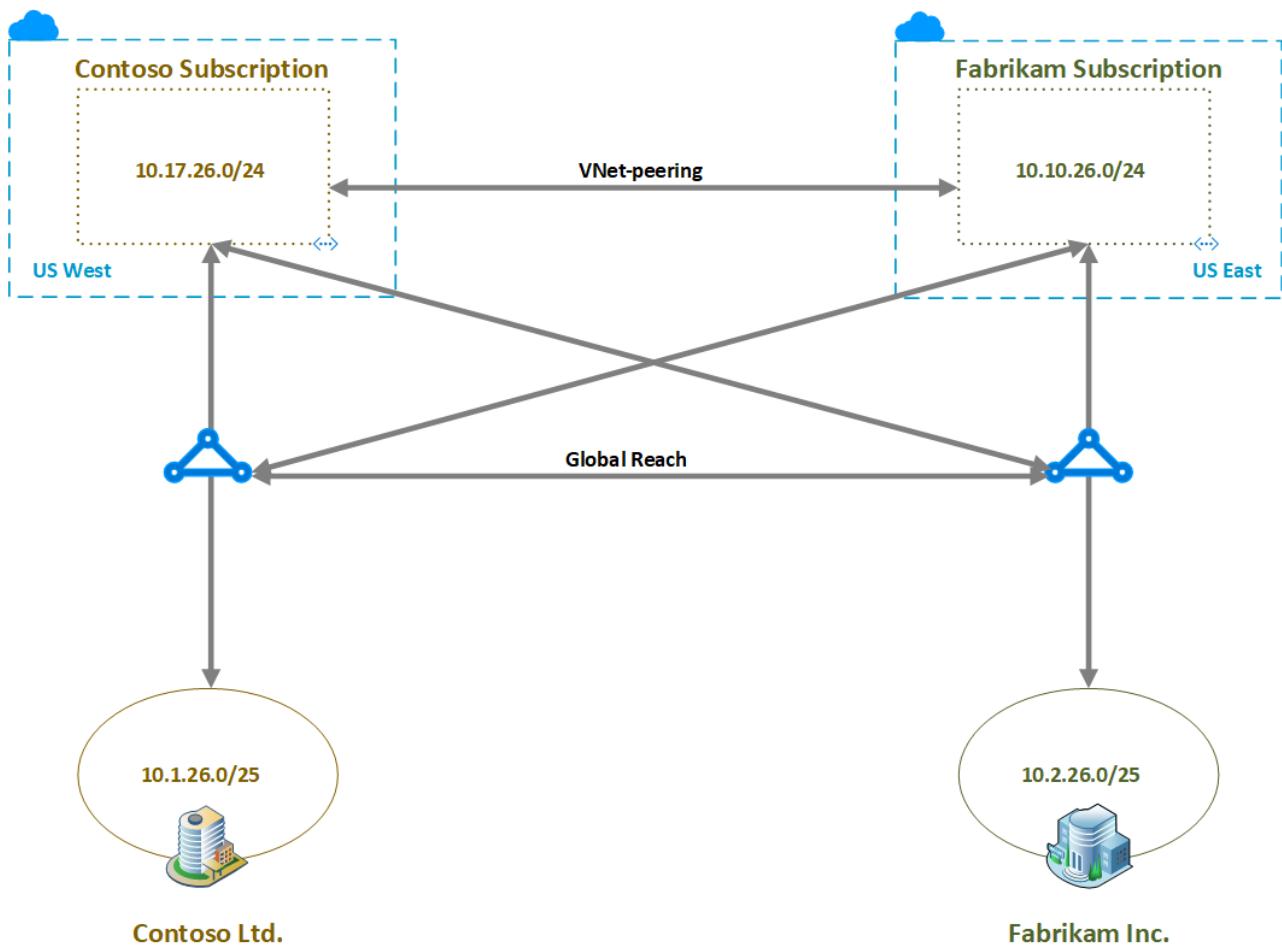
NOTE

In either the Fabrikam and/or Contoso subscriptions you can also have spoke VNets to the respective hub VNet (a hub and spoke design is not illustrated in the architecture diagrams in this article). The cross connections between the hub VNet gateways to ExpressRoute will also allow communication between East and West hubs and spokes.

Cross connecting on-premises networks

ExpressRoute Global Reach provides connectivity between on-premises networks that are connected to different ExpressRoute circuits. Let's configure Global Reach between Contoso and Fabrikam ExpressRoute circuits. Because the ExpressRoute circuits are in different subscriptions, we need to create and use an authorization. See [Configure ExpressRoute Global Reach](#) article for step by step guidance.

The following picture shows the network architecture after configuring Global Reach.



The following table shows the route table of the private peering of the ExpressRoute of Contoso Ltd., after configuring Global Reach. See that the route table has routes belonging to both the on-premises networks.

Route table (Primary)							
AzurePrivatePeering - Contoso-ER							
Download		Show secondary					
Showing only top 200 primary records, click Download above to see all.							
NETWORK	NEXT HOP	LOCPRF	WEIGHT	PATH			
10.1.26.0/25	192.168.26.17		0	65020			
10.2.26.0/25	192.168.26.51	10	0	65021			
10.10.26.0/24	192.168.26.51	10	0	65515			
	10.10.26.141		0	65515			
	10.10.26.140		0	65515			
10.17.26.0/24	192.168.26.51	10	0	65515			
	10.17.26.140		0	65515			
	10.17.26.141		0	65515			

The following table shows the route table of the private peering of the ExpressRoute of Fabrikam Inc., after configuring Global Reach. See that the route table has routes belonging to both the on-premises networks.

Route table (Primary)

AzurePrivatePeering - Fabrikam-ER



[Download](#)

[Show secondary](#)

i Showing only top 200 primary records, click Download above to see all.

NETWORK	↑↓	NEXT HOP	↑↓	LOCPRF	↑↓	WEIGHT	↑↓	PATH	↑↓
10.1.26.0/25		192.168.26.49		10		0		65020	
10.2.26.0/25		192.168.26.17				0		65021	
10.10.26.0/24		192.168.26.49		10		0		65515	
		10.10.26.140				0		65515	
		10.10.26.141				0		65515	
10.17.26.0/24		192.168.26.49		10		0		65515	
		10.17.26.140				0		65515	
		10.17.26.141				0		65515	

Next steps

See [virtual network FAQ](#), for any further questions on VNet and VNet-peering. See [ExpressRoute FAQ](#) for any further questions on ExpressRoute and virtual network connectivity.

Global Reach is rolled out on a country/region by country/region basis. To see if Global Reach is available in the countries/regions that you want, see [ExpressRoute Global Reach](#).

Security controls for Azure ExpressRoute

11/14/2019 • 2 minutes to read • [Edit Online](#)

This article documents the security controls built into Azure ExpressRoute.

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

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

Network

SECURITY CONTROL	YES/NO	NOTES
Service endpoint support	N/A	
VNet injection support	N/A	
Network isolation and firewalling support	Yes	Each customer is contained in its own routing domain and tunneled to its own VNet
Forced tunneling support	N/A	Via Border Gateway Protocol (BGP).

Monitoring & logging

SECURITY CONTROL	YES/NO	NOTES
Azure monitoring support (Log analytics, App insights, etc.)	Yes	See ExpressRoute monitoring, metrics, and alerts .
Control and management plane logging and audit	Yes	
Data plane logging and audit	No	

Identity

SECURITY CONTROL	YES/NO	NOTES
Authentication	Yes	Service account for Gateway for Microsoft (GWM) (controller); Just in Time (JIT) access for Dev and OP.
Authorization	Yes	Service account for Gateway for Microsoft (GWM) (controller); Just in Time (JIT) access for Dev and OP.

Data protection

SECURITY CONTROL	YES/NO	NOTES
Server-side encryption at rest: Microsoft-managed keys	N/A	ExpressRoute does not store customer data.
Server-side encryption at rest: customer-managed keys (BYOK)	N/A	
Column level encryption (Azure Data Services)	N/A	
Encryption in transit (such as ExpressRoute encryption, in VNet encryption, and VNet-VNet encryption)	No	
API calls encrypted	Yes	Through Azure Resource Manager and HTTPS.

Configuration management

SECURITY CONTROL	YES/NO	NOTES
Configuration management support (versioning of configuration, etc.)	Yes	Via the Network Resource Provider (NRP).

Next steps

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

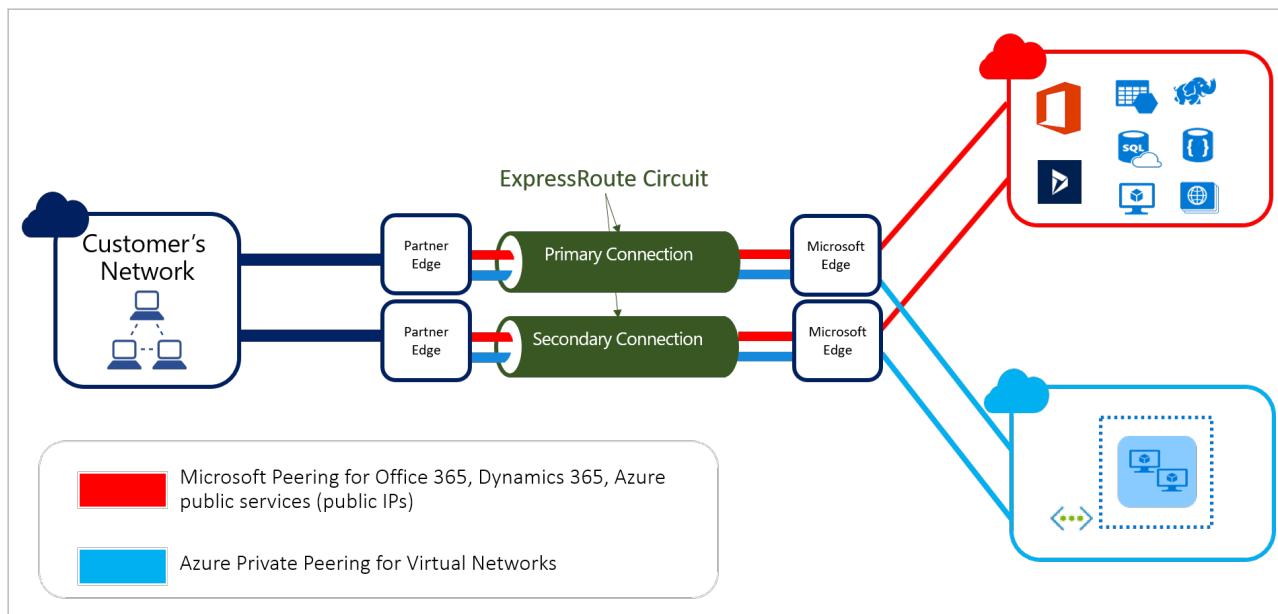
Designing for high availability with ExpressRoute

11/13/2019 • 5 minutes to read • [Edit Online](#)

ExpressRoute is designed for high availability to provide carrier grade private network connectivity to Microsoft resources. In other words, there is no single point of failure in the ExpressRoute path within Microsoft network. To maximize the availability, the customer and the service provider segment of your ExpressRoute circuit should also be architected for high availability. In this article, first let's look into network architecture considerations for building robust network connectivity using an ExpressRoute, then let's look into the fine-tuning features that help you to improve the high availability of your ExpressRoute circuit.

Architecture considerations

The following figure illustrates the recommended way to connect using an ExpressRoute circuit for maximizing the availability of an ExpressRoute circuit.

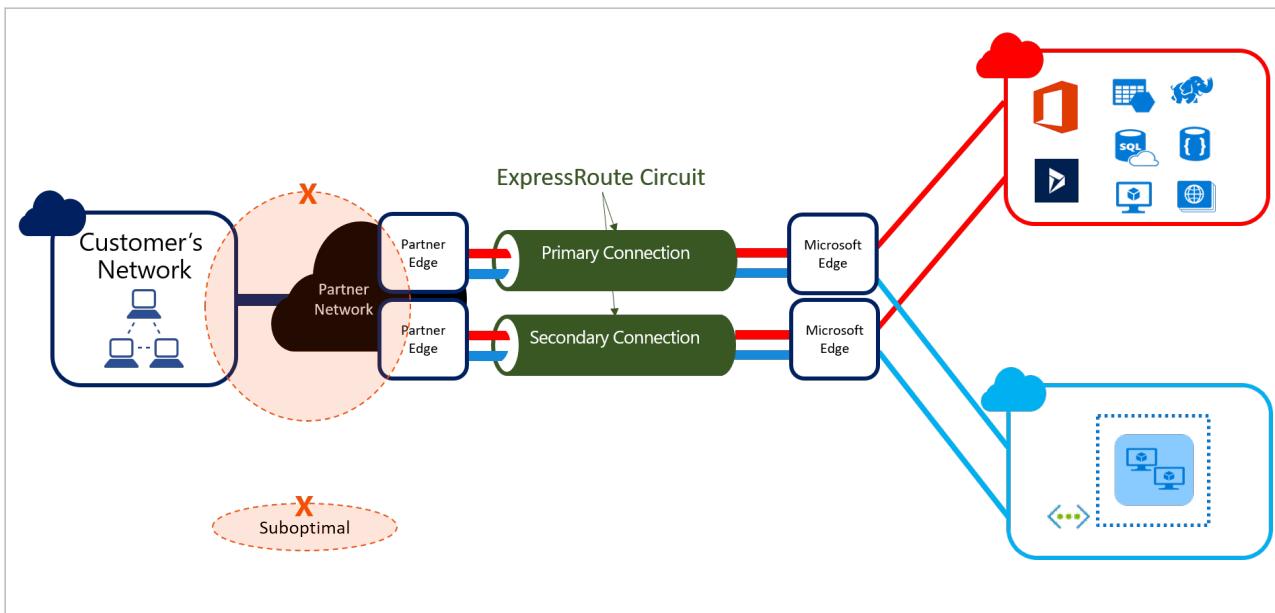


For high availability, it's essential to maintain the redundancy of the ExpressRoute circuit throughout the end-to-end network. In other words, you need to maintain redundancy within your on-premises network, and shouldn't compromise redundancy within your service provider network. Maintaining redundancy at the minimum implies avoiding single point of network failures. Having redundant power and cooling for the network devices will further improve the high availability.

First mile physical layer design considerations

If you terminate both the primary and secondary connections of an ExpressRoute circuits on the same Customer Premises Equipment (CPE), you're compromising the high availability within your on-premises network.

Additionally, if you configure both the primary and secondary connections via the same port of a CPE (either by terminating the two connections under different subinterfaces or by merging the two connections within the partner network), you're forcing the partner to compromise high availability on their network segment as well. This compromise is illustrated in the following figure.



On the other hand, if you terminate the primary and the secondary connections of an ExpressRoute circuits in different geographical locations, then you could be compromising the network performance of the connectivity. If traffic is actively load balanced across the primary and the secondary connections that are terminated on different geographical locations, potential substantial difference in network latency between the two paths would result in suboptimal network performance.

For geo-redundant design considerations, see [Designing for disaster recovery with ExpressRoute](#).

Active-active connections

Microsoft network is configured to operate the primary and secondary connections of ExpressRoute circuits in active-active mode. However, through your route advertisements, you can force the redundant connections of an ExpressRoute circuit to operate in active-passive mode. Advertising more specific routes and BGP AS path prepending are the common techniques used to make one path preferred over the other.

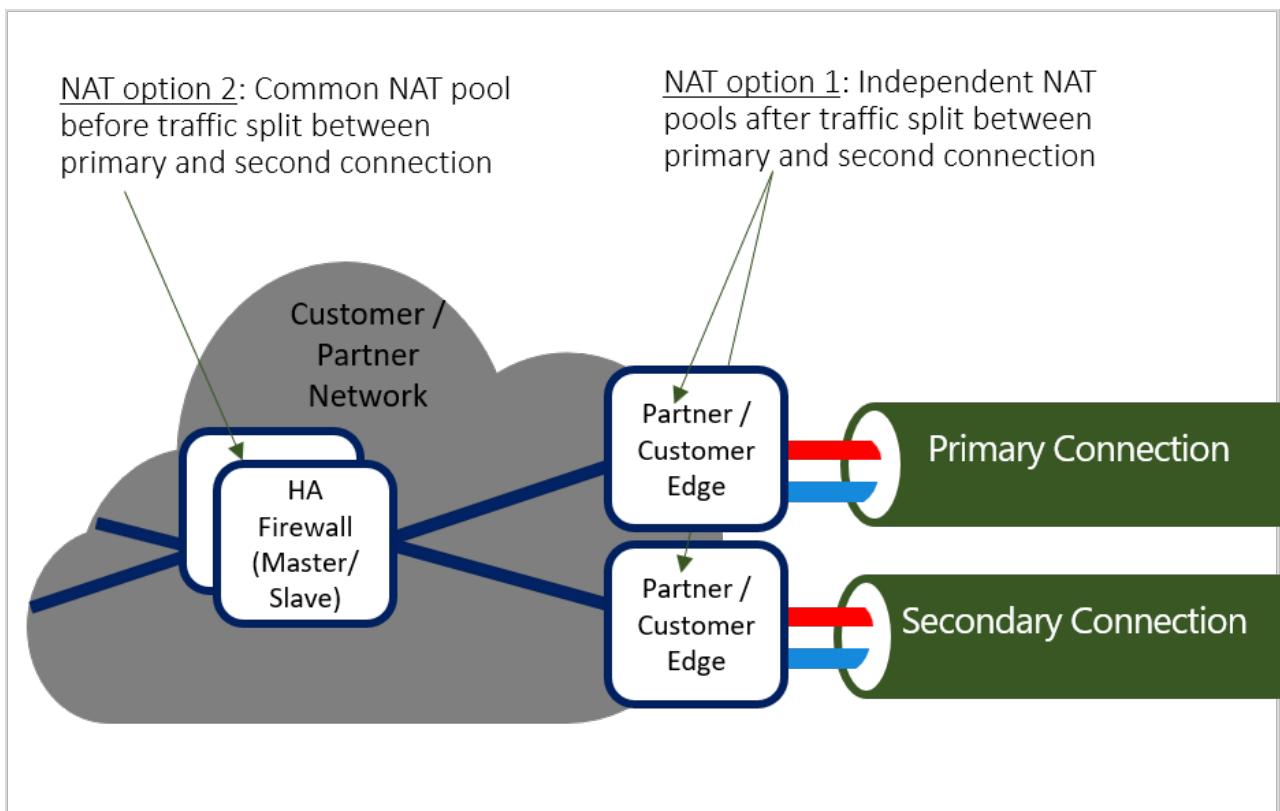
To improve high availability, it's recommended to operate both the connections of an ExpressRoute circuit in active-active mode. If you let the connections operate in active-active mode, Microsoft network will load balance the traffic across the connections on per-flow basis.

Running the primary and secondary connections of an ExpressRoute circuit in active-passive mode face the risk of both the connections failing following a failure in the active path. The common causes for failure on switching over are lack of active management of the passive connection, and passive connection advertising stale routes.

Alternatively, running the primary and secondary connections of an ExpressRoute circuit in active-active mode, results in only about half the flows failing and getting rerouted, following an ExpressRoute connection failure. Thus, active-active mode will significantly help improve the Mean Time To Recover (MTTR).

NAT for Microsoft peering

Microsoft peering is designed for communication between public end-points. So commonly, on-premises private endpoints are Network Address Translated (NATed) with public IP on the customer or partner network before they communicate over Microsoft peering. Assuming you use both the primary and secondary connections in active-active mode, where and how you NAT has an impact on how quickly you recover following a failure in one of the ExpressRoute connections. Two different NAT options are illustrated in the following figure:



In the option 1, NAT is applied after splitting the traffic between the primary and secondary connections of the ExpressRoute. To meet the stateful requirements of NAT, independent NAT pools are used between the primary and the secondary devices so that the return traffic would arrive to the same edge device through which the flow egressed.

In the option 2, a common NAT pool is used before splitting the traffic between the primary and secondary connections of the ExpressRoute. It's important to make the distinction that the common NAT pool before splitting the traffic does not mean introducing single-point of failure thereby compromising high-availability.

With the option 1, following an ExpressRoute connection failure, ability to reach the corresponding NAT pool is broken. Therefore, all the broken flows have to be re-established either by TCP or application layer following the corresponding window timeout. If either of the NAT pools are used to frontend any of the on-premises servers and if the corresponding connectivity were to fail, the on-premises servers cannot be reached from Azure until the connectivity is fixed.

Whereas with the option 2, the NAT is reachable even after a primary or secondary connection failure. Therefore, the network layer itself can reroute the packets and help faster recovery following the failure.

NOTE

If you use NAT option 1 (independent NAT pools for primary and secondary ExpressRoute connections) and map a port of an IP address from one of the NAT pool to an on-premises server, the server will not be reachable via the ExpressRoute circuit when the corresponding connection fails.

Fine-tuning features for private peering

In this section, let us review optional (depending on your Azure deployment and how sensitive you're to MTTR) features that help improve high availability of your ExpressRoute circuit. Specifically, let's review zone-aware deployment of ExpressRoute virtual network gateways, and Bidirectional Forwarding Detection (BFD).

Availability Zone aware ExpressRoute virtual network gateways

An Availability Zone in an Azure region is a combination of a fault domain and an update domain. If you opt for

zone-redundant Azure IaaS deployment, you may also want to configure zone-redundant virtual network gateways that terminate ExpressRoute private peering. To learn further, see [About zone-redundant virtual network gateways in Azure Availability Zones](#). To configure zone-redundant virtual network gateway, see [Create a zone-redundant virtual network gateway in Azure Availability Zones](#).

Improving failure detection time

ExpressRoute supports BFD over private peering. BFD reduces detection time of failure over the Layer 2 network between Microsoft Enterprise Edge (MSEEs) and their BGP neighbors on the on-premises side from about 3 minutes (default) to less than a second. Quick failure detection time helps hastening failure recovery. To learn further, see [Configure BFD over ExpressRoute](#).

Next steps

In this article, we discussed how to design for high availability of an ExpressRoute circuit connectivity. An ExpressRoute circuit peering point is pinned to a geographical location and therefore could be impacted by catastrophic failure that impacts the entire location.

For design considerations to build geo-redundant network connectivity to Microsoft backbone that can withstand catastrophic failures, which impact an entire region, see [Designing for disaster recovery with ExpressRoute private peering](#).

Designing for disaster recovery with ExpressRoute private peering

11/13/2019 • 7 minutes to read • [Edit Online](#)

ExpressRoute is designed for high availability to provide carrier grade private network connectivity to Microsoft resources. In other words, there is no single point of failure in the ExpressRoute path within Microsoft network. For design considerations to maximize the availability of an ExpressRoute circuit, see [Designing for high availability with ExpressRoute](#).

However, taking Murphy's popular adage--*if anything can go wrong, it will*--into consideration, in this article let us focus on solutions that go beyond failures that can be addressed using a single ExpressRoute circuit. In other words, in this article let us look into network architecture considerations for building robust backend network connectivity for disaster recovery using geo-redundant ExpressRoute circuits.

Need for redundant connectivity solution

There are possibilities and instances where an entire regional service (be it that of Microsoft, network service providers, customer, or other cloud service providers) gets degraded. The root cause for such regional wide service impact include natural calamity. Therefore, for business continuity and mission critical applications it is important to plan for disaster recovery.

Irrespective of whether you run your mission critical applications in an Azure region or on-premises or anywhere else, you can use another Azure region as your failover site. The following articles addresses disaster recovery from applications and frontend access perspectives:

- [Enterprise-scale disaster recovery](#)
- [SMB disaster recovery with Azure Site Recovery](#)

If you rely on ExpressRoute connectivity between your on-premises network and Microsoft for mission critical operations, your disaster recovery plan should also include geo-redundant network connectivity.

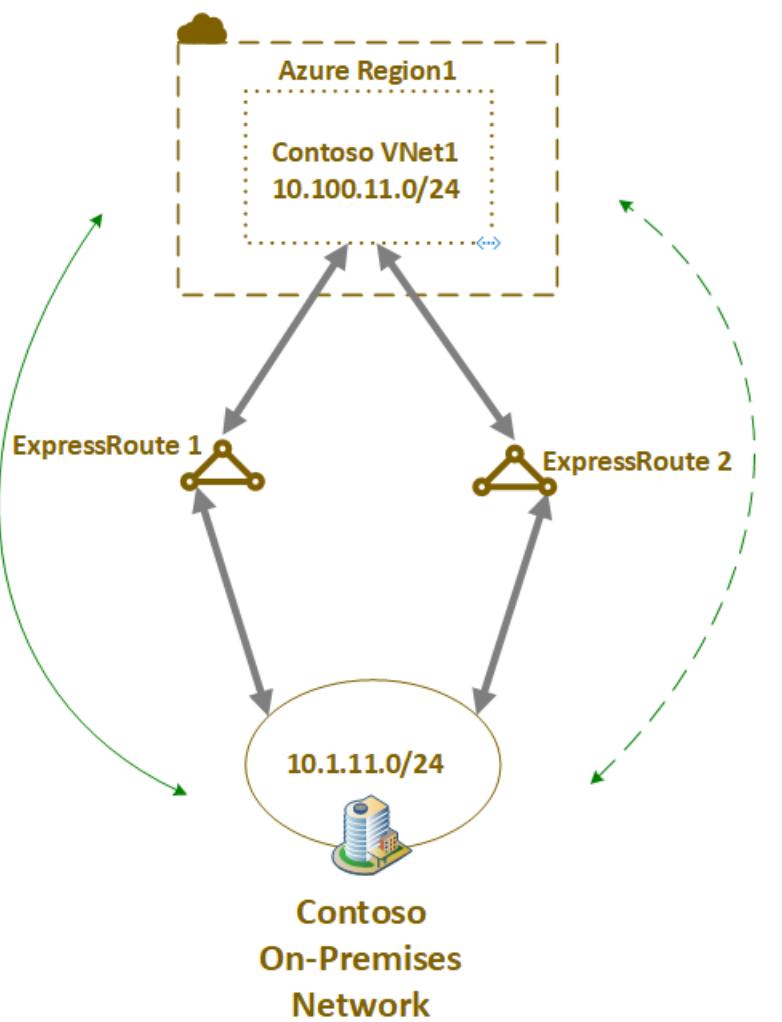
Challenges of using multiple ExpressRoute circuits

When you interconnect the same set of networks using more than one connection, you introduce parallel paths between the networks. Parallel paths, when not properly architected, could lead to asymmetrical routing. If you have stateful entities (for example, NAT, firewall) in the path, asymmetrical routing could block traffic flow. Typically, over the ExpressRoute private peering path you won't come across stateful entities such as NAT or Firewalls. Therefore, asymmetrical routing over ExpressRoute private peering does not necessarily block traffic flow.

However, if you load balance traffic across geo-redundant parallel paths, irrespective of whether you have stateful entities or not, you would experience inconsistent network performance. In this article, let's discuss how to address these challenges.

Small to medium on-premises network considerations

Let's consider the example network illustrated in the following diagram. In the example, geo-redundant ExpressRoute connectivity is established between a Contoso's on-premises location and Contoso's VNet in an Azure region. In the diagram, solid green line indicates preferred path (via ExpressRoute 1) and the dotted one represents stand-by path (via ExpressRoute 2).



When you are designing ExpressRoute connectivity for disaster recovery, you need to consider:

- using geo-redundant ExpressRoute circuits
- using diverse service provider network(s) for different ExpressRoute circuit
- designing each of the ExpressRoute circuit for [high availability](#)
- terminating the different ExpressRoute circuit in different location on the customer network

By default, if you advertise routes identically over all the ExpressRoute paths, Azure will load-balance on-premises bound traffic across all the ExpressRoute paths using Equal-cost multi-path (ECMP) routing.

However, with the geo-redundant ExpressRoute circuits we need to take into consideration different network performances with different network paths (particularly for network latency). To get more consistent network performance during normal operation, you may want to prefer the ExpressRoute circuit that offers the minimal latency.

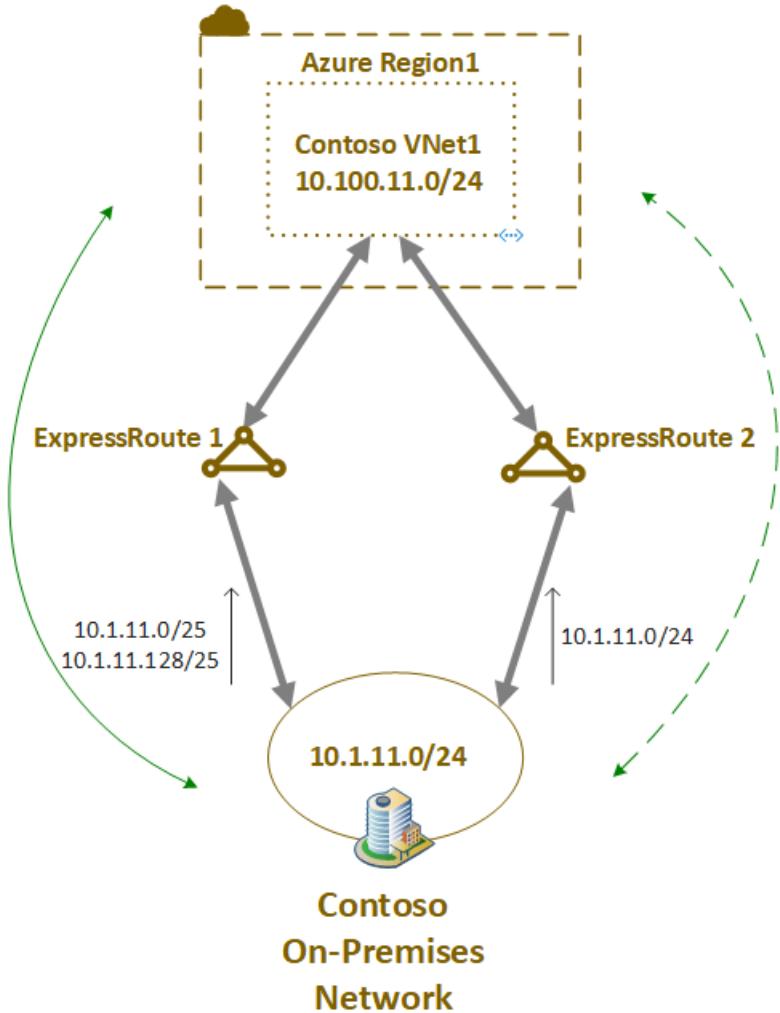
You can influence Azure to prefer one ExpressRoute circuit over another one using one of the following techniques (listed in the order of effectiveness):

- advertising more specific route over the preferred ExpressRoute circuit compared to other ExpressRoute circuit(s)
- configuring higher Connection Weight on the connection that links the virtual network to the preferred ExpressRoute circuit
- advertising the routes over less preferred ExpressRoute circuit with longer AS Path (AS Path prepend)

More specific route

The following diagram illustrates influencing ExpressRoute path selection using more specific route advertisement. In the illustrated example, Contoso on-premises /24 IP range is advertised as two /25 address ranges via the

preferred path (ExpressRoute 1) and as /24 via the stand-by path (ExpressRoute 2).



Because /25 is more specific, compared to /24, Azure would send the traffic destined to 10.1.11.0/24 via ExpressRoute 1 in the normal state. If both the connections of ExpressRoute 1 go down, then the VNet would see the 10.1.11.0/24 route advertisement only via ExpressRoute 2; and therefore the standby circuit is used in this failure state.

Connection weight

The following screenshot illustrates configuring the weight of an ExpressRoute connection via Azure portal.



ASH-Cust11-VNet01-ERGW-Con - Configuration

Connection

 Search (Ctrl+ /)[Save](#) [Discard](#)

★ Routing weight

[Overview](#)[Activity log](#)[Access control \(IAM\)](#)[Tags](#)

Settings

[Configuration](#)[Properties](#)[Locks](#)[Export template](#)

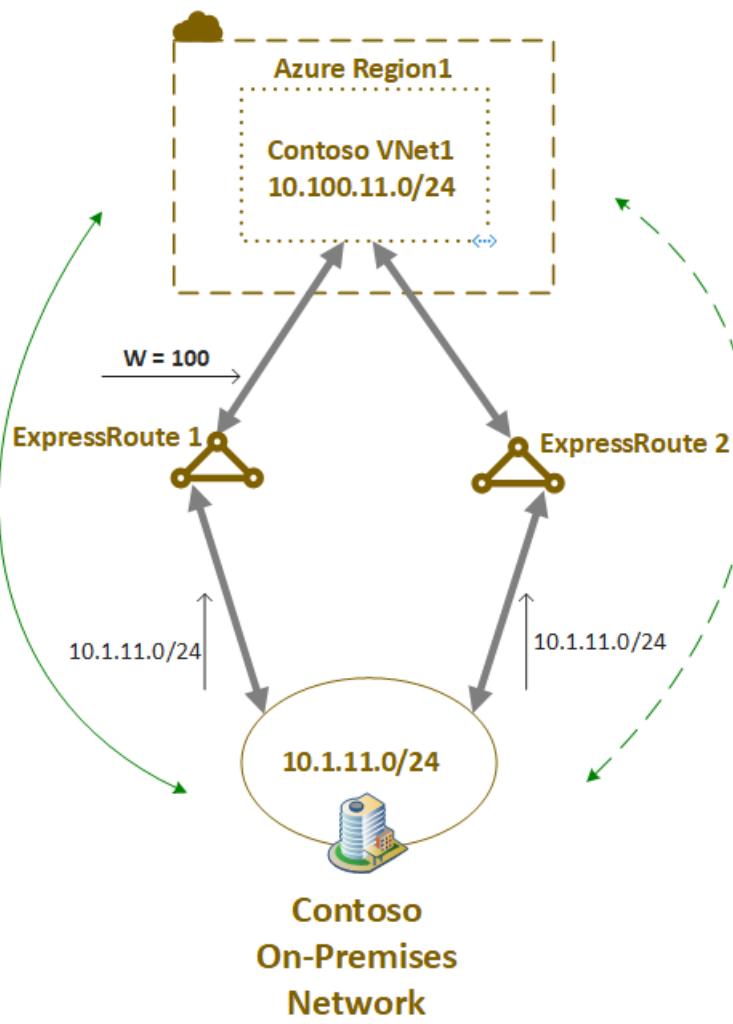
Monitoring

[Metrics](#)

Support + troubleshooting

[Resource health](#)[New support request](#)

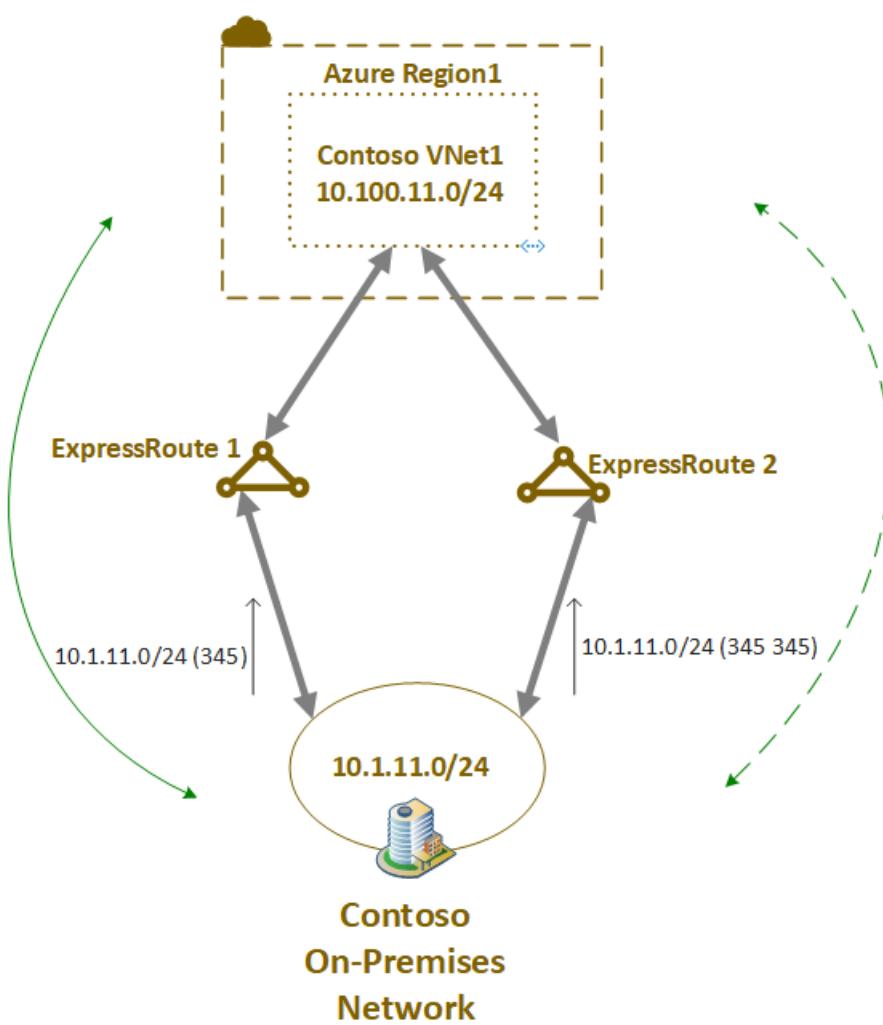
The following diagram illustrates influencing ExpressRoute path selection using connection weight. The default connection weight is 0. In the example below, the weight of the connection for ExpressRoute 1 is configured as 100. When a VNet receives a route prefix advertised via more than one ExpressRoute circuit, the VNet will prefer the connection with the highest weight.



If both the connections of ExpressRoute 1 go down, then the VNet would see the 10.1.11.0/24 route advertisement only via ExpressRoute 2; and therefore the standby circuit is used in this failure state.

AS path prepend

The following diagram illustrates influencing ExpressRoute path selection using AS path prepend. In the diagram, the route advertisement over ExpressRoute 1 indicates the default behavior of eBGP. On the route advertisement over ExpressRoute 2, the on-premises network's ASN is prepended additionally on the route's AS path. When the same route is received through multiple ExpressRoute circuits, per the eBGP route selection process, VNet would prefer the route with the shortest AS path.



If both the connections of ExpressRoute 1 go down, then the VNet would see the 10.1.11.0/24 route advertisement only via ExpressRoute 2. Consequentially, the longer AS path would become irrelevant. Therefore, the standby circuit would be used in this failure state.

Using any of the techniques, if you influence Azure to prefer one of your ExpressRoute over others, you also need to ensure the on-premises network also prefer the same ExpressRoute path for Azure bound traffic to avoid asymmetric flows. Typically, local preference value is used to influence on-premises network to prefer one ExpressRoute circuit over others. Local preference is an internal BGP (iBGP) metric. The BGP route with the highest local preference value is preferred.

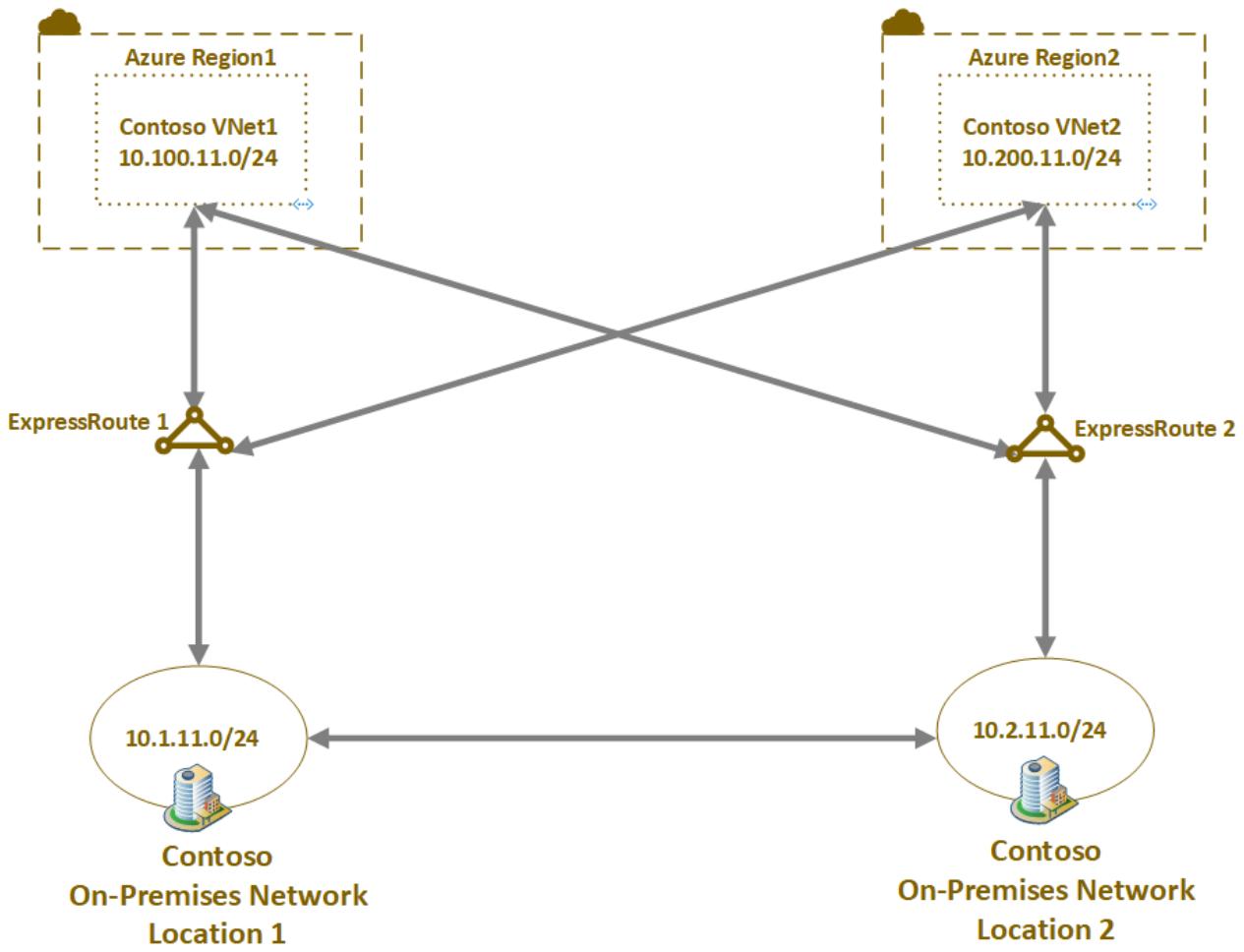
IMPORTANT

When you use certain ExpressRoute circuits as stand-by, you need to actively manage them and periodically test failover operation.

Large distributed enterprise network

When you have a large distributed enterprise network, you're likely to have multiple ExpressRoute circuits. In this section, let's see how to design disaster recovery using the active-active ExpressRoute circuits, without needing additional stand-by circuits.

Let's consider the example illustrated in the following diagram. In the example, Contoso has two on-premises locations connected to two Contoso IaaS deployment in two different Azure regions via ExpressRoute circuits in two different peering locations.

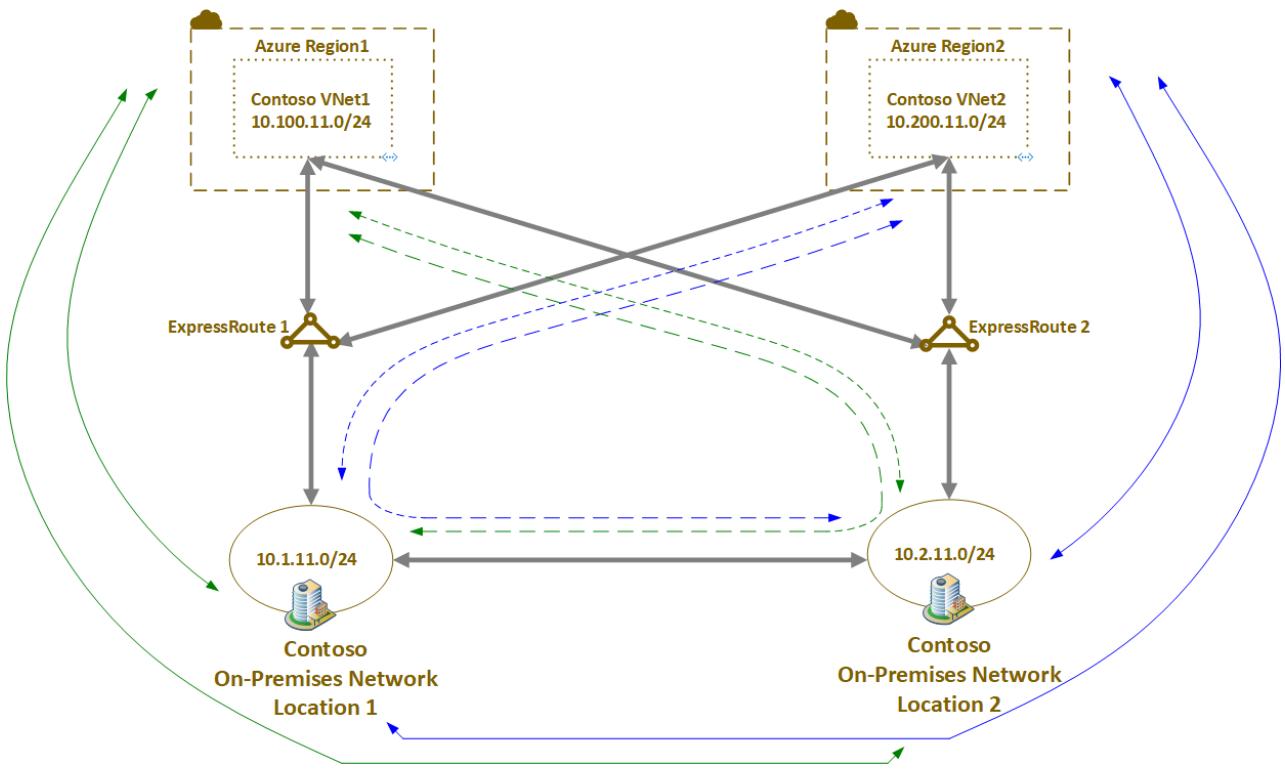


How we architect the disaster recovery has an impact on how cross regional to cross location (region1/region2 to location2/location1) traffic is routed. Let's consider two different disaster architectures that routes cross region-location traffic differently.

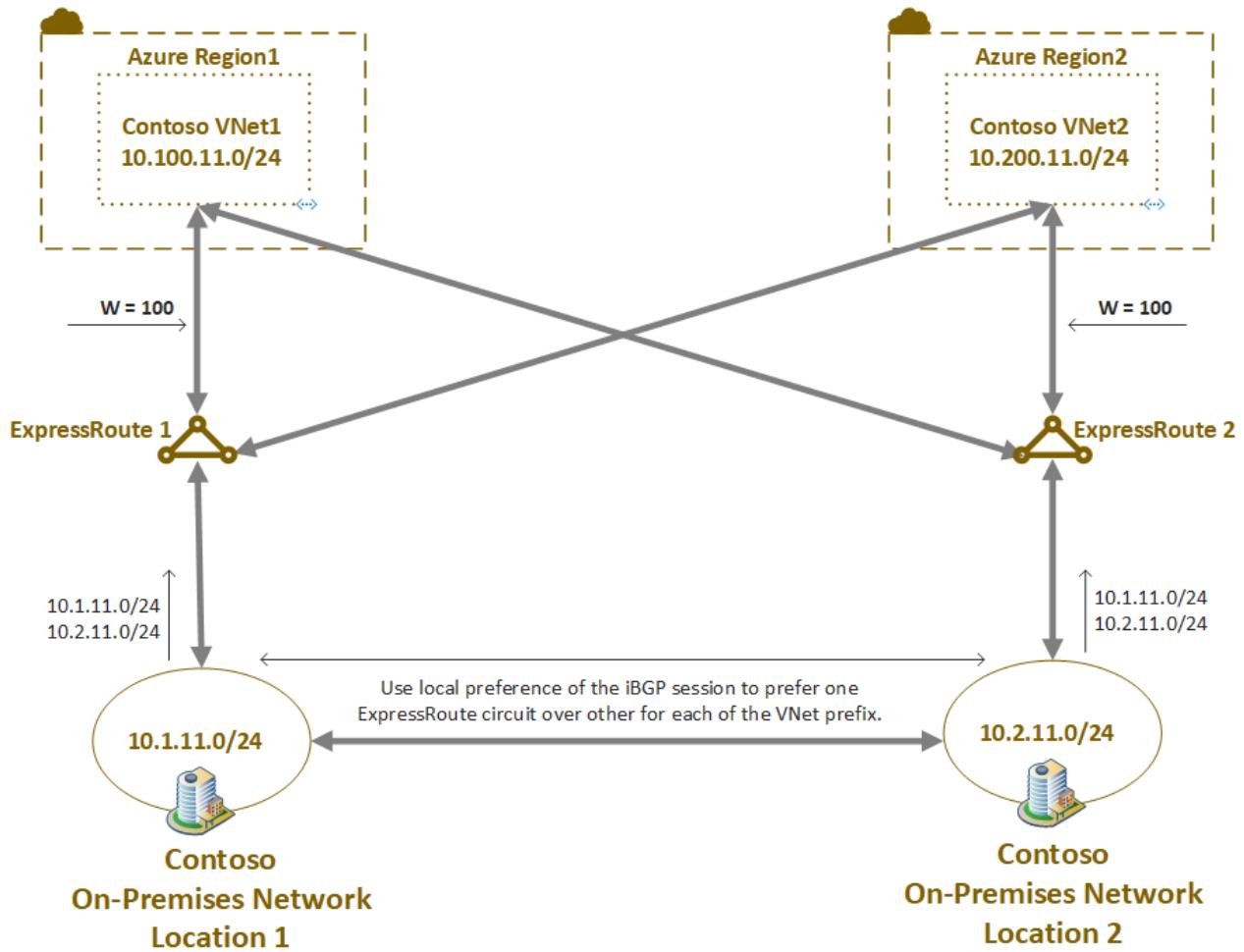
Scenario 1

In the first scenario, let's design disaster recovery such that all the traffic between an Azure region and on-premises network flow through the local ExpressRoute circuit in the steady state. If the local ExpressRoute circuit fails, then the remote ExpressRoute circuit is used for all the traffic flows between Azure and on-premises network.

Scenario 1 is illustrated in the following diagram. In the diagram, green lines indicate paths for traffic flow between VNet1 and on-premises networks. The blue lines indicate paths for traffic flow between VNet2 and on-premises networks. Solid lines indicate desired path in the steady-state and the dashed lines indicate traffic path in the failure of the corresponding ExpressRoute circuit that carries steady-state traffic flow.

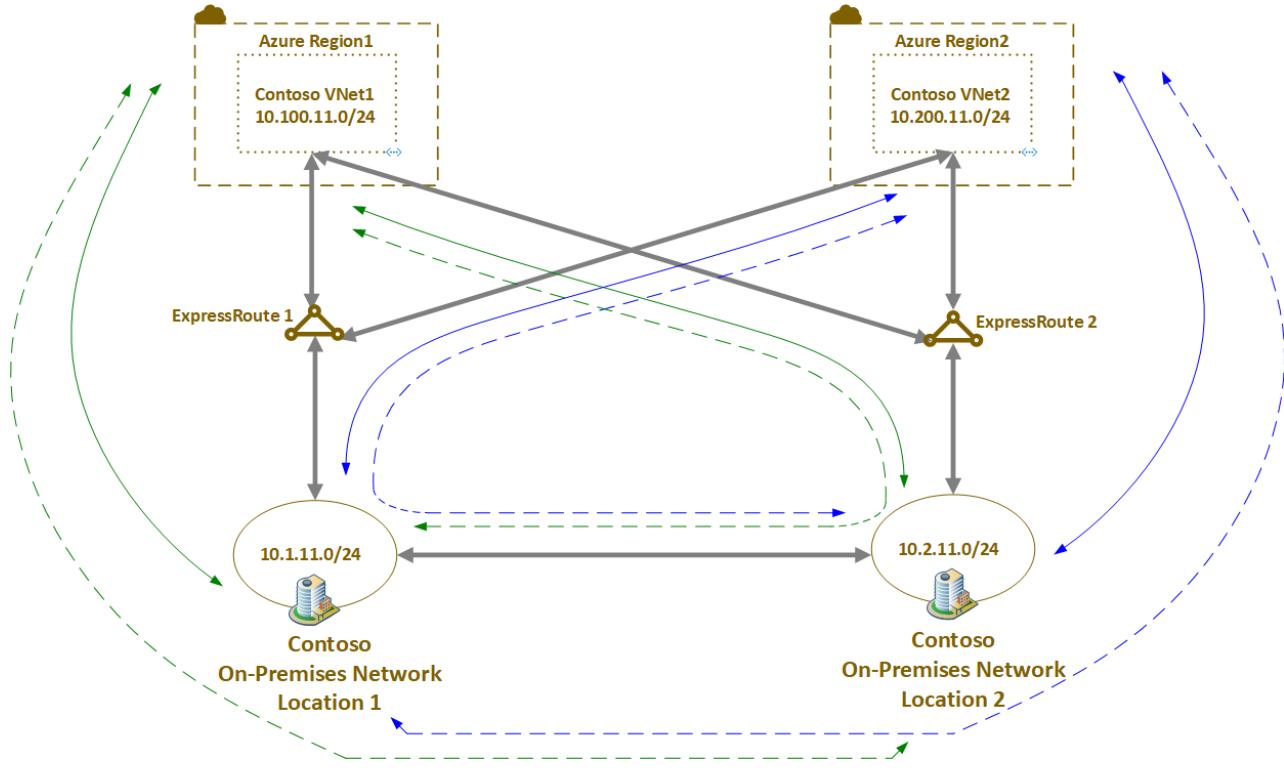


You can architect the scenario using connection weight to influence VNets to prefer connection to local peering location ExpressRoute for on-premises network bound traffic. To complete the solution, you need to ensure symmetrical reverse traffic flow. You can use local preference on the iBGP session between your BGP routers (on which ExpressRoute circuits are terminated on on-premises side) to prefer a ExpressRoute circuit. The solution is illustrated in the following diagram.

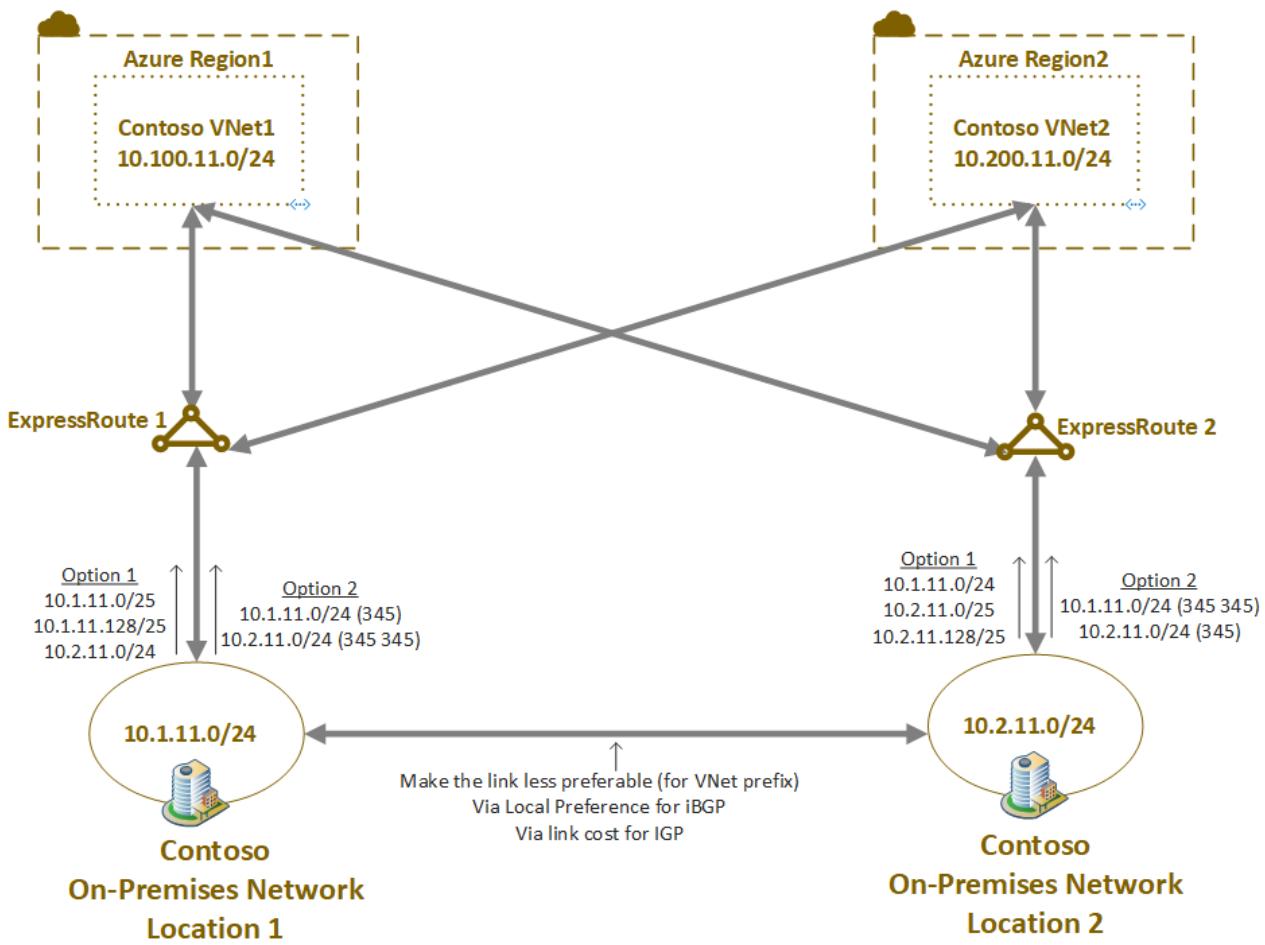


Scenario 2

The Scenario 2 is illustrated in the following diagram. In the diagram, green lines indicate paths for traffic flow between VNet1 and on-premises networks. The blue lines indicate paths for traffic flow between VNet2 and on-premises networks. In the steady-state (solid lines in the diagram), all the traffic between VNets and on-premises locations flow via Microsoft backbone for the most part, and flows through the interconnection between on-premises locations only in the failure state (dotted lines in the diagram) of an ExpressRoute.



The solution is illustrated in the following diagram. As illustrated, you can architect the scenario either using more specific route (Option 1) or AS-path prepend (Option 2) to influence VNet path selection. To influence on-premises network route selection for Azure bound traffic, you need configure the interconnection between the on-premises location as less preferable. Howe you configure the interconnection link as preferable depends on the routing protocol used within the on-premises network. You can use local preference with iBGP or metric with IGP (OSPF or IS-IS).



Next steps

In this article, we discussed how to design for disaster recovery of an ExpressRoute circuit private peering connectivity. The following articles addresses disaster recovery from applications and frontend access perspectives:

- [Enterprise-scale disaster recovery](#)
- [SMB disaster recovery with Azure Site Recovery](#)

Using S2S VPN as a backup for ExpressRoute private peering

2/7/2020 • 10 minutes to read • [Edit Online](#)

In the article titled [Designing for disaster recovery with ExpressRoute private peering](#), we discussed the need for backup connectivity solution for an ExpressRoute private peering connectivity and how to use geo-redundant ExpressRoute circuits for the purpose. In this article, let us consider how to leverage and maintain site-to-site (S2S) VPN as a back for ExpressRoute private peering.

Unlike geo-redundant ExpressRoute circuits, you can use ExpressRoute-VPN disaster recovery combination only in active-passive mode. A major challenge of using any backup network connectivity in the passive mode is that the passive connection would often fail alongside the primary connection. The common reason for the failures of the passive connection is lack of active maintenance. Therefore, in this article let's focus on how to verify and actively maintain S2S VPN connectivity that is backing up an ExpressRoute private peering.

NOTE

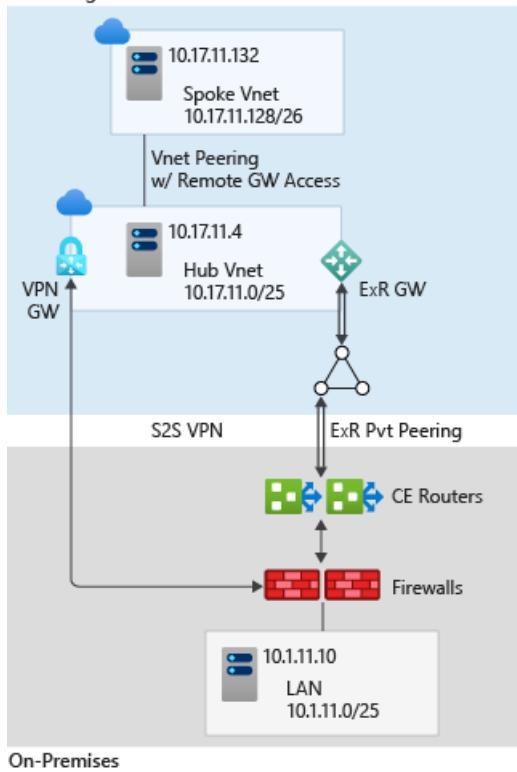
When a given route is advertised via both ExpressRoute and VPN, Azure would prefer routing over ExpressRoute.

In this article, let's see how to verify the connectivity both from the Azure perspective and the customer side network edge perspective. Ability to validate from either end will help irrespective of whether or not you manage the customer side network devices that peer with the Microsoft network entities.

Example topology

In our setup, we have an on-premises network connected to an Azure hub VNet via both an ExpressRoute circuit and a S2S VPN connection. The Azure hub VNet is in turn peered to a spoke VNet, as shown in the diagram below:

Azure Region 1



In the setup, the ExpressRoute circuit is terminated on a pair of "Customer Edge" (CE) routers at the on-premises. The on-premises LAN is connected to the CE routers via a pair of firewalls that operate in leader-follower mode. The S2S VPN is directly terminated on the firewalls.

The following table lists the key IP prefixes of the topology:

ENTITY	PREFIX
On-premises LAN	10.1.11.0/25
Azure Hub VNet	10.17.11.0/25
Azure spoke VNet	10.17.11.128/26
On-premises test server	10.1.11.10
Spoke VNet test VM	10.17.11.132
ExpressRoute primary connection p2p subnet	192.168.11.16/30
ExpressRoute secondary connection p2p subnet	192.168.11.20/30
VPN gateway primary BGP peer IP	10.17.11.76
VPN gateway secondary BGP peer IP	10.17.11.77
On-premises firewall VPN BGP peer IP	192.168.11.88
Primary CE router i/f towards firewall IP	192.168.11.0/31
Firewall i/f towards primary CE router IP	192.168.11.1/31

ENTITY	PREFIX
Secondary CE router i/f towards firewall IP	192.168.11.2/31
Firewall i/f towards secondary CE router IP	192.168.11.3/31

The following table lists the ASNs of the topology:

AUTONOMOUS SYSTEM	ASN
On-premises	65020
Microsoft Enterprise Edge	12076
Virtual Network GW (ExR)	65515
Virtual Network GW (VPN)	65515

High availability without asymmetry

Configuring for high availability

[Configure ExpressRoute and Site-to-Site coexisting connections](#) discusses how to configure the coexisting ExpressRoute circuit and S2S VPN connections. As we discussed in [Designing for high availability with ExpressRoute](#), to improve ExpressRoute high availability our setup maintains the network redundancy (avoids single point of failure) all the way up to the endpoints. Also both the primary and secondary connections of the ExpressRoute circuits are configured to operate in active-active mode by advertising the on-premises prefixes the same way through both the connections.

The on-premises route advertisement of the primary CE router through the primary connection of the ExpressRoute circuit is show below (Junos commands):

```
user@SEA-MX03-01> show route advertising-protocol bgp 192.168.11.18

Cust11.inet.0: 8 destinations, 8 routes (7 active, 0 holddown, 1 hidden)
  Prefix          Nexthop          MED      Lclpref      AS path
* 10.1.11.0/25      Self           I
```

The on-premises route advertisement of the secondary CE router through the secondary connection of the ExpressRoute circuit is show below (Junos commands):

```
user@SEA-MX03-02> show route advertising-protocol bgp 192.168.11.22

Cust11.inet.0: 8 destinations, 8 routes (7 active, 0 holddown, 1 hidden)
  Prefix          Nexthop          MED      Lclpref      AS path
* 10.1.11.0/25      Self           I
```

To improve the high availability of the backup connection, the S2S VPN is also configured in the active-active mode. The Azure VPN gateway configuration is shown below. Note as part of the VPN configuration VPN the BGP peer IP addresses of the gateway--10.17.11.76 and 10.17.11.77--are also listed.

The screenshot shows the Azure portal interface for managing a Virtual Network Gateway. The main title is "SEA-Cust11-VNet01-gw-vpn - Configuration". On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration (which is selected), Connections, Point-to-site configuration, Properties, Locks, and Export template. The main pane shows "Generation" set to Generation2, "SKU" set to VpnGw2, "Active-active mode" is enabled, and a checked checkbox for "Configure BGP ASN". Below that, the Autonomous system number (ASN) is set to 65515, and the BGP peer IP address(es) are listed as 10.17.11.76, 10.17.11.77.

The on-premises route is advertised by the firewalls to the primary and secondary BGP peers of the VPN gateway. The route advertisements are shown below (Junos):

```
user@SEA-SRX42-01> show route advertising-protocol bgp 10.17.11.76

Cust11.inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
  Prefix          Nexthop          MED      Lclpref    AS path
* 10.1.11.0/25      Self           I

{primary:node0}
user@SEA-SRX42-01> show route advertising-protocol bgp 10.17.11.77

Cust11.inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
  Prefix          Nexthop          MED      Lclpref    AS path
* 10.1.11.0/25      Self           I
```

NOTE

Configuring the S2S VPN in active-active mode not only provides high-availability to your disaster recovery backup network connectivity, but also provides higher throughput to the backup connectivity. In other words, configuring S2S VPN in active-active mode is recommended as it forces to create multiple underlying tunnels.

Configuring for symmetric traffic flow

We noted that when a given on-premises route is advertised via both ExpressRoute and S2S VPN, Azure would prefer the ExpressRoute path. To force Azure to prefer S2S VPN path over the coexisting ExpressRoute, you need to advertise more specific routes (longer prefix with bigger subnet mask) via the VPN connection. Our objective here is to use the VPN connections as back only. So, the default path selection behavior of Azure is in-line with our objective.

It is our responsibility to ensure that the traffic destined to Azure from on-premises also prefers ExpressRoute path over S2S VPN. The default local preference of the CE routers and firewalls in our on-premises setup is 100. So, by configuring the local preference of the routes received through the ExpressRoute private peerings greater than 100 (say 150), we can make the traffic destined to Azure prefer ExpressRoute circuit in the steady state.

The BGP configuration of the primary CE router that terminates the primary connection of the ExpressRoute circuit is shown below. Note the value of the local preference of the routes advertised over the iBGP session is configured to be 150. Similarly, we need to ensure the local preference of the secondary CE router that terminates the secondary connection of the ExpressRoute circuit is also configured to be 150.

```
user@SEA-MX03-01> show configuration routing-instances Cust11
description "Customer 11 VRF";
instance-type virtual-router;
interface xe-0/0/0:0.110;
interface ae0.11;
protocols {
    bgp {
        group ibgp {
            type internal;
            local-preference 150;
            neighbor 192.168.11.1;
        }
        group ebgp {
            peer-as 12076;
            bfd-liveness-detection {
                minimum-interval 300;
                multiplier 3;
            }
            neighbor 192.168.11.18;
        }
    }
}
```

The routing table of the on-premises firewalls confirms (shown below) that for the on-premises traffic that is destined to Azure the preferred path is over ExpressRoute in the steady state.

```

user@SEA-SRX42-01> show route table Cust11.inet.0 10.17.11.0/24

Cust11.inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.17.11.0/25      *[BGP/170] 2d 00:34:04, localpref 150
                     AS path: 12076 I, validation-state: unverified
                     > to 192.168.11.0 via reth1.11
                     to 192.168.11.2 via reth2.11
                     [BGP/170] 2d 00:34:01, localpref 150
                     AS path: 12076 I, validation-state: unverified
                     > to 192.168.11.2 via reth2.11
                     [BGP/170] 2d 21:12:13, localpref 100, from 10.17.11.76
                     AS path: 65515 I, validation-state: unverified
                     > via st0.118
                     [BGP/170] 2d 00:41:51, localpref 100, from 10.17.11.77
                     AS path: 65515 I, validation-state: unverified
                     > via st0.119
10.17.11.76/32    *[Static/5] 2d 21:12:16
                     > via st0.118
10.17.11.77/32    *[Static/5] 2d 00:41:56
                     > via st0.119
10.17.11.128/26   *[BGP/170] 2d 00:34:04, localpref 150
                     AS path: 12076 I, validation-state: unverified
                     > to 192.168.11.0 via reth1.11
                     to 192.168.11.2 via reth2.11
                     [BGP/170] 2d 00:34:01, localpref 150
                     AS path: 12076 I, validation-state: unverified
                     > to 192.168.11.2 via reth2.11
                     [BGP/170] 2d 21:12:13, localpref 100, from 10.17.11.76
                     AS path: 65515 I, validation-state: unverified
                     > via st0.118
                     [BGP/170] 2d 00:41:51, localpref 100, from 10.17.11.77
                     AS path: 65515 I, validation-state: unverified
                     > via st0.119

```

In the above route table, for the hub and spoke VNet routes--10.17.11.0/25 and 10.17.11.128/26--we see ExpressRoute circuit is preferred over VPN connections. The 192.168.11.0 and 192.168.11.2 are IPs on firewall interface towards CE routers.

Validation of route exchange over S2S VPN

Earlier in this article, we verified on-premises route advertisement of the firewalls to the primary and secondary BGP peers of the VPN gateway. Additionally, let's confirm Azure routes received by the firewalls from the primary and secondary BGP peers of the VPN gateway.

```

user@SEA-SRX42-01> show route receive-protocol bgp 10.17.11.76 table Cust11.inet.0

Cust11.inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
  Prefix          Nexthop          MED      Lclpref      AS path
  10.17.11.0/25    10.17.11.76                65515 I
  10.17.11.128/26  10.17.11.76                65515 I

{primary:node0}
user@SEA-SRX42-01> show route receive-protocol bgp 10.17.11.77 table Cust11.inet.0

Cust11.inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
  Prefix          Nexthop          MED      Lclpref      AS path
  10.17.11.0/25    10.17.11.77                65515 I
  10.17.11.128/26  10.17.11.77                65515 I

```

Similarly let's verify for on-premises network route prefixes received by the Azure VPN gateway.

```
PS C:\Users\user> Get-AzVirtualNetworkGatewayLearnedRoute -ResourceGroupName SEA-Cust11 -VirtualNetworkGatewayName SEA-Cust11-VNet01-gw-vpn | where {$_.Network -eq "10.1.11.0/25"} | select Network, NextHop, AsPath, Weight

Network      NextHop      AsPath      Weight
-----      -----      -----      -----
10.1.11.0/25 192.168.11.88 65020      32768
10.1.11.0/25 10.17.11.76 65020      32768
10.1.11.0/25 10.17.11.69 12076-65020 32769
10.1.11.0/25 10.17.11.69 12076-65020 32769
10.1.11.0/25 192.168.11.88 65020      32768
10.1.11.0/25 10.17.11.77 65020      32768
10.1.11.0/25 10.17.11.69 12076-65020 32769
10.1.11.0/25 10.17.11.69 12076-65020 32769
```

As seen above, the VPN gateway has routes received both by the primary and secondary BGP peers of the VPN gateway. It also has visibility over the routes received via primary and secondary ExpressRoute connections (the ones with AS-path prepended with 12076). To confirm the routes received via VPN connections, we need to know the on-premises BGP peer IP of the connections. In our setup under consideration, it is 192.168.11.88 and we do see the routes received from it.

Next, let's verify the routes advertised by the Azure VPN gateway to the on-premises firewall BGP peer (192.168.11.88).

```
PS C:\Users\user> Get-AzVirtualNetworkGatewayAdvertisedRoute -Peer 192.168.11.88 -ResourceGroupName SEA-Cust11 -VirtualNetworkGatewayName SEA-Cust11-VNet01-gw-vpn | select Network, NextHop, AsPath, Weight

Network      NextHop      AsPath      Weight
-----      -----      -----      -----
10.17.11.0/25 10.17.11.76 65515      0
10.17.11.128/26 10.17.11.76 65515      0
10.17.11.0/25 10.17.11.77 65515      0
10.17.11.128/26 10.17.11.77 65515      0
```

Failure to see route exchanges indicate connection failure. See [Troubleshooting: An Azure site-to-site VPN connection cannot connect and stops working](#) for help with troubleshooting the VPN connection.

Testing failover

Now that we have confirmed successful route exchanges over the VPN connection (control plane), we are set to switch traffic (data plane) from the ExpressRoute connectivity to the VPN connectivity.

NOTE

In production environments failover testing has to be done during scheduled network maintenance work-window as it can be service disruptive.

Prior to do the traffic switch, let's trace route the current path in our setup from the on-premises test server to the test VM in the spoke VNet.

```
C:\Users\PathLabUser>tracert 10.17.11.132

Tracing route to 10.17.11.132 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.1.11.1
 2  <1 ms    <1 ms    11 ms  192.168.11.0
 3  <1 ms    <1 ms    <1 ms  192.168.11.18
 4  *        *        *      Request timed out.
 5  6 ms    6 ms    5 ms   10.17.11.132

Trace complete.
```

The primary and secondary ExpressRoute point-to-point connection subnets of our setup are, respectively, 192.168.11.16/30 and 192.168.11.20/30. In the above trace route, in step 3 we see that we are hitting 192.168.11.18, which is the interface IP of the primary MSEE. Presence of MSEE interface confirms that as expected our current path is over the ExpressRoute.

As reported in the [Reset ExpressRoute circuit peerings](#), let's use the following powershell commands to disable both the primary and secondary peering of the ExpressRoute circuit.

```
$ckt = Get-AzExpressRouteCircuit -Name "expressroute name" -ResourceGroupName "SEA-Cust11"
$ckt.Peerings[0].State = "Disabled"
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The failover switch time depends on the BGP convergence time. In our setup, the failover switch takes a few seconds (less than 10). After the switch, repeat of the traceroute shows the following path:

```
C:\Users\PathLabUser>tracert 10.17.11.132

Tracing route to 10.17.11.132 over a maximum of 30 hops

 1  <1 ms    <1 ms    <1 ms  10.1.11.1
 2  *        *        *      Request timed out.
 3  6 ms    7 ms    9 ms   10.17.11.132

Trace complete.
```

The traceroute result confirms that the backup connection via S2S VPN is active and can provide service continuity if both the primary and secondary ExpressRoute connections fail. To complete the failover testing, let's enable the ExpressRoute connections back and normalize the traffic flow, using the following set of commands.

```
$ckt = Get-AzExpressRouteCircuit -Name "expressroute name" -ResourceGroupName "SEA-Cust11"
$ckt.Peerings[0].State = "Enabled"
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To confirm the traffic is switched back to ExpressRoute, repeat the traceroute and ensure that it is going through the ExpressRoute private peering.

Next steps

ExpressRoute is designed for high availability with no single point of failure within the Microsoft network. Still an ExpressRoute circuit is confined to a single geographical region and to a service provider. S2S VPN can be a good disaster recovery passive backup solution to an ExpressRoute circuit. For a dependable passive backup connection solution, regular maintenance of the passive configuration and periodical validation the connection are important. It is essential not to let the VPN configuration become stale, and to periodically (say every quarter) repeat the validation and failover test steps described in this article during maintenance window.

To enable monitoring and alerts based on VPN gateway metrics, see [Set up alerts on VPN Gateway metrics](#).

To expedite BGP convergence following an ExpressRoute failure, [Configure BFD over ExpressRoute](#).

Tutorial: Create and modify an ExpressRoute circuit

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

This article helps you create an ExpressRoute circuit using the Azure portal and the Azure Resource Manager deployment model. You can also check the status, update, delete, or deprovision a circuit.

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.

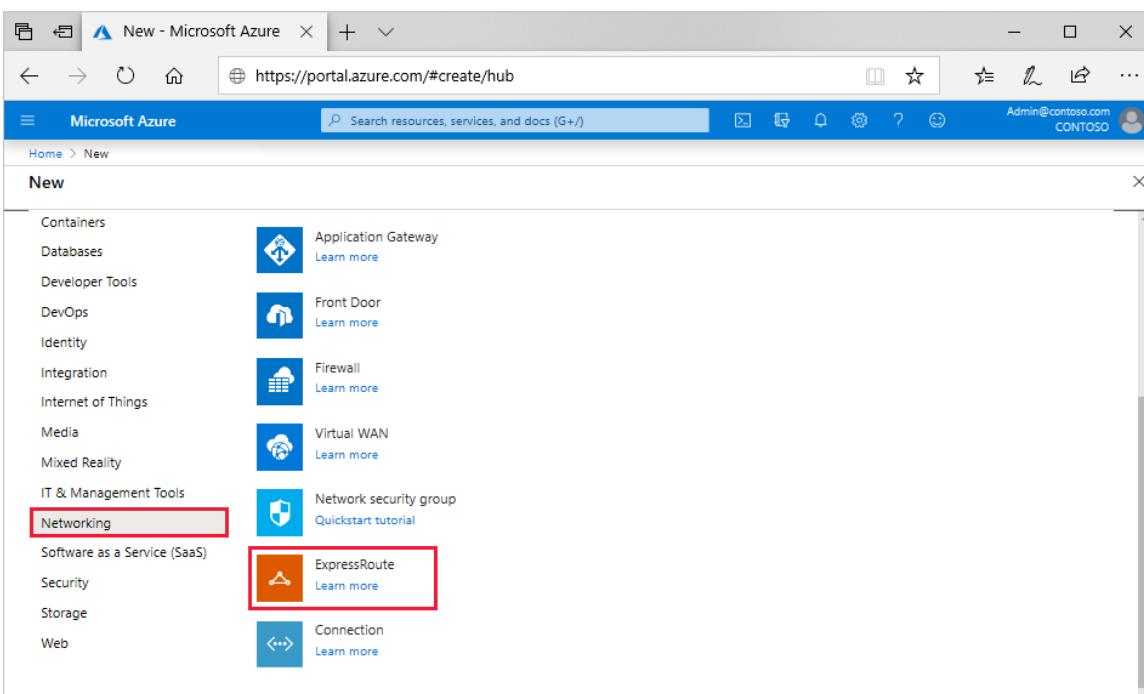
2. Create a new ExpressRoute circuit

IMPORTANT

Your ExpressRoute circuit is 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.

You can create an ExpressRoute circuit by selecting the option to create a new resource.

1. On the Azure portal menu or from the **Home** page, select **Create a resource**. Select **Networking > ExpressRoute**, as shown in the following image:



2. After you click **ExpressRoute**, you'll see the **Create ExpressRoute circuit** page. When you're filling in

the values on this page, make sure that you specify the correct SKU tier (Standard, or Premium) and data metering billing model (Unlimited or Metered).

Create ExpressRoute circuit X

Create new or import from classic i

Create new Import

* Circuit name
TestCkt ✓

* Provider i
Equinix

* Peering location i
Seattle

* Bandwidth i
100Mbps

* SKU i
Standard Premium

* Billing model i
Unlimited Metered

Allow classic operations i

* Subscription
Windows Azure Internal Consumption

* Resource group
 Create new Use existing
DemoRG ✓

* Location
West US

- **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**.

IMPORTANT

You can't change the type from **Unlimited** to **Metered**.

- **Peering Location** is the physical location where you are peering with Microsoft.

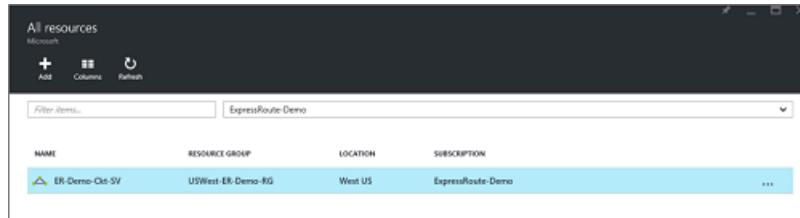
IMPORTANT

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

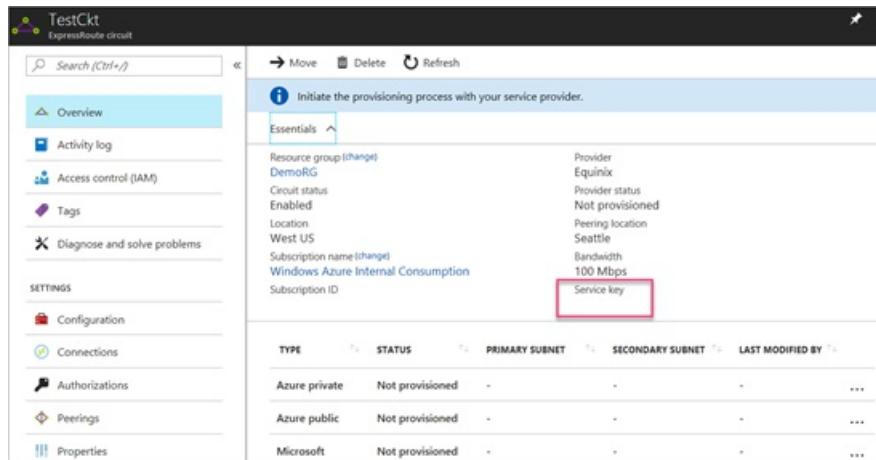
You can view all the circuits that you created by selecting **All resources** on the left-side menu.



A screenshot of the Azure portal's 'All resources' blade. At the top, there are buttons for 'Add', 'Columns', and 'Refresh'. Below that is a search bar with 'Filter items...' and a dropdown set to 'ExpressRoute-Demo'. The main table has columns for NAME, RESOURCE GROUP, LOCATION, and SUBSCRIPTION. One row is visible, showing 'ER-Demo-Okt-SV' under NAME, 'USWest-ER-Demo-RG' under RESOURCE GROUP, 'West US' under LOCATION, and 'ExpressRoute-Demo' under SUBSCRIPTION.

View the properties

You can view the properties of the circuit by selecting it. On the **Overview** page for your circuit, the service key appears in the service key field. You must copy the service key for your circuit and pass it down to the service provider to complete the provisioning process. The circuit service key is specific to your circuit.



A screenshot of the Azure portal showing the 'TestCkt' circuit overview page. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Connections, Authorizations, Peerings, and Properties. The main area shows a message: 'Initiate the provisioning process with your service provider.' Below this is a table with the following data:

Resource group (change)	Provider
DemoRG	Equinix
Circuit status	Provider status
Enabled	Not provisioned
Location	Peering location
West US	Seattle
Subscription name (change)	Bandwidth
Windows Azure Internal Consumption	100 Mbps
Subscription ID	Service key

The 'Service key' field is highlighted with a red box. Below the table is a table of peerings:

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

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

On this page, **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 is in the following state:

Provider status: Not provisioned

Circuit status: Enabled

Type	Status	Primary Subnet	Secondary Subnet	Last Modified By
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

The circuit changes 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 configures and manages 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 and viewing the Overview page.

Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity. You can modify the bandwidth, SKU, billing model and allow classic operations on the **Configuration** page. For information on limits and limitations, see the [ExpressRoute FAQ](#).

You can perform the following tasks with no downtime:

- Enable or disable an ExpressRoute Premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit, provided there is capacity available on the port.

IMPORTANT

Downgrading the bandwidth of a circuit is not supported.

- Change the metering plan from *Metered Data* to *Unlimited Data*.

IMPORTANT

Changing the metering plan from Unlimited Data to Metered Data is not supported.

- You can enable and disable *Allow Classic Operations*.

IMPORTANT

You may have to recreate the ExpressRoute circuit if there is inadequate capacity on the existing port. You cannot upgrade the circuit if there is no additional capacity available at that location.

Although you can seamlessly upgrade the bandwidth, 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.

Disabling the Premium add-on operation can fail if you're using resources that are greater than what is permitted for the standard circuit.

To modify an ExpressRoute circuit, click **Configuration**.

The screenshot shows the Azure portal interface for managing an ExpressRoute circuit. The left sidebar has a search bar and links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration (selected and highlighted with a red box), Connections, Authorizations, Peerings, Properties, Locks, and Automation script. The main content area shows configuration settings: Bandwidth (100 Mbps), SKU (Standard Premium), Billing model (Unlimited Metered), and Allow classic operations (Disabled Enabled). A red box highlights the SKU, Billing model, and Allow classic operations settings.

Deprovisioning and deleting an ExpressRoute circuit

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

- 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 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 delete the circuit. This stops billing for the circuit.

Next steps

After you create your circuit, continue with the following next steps:

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Create and modify an ExpressRoute circuit using PowerShell

1/8/2020 • 10 minutes to read • [Edit Online](#)

This article helps you create an ExpressRoute circuit using PowerShell cmdlets and the Azure Resource Manager deployment model. You can also check the status, update, delete, or deprovision a circuit.

Before you begin

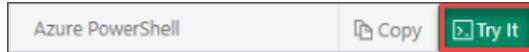
Before you begin, review the [prerequisites](#) and [workflows](#) before you begin configuration.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Create and provision an ExpressRoute circuit

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

If you are using the Azure Cloud Shell, you sign in to your Azure account automatically after clicking 'Try it'. To sign in locally, open your PowerShell console with elevated privileges and run the cmdlet to connect.

```
Connect-AzAccount
```

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

```
Get-AzSubscription
```

Specify the subscription that you want to use.

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

2. 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-AzExpressRouteServiceProvider** returns this information, which you'll use in later steps:

```
Get-AzExpressRouteServiceProvider
```

Check to see if your connectivity provider is listed there. Make a note of the following information, which you need 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-AzResourceGroup -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. Use the following example to request a new service key:

```
New-AzExpressRouteCircuit -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 circuit is [Local](#), [Standard](#) or [Premium](#). You can specify *Local*, *Standard* or *Premium*.
- SKU family determines the billing type. You can specify *Metereddata* for a metered data plan and *Unlimiteddata* for an unlimited data plan. You can change the billing type from *Metereddata* to *Unlimiteddata*, but you can't change the type from *Unlimiteddata* to *Metereddata*. A *Local* circuit is always *Unlimiteddata*.

IMPORTANT

Your ExpressRoute circuit is 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 command:

```
get-help New-AzExpressRouteCircuit -detailed
```

4. List all ExpressRoute circuits

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

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

The response looks similar to the following example:

```
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 : NotProvisioned
ServiceProviderNotes :
ServiceProviderProperties :
    "ServiceProviderName": "Equinix",
    "PeeringLocation": "Silicon Valley",
    "BandwidthInMbps": 200
}
ServiceKey :
*****
Peerings : []
```

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

```
Get-AzExpressRouteCircuit
```

The response looks similar to the following example:

```

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 : NotProvisioned
ServiceProviderNotes :
ServiceProviderProperties :
    "ServiceProviderName": "Equinix",
    "PeeringLocation": "Silicon Valley",
    "BandwidthInMbps": 200
}
ServiceKey :
*****
Peerings : []

```

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 [Workflows](#).

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

```

ServiceProviderProvisioningState : NotProvisioned
CircuitProvisioningState : Enabled

```

The circuit changes 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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup"
```

The response looks similar to the following example:

```

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 : []

```

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 configures and manages 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-AzExpressRouteCircuit** cmdlet. Making the call with no parameters lists all the circuits.

```
Get-AzExpressRouteCircuit
```

The response is similar to the following example:

```

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 : []

```

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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName "ExpressRouteResourceGroup"
```

The response looks similar to the following example:

```

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 : []

```

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

```
get-help Get-AzExpressRouteCircuit -detailed
```

Modifying an ExpressRoute circuit

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

You can do the following tasks with no downtime:

- Enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit provided there is capacity available on the port. Downgrading the bandwidth of a circuit is not supported.
- Change the metering plan from Metered Data to Unlimited Data. 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, see 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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
$ckt.Sku.Tier = "Premium"  
$ckt.sku.Name = "Premium_MeteredData"  
  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The circuit now has the ExpressRoute premium add-on features enabled. We 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

If you're using resources that are greater than what is permitted for the standard circuit, this operation can fail.

Note the following information:

- 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, your update request fails, and we bill you at premium rates.
- You must unlink all virtual networks in other geopolitical regions. If you don't do this, your update request fails, and we 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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
$ckt.Sku.Tier = "Standard"  
$ckt.sku.Name = "Standard_MeteredData"  
  
Set-AzExpressRouteCircuit -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 may have to recreate the ExpressRoute circuit if there is inadequate capacity on the existing port. You cannot upgrade the circuit if there is no additional capacity available at that location.

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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
$ckt.ServiceProviderProperties.BandwidthInMbps = 1000  
  
Set-AzExpressRouteCircuit -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-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
$ckt.Sku.Family = "UnlimitedData"  
$ckt.sku.Name = "Premium_UnlimitedData"  
  
Set-AzExpressRouteCircuit -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 information:

- 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 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 delete the circuit. This stops billing for the circuit.

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

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

Next steps

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

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Create and modify an ExpressRoute circuit using CLI

11/14/2019 • 8 minutes to read • [Edit Online](#)

This article describes how to create an Azure ExpressRoute circuit by using the Command Line Interface (CLI). This article also shows you how to check the status, update, or delete and deprovision a circuit. If you want to use a different method to work with ExpressRoute circuits, you can select the article from the following list:

Before you begin

- Before beginning, install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install the Azure CLI](#) and [Get Started with Azure CLI](#).
- 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. If you use the CloudShell "Try It", you are signed in automatically. Use the following examples to help you connect:

```
az login
```

Check the subscriptions for the account.

```
az account list
```

Select the subscription for which you want to create an ExpressRoute circuit.

```
az account set --subscription "<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 CLI command `az network express-route list-service-providers` returns this information, which you'll use in later steps:

```
az network express-route list-service-providers
```

The response is similar to the following example:

```
[
  {
    "bandwidthsOffered": [
      {
        "offerName": "50Mbps",
        "valueInMbps": 50
      },
      {
        "offerName": "100Mbps",
        "valueInMbps": 100
      },
      {
        "offerName": "200Mbps",
        "valueInMbps": 200
      },
      {
        "offerName": "500Mbps",
        "valueInMbps": 500
      },
      {
        "offerName": "1Gbps",
        "valueInMbps": 1000
      },
      {
        "offerName": "2Gbps",
        "valueInMbps": 2000
      },
      {
        "offerName": "5Gbps",
        "valueInMbps": 5000
      },
      {
        "offerName": "10Gbps",
        "valueInMbps": 10000
      }
    ],
    "id": "/subscriptions//resourceGroups//providers/Microsoft.Network/expressRouteServiceProviders/",
    "location": null,
    "name": "AARNet",
    "peeringLocations": [
      "Melbourne",
      "Sydney"
    ],
    "provisioningState": "Succeeded",
    "resourceGroup": "",
    "tags": null,
    "type": "Microsoft.Network/expressRouteServiceProviders"
  },
]
```

Check the response to see if your connectivity provider is listed. Make a note of the following information, which you will need when you create a circuit:

- Name
- PeeringLocations
- BandwidthsOffered

You're now ready to create an ExpressRoute circuit.

3. Create an ExpressRoute circuit

IMPORTANT

Your ExpressRoute circuit is billed from the moment a service key is issued. Perform this operation when the connectivity provider is ready to provision the circuit.

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

```
az group create -n ExpressRouteResourceGroup -l "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.

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

- SKU tier determines whether an ExpressRoute circuit is [Local](#), Standard or [Premium](#). You can specify *Local*, *Standard* or *Premium*.
- SKU family determines the billing type. You can specify *Metereddata* for a metered data plan and *Unlimiteddata* for an unlimited data plan. You can change the billing type from *Metereddata* to *Unlimiteddata*, but you can't change the type from *Unlimiteddata* to *Metereddata*. A *Local* circuit is *Unlimiteddata* only.

Your ExpressRoute circuit is billed from the moment a service key is issued. The following example is a request for a new service key:

```
az network express-route create --bandwidth 200 -n MyCircuit --peering-location "Silicon Valley" -g ExpressRouteResourceGroup --provider "Equinix" -l "West US" --sku-family MeteredData --sku-tier Standard
```

The response contains the service key.

4. List all ExpressRoute circuits

To get a list of all the ExpressRoute circuits that you created, run the `az network express-route list` command. You can retrieve this information at any time by using this command. To list all circuits, make the call with no parameters.

```
az network express-route list
```

Your service key is listed in the *ServiceKey* field of the response.

```
"allowClassicOperations": false,
"authorizations": [],
"circuitProvisioningState": "Enabled",
"etag": "W/\"1262c492-ffef-4a63-95a8-a6002736b8c4\"",
"gatewayManagerEtag": null,
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit",
"location": "westus",
"name": "MyCircuit",
"peerings": [],
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"serviceKey": "1d05cf70-1db5-419f-ad86-1ca62c3c125b",
"serviceProviderNotes": null,
"serviceProviderProperties": {
    "bandwidthInMbps": 200,
    "peeringLocation": "Silicon Valley",
    "serviceProviderName": "Equinix"
},
"serviceProviderProvisioningState": "NotProvisioned",
"sku": {
    "family": "UnlimitedData",
    "name": "Standard_MeteredData",
    "tier": "Standard"
},
"tags": null,
"type": "Microsoft.Network/expressRouteCircuits"]
```

You can get detailed descriptions of all the parameters by running the command using the '-h' parameter.

```
az network express-route list -h
```

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. The status provides the state on the Microsoft side. For more information, see the [Workflows article](#).

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

```
"serviceProviderProvisioningState": "NotProvisioned"
"circuitProvisioningState": "Enabled"
```

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

```
"serviceProviderProvisioningState": "Provisioning"
"circuitProvisioningState": "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:

```
az network express-route show --resource-group ExpressRouteResourceGroup --name MyCircuit
```

The response is similar to the following example:

```
"allowClassicOperations": false,
"authorizations": [],
"circuitProvisioningState": "Enabled",
"etag": "W/\"1262c492-ffef-4a63-95a8-a6002736b8c4\"",
"gatewayManagerEtag": null,
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit",
"location": "westus",
"name": "MyCircuit",
"peerings": [],
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"serviceKey": "1d05cf70-1db5-419f-ad86-1ca62c3c125b",
"serviceProviderNotes": null,
"serviceProviderProperties": {
    "bandwidthInMbps": 200,
    "peeringLocation": "Silicon Valley",
    "serviceProviderName": "Equinix"
},
"serviceProviderProvisioningState": "NotProvisioned",
"sku": {
    "family": "UnlimitedData",
    "name": "Standard_MeteredData",
    "tier": "Standard"
},
"tags": null,
"type": "Microsoft.Network/expressRouteCircuits"]
```

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 configures and manages 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.

Modifying an ExpressRoute circuit

You can modify certain properties of an ExpressRoute circuit without impacting connectivity. You can make the following changes with no downtime:

- You can enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- You can increase the bandwidth of your ExpressRoute circuit provided there is capacity available on the port. However, downgrading the bandwidth of a circuit is not supported.
- You can change the metering plan from Metered Data to Unlimited Data. However, 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, see 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 command:

```
az network express-route update -n MyCircuit -g ExpressRouteResourceGroup --sku-tier Premium
```

The circuit now has the ExpressRoute premium add-on features enabled. We 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.

Before disabling the ExpressRoute premium add-on, understand the following criteria:

- Before you downgrade from premium to standard, you must make sure that you have fewer than 10 virtual networks linked to the circuit. If you have more than 10, your update request fails, and we bill you at premium rates.
- You must unlink all virtual networks in other geopolitical regions. If you don't unlink all your virtual networks, your update request fails and we 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. The session won't be reenabled until the number of advertised prefixes is below 4,000.

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

```
az network express-route update -n MyCircuit -g ExpressRouteResourceGroup --sku-tier Standard
```

To update the ExpressRoute circuit bandwidth

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

IMPORTANT

If there is inadequate capacity on the existing port, you may have to recreate the ExpressRoute circuit. You cannot upgrade the circuit if there is no additional capacity available at that location.

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

```
az network express-route update -n MyCircuit -g ExpressRouteResourceGroup --bandwidth 1000
```

Your circuit is sized up on the Microsoft side. Next, you must contact your connectivity provider to update configurations on their side to match this change. After you make this notification, we 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 example:

```
az network express-route update -n MyCircuit -g ExpressRouteResourceGroup --sku-family UnlimitedData
```

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

To deprovision and delete an ExpressRoute circuit, make sure you understand the following criteria:

- 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 continue to reserve resources and bill you until the service provider completes deprovisioning the circuit and notifies us.
- You can delete the circuit if the service provider has deprovisioned the circuit. When a circuit is deprovisioned, the service provider provisioning state is set to **Not provisioned**. This stops billing for the circuit.

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

```
az network express-route delete -n MyCircuit -g ExpressRouteResourceGroup
```

Next steps

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

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Create an ExpressRoute circuit by using Azure Resource Manager template

1/14/2020 • 4 minutes to read • [Edit Online](#)

Learn how to create an ExpressRoute circuit by deploying an Azure Resource Manager template by using Azure PowerShell. For more information on developing Resource Manager templates, see [Resource Manager documentation](#) and the [template reference](#).

Before you begin

- Review the [prerequisites](#) and [workflows](#) before you begin configuration.
- 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

Azure Quickstart templates has a good collection of Resource Manager template. You use one of the [existing templates](#) to create an ExpressRoute circuit.

```
{  
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "parameters": {  
        "circuitName": {  
            "type": "string",  
            "metadata": {  
                "description": "This is the name of the ExpressRoute circuit"  
            }  
        },  
        "serviceProviderName": {  
            "type": "string",  
            "metadata": {  
                "description": "This is the name of the ExpressRoute Service Provider. It must exactly match one of  
the Service Providers from List ExpressRoute Service Providers API call."  
            }  
        },  
        "peeringLocation": {  
            "type": "string",  
            "metadata": {  
                "description": "This is the name of the peering location and not the ARM resource location. It must  
exactly match one of the available peering locations from List ExpressRoute Service Providers API call."  
            }  
        },  
        "bandwidthInMbps": {  
            "type": "int",  
            "metadata": {  
                "description": "This is the bandwidth in Mbps of the circuit being created. It must exactly match one  
of the available bandwidth offers List ExpressRoute Service Providers API call."  
            }  
        },  
        "sku_tier": {  
            "type": "string",  
            "defaultValue": "Standard",  
            "allowedValues": [  
                "Standard",  
                "Premium"  
            ]  
        }  
    }  
}
```

```

    ],
    "metadata": {
      "description": "Chosen SKU Tier of ExpressRoute circuit. Choose from Premium or Standard SKU tiers."
    }
  },
  "sku_family": {
    "type": "string",
    "defaultValue": "MeteredData",
    "allowedValues": [
      "MeteredData",
      "UnlimitedData"
    ],
    "metadata": {
      "description": "Chosen SKU family of ExpressRoute circuit. Choose from MeteredData or UnlimitedData SKU families."
    }
  },
  "location": {
    "type": "string",
    "defaultValue": "[resourceGroup().location]",
    "metadata": {
      "description": "Location for all resources."
    }
  }
},
"resources": [
  {
    "apiVersion": "2019-04-01",
    "type": "Microsoft.Network/expressRouteCircuits",
    "name": "[parameters('circuitName')]",
    "location": "[parameters('location')]",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    },
    "sku": {
      "name": "[concat(parameters('sku_tier'), '_', parameters('sku_family'))]",
      "tier": "[parameters('sku_tier')]",
      "family": "[parameters('sku_family')]"
    },
    "properties": {
      "serviceProviderProperties": {
        "serviceProviderName": "[parameters('serviceProviderName')]",
        "peeringLocation": "[parameters('peeringLocation')]",
        "bandwidthInMbps": "[parameters('bandwidthInMbps')]"
      }
    }
  }
]
}

```

To see more related templates, select [here](#).

To create an ExpressRoute Circuit by deploying a template:

1. Select **Try it** from the following code block, and then follow the instructions to sign in to the Azure Cloud shell.

```

$circuitName = Read-Host -Prompt "Enter a circuit name"
$location = Read-Host -Prompt "Enter the location (i.e. centralus)"
$resourceGroupName = "${circuitName}rg"
$templateUri = "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-expressroute-circuit-create/azuredploy.json"

$serviceProviderName = "Equinix"
$peeringLocation = "Silicon Valley"
$bandwidthInMbps = 500
$sku_tier = "Premium"
$sku_family = "MeteredData"

New-AzResourceGroup -Name $resourceGroupName -Location $location
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateUri $templateUri -
circuitName $circuitName -serviceName $serviceProviderName -peeringLocation $peeringLocation -
bandwidthInMbps $bandwidthInMbps -sku_tier $sku_tier -sku_family $sku_family

Write-Host "Press [ENTER] to continue ..."

```

- **SKU tier** determines whether an ExpressRoute circuit is [Local](#), Standard or [Premium](#). You can specify *Local*, *Standard* or *Premium*.
- **SKU family** determines the billing type. You can specify *Metereddata* for a metered data plan and *Unlimiteddata* for an unlimited data plan. You can change the billing type from *Metereddata* to *Unlimiteddata*, but you can't change the type from *Unlimiteddata* to *Metereddata*. A *Local* circuit is *Unlimiteddata* only.
- **Peering Location** is the physical location where you are peering with Microsoft.

IMPORTANT

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.

The resource group name is the service bus namespace name with **rg** appended.

2. Select **Copy** to copy the PowerShell script.
3. Right-click the shell console, and then select **Paste**.

It takes a few moments to create an event hub.

Azure PowerShell is used to deploy the template in this tutorial. For other template deployment methods, see:

- [By using the Azure portal](#).
- [By using Azure CLI](#).
- [By using REST API](#).

Deprovisioning and deleting an ExpressRoute circuit

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

- 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 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 delete the circuit. This stops billing for the circuit.

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

```
$circuitName = Read-Host -Prompt "Enter the same circuit name that you used earlier"  
$resourceGroupName = "${circuitName}rg"  
  
Remove-AzExpressRouteCircuit -ResourceGroupName $resourceGroupName -Name $circuitName
```

Next steps

After you create your circuit, continue with the following next steps:

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Create and modify peering for an ExpressRoute circuit

12/17/2019 • 7 minutes to read • [Edit Online](#)

This article helps you create and manage routing configuration for an Azure Resource Manager (ARM) ExpressRoute circuit, using the Azure portal. You can also check the status, update, or delete and deprovision peerings for an ExpressRoute circuit. If you want to use a different method to work with your circuit, select an article from the following list:

You can configure private peering and Microsoft peering for an ExpressRoute circuit (Azure public peering is deprecated for new circuits). Peerings can be configured in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time. For more information about routing domains and peerings, see [ExpressRoute routing domains](#). For information about public peering, see [ExpressRoute public peering](#).

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. In order to configure peering(s), the ExpressRoute circuit must be in a provisioned and enabled state.
- If you plan to use a shared key/MD5 hash, be sure to use this on both sides of the tunnel and limit the number of alphanumeric characters to a maximum of 25. Special characters are not supported.

These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider that offers managed Layer 3 services (typically an IPVPN, like MPLS), your connectivity provider configures and manages 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. Check with your service provider before configuring BGP peerings.

Microsoft peering

This section helps you create, get, update, and delete the Microsoft peering configuration for an ExpressRoute circuit.

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through the Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit. For more information, see [Configure a route filter for Microsoft peering](#).

To create Microsoft peering

1. Configure the ExpressRoute circuit. Check the **Provider status** to ensure that the circuit is fully

provisioned by the connectivity provider before continuing further.

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

Circuit - Provider status: Not provisioned

The screenshot shows the Azure portal interface for an ExpressRoute circuit named 'ER-Demo-Ckt'. The left sidebar has navigation links like Home, Search, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration, Connections, Authorizations, Peering, Properties, Locks), and Help & support. The main content area displays the circuit's details: Resource group (USWest-ER-Demo-RG), Circuit status (Enabled), Location (West US 2), Subscription (ExpressRoute-Lab), Subscription ID (4bffb15-d414-4874-a2e4), and Tags (Click here to add tags). On the right, there's a table for Peering settings with rows for Azure private, Azure public, and Microsoft, all showing 'Not provisioned' status. A callout box highlights the 'Provider status' field as 'Not provisioned'.

Circuit - Provider status: Provisioned

The screenshot shows the same 'ER-Demo-Ckt' ExpressRoute circuit overview as the previous one, but with a different provider status. The 'Provider status' field is now highlighted in red as 'Provisioned'. The rest of the circuit details and the Peering table remain the same.

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. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID. For both Primary and Secondary links you must use 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. If you plan to send a set of prefixes, you can send a comma-separated list. These prefixes must be registered to you in an RIR / IRR.
- **Optional** - 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.
- Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
- **Optional** - An MD5 hash if you choose to use one.

3. You can select the peering you wish to configure, as shown in the following example. Select the Microsoft peering row.

Resource group (change) : USWest-ER-Demo-RG
Circuit status : Enabled
Location : West US 2
Subscription (change) : ExpressRoute-Lab
Subscription ID : 4bffb15-d414-4874
Tags (change) : Click here to add tags

Provider : Equinix
Provider status : Provisioned
Peering location : Seattle
Bandwidth : 200 Mbps
Service key : 74b7c74d-4b76-4a23

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	-
Azure public	Not provisioned	-	-	-
Microsoft	Not provisioned	-	-	-

4. Configure Microsoft peering. **Save** the configuration once you have specified all parameters. The following image shows an example configuration:

Microsoft peering

To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. Learn More.

* Peer ASN

* Primary subnet

* Secondary subnet

* VLAN ID

* Advertised public prefixes Status: Not configured

Customer ASN

Routing registry name

Shared key

IMPORTANT

Microsoft verifies if the specified 'Advertised public prefixes' and 'Peer ASN' (or 'Customer ASN') are assigned to you in the Internet Routing Registry. If you are getting the public prefixes from another entity and if the assignment is not recorded with the routing registry, the automatic validation will not complete and will require manual validation. If the automatic validation fails, you will see the message 'Validation needed'.

If you see the message 'Validation needed', collect the document(s) that show the public prefixes are assigned to your organization by the entity that is listed as the owner of the prefixes in the routing registry and submit these documents for manual validation by opening a support ticket as shown below.

If your circuit gets to a 'Validation needed' state, 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 in the

following example:

Home > Help + support - New support request

Help + support - New support request

Search (Ctrl+ /)

Overview

Support

New support request

All support requests

Support Plans

Service Health

Advisor

Get started with Azure

Basics Solutions Details Review + create

Create a new support request to get assistance with billing, subscription, technical or quota management issues. Complete the Basics tab by selecting the options that best describe your problem. Providing detailed, accurate information can help to solve your issues faster.

* Issue type: Technical

* Subscription: ExpressRoute-Lab (4bfffbb15-d414-4874-a2e4-c548c6d45e...) Can't find your subscription? Show more ⓘ

* Service: My services (ExpressRoute)

* Resource: ER-Demo-Ckt

* Problem type: Configuration and Setup

* Problem subtype: ExpressRoute Peerings

* Subject: Proof of ownership for public prefixes.

- After the configuration has been accepted successfully, you'll see something similar to the following image:

Home > ER-Demo-Ckt > Microsoft peering

Microsoft peering

ER-Demo-Ckt

Save Discard Delete

To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. Learn More.

* Peer ASN ⓘ 394749 ✓

* Primary subnet ⓘ 64.191.192.240/30 ✓

* Secondary subnet ⓘ 64.191.192.244/30 ✓

* VLAN ID ⓘ 152 ✓

* Advertised public prefixes ⓘ 64.191.192.224/28 ✓ Status: Configured

Customer ASN ⓘ 394749 ✓

Routing registry name ⓘ ARIN ✓

Shared key

To view Microsoft peering details

You can view the properties of Microsoft peering by selecting the row for the peering.

The screenshot shows the Azure portal interface for managing a circuit named 'ER-Demo-Ckt'. In the left sidebar, under 'Settings', the 'Peering' section is selected. The main content area displays the 'Peerings' table. A single row for 'Microsoft' peering is highlighted with a red border. The table columns include TYPE, STATUS, PRIMARY SUBNET, SECONDARY SUBNET, and LAST MODIFIED BY. The 'Microsoft' row shows STATUS as 'Provisioned', PRIMARY SUBNET as '64.191.192.240/30', SECONDARY SUBNET as '64.191.192.244/30', and LAST MODIFIED BY as 'Customer'.

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Provisioned	64.191.192.240/30	64.191.192.244/30	Customer

To update Microsoft peering configuration

You can select the row for the peering that you want to modify, then modify the peering properties and save your modifications.

The screenshot shows the 'Microsoft peering' configuration page for the 'ER-Demo-Ckt' circuit. The page includes fields for Peer ASN (394749), Primary subnet (64.191.192.240/30), Secondary subnet (64.191.192.244/30), VLAN ID (152), Advertised public prefixes (64.191.192.224/28 with status 'Configured'), Customer ASN (394749), and Routing registry name (APNIC). A note at the top states: 'To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. Learn More.' A 'Save' button is visible at the top left.

To delete Microsoft peering

You can remove your peering configuration by clicking the delete icon, as shown in the following image:

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Microsoft peering

Microsoft peering

ER-Demo-Ckt

To receive route advertisements on Microsoft peering, attach route filters to the circuit after creating Microsoft Peering. [Learn More.](#)

* Peer ASN [?](#)
394749

* Primary subnet [?](#)
64.191.192.240/30

* Secondary subnet [?](#)
64.191.192.244/30

* VLAN ID [?](#)
152

* Advertised public prefixes [?](#)
64.191.192.224/28

Status: Configured

Customer ASN [?](#)
394749

Routing registry name [?](#)
ARIN

Shared key

[Get ARP records](#)
[Get route table](#)
[Get route table summary](#)

Azure private peering

This section helps you 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.

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

Circuit - Provider status: Not provisioned

Resource group (change) : USWest-ER-Demo-RG

Circuit status : Enabled

Location : West US 2

Subscription (change) : ExpressRoute-Lab

Subscription ID : 4bffb15-d414-4874-a2e4

Tags (change) : Click here to add tags

Provider : Equinix

Provider status : Not provisioned

Peering location : Seattle

Bandwidth : 200 Mbps

Service key : 74b7c74d-4b76-4a23

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

Circuit - Provider status: Provisioned

Resource group (change) : USWest-ER-Demo-RG

Circuit status : Enabled

Location : West US 2

Subscription (change) : ExpressRoute-Lab

Subscription ID : 4bffb15-d414-4874-a2e4

Tags (change) : Click here to add tags

Provider : Equinix

Provider status : Provisioned

Peering location : Seattle

Bandwidth : 200 Mbps

Service key : 74b7c74d-4b76-4a23

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

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. The subnet must not be part of any address space reserved for virtual networks. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A /30 subnet for the secondary link. The subnet must not be part of any address space reserved for virtual networks. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID. For both Primary and Secondary links you must use 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 except for the number from 65515 to 65520, inclusively.
- You must advertise the routes from your on-premises Edge router to Azure via BGP when you set up the private peering.
- **Optional** - An MD5 hash if you choose to use one.

3. Select the Azure private peering row, as shown in the following example:

Resource group (change) : USWest-ER-Demo-RG

Circuit status : Enabled

Location : West US 2

Subscription (change) : ExpressRoute-Lab

Subscription ID : 4bffb15-d414-4874-a2e4

Tags (change) : Click here to add tags

Provider : Equinix

Provider status : Provisioned

Peering location : Seattle

Bandwidth : 200 Mbps

Service key : 74b7c74d-4b76-4a23

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Not provisioned	-	-	...
Azure public	Not provisioned	-	-	...
Microsoft	Not provisioned	-	-	...

4. Configure private peering. **Save** the configuration once you have specified all parameters.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

* Peer ASN ⓘ
394749 ✓

* Primary subnet ⓘ
172.16.0.0/30 ✓

* Secondary subnet ⓘ
172.16.0.4/30 ✓

* VLAN ID ⓘ
154 ✓

Shared key

5. After the configuration has been accepted successfully, you see something similar to the following example:

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

* Peer ASN ⓘ
394749

* Primary subnet ⓘ
172.16.0.0/30

* Secondary subnet ⓘ
172.16.0.4/30

* VLAN ID ⓘ
154

Shared key

[Get ARP records](#)
[Get route table](#)
[Get route table summary](#)

To view Azure private peering details

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

Resource group (change) : USWest-ER-Demo-RG

Circuit status : Enabled

Location : West US 2

Subscription (change) : ExpressRoute-Lab

Subscription ID : 4bffb15-d414-4874-

Tags (change) : Click here to add tags

Provider : Equinix

Provider status : Provisioned

Peering location : Seattle

Bandwidth : 200 Mbps

Service key : 74b7c74d-4b76-4a23

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED BY
Azure private	Provisioned	172.16.0.0/30	172.16.0.4/30	Customer
Azure public	Not provisioned	-	-	...
Microsoft	Provisioned	64.191.192.240/30	64.191.192.244/30	Customer

To update Azure private peering configuration

You can select the row for peering and modify the peering properties. After updating, save your changes.

Private peering

ER-Demo-Ckt

Save Discard Delete

* Peer ASN

* Primary subnet

* Secondary subnet

* VLAN ID

Shared key

[Get ARP records](#)

[Get route table](#)

[Get route table summary](#)

To delete Azure private peering

You can remove your peering configuration by selecting the delete icon, as shown in the following image:

WARNING

You must ensure that all virtual networks and ExpressRoute Global Reach connections are removed before running this example.

Home > Resource groups > USWest-ER-Demo-RG > ER-Demo-Ckt > Private peering

Private peering

ER-Demo-Ckt

* Peer ASN

* Primary subnet

* Secondary subnet

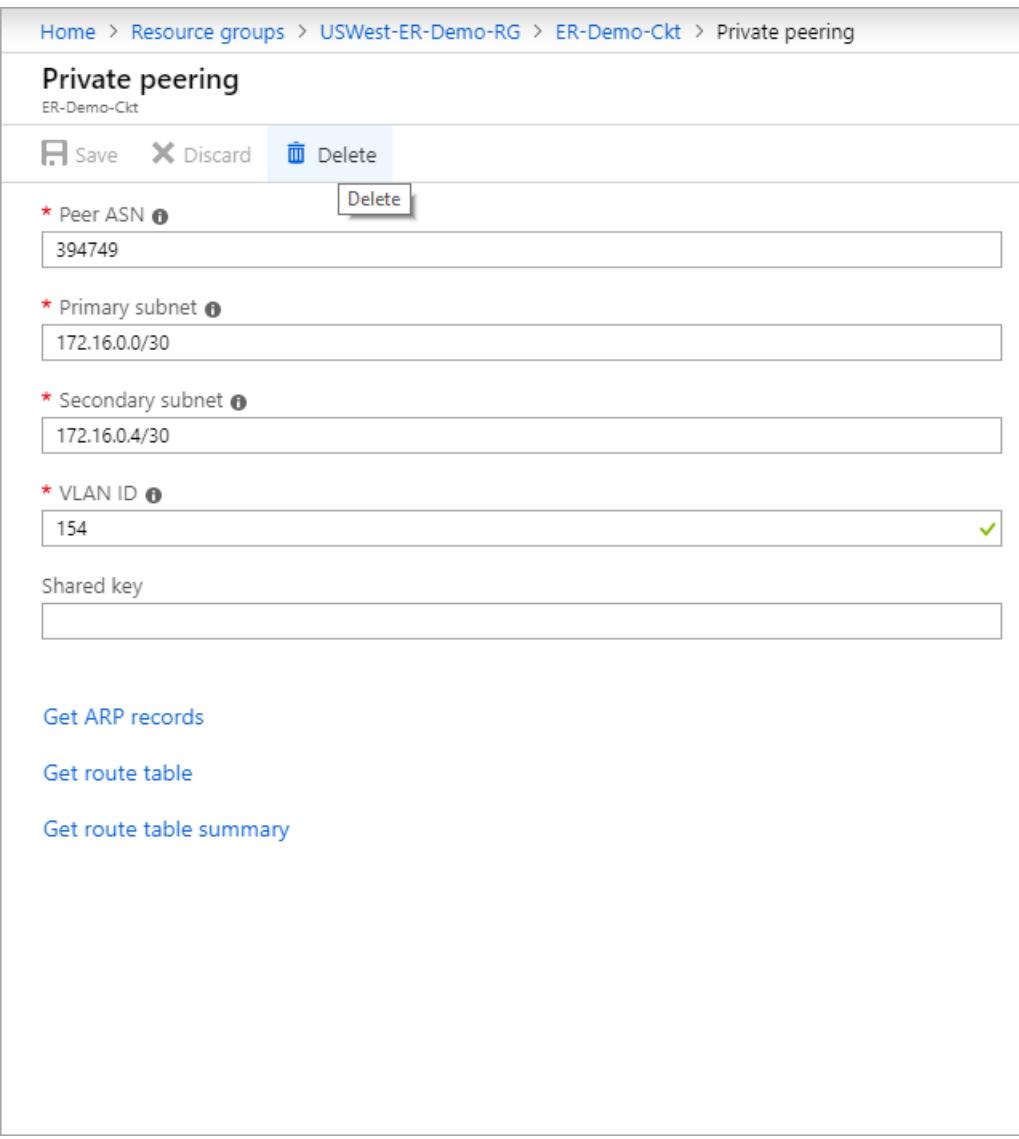
* VLAN ID

Shared key

[Get ARP records](#)

[Get route table](#)

[Get route table summary](#)



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 peering for an ExpressRoute circuit using PowerShell

12/17/2019 • 9 minutes to read • [Edit Online](#)

This article helps you create and manage routing configuration for an ExpressRoute circuit in the Resource Manager deployment model using PowerShell. You can also check the status, update, or delete and deprovision peerings for an ExpressRoute circuit. If you want to use a different method to work with your circuit, select an article from the following list:

These instructions only apply to circuits created with service providers offering Layer 2 connectivity services. If you are using a service provider that offers 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. Check with your service provider before configuring BGP peerings.

You can configure private peering and Microsoft peering for an ExpressRoute circuit (Azure public peering is deprecated for new circuits). Peerings can be configured in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time. For more information about routing domains and peerings, see [ExpressRoute routing domains](#). For information about public peering, see [ExpressRoute public peering](#).

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 in this article.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Azure PowerShell

Copy

Try It

Open Cloud Shell in your browser.

Launch Cloud Shell

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



Microsoft peering

This section helps you create, get, update, and delete the Microsoft peering configuration for an ExpressRoute circuit.

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through the Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit. For more information, see [Configure a route filter for Microsoft peering](#).

To create Microsoft peering

1. Sign in and select your subscription.

If you installed PowerShell locally, sign in. If you are using Azure Cloud Shell, you can skip this step.

```
Connect-AzAccount
```

Select the subscription you want to create ExpressRoute circuit.

```
Select-AzSubscription -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 ask your connectivity provider to enable Microsoft 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, continue your configuration using the next steps.

3. Check the ExpressRoute circuit to make sure it is provisioned and also enabled. Use the following example:

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

The response is similar to the following example:

```

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 or /126 subnet for the primary link. This must be a valid public IPv4 or IPv6 prefix owned by you and registered in an RIR / IRR.
- A /30 or /126 subnet for the secondary link. This must be a valid public IPv4 or IPv6 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. If you plan to send a set of prefixes, you can send a comma-separated list. These prefixes must be registered to you in an RIR / IRR. IPv4 BGP sessions require IPv4 advertised prefixes and IPv6 BGP sessions require IPv6 advertised prefixes.
- Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
- Optional:
 - 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.
 - An MD5 hash if you choose to use one.

IMPORTANT

Microsoft verifies if the specified 'Advertised public prefixes' and 'Peer ASN' (or 'Customer ASN') are assigned to you in the Internet Routing Registry. If you are getting the public prefixes from another entity and if the assignment is not recorded with the routing registry, the automatic validation will not complete and will require manual validation. If the automatic validation fails, you will see 'AdvertisedPublicPrefixesState' as 'Validation needed' on the output of "Get-AzExpressRouteCircuitPeeringConfig" (see "To get Microsoft peering details" below) command.

If you see the message 'Validation needed', collect the document(s) that show the public prefixes are assigned to your organization by the entity that is listed as the owner of the prefixes in the routing registry and submit these documents for manual validation by opening a support ticket as shown below.

Use the following example to configure Microsoft peering for your circuit:

```
Add-AzExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType MicrosoftPeering -PeerASN 100 -PeerAddressType IPv4 -PrimaryPeerAddressPrefix "123.0.0.0/30" - SecondaryPeerAddressPrefix "123.0.0.4/30" -VlanId 300 -MicrosoftConfigAdvertisedPublicPrefixes "123.1.0.0/24" -MicrosoftConfigCustomerAsn 23 -MicrosoftConfigRoutingRegistryName "ARIN"

Add-AzExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType MicrosoftPeering -PeerASN 100 -PeerAddressType IPv6 -PrimaryPeerAddressPrefix "3FFE:FFFF:0:CD30::/126" - SecondaryPeerAddressPrefix "3FFE:FFFF:0:CD30::4/126" -VlanId 300 -MicrosoftConfigAdvertisedPublicPrefixes "3FFE:FFFF:0:CD31::/120" -MicrosoftConfigCustomerAsn 23 -MicrosoftConfigRoutingRegistryName "ARIN"

Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To get Microsoft peering details

You can get configuration details using the following example:

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

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

To update Microsoft peering configuration

You can update any part of the configuration using the following example:

```
Set-AzExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType MicrosoftPeering -PeerASN 100 -PeerAddressType IPv4 -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-AzExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $ckt -PeeringType MicrosoftPeering -PeerASN 100 -PeerAddressType IPv6 -PrimaryPeerAddressPrefix "3FFE:FFFF:0:CD30::/126" - SecondaryPeerAddressPrefix "3FFE:FFFF:0:CD30::4/126" -VlanId 300 -MicrosoftConfigAdvertisedPublicPrefixes "3FFE:FFFF:0:CD31::/120" -MicrosoftConfigCustomerAsn 23 -MicrosoftConfigRoutingRegistryName "ARIN"

Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To delete Microsoft peering

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

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

Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Azure private peering

This section helps you 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 Az
```

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

```
Import-Module Az
```

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

```
Import-Module Az.Network
```

Sign in to your account.

```
Connect-AzAccount
```

Select the subscription you want to create ExpressRoute circuit.

```
Select-AzSubscription -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 ask 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, continue your configuration using the next steps.

3. Check the ExpressRoute circuit to make sure it is provisioned and also enabled. Use the following example:

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

The response is similar to the following example:

```

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. The subnet must not be part of any address space reserved for virtual networks.
- A /30 subnet for the secondary link. The subnet 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.
- Optional:
 - An MD5 hash if you choose to use one.

Use the following example to configure Azure private peering for your circuit:

```

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

Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt

```

If you choose to use an MD5 hash, use the following example:

```

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

```

IMPORTANT

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

To get Azure private peering details

You can get configuration details by using the following example:

```
$ckt = Get-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
Get-AzExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -ExpressRouteCircuit $ckt
```

To update Azure private peering configuration

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

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

To delete Azure private peering

You can remove your peering configuration by running the following example:

WARNING

You must ensure that all virtual networks and ExpressRoute Global Reach connections are removed before running this example.

```
Remove-AzExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -ExpressRouteCircuit $ckt  
  
Set-AzExpressRouteCircuit -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 peering for an ExpressRoute circuit using CLI

12/17/2019 • 8 minutes to read • [Edit Online](#)

This article helps you create and manage routing configuration/peering for an ExpressRoute circuit in the Resource Manager deployment model using CLI. You can also check the status, update, or delete and deprovision peerings for an ExpressRoute circuit. If you want to use a different method to work with your circuit, select an article from the following list:

Configuration prerequisites

- Before beginning, install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install the Azure CLI](#).
- Make sure that you have reviewed the [prerequisites](#), [routing requirements](#), and [workflow](#) pages 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 commands in this article.

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

You can configure private peering and Microsoft peering for an ExpressRoute circuit (Azure public peering is deprecated for new circuits). Peerings can be configured in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time. For more information about routing domains and peerings, see [ExpressRoute routing domains](#). For information about public peering, see [ExpressRoute public peering](#).

Microsoft peering

This section helps you create, get, update, and delete the Microsoft peering configuration for an ExpressRoute circuit.

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through the Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit. For more information, see [Configure a route filter for Microsoft peering](#).

To create Microsoft peering

1. Install the latest version of Azure CLI. Use the latest version of the Azure Command-line Interface (CLI).* Review the [prerequisites](#) and [workflows](#) before you begin configuration.

```
az login
```

Select the subscription for which you want to create ExpressRoute circuit.

```
az account set --subscription "<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 ask your connectivity provider to enable Microsoft 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, continue your configuration using the next steps.

3. Check the ExpressRoute circuit to make sure it is provisioned and also enabled. Use the following example:

```
az network express-route list
```

The response is similar to the following example:

```
"allowClassicOperations": false,
"authorizations": [],
"circuitProvisioningState": "Enabled",
"etag": "W/\"1262c492-ffef-4a63-95a8-a6002736b8c4\"",
"gatewayManagerEtag": null,
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit",
"location": "westus",
"name": "MyCircuit",
"peerings": [],
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"serviceKey": "1d05cf70-1db5-419f-ad86-1ca62c3c125b",
"serviceProviderNotes": null,
"serviceProviderProperties": {
    "bandwidthInMbps": 200,
    "peeringLocation": "Silicon Valley",
    "serviceProviderName": "Equinix"
},
"serviceProviderProvisioningState": "Provisioned",
"sku": {
    "family": "UnlimitedData",
    "name": "Standard_MeteredData",
    "tier": "Standard"
},
"tags": null,
"type": "Microsoft.Network/expressRouteCircuits"]
```

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. If you plan to send a set of prefixes, you can send a comma-separated list. These prefixes must be registered to you in an RIR / IRR.
- **Optional** - 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.

- Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
- **Optional** - An MD5 hash if you choose to use one.

Run the following example to configure Microsoft peering for your circuit:

```
az network express-route peering create --circuit-name MyCircuit --peer-asn 100 --primary-peer-subnet 123.0.0.0/30 -g ExpressRouteResourceGroup --secondary-peer-subnet 123.0.0.4/30 --vlan-id 300 --peering-type MicrosoftPeering --advertised-public-prefixes 123.1.0.0/24
```

To view Microsoft peering details

You can get configuration details by using the following example:

```
az network express-route peering show -g ExpressRouteResourceGroup --circuit-name MyCircuit --name AzureMicrosoftPeering
```

IMPORTANT

Microsoft verifies if the specified 'Advertised public prefixes' and 'Peer ASN' (or 'Customer ASN') are assigned to you in the Internet Routing Registry. If you are getting the public prefixes from another entity and if the assignment is not recorded with the routing registry, the automatic validation will not complete and will require manual validation. If the automatic validation fails, you will see 'AdvertisedPublicPrefixesState' as 'Validation needed' on the output of the above command.

If you see the message 'Validation needed', collect the document(s) that show the public prefixes are assigned to your organization by the entity that is listed as the owner of the prefixes in the routing registry and submit these documents for manual validation by opening a support ticket as shown below.

The output is similar to the following example:

```
{
  "azureAsn": 12076,
  "etag": "W/\"2e97be83-a684-4f29-bf3c-96191e270666\"",
  "gatewayManagerEtag": "18",
  "id": "/subscriptions/9a0c2943-e0c2-4608-876c-e0ddffd1211b/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit/peerings/AzureMicrosoftPeering",
  "lastModifiedBy": "Customer",
  "microsoftPeeringConfig": {
    "advertisedPublicPrefixes": [
      ""
    ],
    "advertisedPublicPrefixesState": "",
    "customerASN": ,
    "routingRegistryName": ""
  }
}
"name": "AzureMicrosoftPeering",
"peerAsn": ,
"peeringType": "AzureMicrosoftPeering",
"primaryAzurePort": "",
"primaryPeerAddressPrefix": "",
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"routeFilter": null,
"secondaryAzurePort": "",
"secondaryPeerAddressPrefix": "",
"sharedKey": null,
"state": "Enabled",
"stats": null,
"vlanId": 100
}
```

To update Microsoft peering configuration

You can update any part of the configuration. The advertised prefixes of the circuit are being updated from 123.1.0.0/24 to 124.1.0.0/24 in the following example:

```
az network express-route peering update --circuit-name MyCircuit -g ExpressRouteResourceGroup --peering-type MicrosoftPeering --advertised-public-prefixes 124.1.0.0/24
```

To add IPv6 Microsoft peering settings to an existing IPv4 configuration

```
az network express-route peering update -g ExpressRouteResourceGroup --circuit-name MyCircuit --peering-type MicrosoftPeering --ip-version ipv6 --primary-peer-subnet 2002:db00::/126 --secondary-peer-subnet 2003:db00::/126 --advertised-public-prefixes 2002:db00::/126
```

To delete Microsoft peering

You can remove your peering configuration by running the following example:

```
az network express-route peering delete -g ExpressRouteResourceGroup --circuit-name MyCircuit --name MicrosoftPeering
```

Azure private peering

This section helps you create, get, update, and delete the Azure private peering configuration for an ExpressRoute circuit.

To create Azure private peering

1. Install the latest version of Azure CLI. You must use the latest version of the Azure Command-line

Interface (CLI).* Review the [prerequisites](#) and [workflows](#) before you begin configuration.

```
az login
```

Select the subscription you want to create ExpressRoute circuit

```
az account set --subscription "<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 ask 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, continue your configuration using the next steps.

3. Check the ExpressRoute circuit to make sure it is provisioned and also enabled. Use the following example:

```
az network express-route show --resource-group ExpressRouteResourceGroup --name MyCircuit
```

The response is similar to the following example:

```
"allowClassicOperations": false,
"authorizations": [],
"circuitProvisioningState": "Enabled",
"etag": "W/\\"1262c492-ffef-4a63-95a8-a6002736b8c4\\\"",
"gatewayManagerEtag": null,
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-
f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits
/MyCircuit",
"location": "westus",
"name": "MyCircuit",
"peerings": [],
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"serviceKey": "1d05cf70-1db5-419f-ad86-1ca62c3c125b",
"serviceProviderNotes": null,
"serviceProviderProperties": {
"bandwidthInMbps": 200,
"peeringLocation": "Silicon Valley",
"serviceProviderName": "Equinix"
},
"serviceProviderProvisioningState": "Provisioned",
"sku": {
"family": "UnlimitedData",
"name": "Standard_MeteredData",
"tier": "Standard"
},
"tags": null,
"type": "Microsoft.Network/expressRouteCircuits"]
```

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. The subnet must not be part of any address space reserved for virtual networks.
- A /30 subnet for the secondary link. The subnet 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.
- **Optional** - An MD5 hash if you choose to use one.

Use the following example to configure Azure private peering for your circuit:

```
az network express-route peering create --circuit-name MyCircuit --peer-asn 100 --primary-peer-subnet 10.0.0.0/30 -g ExpressRouteResourceGroup --secondary-peer-subnet 10.0.0.4/30 --vlan-id 200 --peering-type AzurePrivatePeering
```

If you choose to use an MD5 hash, use the following example:

```
az network express-route peering create --circuit-name MyCircuit --peer-asn 100 --primary-peer-subnet 10.0.0.0/30 -g ExpressRouteResourceGroup --secondary-peer-subnet 10.0.0.4/30 --vlan-id 200 --peering-type AzurePrivatePeering --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 by using the following example:

```
az network express-route peering show -g ExpressRouteResourceGroup --circuit-name MyCircuit --name AzurePrivatePeering
```

The output is similar to the following example:

```
{  
    "azureAsn": 12076,  
    "etag": "W/\\"2e97be83-a684-4f29-bf3c-96191e270666\\\"",  
    "gatewayManagerEtag": "18",  
    "id": "/subscriptions/9a0c2943-e0c2-4608-876c-e0ddffd1211b/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit/peerings/AzurePrivatePeering",  
    "ipv6PeeringConfig": null,  
    "lastModifiedBy": "Customer",  
    "microsoftPeeringConfig": null,  
    "name": "AzurePrivatePeering",  
    "peerAsn": 7671,  
    "peeringType": "AzurePrivatePeering",  
    "primaryAzurePort": "",  
    "primaryPeerAddressPrefix": "",  
    "provisioningState": "Succeeded",  
    "resourceGroup": "ExpressRouteResourceGroup",  
    "routeFilter": null,  
    "secondaryAzurePort": "",  
    "secondaryPeerAddressPrefix": "",  
    "sharedKey": null,  
    "state": "Enabled",  
    "stats": null,  
    "vlanId": 100  
}
```

To update Azure private peering configuration

You can update any part of the configuration using the following example. In this example, the VLAN ID of the

circuit is being updated from 100 to 500.

```
az network express-route peering update --vlan-id 500 -g ExpressRouteResourceGroup --circuit-name MyCircuit -  
-name AzurePrivatePeering
```

To delete Azure private peering

You can remove your peering configuration by running the following example:

WARNING

You must ensure that all virtual networks and ExpressRoute Global Reach connections are removed before running this example.

```
az network express-route peering delete -g ExpressRouteResourceGroup --circuit-name MyCircuit --name  
AzurePrivatePeering
```

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](#).

Connect a virtual network to an ExpressRoute circuit using the portal

11/13/2019 • 4 minutes to read • [Edit Online](#)

This article helps you create a connection to link a virtual network to an Azure ExpressRoute circuit using the Azure portal. The virtual networks that you connect to your Azure ExpressRoute circuit can either be in the same subscription, or they can be part of another subscription.

Before you begin

- Review the [prerequisites, routing requirements](#), and [workflows](#) before you begin configuration.
- You must have an active ExpressRoute circuit.
 - Follow the instructions to [create an ExpressRoute circuit](#) and have the circuit enabled by your connectivity provider.
 - Ensure that you have Azure private peering configured for your circuit. See the [Create and modify peering for an ExpressRoute circuit](#) article for peering and routing instructions.
 - Ensure that Azure private peering 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 virtual network gateway for ExpressRoute](#). A virtual network gateway for ExpressRoute uses the GatewayType 'ExpressRoute', not VPN.
- You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit.
- A single VNet can be linked to up to four ExpressRoute circuits. Use the process below to create a new connection object for each ExpressRoute circuit you are connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.
- You can link a virtual network 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.
- You can [view a video](#) before beginning to better understand the steps.

Connect a VNet to a circuit - same subscription

NOTE

BGP configuration information will not 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.

To create a connection

1. Ensure that your ExpressRoute circuit and Azure private peering have been configured successfully. Follow the instructions in [Create an ExpressRoute circuit](#) and [Create and modify peering for an ExpressRoute circuit](#). Your ExpressRoute circuit should look like the following image:

The screenshot shows three windows from the Azure portal:

- ER-Demo-Ckt-SV**: Shows basic circuit details like Resource group (USWest-ER-Demo-RG), Provider (Equinix), and Peering status.
- Settings**: Shows the Peering configuration for the circuit. The "Peering" section is highlighted with a red box.
- Peerings**: A table showing peering configurations. One row for "Azure private" is highlighted with a red box.

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** page, and then configure the values.

The screenshot shows three windows from the Azure portal:

- Settings**: Shows the "Connections" section, which is highlighted with a red box. A blue "Add" button is also highlighted.
- Connections**: An empty list of connections.
- Add connection**: A configuration dialog with fields for Name (ER-VNet-Connection), Connection type (ExpressRoute), Virtual network gateway (Demo-VNet-GW), ExpressRoute circuit (ER-Demo-Ckt-SV), Subscription (ExpressRoute-Demo), Resource group (USWest-ER-Demo-RG), and Location (West US).

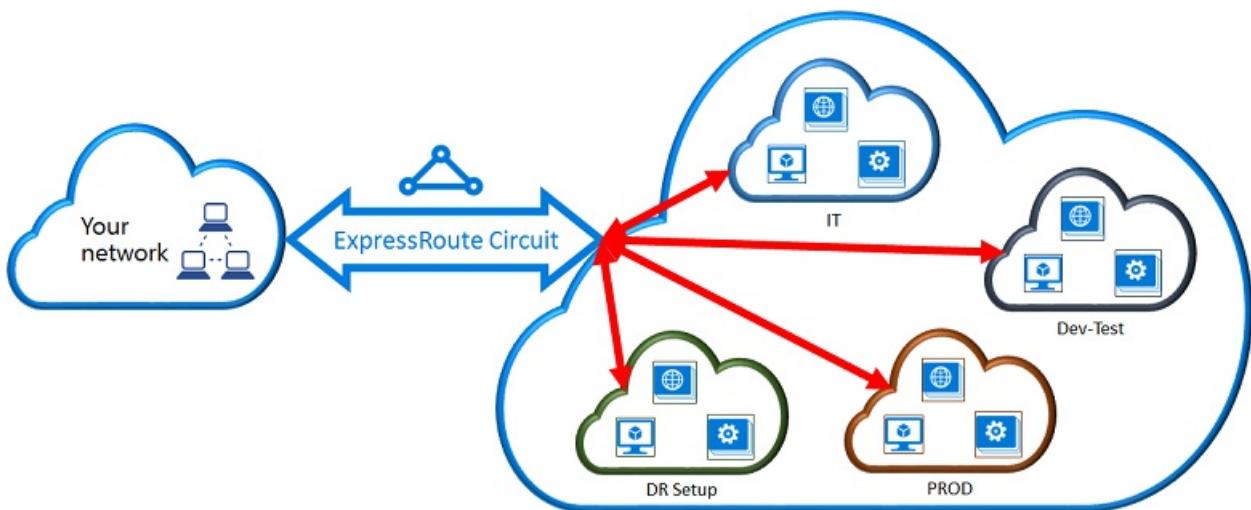
3. After your connection has been successfully configured, your connection object will show the information for the connection.

The screenshot shows two windows from the Azure portal:

- Connections**: Shows the "ER-Demo-Ckt-SV" connection, which is highlighted with a red box.
- ER-VNet-Connection**: A detailed view of the connection object, showing its status (Succeeded), connection type (ExpressRoute), peer (Demo-VNet-GW), and essential properties like Resource group (USWest-ER-Demo-RG), Status (Succeeded), Location (West US), and Subscriptions. The "Virtual network" section is highlighted with a red box.

Connect a VNet to a circuit - different subscription

You can share an ExpressRoute circuit across multiple subscriptions. The figure below 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 and authorizations associated to the circuit, including subscriptions linked to other Azure Active Directory tenants and Enterprise Agreement enrollments.

NOTE

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

Administration - About circuit owners and circuit users

The 'circuit owner' is an authorized Power User of the ExpressRoute circuit resource. The circuit owner can create authorizations that can be redeemed by 'circuit users'. Circuit users are owners of virtual network gateways that 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

To create a connection 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.

NOTE

Each connection requires a separate authorization.

1. In the ExpressRoute page, Click **Authorizations** and then type a **name** for the authorization and click **Save**.

- Once the configuration is saved, copy the **Resource ID** and the **Authorization Key**.

To delete a connection authorization

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

Circuit user operations

The circuit user needs the resource ID and an authorization key from the circuit owner.

To redeem a connection authorization

- Click the **+New** button.

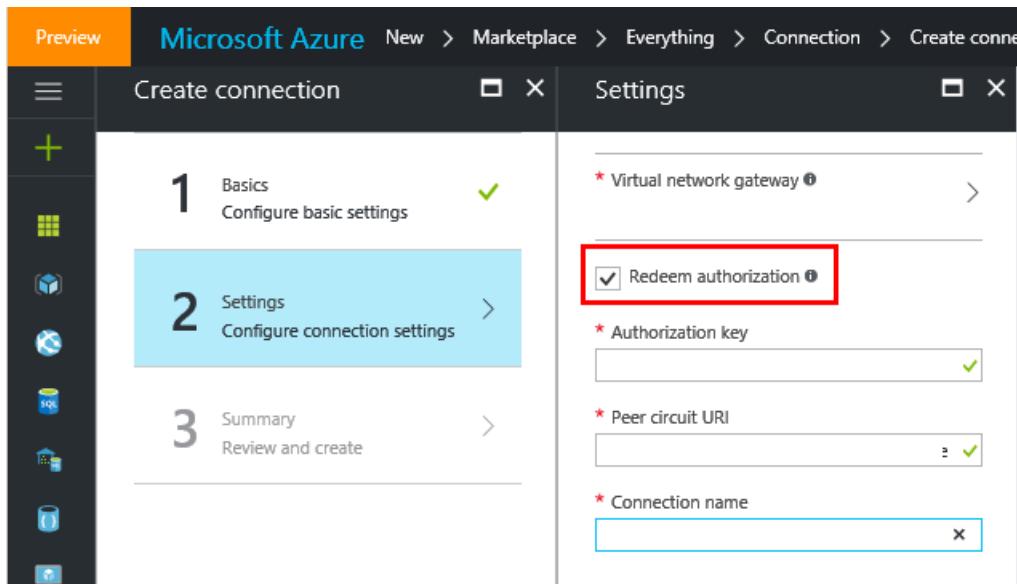
- Search for "**Connection**" in the Marketplace, select it, and click **Create**.

3. Make sure the **Connection type** is set to "ExpressRoute".

4. Fill in the details, then click **OK** in the Basics page.

5. In the **Settings** page, Select the **Virtual network gateway** and check the **Redeem authorization** check box.

6. Enter the **Authorization key** and the **Peer circuit URI** and give the connection a name. Click **OK**. The **Peer Circuit URI** is the Resource ID of the ExpressRoute circuit (which you can find under the Properties Setting pane of the ExpressRoute Circuit).



7. Review the information in the **Summary** page and click **OK**.

To release a connection authorization

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

Delete a connection to unlink a VNet

You can delete a connection and unlink your VNet to an ExpressRoute circuit by selecting the **Delete** icon on the page for your connection.

Next steps

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

Connect a virtual network to an ExpressRoute circuit

11/13/2019 • 6 minutes to read • [Edit Online](#)

This article helps 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. This article also shows you how to update a virtual network link.

- You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit.
- A single VNet can be linked to up to four ExpressRoute circuits. Use the steps in this article to create a new connection object for each ExpressRoute circuit you are connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.
- You can link 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.

Before you begin

- 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 virtual network gateway for ExpressRoute](#). A virtual network gateway for ExpressRoute uses the GatewayType 'ExpressRoute', not VPN.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Azure PowerShell

Copy

Try It

Open Cloud Shell in your browser.

Launch Cloud Shell

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



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

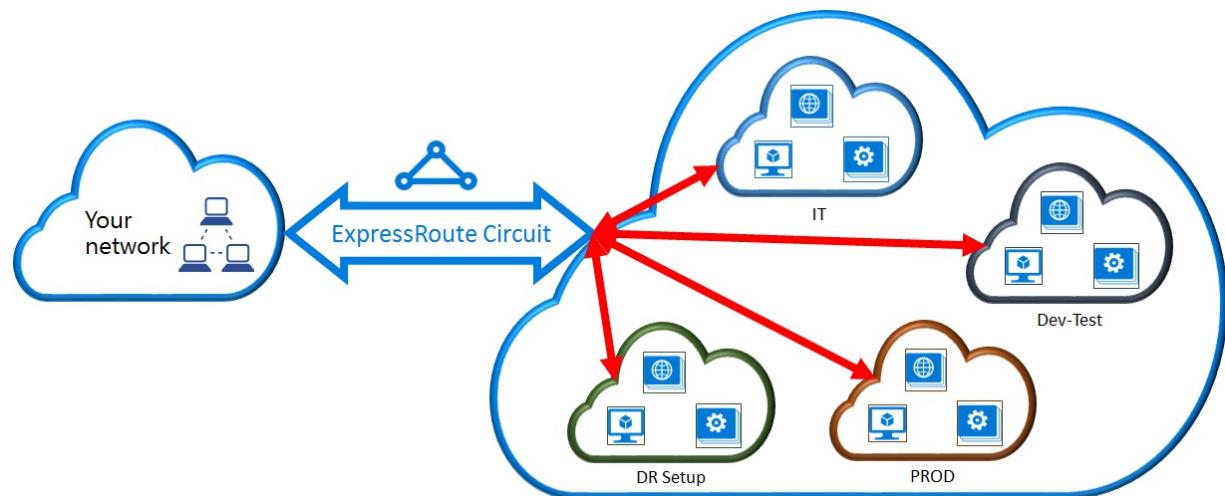
Connect a virtual network in a different subscription to a circuit

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 ExpressRoute circuit will be applied to the subscription owner. All virtual networks share the same bandwidth.



Administration - circuit owners and circuit users

The 'circuit owner' is an authorized Power User of the ExpressRoute circuit resource. The circuit owner can

create authorizations that can be redeemed by 'circuit users'. Circuit users are owners of virtual network gateways that 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

To create 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-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
Add-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization1"
Set-AzExpressRouteCircuit -ExpressRouteCircuit $circuit

$circuit = Get-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$auth1 = Get-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization1"
```

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

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

To review authorizations

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

```
$circuit = Get-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$authorizations = Get-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit
```

To add authorizations

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

```
$circuit = Get-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
Add-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit -Name "MyAuthorization2"
Set-AzExpressRouteCircuit -ExpressRouteCircuit $circuit

$circuit = Get-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"
$authorizations = Get-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $circuit
```

To delete authorizations

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

```
Remove-AzExpressRouteCircuitAuthorization -Name "MyAuthorization2" -ExpressRouteCircuit $circuit
Set-AzExpressRouteCircuit -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 can be checked from the following command:

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

To redeem a connection authorization

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

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

To release a connection authorization

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

Modify a virtual network connection

You can update certain properties of a virtual network connection.

To update the connection weight

Your virtual network can be connected to multiple ExpressRoute circuits. You may receive the same prefix from more than one ExpressRoute circuit. To choose which connection to send traffic destined for this prefix, you can change *RoutingWeight* of a connection. Traffic will be sent on the connection with the highest *RoutingWeight*.

```
$connection = Get-AzVirtualNetworkGatewayConnection -Name "MyVirtualNetworkConnection" -ResourceGroupName "MyRG"
$connection.RoutingWeight = 100
Set-AzVirtualNetworkGatewayConnection -VirtualNetworkGatewayConnection $connection
```

The range of *RoutingWeight* is 0 to 32000. The default value is 0.

Configure ExpressRoute FastPath

You can enable [ExpressRoute FastPath](#) if your ExpressRoute circuit is on [ExpressRoute Direct](#) and your virtual network gateway is Ultra Performance or ErGw3AZ. FastPath improves data path performance such as packets per second and connections per second between your on-premises network and your virtual network.

NOTE

If you already have a virtual network connection but haven't enabled FastPath you need to delete the virtual network connection and create a new one.

```
$circuit = Get-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "MyRG"  
$gw = Get-AzVirtualNetworkGateway -Name "MyGateway" -ResourceGroupName "MyRG"  
$connection = New-AzVirtualNetworkGatewayConnection -Name "MyConnection" -ResourceGroupName "MyRG" -  
ExpressRouteGatewayBypass -VirtualNetworkGateway1 $gw -PeerId $circuit.Id -ConnectionType ExpressRoute -  
Location "MyLocation"
```

Next steps

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

Connect a virtual network to an ExpressRoute circuit using CLI

11/13/2019 • 5 minutes to read • [Edit Online](#)

This article helps you link virtual networks (VNets) to Azure ExpressRoute circuits using CLI. To link using Azure CLI, the virtual networks must be created using the Resource Manager deployment model. They can either be in the same subscription, or part of another subscription. If you want to use a different method to connect your VNet to an ExpressRoute circuit, you can select an article from the following list:

Configuration prerequisites

- You need the latest version of the command-line interface (CLI). For more information, see [Install the Azure CLI](#).
- 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. The BGP peering between your network and Microsoft must be 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 [Configure a virtual network gateway for ExpressRoute](#). Be sure to use `--gateway-type 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.
- A single VNet can be linked to up to four ExpressRoute circuits. Use the process below to create a new connection object for each ExpressRoute circuit you are connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.
- If you enable the ExpressRoute premium add-on, you can link a virtual network outside of the geopolitical region of the ExpressRoute circuit, or connect a larger number of virtual networks to your ExpressRoute circuit. For more information about the premium add-on, see the [FAQ](#).

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 example. Make sure that the virtual network gateway is created and is ready for linking before you run the command.

```
az network vpn-connection create --name ERConnection --resource-group ExpressRouteResourceGroup --vnet-gateway1 VNet1GW --express-route-circuit2 MyCircuit
```

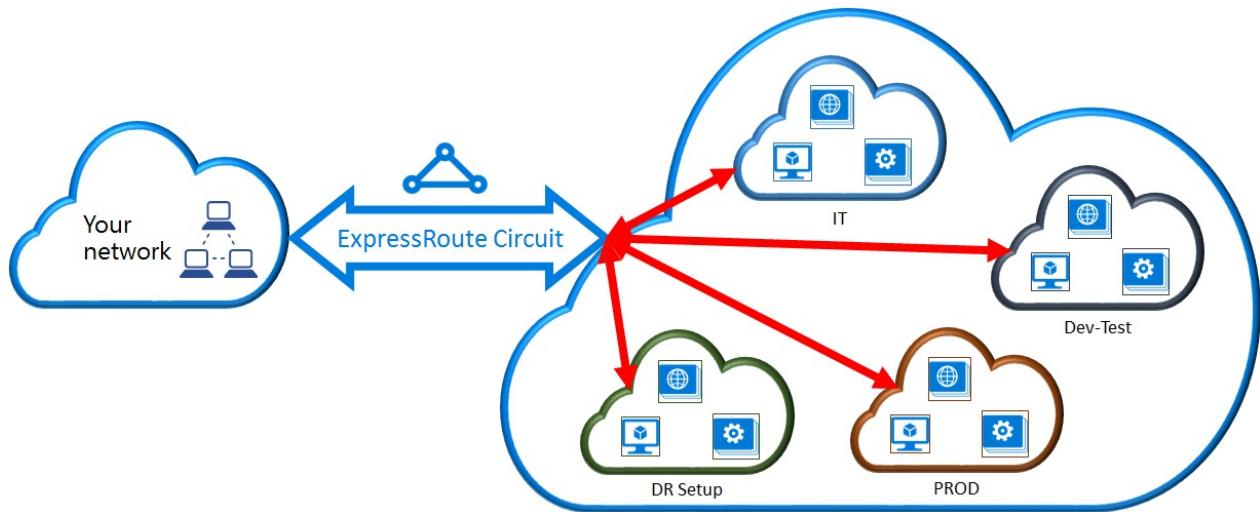
Connect a virtual network in a different subscription to a circuit

You can share an ExpressRoute circuit across multiple subscriptions. The figure below 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 - Circuit Owners and Circuit Users

The 'Circuit Owner' is an authorized Power User of the ExpressRoute circuit resource. The Circuit Owner can create authorizations that can be redeemed by 'Circuit Users'. Circuit Users are owners of virtual network gateways that 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. When an authorization is revoked, all link connections are deleted from the subscription whose access was revoked.

Circuit Owner operations

To create an authorization

The Circuit Owner creates an authorization, which creates 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 example shows how to create an authorization:

```
az network express-route auth create --circuit-name MyCircuit -g ExpressRouteResourceGroup -n MyAuthorization
```

The response contains the authorization key and status:

```
"authorizationKey": "0a7f3020-541f-4b4b-844a-5fb43472e3d7",
"authorizationUseStatus": "Available",
"etag": "W/\"010353d4-8955-4984-807a-585c21a22ae0\",
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-
f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCirc
uit/authorizations/MyAuthorization1",
"name": "MyAuthorization1",
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup"
```

To review authorizations

The Circuit Owner can review all authorizations that are issued on a particular circuit by running the following example:

```
az network express-route auth list --circuit-name MyCircuit -g ExpressRouteResourceGroup
```

To add authorizations

The Circuit Owner can add authorizations by using the following example:

```
az network express-route auth create --circuit-name MyCircuit -g ExpressRouteResourceGroup -n
MyAuthorization1
```

To delete authorizations

The Circuit Owner can revoke/delete authorizations to the user by running the following example:

```
az network express-route auth delete --circuit-name MyCircuit -g ExpressRouteResourceGroup -n
MyAuthorization1
```

Circuit User operations

The Circuit User needs the peer ID and an authorization key from the Circuit Owner. The authorization key is a GUID.

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

To redeem a connection authorization

The Circuit User can run the following example to redeem a link authorization:

```
az network vpn-connection create --name ERConnection --resource-group ExpressRouteResourceGroup --vnet-
gateway1 VNet1GW --express-route-circuit2 MyCircuit --authorization-key
"^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"
```

To release a connection authorization

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

Modify a virtual network connection

You can update certain properties of a virtual network connection.

To update the connection weight

Your virtual network can be connected to multiple ExpressRoute circuits. You may receive the same prefix from more than one ExpressRoute circuit. To choose which connection to send traffic destined for this prefix, you can change *RoutingWeight* of a connection. Traffic will be sent on the connection with the highest *RoutingWeight*.

```
az network vpn-connection update --name ERConnection --resource-group ExpressRouteResourceGroup --routing-weight 100
```

The range of *RoutingWeight* is 0 to 32000. The default value is 0.

Configure ExpressRoute FastPath

You can enable [ExpressRoute FastPath](#) if your ExpressRoute circuit is on [ExpressRoute Direct](#) and your virtual network gateway is Ultra Performance or ErGw3AZ. FastPath improves data path performance such as packets per second and connections per second between your on-premises network and your virtual network.

NOTE

If you already have a virtual network connection but haven't enabled FastPath you need to delete the virtual network connection and create a new one.

```
az network vpn-connection create --name ERConnection --resource-group ExpressRouteResourceGroup --express-route-gateway-bypass true --vnet-gateway1 VNet1GW --express-route-circuit2 MyCircuit
```

Next steps

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

Configure a site-to-site VPN over ExpressRoute Microsoft peering

12/23/2019 • 16 minutes to read • [Edit Online](#)

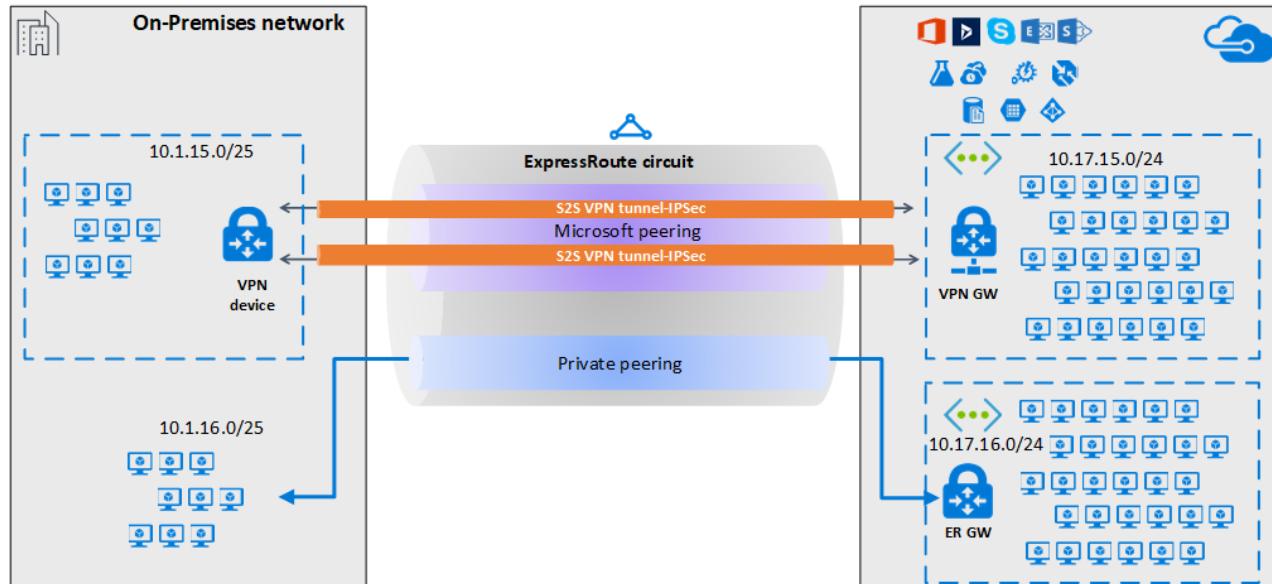
This article helps you configure secure encrypted connectivity between your on-premises network and your Azure virtual networks (VNets) over an ExpressRoute private connection. You can use Microsoft peering to establish a site-to-site IPsec/IKE VPN tunnel between your selected on-premises networks and Azure VNets. Configuring a secure tunnel over ExpressRoute allows for data exchange with confidentiality, anti-replay, authenticity, and integrity.

NOTE

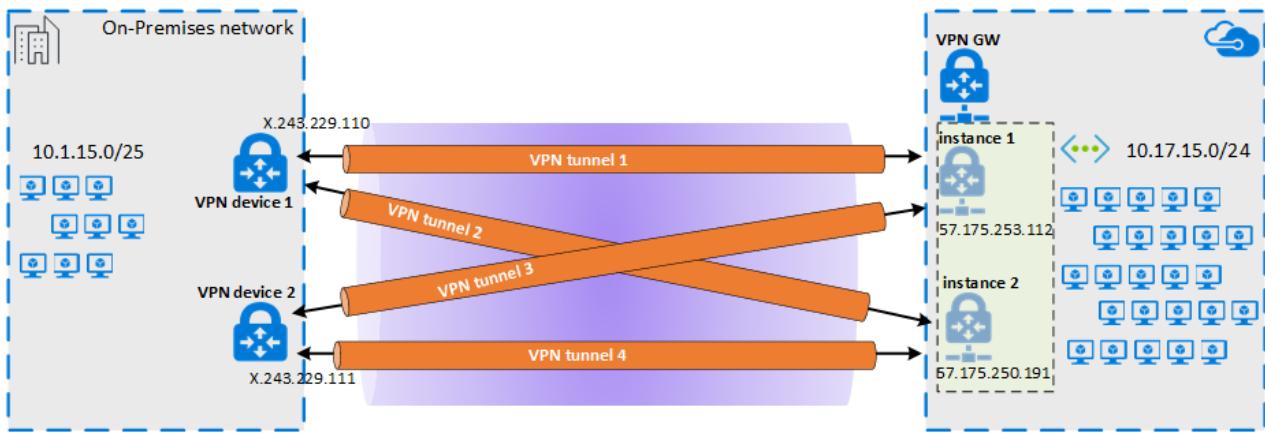
When you set up site-to-site VPN over Microsoft peering, you are charged for the VPN gateway and VPN egress. For more information, see [VPN Gateway pricing](#).

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

Architecture



For high availability and redundancy, you can configure multiple tunnels over the two MSEE-PE pairs of a ExpressRoute circuit and enable load balancing between the tunnels.



VPN tunnels over Microsoft peering can be terminated either using VPN gateway, or using an appropriate Network Virtual Appliance (NVA) available through Azure Marketplace. You can exchange routes statically or dynamically over the encrypted tunnels without exposing the route exchange to the underlying Microsoft peering. In the examples in this article, BGP (different from the BGP session used to create the Microsoft peering) is used to dynamically exchange prefixes over the encrypted tunnels.

IMPORTANT

For the on-premises side, typically Microsoft peering is terminated on the DMZ and private peering is terminated on the core network zone. The two zones would be segregated using firewalls. If you are configuring Microsoft peering exclusively for enabling secure tunneling over ExpressRoute, remember to filter through only the public IPs of interest that are getting advertised via Microsoft peering.

Workflow

1. Configure Microsoft peering for your ExpressRoute circuit.
2. Advertise selected Azure regional public prefixes to your on-premises network via Microsoft peering.
3. Configure a VPN gateway and establish IPsec tunnels
4. Configure the on-premises VPN device.
5. Create the site-to-site IPsec/IKE connection.
6. (Optional) Configure firewalls/filtering on the on-premises VPN device.
7. Test and validate the IPsec communication over the ExpressRoute circuit.

1. Configure Microsoft peering

To configure a site-to-site VPN connection over ExpressRoute, you must leverage ExpressRoute Microsoft peering.

- To configure a new ExpressRoute circuit, start with the [ExpressRoute prerequisites](#) article, and then [Create and modify an ExpressRoute circuit](#).
- If you already have an ExpressRoute circuit, but do not have Microsoft peering configured, configure Microsoft peering using the [Create and modify peering for an ExpressRoute circuit](#) article.

Once you have configured your circuit and Microsoft peering, you can easily view it using the **Overview** page in the Azure portal.

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	LAST MODIFIED...
Azure private	Provisioned	10.100.11.0/30	10.100.11.4/30	Customer
Azure public	Not provisioned	-	-	
Microsoft	Provisioned	243.229.32/30	243.229.36/30	Customer

2. Configure route filters

A route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. It is essentially an allow list of all the BGP community values.

NAME	VALUE
Azure West US 2	12076:51026

In this example, the deployment is only in the *Azure West US 2* region. A route filter rule is added to allow only the advertisement of *Azure West US 2* regional prefixes, which has the BGP community value *12076:51026*. You specify the regional prefixes that you want to allow by selecting **Manage rule**.

Within the route filter, you also need to choose the ExpressRoute circuits for which the route filter applies. You can choose the ExpressRoute circuits by selecting **Add circuit**. In the previous figure, the route filter is associated to the example ExpressRoute circuit.

2.1 Configure the route filter

Configure a route filter. For steps, see [Configure route filters for Microsoft peering](#).

2.2 Verify BGP routes

Once you have successfully created Microsoft peering over your ExpressRoute circuit and associated a route filter with the circuit, you can verify the BGP routes received from MSEEs on the PE devices that are peering with the MSEEs. The verification command varies, depending on the operating system of your PE devices.

Cisco examples

This example uses a Cisco IOS-XE command. In the example, a virtual routing and forwarding (VRF) instance is used to isolate the peering traffic.

```
show ip bgp vpng4 vrf 10 summary
```

The following partial output shows that 68 prefixes were received from the neighbor *.243.229.34 with the ASN 12076 (MSEE):

```
...
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
X.243.229.34	4	12076	17671	17650	25228	0	0	1w4d	68

To see the list of prefixes received from the neighbor, use the following example:

```
sh ip bgp vpng4 vrf 10 neighbors X.243.229.34 received-routes
```

To confirm that you are receiving the correct set of prefixes, you can cross-verify. The following Azure PowerShell command output lists the prefixes advertised via Microsoft peering for each of the services and for each of the Azure region:

```
Get-AzBgpServiceCommunity
```

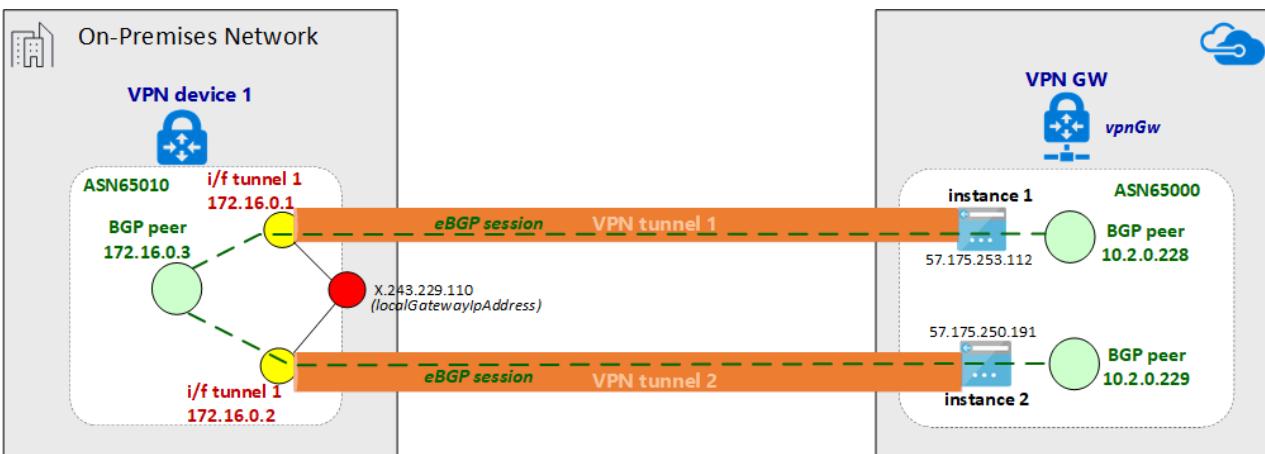
3. Configure the VPN gateway and IPsec tunnels

In this section, IPsec VPN tunnels are created between the Azure VPN gateway and the on-premises VPN device. The examples use Cisco Cloud Service Router (CSR1000) VPN devices.

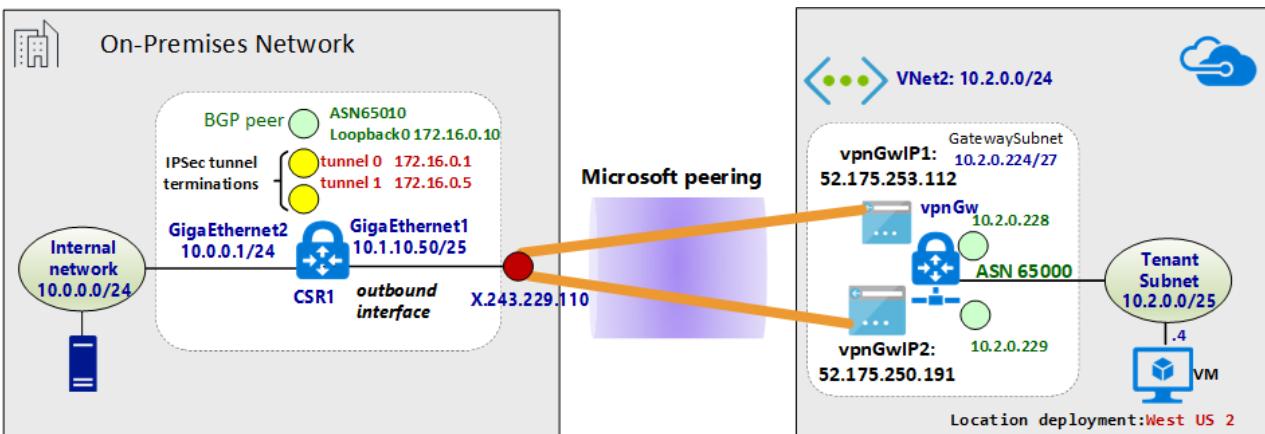
The following diagram shows the IPsec VPN tunnels established between on-premises VPN device 1, and the Azure VPN gateway instance pair. The two IPsec VPN tunnels established between the on-premises VPN device 2 and the Azure VPN gateway instance pair isn't illustrated in the diagram, and the configuration details are not listed. However, having additional VPN tunnels improves high availability.



Over the IPsec tunnel pair, an eBGP session is established to exchange private network routes. The following diagram shows the eBGP session established over the IPsec tunnel pair:



The following diagram shows the abstracted overview of the example network:



About the Azure Resource Manager template examples

In the examples, the VPN gateway and the IPsec tunnel terminations are configured using an Azure Resource Manager template. If you are new to using Resource Manager templates, or to understand the Resource Manager template basics, see [Understand the structure and syntax of Azure Resource Manager templates](#). The template in this section creates a greenfield Azure environment (VNet). However, if you have an existing VNet, you can reference it in the template. If you are not familiar with VPN gateway IPsec/IKE site-to-site configurations, see [Create a site-to-site connection](#).

NOTE

You do not need to use Azure Resource Manager templates in order to create this configuration. You can create this configuration using the Azure portal, or PowerShell.

3.1 Declare the variables

In this example, the variable declarations correspond to the example network. When declaring variables, modify this section to reflect your environment.

- The variable **localAddressPrefix** is an array of on-premises IP addresses to terminate the IPsec tunnels.
- The **gatewaySku** determines the VPN throughput. For more information about gatewaySku and vpnType, see [VPN Gateway configuration settings](#). For pricing, see [VPN Gateway pricing](#).
- Set the **vpnType** to **RouteBased**.

```

"variables": {
    "virtualNetworkName": "SecureVNet",           // Name of the Azure VNet
    "azureVNetAddressPrefix": "10.2.0.0/24",      // Address space assigned to the VNet
    "subnetName": "Tenant",                      // subnet name in which tenants exists
    "subnetPrefix": "10.2.0.0/25",                // address space of the tenant subnet
    "gatewaySubnetPrefix": "10.2.0.224/27",       // address space of the gateway subnet
    "localGatewayName": "localGW1",              // name of remote gateway (on-premises)
    "localGatewayIpAddress": "X.243.229.110",     // public IP address of the on-premises VPN device
    "localAddressPrefix": [
        "172.16.0.1/32",                         // termination of IPsec tunnel-1 on-premises
        "172.16.0.2/32"                          // termination of IPsec tunnel-2 on-premises
    ],
    "gatewayPublicIPName1": "vpnGwVIP1",          // Public address name of the first VPN gateway instance
    "gatewayPublicIPName2": "vpnGwVIP2",          // Public address name of the second VPN gateway instance
    "gatewayName": "vpnGw",                      // Name of the Azure VPN gateway
    "gatewaySKU": "VpnGw1",                      // Azure VPN gateway SKU
    "vpnType": "RouteBased",                    // type of VPN gateway
    "sharedKey": "string",                      // shared secret needs to match with on-premises configuration
    "asnVpnGateway": 65000,                      // BGP Autonomous System number assigned to the VPN Gateway
    "asnRemote": 65010,                          // BGP Autonomous System number assigned to the on-premises device
    "bgpPeeringAddress": "172.16.0.3",          // IP address of the remote BGP peer on-premises
    "connectionName": "vpn2local1",
    "vnetID": "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]",
    "gatewaySubnetRef": "[concat(variables('vnetID'), '/subnets/', 'GatewaySubnet')]",
    "subnetRef": "[concat(variables('vnetID'), '/subnets/', variables('subnetName'))]",
    "api-version": "2017-06-01"
},

```

3.2 Create virtual network (VNet)

If you are associating an existing VNet with the VPN tunnels, you can skip this step.

```

{
    "apiVersion": "[variables('api-version')]",
    "type": "Microsoft.Network/virtualNetworks",
    "name": "[variables('virtualNetworkName')]",
    "location": "[resourceGroup().location]",
    "properties": {
        "addressSpace": {
            "addressPrefixes": [
                "[variables('azureVNetAddressPrefix')]"
            ]
        },
        "subnets": [
            {
                "name": "[variables('subnetName')]",
                "properties": {
                    "addressPrefix": "[variables('subnetPrefix')]"
                }
            },
            {
                "name": "GatewaySubnet",
                "properties": {
                    "addressPrefix": "[variables('gatewaySubnetPrefix')]"
                }
            }
        ]
    },
    "comments": "Create a Virtual Network with Subnet1 and Gatewaysubnet"
},

```

3.3 Assign public IP addresses to VPN gateway instances

Assign a public IP address for each instance of a VPN gateway.

```
{
  "apiVersion": "[variables('api-version')]",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "[variables('gatewayPublicIPName1')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "publicIPAllocationMethod": "Dynamic"
  },
  "comments": "Public IP for the first instance of the VPN gateway"
},
{
  "apiVersion": "[variables('api-version')]",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "[variables('gatewayPublicIPName2')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "publicIPAllocationMethod": "Dynamic"
  },
  "comments": "Public IP for the second instance of the VPN gateway"
},
```

3.4 Specify the on-premises VPN tunnel termination (local network gateway)

The on-premises VPN devices are referred to as the **local network gateway**. The following json snippet also specifies remote BGP peer details:

```
{
  "apiVersion": "[variables('api-version')]",
  "type": "Microsoft.Network/localNetworkGateways",
  "name": "[variables('localGatewayName')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "localNetworkAddressSpace": {
      "addressPrefixes": "[variables('localAddressPrefix')]"
    },
    "gatewayIpAddress": "[variables('localGatewayIpAddress')]",
    "bgpSettings": {
      "asn": "[variables('asnRemote')]",
      "bgpPeeringAddress": "[variables('bgpPeeringAddress')]",
      "peerWeight": 0
    }
  },
  "comments": "Local Network Gateway (referred to your on-premises location) with IP address of remote tunnel peering and IP address of remote BGP peer"
},
```

3.5 Create the VPN gateway

This section of the template configures the VPN gateway with the required settings for an active-active configuration. Keep in mind the following requirements:

- Create the VPN gateway with a "**RouteBased**" VpnType. This setting is mandatory if you want to enable the BGP routing between the VPN gateway, and the VPN on-premises.
- To establish VPN tunnels between the two instances of the VPN gateway and a given on-premises device in active-active mode, the "**activeActive**" parameter is set to **true** in the Resource Manager template. To understand more about highly available VPN gateways, see [Highly available VPN gateway connectivity](#).
- To configure eBGP sessions between the VPN tunnels, you must specify two different ASNs on either side. It is preferable to specify private ASN numbers. For more information, see [Overview of BGP and Azure VPN gateways](#).

```
{
  "apiVersion": "[variables('api-version')]",
  "type": "Microsoft.Network/virtualNetworkGateways",
  "name": "[variables('gatewayName')]",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Network/publicIPAddresses/', variables('gatewayPublicIPName1'))]",
    "[concat('Microsoft.Network/publicIPAddresses/', variables('gatewayPublicIPName2'))]",
    "[concat('Microsoft.Network/virtualNetworks/', variables('virtualNetworkName'))]"
  ],
  "properties": {
    "ipConfigurations": [
      {
        "properties": {
          "privateIPAllocationMethod": "Dynamic",
          "subnet": {
            "id": "[variables('gatewaySubnetRef')]"
          },
          "publicIPAddress": {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses',variables('gatewayPublicIPName1'))]"
          }
        },
        "name": "vnetGtwConfig1"
      },
      {
        "properties": {
          "privateIPAllocationMethod": "Dynamic",
          "subnet": {
            "id": "[variables('gatewaySubnetRef')]"
          },
          "publicIPAddress": {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses',variables('gatewayPublicIPName2'))]"
          }
        },
        "name": "vnetGtwConfig2"
      }
    ],
    "sku": {
      "name": "[variables('gatewaySku')]",
      "tier": "[variables('gatewaySku')]"
    },
    "gatewayType": "Vpn",
    "vpnType": "[variables('vpnType')]",
    "enableBgp": true,
    "activeActive": true,
    "bgpSettings": {
      "asn": "[variables('asnVpnGateway')]"
    }
  },
  "comments": "VPN Gateway in active-active configuration with BGP support"
}
}
```

3.6 Establish the IPsec tunnels

The final action of the script creates IPsec tunnels between the Azure VPN gateway and the on-premises VPN device.

```
{
  "apiVersion": "[variables('api-version')]",
  "name": "[variables('connectionName')]",
  "type": "Microsoft.Network/connections",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Network/virtualNetworkGateways/', variables('gatewayName'))]",
    "[concat('Microsoft.Network/localNetworkGateways/', variables('localGatewayName'))]"
  ],
  "properties": {
    "virtualNetworkGateway1": {
      "id": "[resourceId('Microsoft.Network/virtualNetworkGateways', variables('gatewayName'))]"
    },
    "localNetworkGateway2": {
      "id": "[resourceId('Microsoft.Network/localNetworkGateways', variables('localGatewayName'))]"
    },
    "connectionType": "IPsec",
    "routingWeight": 0,
    "sharedKey": "[variables('sharedKey')]",
    "enableBGP": "true"
  },
  "comments": "Create a Connection type site-to-site (IPsec) between the Azure VPN Gateway and the VPN device on-premises"
}
```

4. Configure the on-premises VPN device

The Azure VPN gateway is compatible with many VPN devices from different vendors. For configuration information and devices that have been validated to work with VPN gateway, see [About VPN devices](#).

When configuring your VPN device, you need the following items:

- A shared key. This is the same shared key that you specify when creating your site-to-site VPN connection. The examples use a basic shared key. We recommend that you generate a more complex key to use.
- The Public IP address of your VPN gateway. You can view the public IP address by using the Azure portal, PowerShell, or CLI. To find the Public IP address of your VPN gateway using the Azure portal, navigate to Virtual network gateways, then click the name of your gateway.

Typically eBGP peers are directly connected (often over a WAN connection). However, when you are configuring eBGP over IPsec VPN tunnels via ExpressRoute Microsoft peering, there are multiple routing domains between the eBGP peers. Use the **ebgp-multihop** command to establish the eBGP neighbor relationship between the two not-directly connected peers. The integer that follows ebpgp-multihop command specifies the TTL value in the BGP packets. The command **maximum-paths eibgp 2** enables load balancing of traffic between the two BGP paths.

Cisco CSR1000 example

The following example shows the configuration for Cisco CSR1000 in a Hyper-V virtual machine as the on-premises VPN device:

```
!
crypto ikev2 proposal az-PROPOSAL
  encryption aes-cbc-256 aes-cbc-128 3des
  integrity sha1
  group 2
!
crypto ikev2 policy az-POLICY
  proposal az-PROPOSAL
!
crypto ikev2 keyring key-peer1
  peer azvpn1
    address 52.175.253.112
    pre-shared-key secret*1234
```

```
!
crypto ikev2 keyring key-peer2
peer azvpn2
  address 52.175.250.191
  pre-shared-key secret*1234
!
!
crypto ikev2 profile az-PROFILE1
  match address local interface GigabitEthernet1
  match identity remote address 52.175.253.112 255.255.255.255
  authentication remote pre-share
  authentication local pre-share
  keyring local key-peer1
!
crypto ikev2 profile az-PROFILE2
  match address local interface GigabitEthernet1
  match identity remote address 52.175.250.191 255.255.255.255
  authentication remote pre-share
  authentication local pre-share
  keyring local key-peer2
!
crypto ikev2 dpd 10 2 on-demand
!
!
crypto ipsec transform-set az-IPSEC-PROPOSAL-SET esp-aes 256 esp-sha-hmac
  mode tunnel
!
crypto ipsec profile az-VTI1
  set transform-set az-IPSEC-PROPOSAL-SET
  set ikev2-profile az-PROFILE1
!
crypto ipsec profile az-VTI2
  set transform-set az-IPSEC-PROPOSAL-SET
  set ikev2-profile az-PROFILE2
!
!
interface Loopback0
  ip address 172.16.0.3 255.255.255.255
!
interface Tunnel0
  ip address 172.16.0.1 255.255.255.255
  ip tcp adjust-mss 1350
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
  tunnel destination 52.175.253.112
  tunnel protection ipsec profile az-VTI1
!
interface Tunnel1
  ip address 172.16.0.2 255.255.255.255
  ip tcp adjust-mss 1350
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
  tunnel destination 52.175.250.191
  tunnel protection ipsec profile az-VTI2
!
interface GigabitEthernet1
  description External interface
  ip address x.243.229.110 255.255.255.252
  negotiation auto
  no mop enabled
  no mop sysid
!
interface GigabitEthernet2
  ip address 10.0.0.1 255.255.255.0
  negotiation auto
  no mop enabled
  no mop sysid
```

```

no more system
!
router bgp 65010
  bgp router-id interface Loopback0
  bgp log-neighbor-changes
  network 10.0.0.0 mask 255.255.255.0
  network 10.1.10.0 mask 255.255.255.128
  neighbor 10.2.0.228 remote-as 65000
  neighbor 10.2.0.228 ebgp-multihop 5
  neighbor 10.2.0.228 update-source Loopback0
  neighbor 10.2.0.228 soft-reconfiguration inbound
  neighbor 10.2.0.228 filter-list 10 out
  neighbor 10.2.0.229 remote-as 65000
  neighbor 10.2.0.229 ebgp-multihop 5
  neighbor 10.2.0.229 update-source Loopback0
  neighbor 10.2.0.229 soft-reconfiguration inbound
  maximum-paths eibgp 2
!
ip route 0.0.0.0 0.0.0.0 10.1.10.1
ip route 10.2.0.228 255.255.255.255 Tunnel0
ip route 10.2.0.229 255.255.255.255 Tunnel1
!
```

5. Configure VPN device filtering and firewalls (optional)

Configure your firewall and filtering according to your requirements.

6. Test and validate the IPsec tunnel

The status of IPsec tunnels can be verified on the Azure VPN gateway by Powershell commands:

```
Get-AzVirtualNetworkGatewayConnection -Name vpn2local1 -ResourceGroupName myRG | Select-Object ConnectionStatus,EgressBytesTransferred,IngressBytesTransferred | fl
```

Example output:

```
ConnectionStatus      : Connected
EgressBytesTransferred : 17734660
IngressBytesTransferred : 10538211
```

To check the status of the tunnels on the Azure VPN gateway instances independently, use the following example:

```
Get-AzVirtualNetworkGatewayConnection -Name vpn2local1 -ResourceGroupName myRG | Select-Object -ExpandProperty TunnelConnectionStatus
```

Example output:

```
Tunnel                  : vpn2local1_52.175.250.191
ConnectionStatus        : Connected
IngressBytesTransferred : 4877438
EgressBytesTransferred  : 8754071
LastConnectionEstablishedUtcTime : 11/04/2017 17:03:30

Tunnel                  : vpn2local1_52.175.253.112
ConnectionStatus        : Connected
IngressBytesTransferred : 5600773
EgressBytesTransferred  : 8980589
LastConnectionEstablishedUtcTime : 11/04/2017 17:03:13
```

You can also check the tunnel status on your on-premises VPN device.

Cisco CSR1000 example:

```
show crypto session detail
show crypto ikev2 sa
show crypto ikev2 session detail
show crypto ipsec sa
```

Example output:

```
csr1#show crypto session detail

Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect

Interface: Tunnel1
Profile: az-PROFILE2
Uptime: 00:52:46
Session status: UP-ACTIVE
Peer: 52.175.250.191 port 4500 fvrf: (none) ivrf: (none)
    Phase1_id: 52.175.250.191
    Desc: (none)
Session ID: 3
IKEv2 SA: local 10.1.10.50/4500 remote 52.175.250.191/4500 Active
    Capabilities:DN connid:3 lifetime:23:07:14
IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
    Active SAs: 2, origin: crypto map
    Inbound: #pkts dec'ed 279 drop 0 life (KB/Sec) 4607976/433
    Outbound: #pkts enc'ed 164 drop 0 life (KB/Sec) 4607992/433

Interface: Tunnel0
Profile: az-PROFILE1
Uptime: 00:52:43
Session status: UP-ACTIVE
Peer: 52.175.253.112 port 4500 fvrf: (none) ivrf: (none)
    Phase1_id: 52.175.253.112
    Desc: (none)
Session ID: 2
IKEv2 SA: local 10.1.10.50/4500 remote 52.175.253.112/4500 Active
    Capabilities:DN connid:2 lifetime:23:07:17
IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
    Active SAs: 2, origin: crypto map
    Inbound: #pkts dec'ed 668 drop 0 life (KB/Sec) 4607926/437
    Outbound: #pkts enc'ed 477 drop 0 life (KB/Sec) 4607953/437
```

The line protocol on the Virtual Tunnel Interface (VTI) does not change to "up" until IKE phase 2 has completed.

The following command verifies the security association:

```

csr1#show crypto ikev2 sa

IPv4 Crypto IKEv2 SA

Tunnel-id Local           Remote           fvrif/ivrf      Status
2       10.1.10.50/4500   52.175.253.112/4500 none/none      READY
    Encr: AES-CBC, keysize: 256, PRF: SHA1, Hash: SHA96, DH Grp:2, Auth sign: PSK, Auth verify: PSK
    Life/Active Time: 86400/3277 sec

Tunnel-id Local           Remote           fvrif/ivrf      Status
3       10.1.10.50/4500   52.175.250.191/4500 none/none      READY
    Encr: AES-CBC, keysize: 256, PRF: SHA1, Hash: SHA96, DH Grp:2, Auth sign: PSK, Auth verify: PSK
    Life/Active Time: 86400/3280 sec

IPv6 Crypto IKEv2 SA

csr1#show crypto ipsec sa | inc encaps|decaps
#pkts encaps: 177, #pkts encrypt: 177, #pkts digest: 177
#pkts decaps: 296, #pkts decrypt: 296, #pkts verify: 296
#pkts encaps: 554, #pkts encrypt: 554, #pkts digest: 554
#pkts decaps: 746, #pkts decrypt: 746, #pkts verify: 746

```

Verify end-to-end connectivity between the inside network on-premises and the Azure VNet

If the IPsec tunnels are up and the static routes are correctly set, you should be able to ping the IP address of the remote BGP peer:

```

csr1#ping 10.2.0.228
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.0.228, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/5 ms

#ping 10.2.0.229
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.0.229, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/6 ms

```

Verify the BGP sessions over IPsec

On the Azure VPN gateway, verify the status of BGP peer:

```
Get-AzVirtualNetworkGatewayBGPPeerStatus -VirtualNetworkGatewayName vpnGtw -ResourceGroupName SEA-C1-VPN-ER | ft
```

Example output:

Asn	ConnectedDuration	LocalAddress	MessagesReceived	MessagesSent	Neighbor	RoutesReceived	State
65010	00:57:19.9003584	10.2.0.228	68	72	172.16.0.10	2	Connected
65000		10.2.0.228	0	0	10.2.0.228	0	Unknown
65000	07:13:51.0109601	10.2.0.228	507	500	10.2.0.229	6	Connected

To verify the list of network prefixes received via eBGP from the VPN concentrator on-premises, you can filter by attribute "Origin":

```
Get-AzVirtualNetworkGatewayLearnedRoute -VirtualNetworkGatewayName vpnGtw -ResourceGroupName myRG | Where-Object Origin -eq "EBgp" | ft
```

In the example output, the ASN 65010 is the BGP autonomous system number in the VPN on-premises.

AsPath	LocalAddress	Network	NextHop	Origin	SourcePeer	Weight
65010	10.2.0.228	10.1.10.0/25	172.16.0.10	EBgp	172.16.0.10	32768
65010	10.2.0.228	10.0.0.0/24	172.16.0.10	EBgp	172.16.0.10	32768

To see the list of advertised routes:

```
Get-AzVirtualNetworkGatewayAdvertisedRoute -VirtualNetworkGatewayName vpnGtw -ResourceGroupName myRG -Peer 10.2.0.228 | ft
```

Example output:

AsPath	LocalAddress	Network	NextHop	Origin	SourcePeer	Weight
	10.2.0.229	10.2.0.0/24	10.2.0.229	Igp		0
	10.2.0.229	172.16.0.10/32	10.2.0.229	Igp		0
	10.2.0.229	172.16.0.5/32	10.2.0.229	Igp		0
	10.2.0.229	172.16.0.1/32	10.2.0.229	Igp		0
65010	10.2.0.229	10.1.10.0/25	10.2.0.229	Igp		0
65010	10.2.0.229	10.0.0.0/24	10.2.0.229	Igp		0

Example for the on-premises Cisco CSR1000:

```
csr1#show ip bgp neighbors 10.2.0.228 routes
BGP table version is 7, local router ID is 172.16.0.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop            Metric LocPrf Weight Path
*>   10.2.0.0/24      10.2.0.228                  0 65000  i
r>   172.16.0.1/32    10.2.0.228                  0 65000  i
r>   172.16.0.2/32    10.2.0.228                  0 65000  i
r>   172.16.0.3/32    10.2.0.228                  0 65000  i

Total number of prefixes 4
```

The list of networks advertised from the on-premises Cisco CSR1000 to the Azure VPN gateway can be listed using the following command:

```
csr1#show ip bgp neighbors 10.2.0.228 advertised-routes
BGP table version is 7, local router ID is 172.16.0.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop            Metric LocPrf Weight Path
*>   10.0.0.0/24      0.0.0.0                  0       32768  i
*>   10.1.10.0/25     0.0.0.0                  0       32768  i

Total number of prefixes 2
```

Next steps

- [Configure Network Performance Monitor for ExpressRoute](#)
- [Add a site-to-site connection to a VNet with an existing VPN gateway connection](#)

Configure IPsec transport mode for ExpressRoute private peering

11/13/2019 • 11 minutes to read • [Edit Online](#)

This article helps you create IPsec tunnels in transport mode over ExpressRoute private peering between Azure VMs running Windows, and on-premises Windows hosts. The steps in this article create this configuration using group policy objects. While it is possible to create this configuration without using organizational units (OUs) and group policy objects (GPOs), the combination of OUs and GPOs will help simplify the control of your security policies and allows you to quickly scale up. The steps in this article assume that you already have an Active Directory configuration and that you are familiar with using OUs and GPOs.

About this configuration

The configuration in the following steps use a single Azure virtual network (VNet) with ExpressRoute private peering. However, this configuration can span more Azure VNets and on-premises networks. This article will help you define an IPsec encryption policy, and apply it to a group of Azure VMs and hosts on-premises that are part of the same OU. You configure encryption between the Azure VMs (vm1 and vm2), and the on-premises host1 only for HTTP traffic with destination port 8080. Different types of IPsec policy can be created based on your requirements.

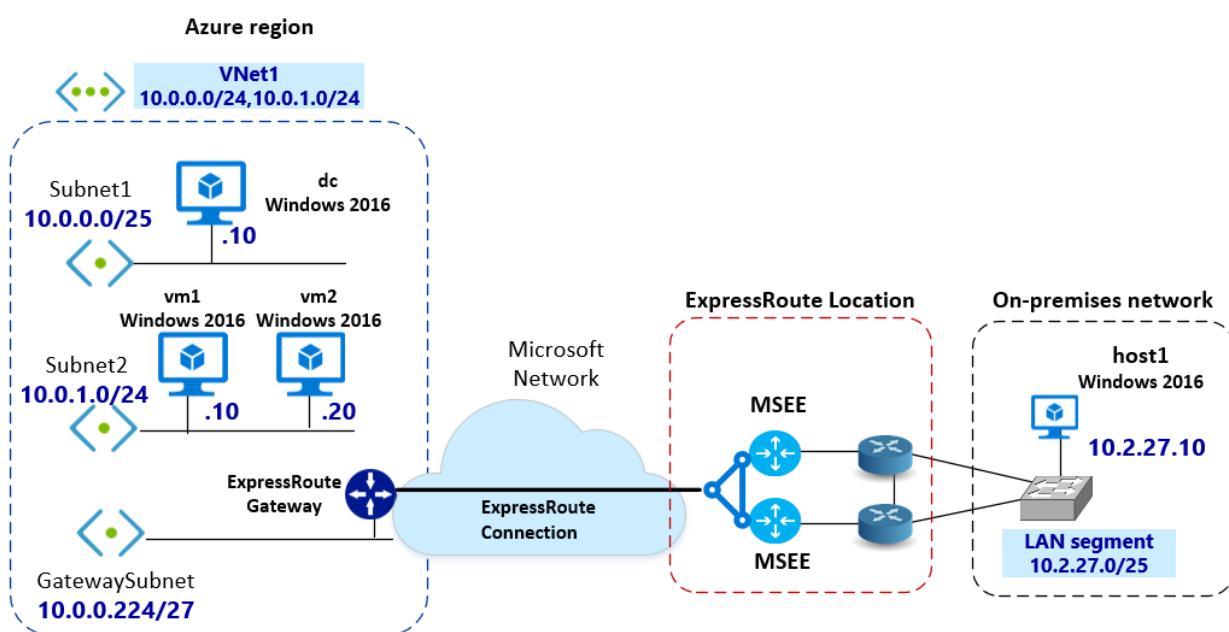
Working with OUs

The security policy associated with an OU is pushed to the computers via GPO. A few advantages to using OUs, rather than applying policies to a single host, are:

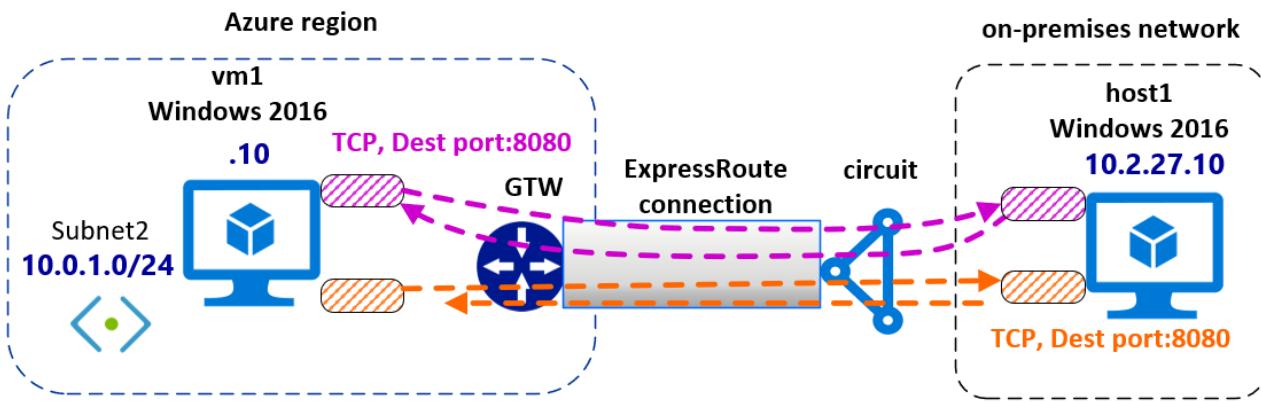
- Associating a policy with an OU guarantees that computers that belong to the same OU get the same policies.
- Changing the security policy associated with OU will apply the changes to all hosts in the OU.

Diagrams

The following diagram shows the interconnection and assigned IP address space. The Azure VMs and the on-premises host are running Windows 2016. The Azure VMs and the on-premises host1 are part of the same domain. The Azure VMs and the on-premises hosts can resolve names properly using DNS.



This diagram shows the IPsec tunnels in transit in ExpressRoute private peering.

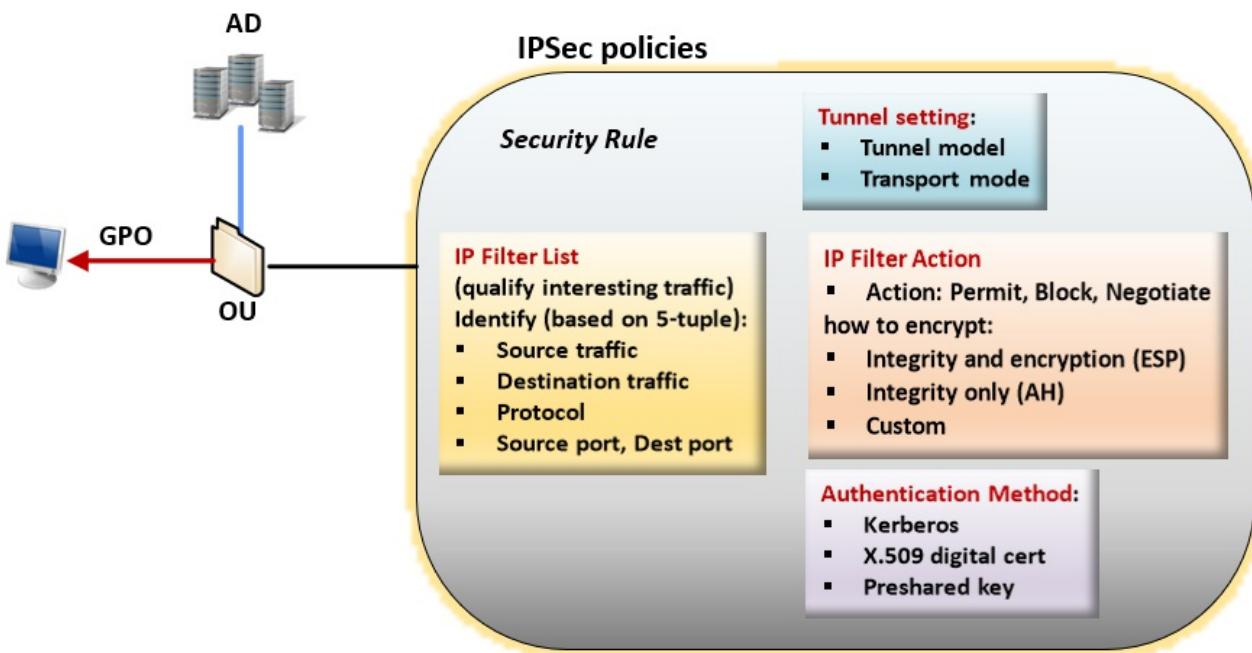


Working with IPsec policy

In Windows, encryption is associated with IPsec policy. IPsec policy determines which IP traffic is secured and the security mechanism applied to the IP packets. **IPSec policies** are composed of the following items: **Filter Lists**, **Filter Actions**, and **Security Rules**.

When configuring IPsec policy, it's important to understand the following IPsec policy terminology:

- **IPsec policy:** A collection of rules. Only one policy can be active ("assigned") at any particular time. Each policy can have one or more rules, all of which can be active simultaneously. A computer can be assigned only one active IPsec policy at given time. However, within the IPsec policy, you can define multiple actions that may be taken in different situations. Each set of IPsec rules is associated with a filter list that affects the type of network traffic to which the rule applies.
- **Filter lists:** Filter lists are bundle of one or more filters. One list can contain multiple filters. Filter defines if the communication is allowed, secured, or blocked, according to the IP address ranges, protocols, or even specific protocol ports. Each filter matches a particular set of conditions; for example, packets sent from a particular subnet to a particular computer on a specific destination port. When network conditions match one or more of those filters, the filter list is activated. Each filter is defined inside a specific filter list. Filters can't be shared between filter lists. However, a given filter list can be incorporated into several IPsec policies.
- **Filter actions:** A security method defines a set of security algorithms, protocols, and key a computer offers during IKE negotiations. Filter actions are lists of security methods, ranked in order of preference. When a computer negotiates an IPsec session, it accepts or sends proposals based on the security setting stored in filter actions list.
- **Security rules:** Rules govern how and when an IPsec policy protects communication. It uses **filter list** and **filter actions** to create an IPsec rule to build the IPsec connection. Each policy can have one or more rules, all of which can be active simultaneously. Each rule contains a list of IP filters and a collection of security actions that take place upon a match with that filter list:
 - IP Filter Actions
 - Authentication methods
 - IP tunnel settings
 - Connection types



Before you begin

Ensure that you meet the following prerequisites:

- You must have a functioning Active Directory configuration that you can use to implement Group Policy settings. For more information about GPOs, see [Group Policy Objects](#).
- You must have an active ExpressRoute circuit.
 - For information about creating an ExpressRoute circuit, see [Create an ExpressRoute circuit](#).
 - Verify that the circuit is enabled by your connectivity provider.
 - Verify that you have Azure private peering configured for your circuit. See the [configure routing](#) article for routing instructions.
 - Verify that you have a VNet and a virtual network gateway created and fully provisioned. Follow the instructions to [create a virtual network gateway for ExpressRoute](#). A virtual network gateway for ExpressRoute uses the GatewayType 'ExpressRoute', not VPN.
- The ExpressRoute virtual network gateway must be connected to the ExpressRoute circuit. For more information, see [Connect a VNet to an ExpressRoute circuit](#).
- Verify that the Azure Windows VMs are deployed to the VNet.
- Verify that there is connectivity between the on-premises hosts and the Azure VMs.
- Verify that the Azure Windows VMs and the on-premises hosts are able to use DNS to properly resolve names.

Workflow

- Create a GPO and associate it to the OU.
- Define an IPsec **Filter Action**.
- Define an IPsec **Filter List**.
- Create an IPsec Policy with **Security Rules**.
- Assign the IPsec GPO to the OU.

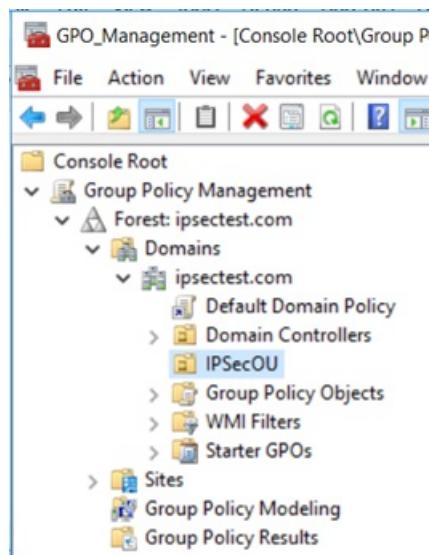
Example values

- Domain Name:** ipsectest.com

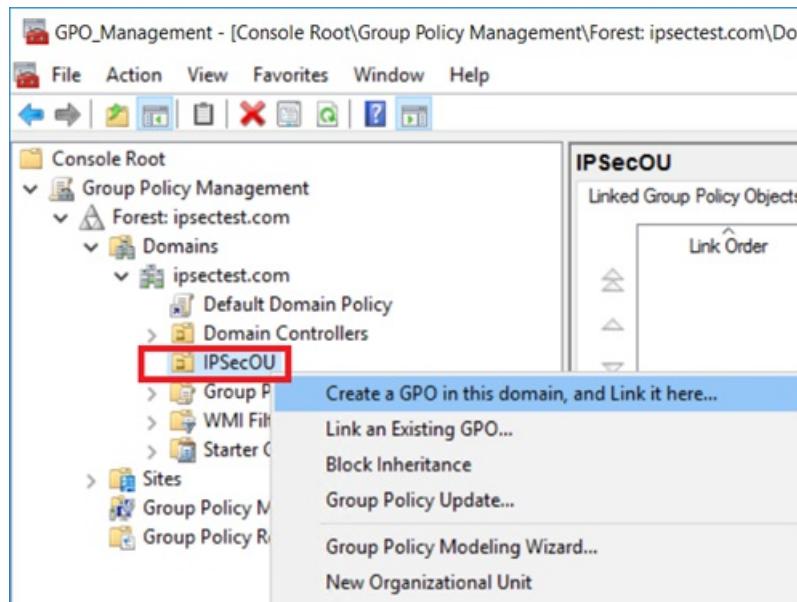
- **OU: IPSecOU**
- **On-premises Windows computer:** host1
- **Azure Windows VMs:** vm1, vm2

1. Create a GPO

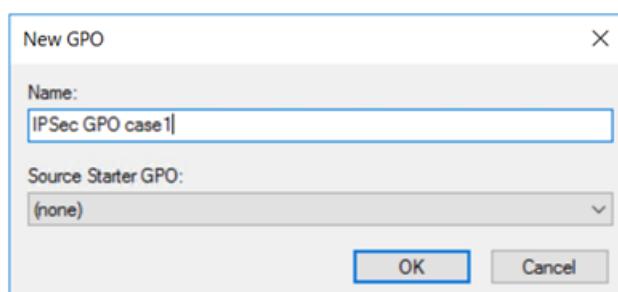
1. To create a new GPO linked to an OU, open the Group Policy Management snap-in and locate the OU to which the GPO will be linked. In the example, the OU is named **IPSecOU**.



2. In the Group Policy Management snap-in, select the OU, and right-click. In the dropdown, click "**Create a GPO in this domain, and Link it here...**".



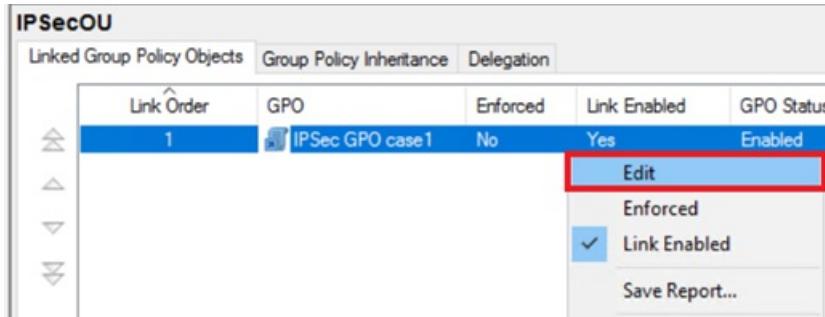
3. Name the GPO an intuitive name so that you can easily locate it later. Click **OK** to create and link the GPO.



2. Enable the GPO link

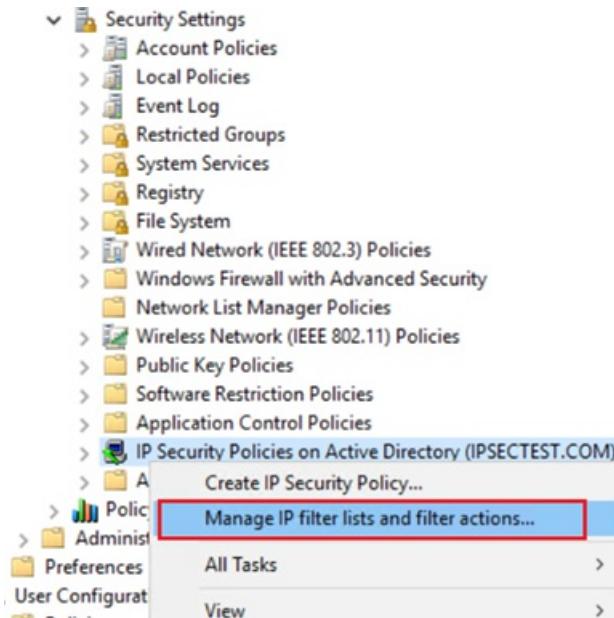
To apply the GPO to the OU, the GPO must not only be linked to the OU, but the link must be also enabled.

1. Locate the GPO that you created, right-click, and select **Edit** from the dropdown.
2. To apply the GPO to the OU, select **Link Enabled**.

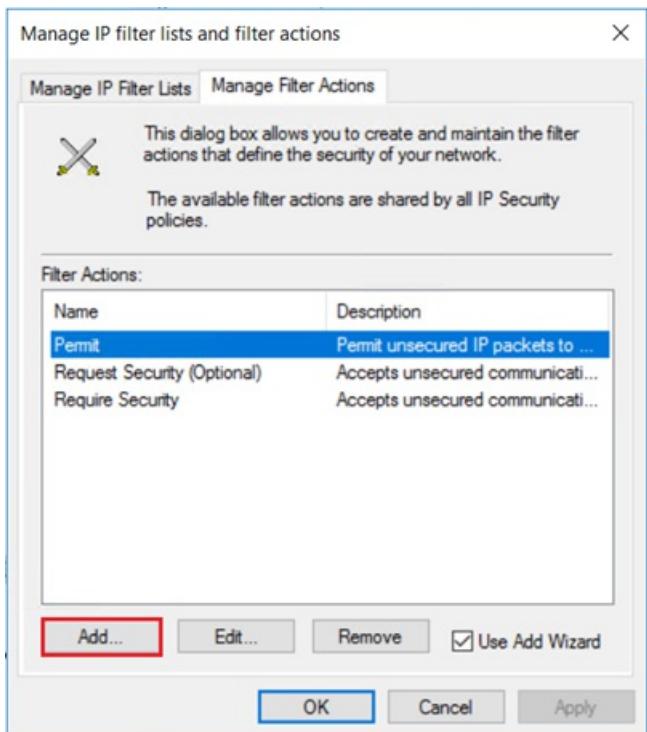


3. Define the IP filter action

1. From the drop-down, right-click **IP Security Policy on Active Directory**, and then click **Manage IP filter lists and filter actions....**



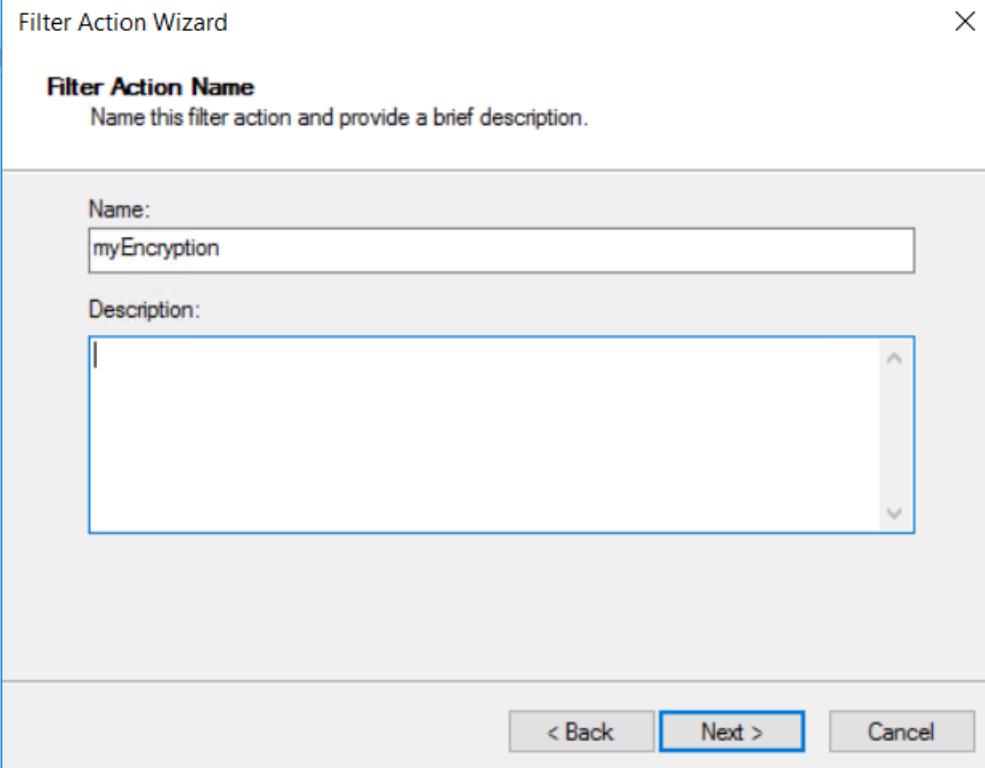
2. On the "Manage filter Actions" tab, click **Add**.



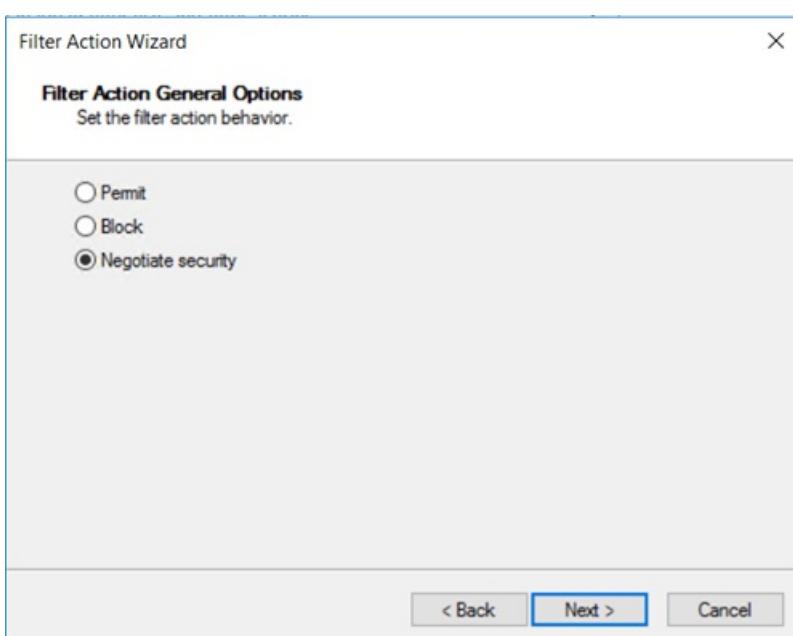
3. On the **IP Security Filter Action wizard**, click **Next**.



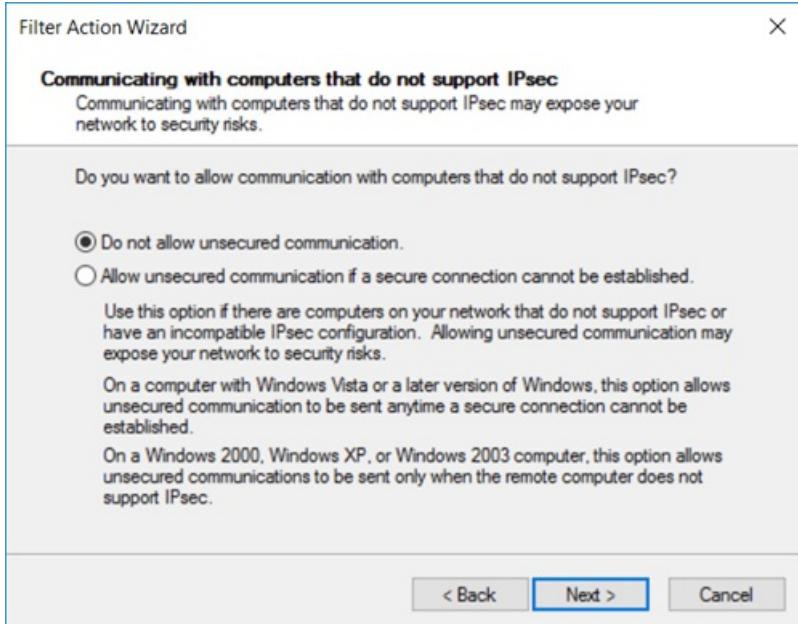
4. Name the filter action an intuitive name so that you can find it later. In this example, the filter action is named **myEncryption**. You can also add a description. Then, click **Next**.



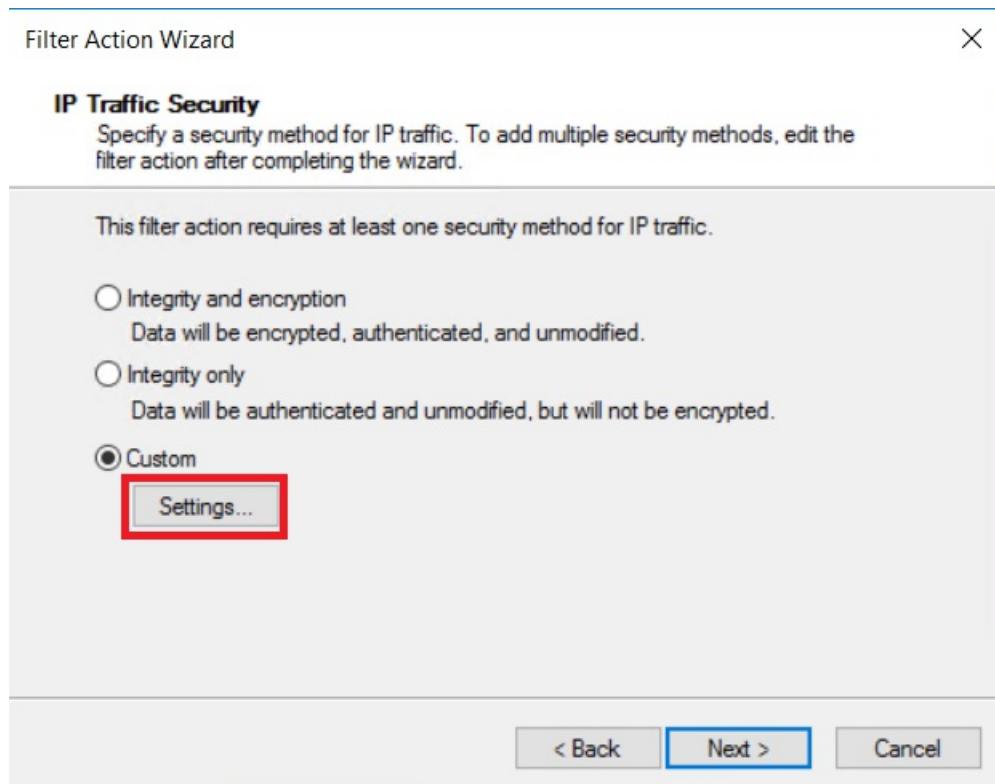
5. **Negotiate security** lets you define the behavior if IPsec can't be established with another computer. Select **Negotiate security**, then click **Next**.



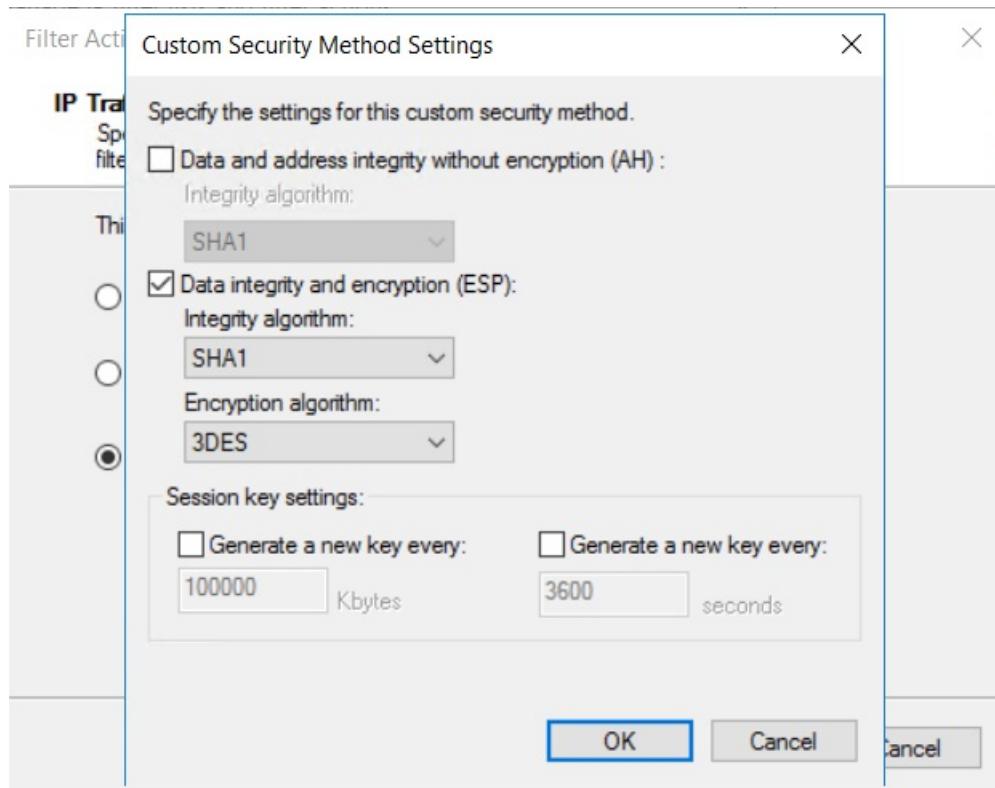
6. On the **Communicating with computers that do not support IPsec** page, select **Do not allow unsecured communication**, then click **Next**.



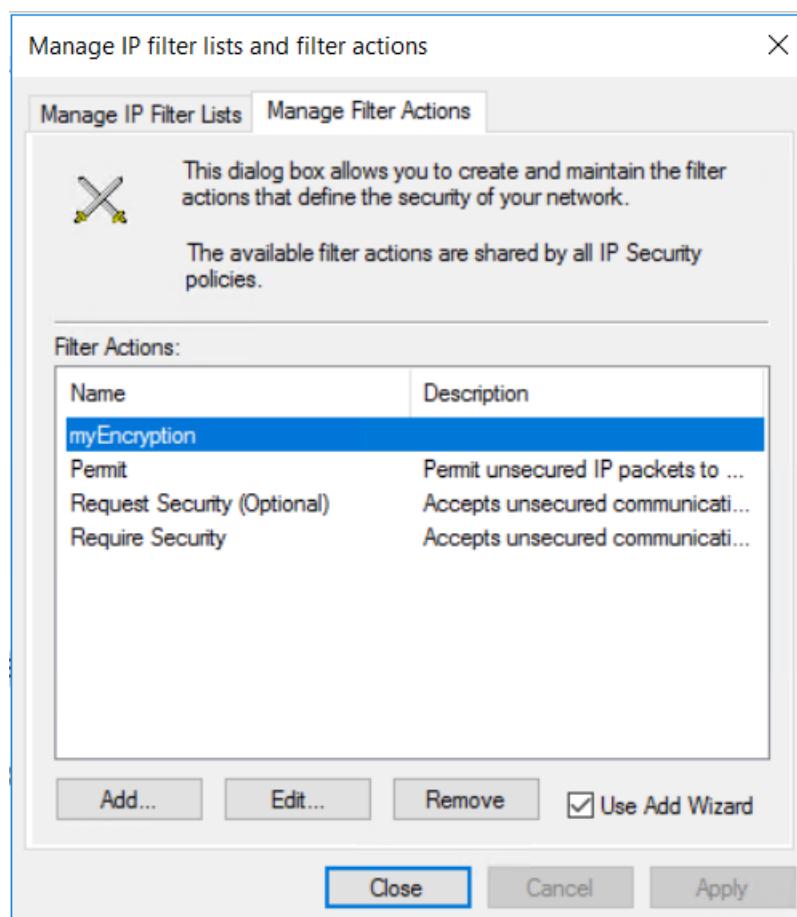
7. On the **IP Traffic and Security** page, select **Custom**, then click **Settings...**



8. On the **Custom Security Method Settings** page, select **Data integrity and encryption (ESP): SHA1, 3DES**. Then, click **OK**.



9. On the **Manage Filter Actions** page, you can see that the **myEncryption** filter was successfully added. Click **Close**.

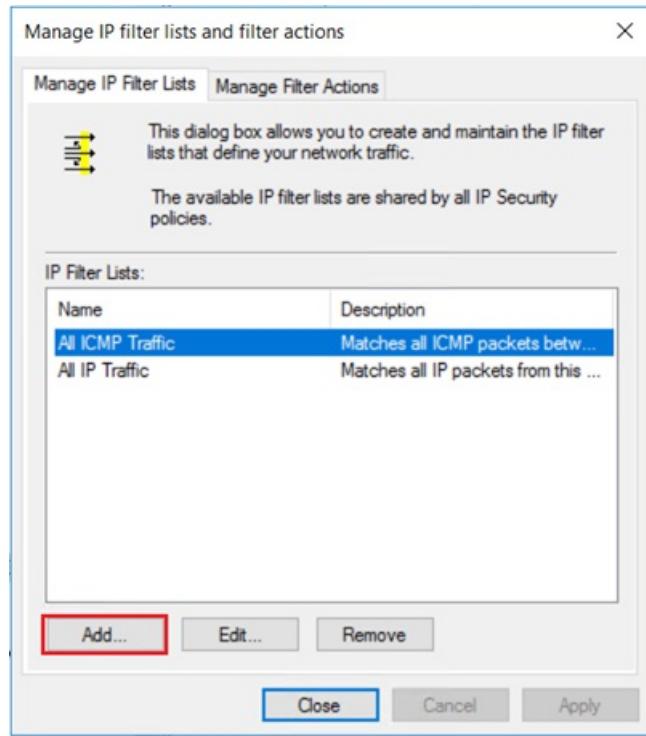


4. Define an IP filter list

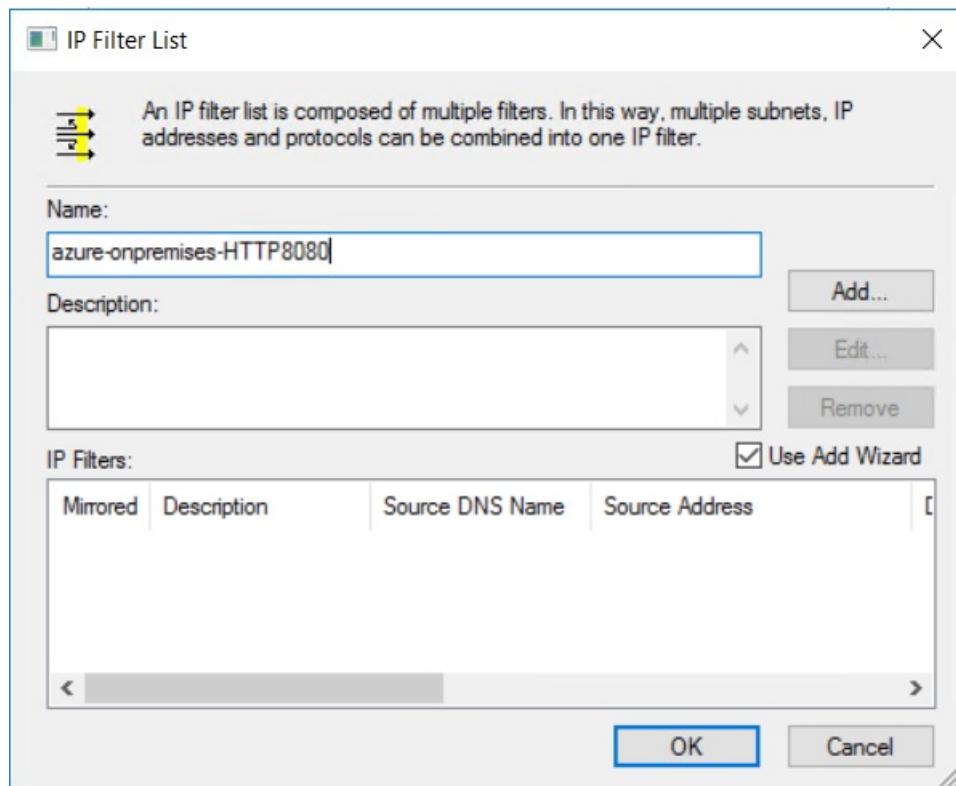
Create a filter list that specifies encrypted HTTP traffic with destination port 8080.

1. To qualify which types of traffic must be encrypted, use an **IP filter list**. In the **Manage IP Filter Lists** tab,

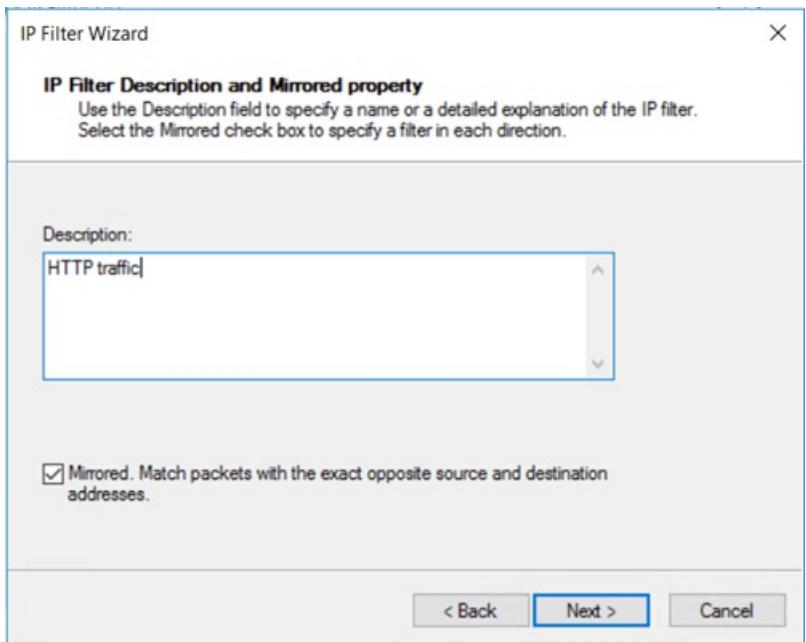
click **Add** to add a new IP filter list.



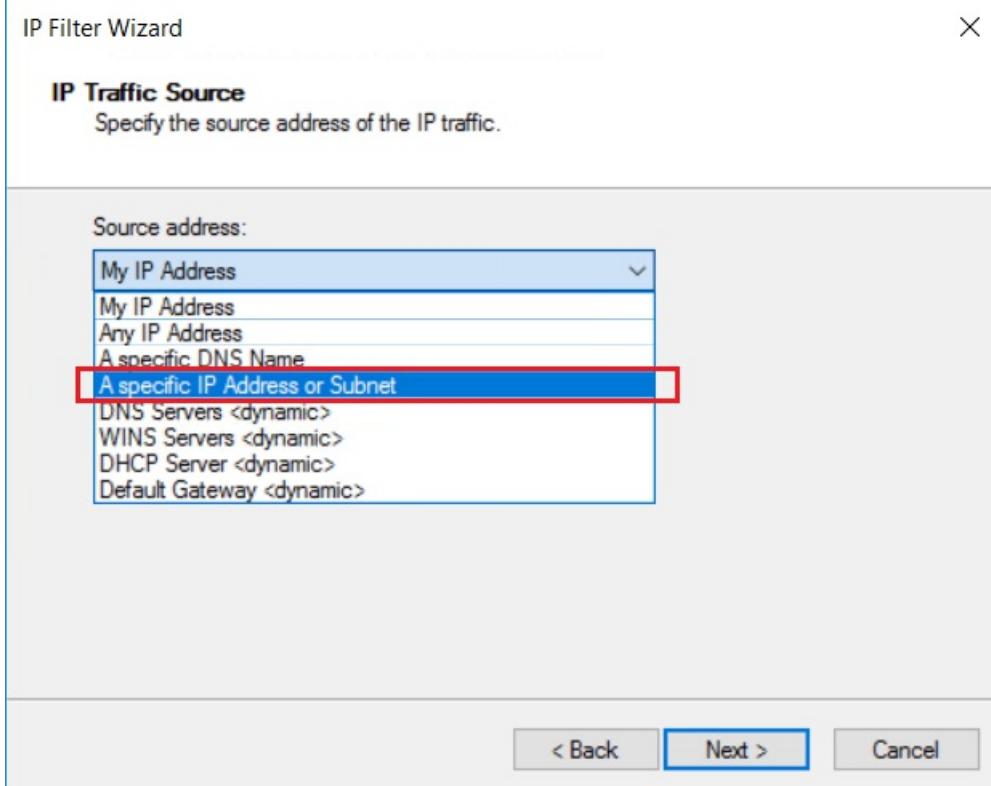
2. In the **Name:** field, type a name for your IP filter list. For example, **azure-onpremises-HTTP8080**. Then, click **Add**.



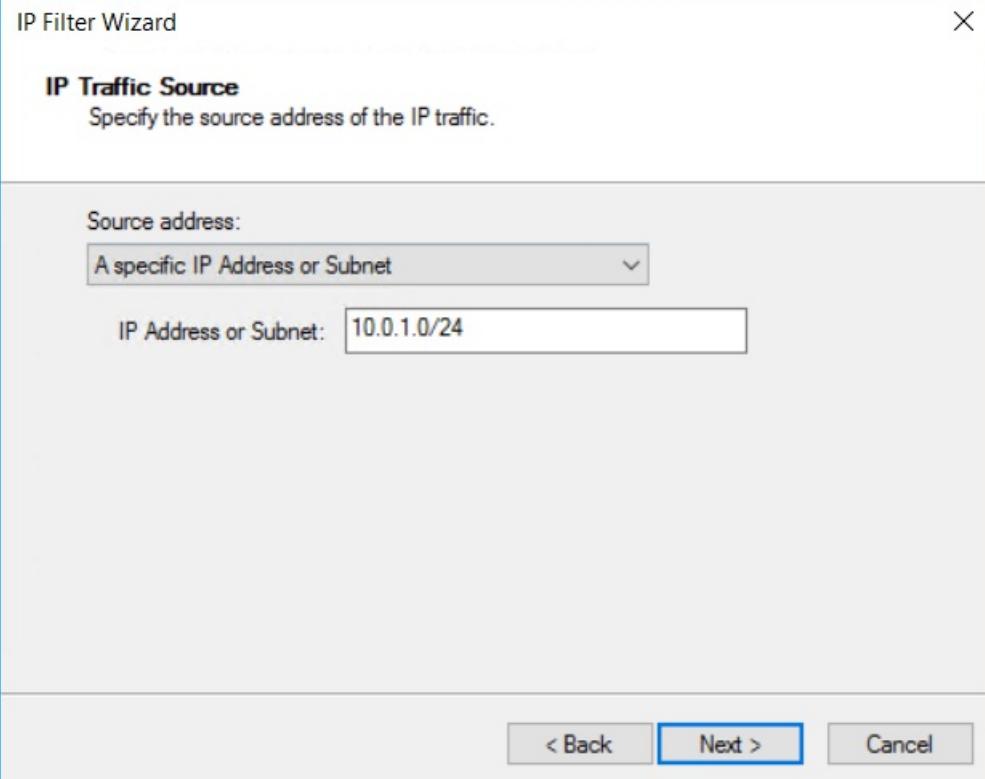
3. On the **IP Filter Description and Mirrored property** page, select **Mirrored**. The mirrored setting matches packets going in both directions, which allows for two-way communication. Then click **Next**.



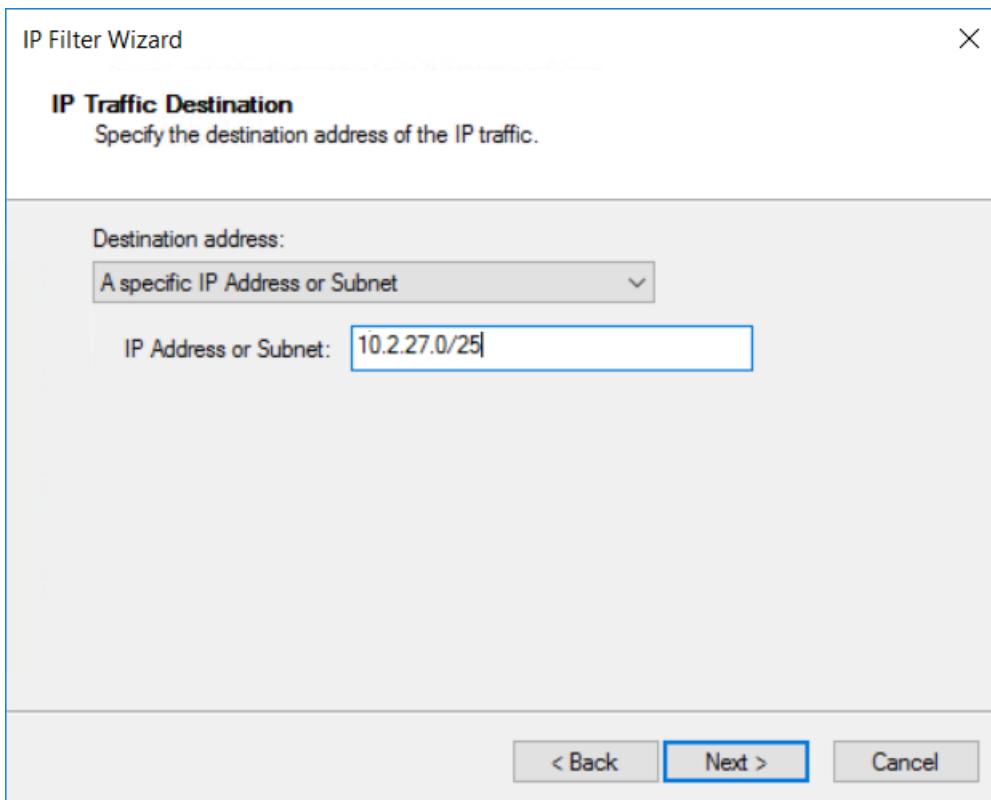
4. On the **IP Traffic Source** page, from the **Source address:** dropdown, choose **A specific IP Address or Subnet**.



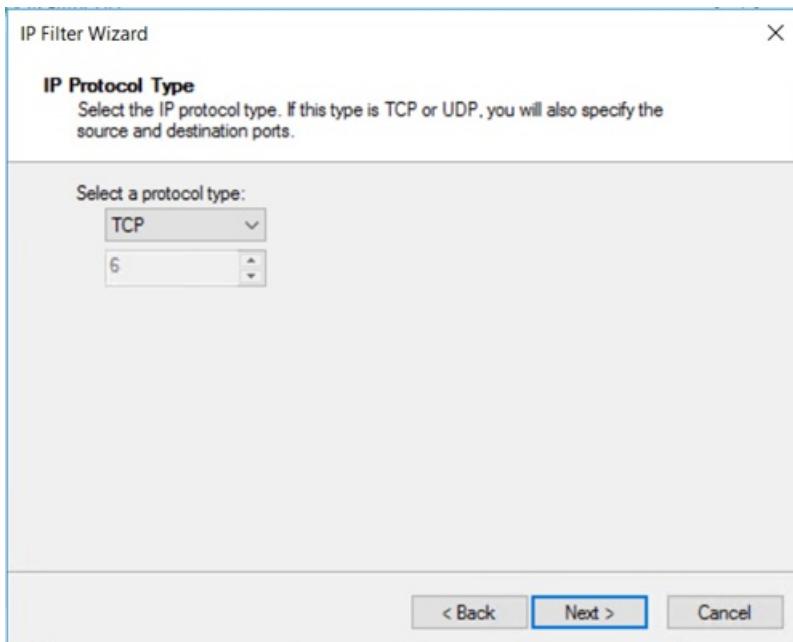
5. Specify the source address **IP Address or Subnet:** of the IP traffic, then click **Next**.



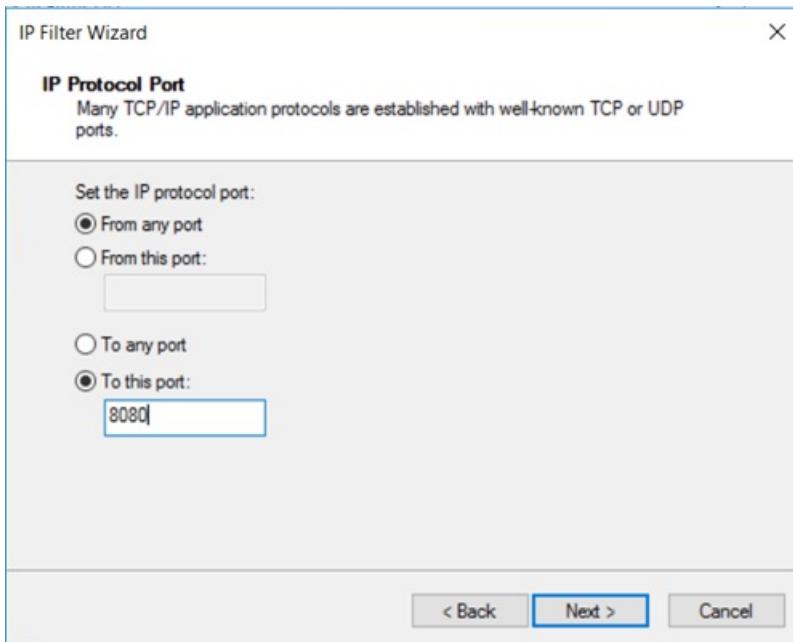
6. Specify the **Destination address**: IP Address or Subnet. Then, click **Next**.



7. On the **IP Protocol Type** page, select **TCP**. Then, click **Next**.

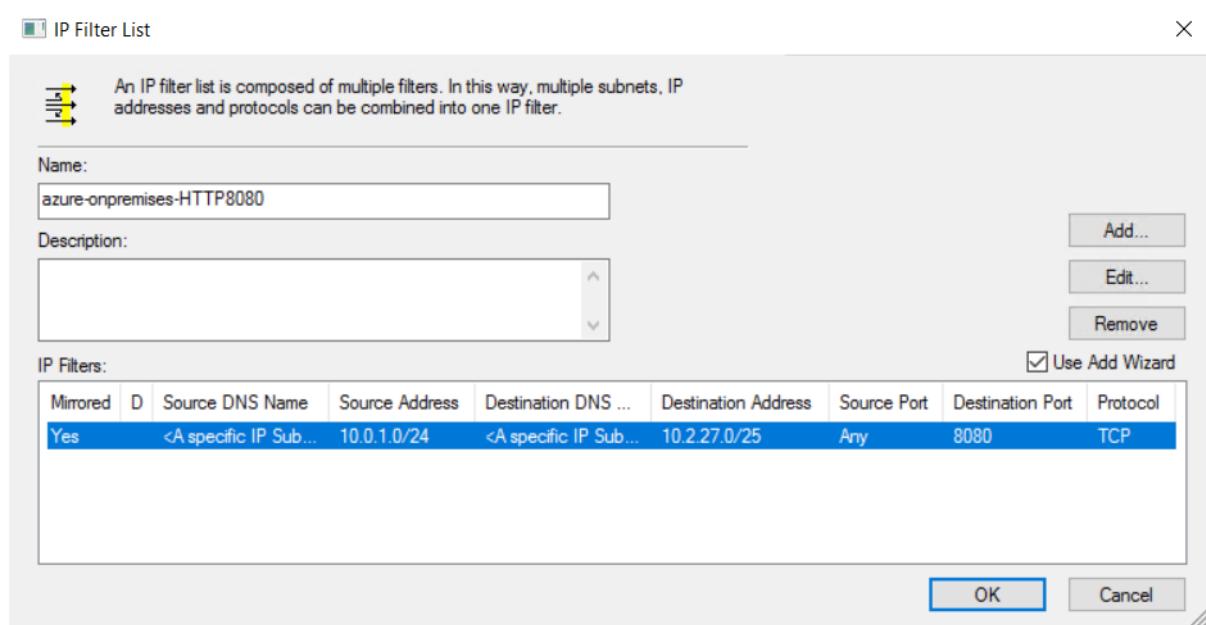


8. On the **IP Protocol Port** page, select **From any port** and **To this port**: Type **8080** in the text box. These settings specify only the HTTP traffic on destination port 8080 will be encrypted. Then, click **Next**.



9. View the IP filter list. The configuration of the IP Filter List **azure-onpremises-HTTP8080** triggers encryption for all traffic that matches the following criteria:

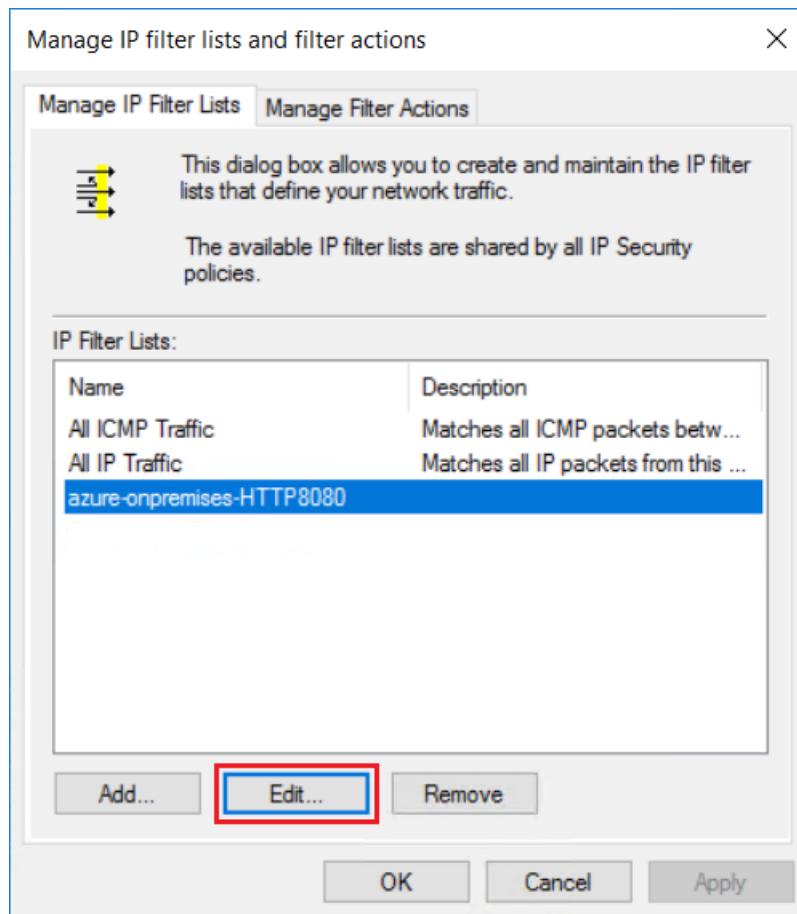
- Any source address in 10.0.1.0/24 (Azure Subnet2)
- Any destination address in 10.2.27.0/25 (on-premises subnet)
- TCP protocol
- Destination port 8080



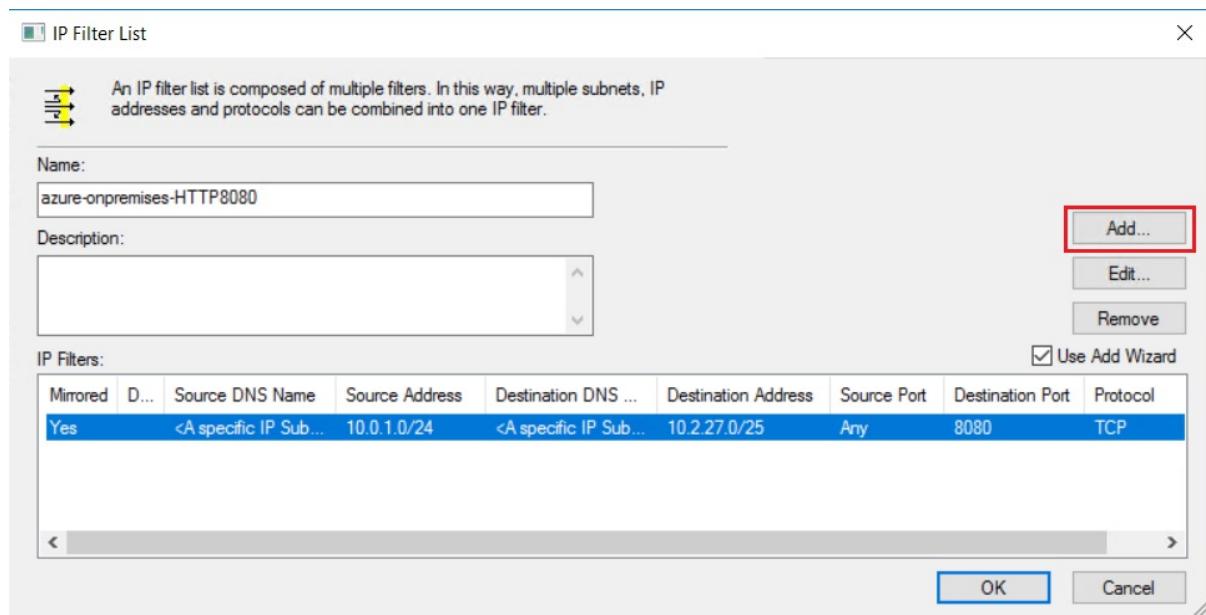
5. Edit the IP filter list

To encrypt the same type of traffic in opposite direction (from the on-premises host to the Azure VM) you need a second IP filter. The process of setting up of the new filter is the same process you used to set up the first IP filter. The only differences are the source subnet and destination subnet.

1. To add a new IP filter to the IP Filter List, select **Edit**.



2. On the **IP Filter List** page, click **Add**.



3. Create a second IP filter using the settings in the following example:

IP Filter Properties

Addresses Protocol Description

Source address:
A specific IP Address or Subnet

Destination address:
A specific IP Address or Subnet

Mirrored. Match packets with the exact opposite source and destination addresses.

OK Cancel

IP Filter Properties

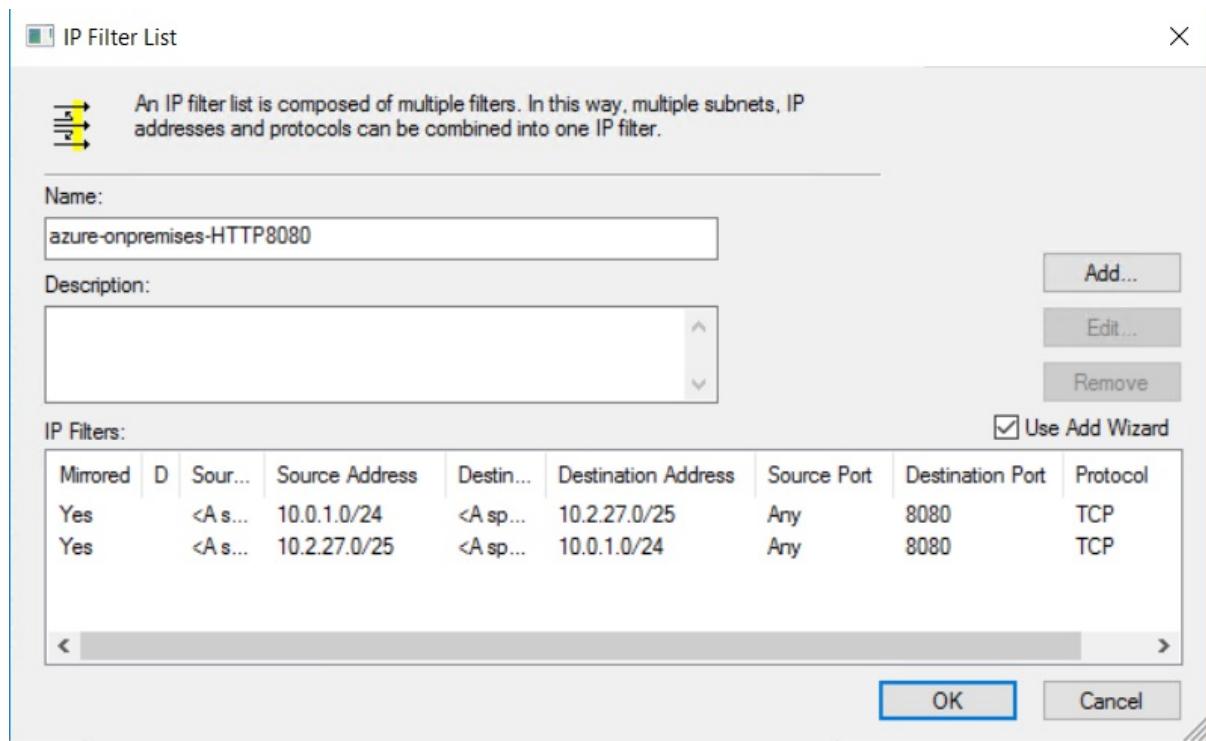
Addresses Protocol Description

Select a protocol type:
TCP
6

Set the IP protocol port:
 From any port
 From this port:
 To any port
 To this port:
8080

OK Cancel

4. After you create the second IP filter, the IP filter list will look like this:

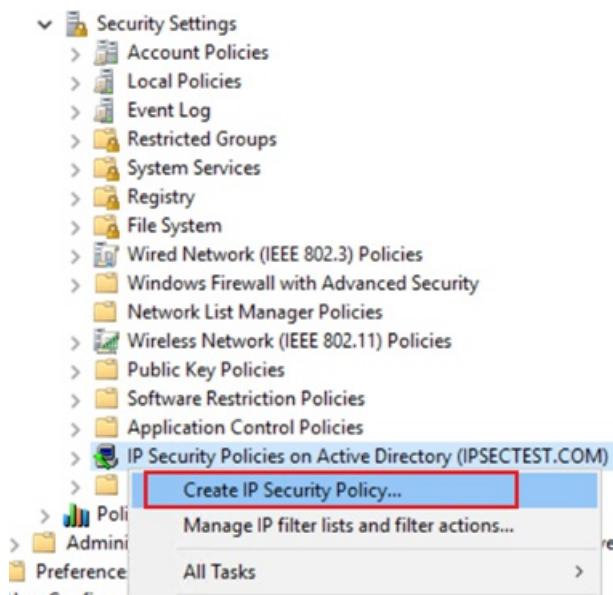


If encryption is required between an on-premises location and an Azure subnet to protect an application, instead of modifying the existing IP filter list, you can add a new IP filter list instead. Associating 2 IP filter lists to the same IPsec policy provides better flexibility because a specific IP filter list can be modified or removed at any time without impacting the other IP filter lists.

6. Create an IPsec security policy

Create an IPsec policy with security rules.

1. Select the **IPSecurity Policies on Active directory** that is associated with the OU. Right-click, and select **Create IP Security Policy**.



2. Name the security policy. For example, **policy-azure-onpremises**. Then, click **Next**.

IP Security Policy Name

Name this IP Security policy and provide a brief description

Name:

policy-azure-onpremises

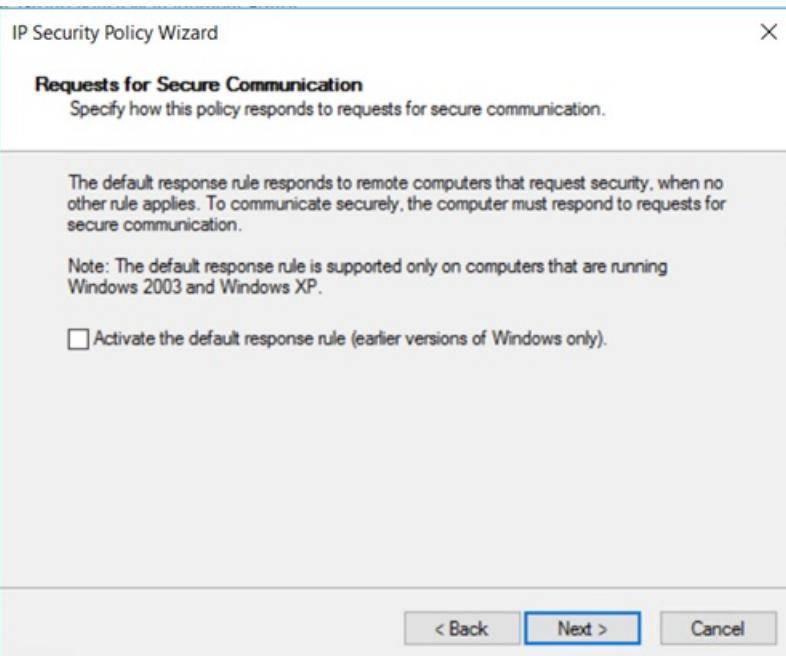
Description:

< Back

Next >

Cancel

3. Click **Next** without selecting the checkbox.



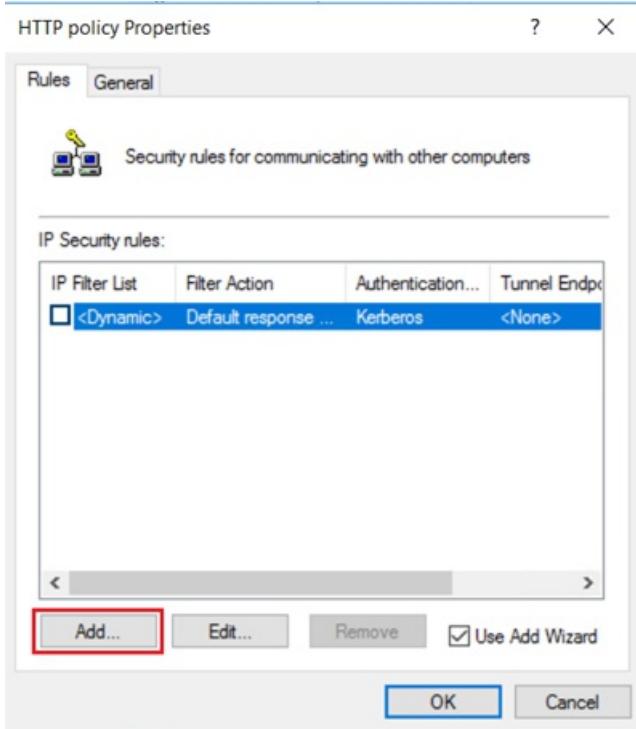
4. Verify that the **Edit properties** checkbox is selected, and then click **Finish**.



7. Edit the IPsec security policy

Add to the IPsec policy the **IP Filter List** and **Filter Action** that you previously configured.

1. On the HTTP policy Properties **Rules** tab, click **Add**.

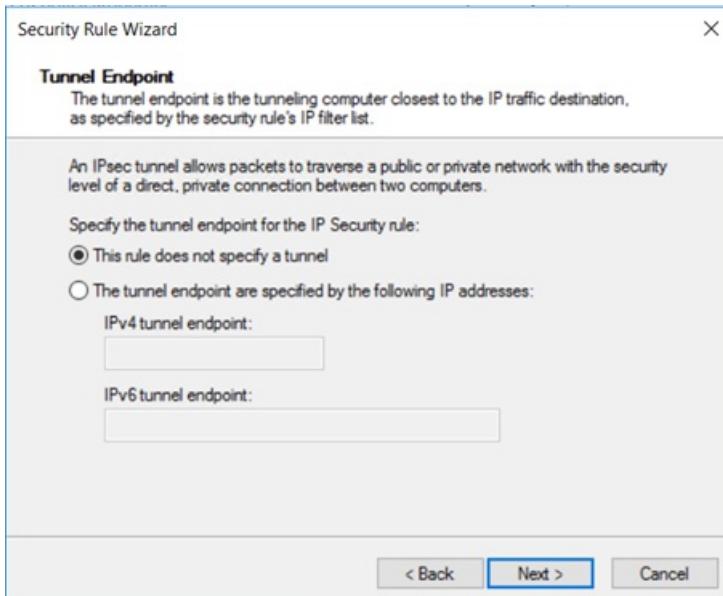


2. On the Welcome page, click **Next**.

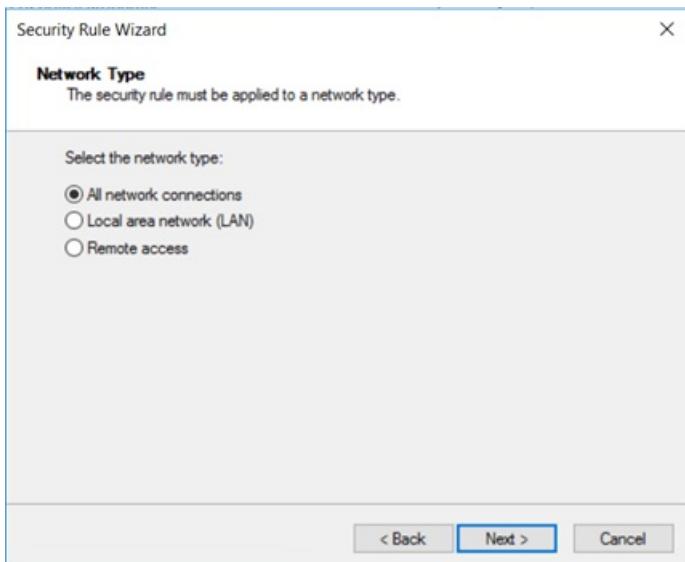


3. A rule provides the option to define the IPsec mode: tunnel mode or transport mode.
- In tunnel mode, the original packet is encapsulated by a set of IP headers. Tunnel mode protects the internal routing information by encrypting the IP header of the original packet. Tunnel mode is widely implemented between gateways in site-to-site VPN scenarios. Tunnel mode is in most of cases used for end-to-end encryption between hosts.
 - Transport mode encrypts only the payload and ESP trailer; the IP header of the original packet isn't encrypted. In transport mode, the IP source and IP destination of the packets are unchanged.

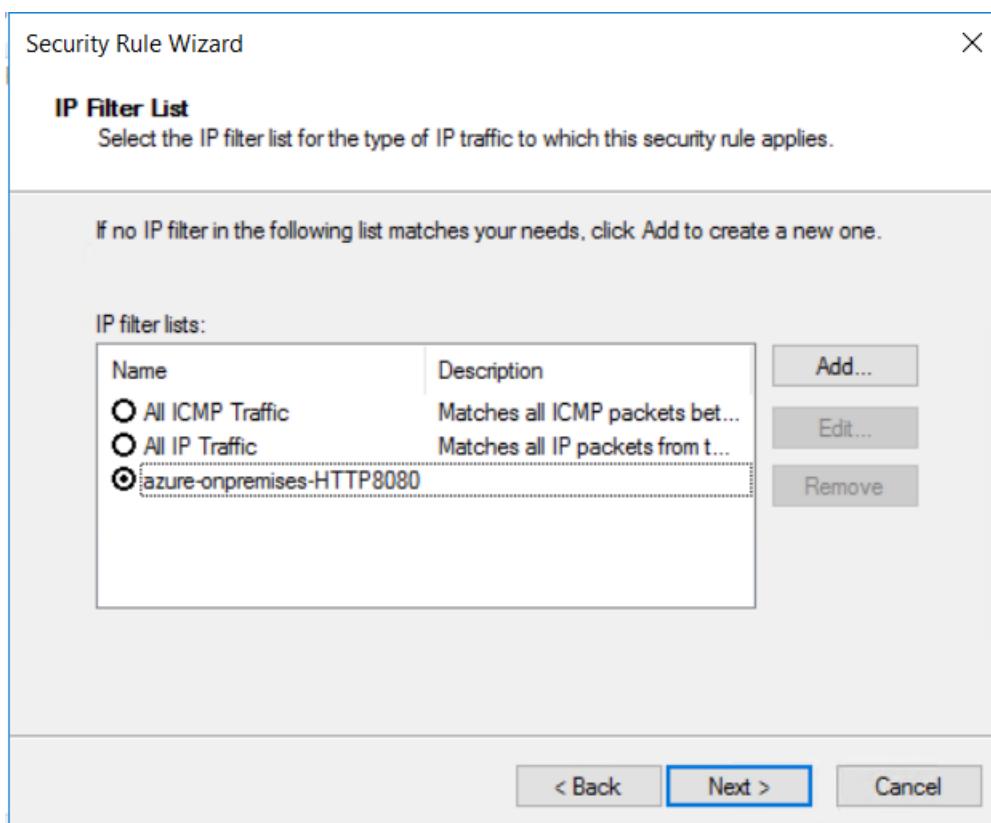
Select **This rule does not specify a tunnel**, and then click **Next**.



4. **Network Type** defines which network connection associates with the security policy. Select **All network connections**, and then click **Next**.



5. Select the IP filter list that you created previously, **azure-onpremises-HTTP8080**, and then click **Next**.



6. Select the existing Filter Action **myEncryption** that you created previously.

Security Rule Wizard

X

Filter Action

Select the filter action for this security rule.

If no filter action in the following list matches your needs, click Add to create a new one.
Select Use Add Wizard to create a filter action using the wizard.

Filter Actions:

Use Add Wizard

Name	Description
<input checked="" type="radio"/> myEncryption	
<input type="radio"/> Permit	Permit unsecured IP packets t...
<input type="radio"/> Request Security (Optional)	Accepts unsecured communi...
<input type="radio"/> Require Security	Accepts unsecured communi...

Add...

Edit...

Remove

< Back

Next >

Cancel

7. Windows supports four distinct types of authentications: Kerberos, certificates, NTLMv2, and pre-shared key. Because we are working with domain-joined hosts, select **Active Directory default (Kerberos V5 protocol)**, and then click **Next**.

Security Rule Wizard

X

Authentication Method

To add multiple authentication methods, edit the security rule after completing the wizard.

Set the initial authentication method for this security rule:

Active Directory default (Kerberos V5 protocol)

Use a certificate from this certification authority (CA):

Browse...

Exclude the CA name from the certificate request

Enable certificate to account mapping

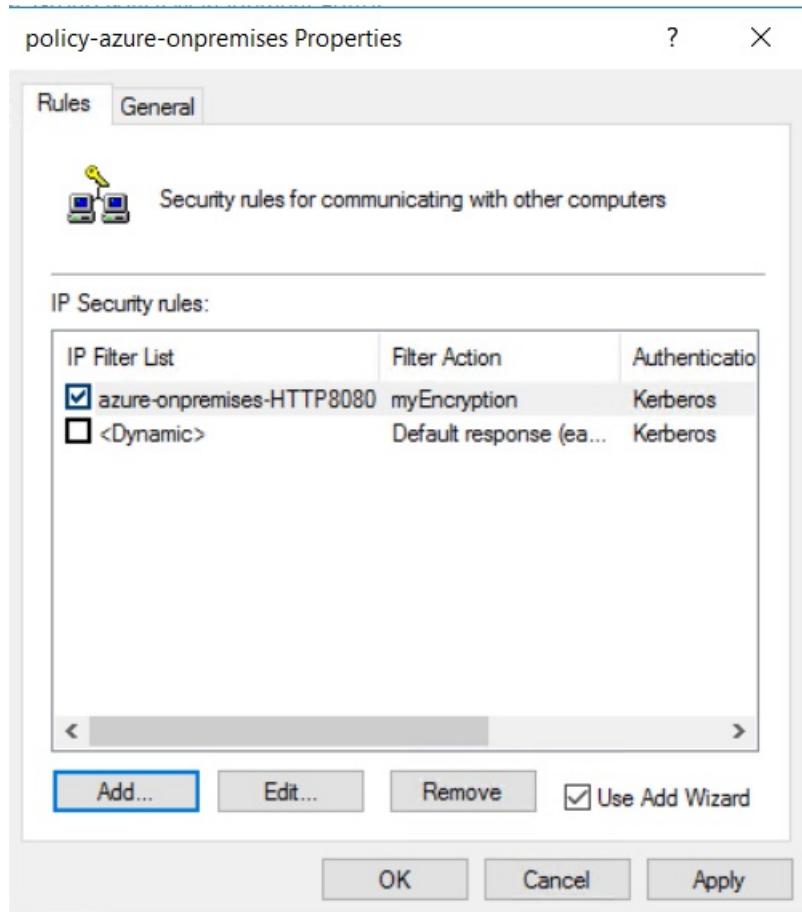
Use this string to protect the key exchange (preshared key):

< Back

Next >

Cancel

8. The new policy creates the security rule: **azure-onpremises-HTTP8080**. Click **OK**.



The IPsec policy requires all HTTP connections on the destination port 8080 to use IPsec transport mode. Because HTTP is a clear text protocol, having the security policy enabled ensures data is encrypted when transferred through the ExpressRoute private peering. IP Security policy for Active Directory is more complex to configure than Windows Firewall with Advanced Security, but it does allow for more customization of the IPsec connection.

8. Assign the IPsec GPO to the OU

1. View the policy. The security group policy is defined, but not yet assigned.

Name	Description	Policy Assigned
Client (Respond Only)	Communicate normally (...)	No
policy-azure-onpremises		No
Secure Server (Require Security)	For all IP traffic, always re...	No
Server (Request Security)	For all IP traffic, always re...	No

2. To assign the security group policy to the OU **IPSecOU**, right-click the security policy and chose **Assign**. Every computer that belongs to the OU will have the security group policy assigned.

Name	Description	Policy Assigned
Client (Respond Only)	Communicate normally (...	No
policy-azure-onpremises		No
Secure Server (Require Security)	For all IP	Assign
Server (Request Security)	For all IP	

Check traffic encryption

To check out the encryption GPO applied on the OU, install IIS on all Azure VMs and in the host1. Every IIS is customized to answer to HTTP requests on port 8080. To verify encryption, you can install a network sniffer (like Wireshark) in all computers in the OU. A powershell script works as an HTTP client to generate HTTP requests on port 8080:

```
$url = "http://10.0.1.20:8080"
while ($true) {
try {
[net.httpWebRequest]
$req = [net.webRequest]::create($url)
$req.method = "GET"
$req.ContentType = "application/x-www-form-urlencoded"
$req.TimeOut = 60000

$start = get-date
[net.httpWebResponse] $res = $req.getResponse()
$timetaken = ((get-date) - $start).TotalMilliseconds

Write-Output $res.Content
Write-Output ("{0} {1} {2}" -f (get-date), $res.StatusCode.value__, $timetaken)
$req = $null
$res.Close()
$res = $null
} catch [Exception] {
Write-Output ("{0} {1}" -f (get-date), $_.ToString())
}
$req = $null

# uncomment the line below and change the wait time to add a pause between requests
#Start-Sleep -Seconds 1
}
```

The following network capture shows the results for on-premises host1 with display filter ESP to match only the encrypted traffic:

No.	Time	Source	Destination	Protocol	Length	Info
8874	214.175249	10.0.1.10	10.2.27.10	ESP	174	ESP (SPI=0xa946b16a)
8875	214.178654	10.2.27.10	10.0.1.10	ESP	1094	ESP (SPI=0x34c296ea)
8876	214.210773	10.2.27.10	10.0.1.10	ESP	1094	ESP (SPI=0x34c296ea)
8877	214.210978	10.0.1.10	10.2.27.10	ESP	102	ESP (SPI=0xa946b16a)
8902	215.187534	10.0.1.10	10.2.27.10	ESP	174	ESP (SPI=0xa946b16a)
8903	215.191057	10.2.27.10	10.0.1.10	ESP	1094	ESP (SPI=0x34c296ea)
8904	215.237499	10.2.27.10	10.0.1.10	ESP	1094	ESP (SPI=0x34c296ea)
8905	215.237703	10.0.1.10	10.2.27.10	ESP	102	ESP (SPI=0xa946b16a)

If you run the powershell script on-premises (HTTP client), the network capture in the Azure VM shows a similar trace.

Next steps

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

Configure ExpressRoute Global Reach

11/14/2019 • 4 minutes to read • [Edit Online](#)

This article helps you configure ExpressRoute Global Reach using PowerShell. For more information, see [ExpressRoute Global Reach](#).

Before you begin

Before you start configuration, confirm the following:

- You understand ExpressRoute circuit provisioning [workflows](#).
- Your ExpressRoute circuits are in a provisioned state.
- Azure private peering is configured on your ExpressRoute circuits.
- If you want to run PowerShell locally, verify that the latest version of Azure PowerShell is installed on your computer.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Identify circuits

1. To start the configuration, sign in to your Azure account and select the subscription that you want to use.

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

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

```
Connect-AzAccount
```

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

```
Get-AzSubscription
```

Specify the subscription that you want to use.

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

2. Identify the ExpressRoute circuits that you want use. You can enable ExpressRoute Global Reach between any two ExpressRoute circuits as long as they're located in the supported countries/regions and were created at different peering locations.
 - If your subscription owns both circuits, you can choose either circuit to run the configuration in the following sections.
 - If the two circuits are in different Azure subscriptions, you need authorization from one Azure subscription. Then you pass in the authorization key when you run the configuration command in the other Azure subscription.

Enable connectivity

Enable connectivity between your on-premises networks. There are separate sets of instructions for circuits that are in the same Azure subscription, and circuits that are different subscriptions.

ExpressRoute circuits in the same Azure subscription

1. Use the following commands to get circuit 1 and circuit 2. The two circuits are in the same subscription.

```
$ckt_1 = Get-AzExpressRouteCircuit -Name "Your_circuit_1_name" -ResourceGroupName "Your_resource_group"  
$ckt_2 = Get-AzExpressRouteCircuit -Name "Your_circuit_2_name" -ResourceGroupName "Your_resource_group"
```

2. Run the following command against circuit 1, and pass in the private peering ID of circuit 2. When running the command, note the following:

- The private peering ID looks similar to the following example:

```
/subscriptions/{your_subscription_id}/resourceGroups/{your_resource_group}/providers/Microsoft.Network/expressRouteCircuits/{your_circuit_name}/peerings/AzurePrivatePeering
```

- *-AddressPrefix* must be a /29 IPv4 subnet, for example, "10.0.0.0/29". We use IP addresses in this subnet to establish connectivity between the two ExpressRoute circuits. You shouldn't use the addresses in this subnet in your Azure virtual networks, or in your on-premises network.

```
Add-AzExpressRouteCircuitConnectionConfig -Name 'Your_connection_name' -ExpressRouteCircuit $ckt_1 -PeerExpressRouteCircuitPeering $ckt_2.Peerings[0].Id -AddressPrefix '_._._._./29'
```

3. Save the configuration on circuit 1 as follows:

```
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt_1
```

When the previous operation completes, you will have connectivity between your on-premises networks on both sides through your two ExpressRoute circuits.

ExpressRoute circuits in different Azure subscriptions

If the two circuits are not in the same Azure subscription, you need authorization. In the following configuration, authorization is generated in the circuit 2 subscription, and the authorization key is passed to circuit 1.

1. Generate an authorization key.

```
$ckt_2 = Get-AzExpressRouteCircuit -Name "Your_circuit_2_name" -ResourceGroupName "Your_resource_group"  
Add-AzExpressRouteCircuitAuthorization -ExpressRouteCircuit $ckt_2 -Name "Name_for_auth_key"  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt_2
```

Make a note of the private peering ID of circuit 2, as well as the authorization key.

2. Run the following command against circuit 1. Pass in the private peering ID of circuit 2 and the authorization key.

```
Add-AzExpressRouteCircuitConnectionConfig -Name 'Your_connection_name' -ExpressRouteCircuit $ckt_1 -  
PeerExpressRouteCircuitPeering "circuit_2_private_peering_id" -AddressPrefix '__.__.__.__/29' -  
AuthorizationKey '#####-###-###-###-#####-#####'
```

3. Save the configuration on circuit 1.

```
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt_1
```

When the previous operation completes, you will have connectivity between your on-premises networks on both sides through your two ExpressRoute circuits.

Verify the configuration

Use the following command to verify the configuration on the circuit where the configuration was made (for example, circuit 1 in the previous example).

```
$ckt1 = Get-AzExpressRouteCircuit -Name "Your_circuit_1_name" -ResourceGroupName "Your_resource_group"
```

If you simply run `$ckt1` in PowerShell, you see *CircuitConnectionStatus* in the output. It tells you whether the connectivity is established, "Connected", or "Disconnected".

Disable connectivity

To disable connectivity between your on-premises networks, run the commands against the circuit where the configuration was made (for example, circuit 1 in the previous example).

```
$ckt1 = Get-AzExpressRouteCircuit -Name "Your_circuit_1_name" -ResourceGroupName "Your_resource_group"  
Remove-AzExpressRouteCircuitConnectionConfig -Name "Your_connection_name" -ExpressRouteCircuit $ckt_1  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt_1
```

You can run the Get operation to verify the status.

After the previous operation is complete, you no longer have connectivity between your on-premises network through your ExpressRoute circuits.

Next steps

1. [Learn more about ExpressRoute Global Reach](#)
2. [Verify ExpressRoute connectivity](#)
3. [Link an ExpressRoute circuit to an Azure virtual network](#)

Configure ExpressRoute Global Reach by using the Azure CLI

11/13/2019 • 3 minutes to read • [Edit Online](#)

This article helps you configure Azure ExpressRoute Global Reach by using the Azure CLI. For more information, see [ExpressRoute Global Reach](#).

Before you start configuration, complete the following requirements:

- Install the latest version of the Azure CLI. See [Install the Azure CLI](#) and [Get started with Azure CLI](#).
- Understand the ExpressRoute circuit-provisioning [workflows](#).
- Make sure your ExpressRoute circuits are in the Provisioned state.
- Make sure Azure private peering is configured on your ExpressRoute circuits.

Sign in to your Azure account

To start configuration, sign in to your Azure account. The following command opens your default browser and prompts you for the sign-in credentials for your Azure account:

```
az login
```

If you have multiple Azure subscriptions, check the subscriptions for the account:

```
az account list
```

Specify the subscription that you want to use:

```
az account set --subscription <your subscription ID>
```

Identify your ExpressRoute circuits for configuration

You can enable ExpressRoute Global Reach between any two ExpressRoute circuits, as long as they're located in supported countries/regions and were created at different peering locations. If your subscription owns both circuits, you can choose either circuit to run the configuration as explained later in this article. If the two circuits are in different Azure subscriptions, you must have authorization from one Azure subscription and must pass in its authorization key when you run the configuration command in the other Azure subscription.

Enable connectivity between your on-premises networks

When running the command to enable connectivity, note the following requirements for parameter values:

- *peer-circuit* should be the full resource ID. For example:

```
/subscriptions/{your_subscription_id}/resourceGroups/{your_resource_group}/providers/Microsoft.Net  
work/expressRouteCircuits/{your_circuit_name}
```

- *address-prefix* must be a "/29" IPv4 subnet (for example, "10.0.0.0/29"). We use IP addresses in this subnet to establish connectivity between the two ExpressRoute circuits. You must not use addresses in this subnet in your Azure virtual networks or in your on-premises networks.

Run the following CLI command to connect two ExpressRoute circuits:

```
az network express-route peering connection create -g <ResourceGroupName> --circuit-name <Circuit1Name> --peering-name AzurePrivatePeering -n <ConnectionName> --peer-circuit <Circuit2ResourceID> --address-prefix <__.__.__.__/29>
```

The CLI output looks like this:

```
{
  "addressPrefix": "<__.__.__.__/29>",
  "authorizationKey": null,
  "circuitConnectionStatus": "Connected",
  "etag": "W/\"48d682f9-c232-4151-a09f-fab7cb56369a\"",
  "expressRouteCircuitPeering": {
    "id": "/subscriptions/<SubscriptionID>/resourceGroups/<ResourceGroupName>/providers/Microsoft.Network/expressRouteCircuits/<Circuit1Name>/peerings/AzurePrivatePeering",
    "resourceGroup": "<ResourceGroupName>"
  },
  "id": "/subscriptions/<SubscriptionID>/resourceGroups/<ResourceGroupName>/providers/Microsoft.Network/expressRouteCircuits/<Circuit1Name>/peerings/AzurePrivatePeering/connections/<ConnectionName>",
  "name": "<ConnectionName>",
  "peerExpressRouteCircuitPeering": {
    "id": "/subscriptions/<SubscriptionID>/resourceGroups/<Circuit2ResourceGroupName>/providers/Microsoft.Network/expressRouteCircuits/<Circuit2Name>/peerings/AzurePrivatePeering",
    "resourceGroup": "<Circuit2ResourceGroupName>"
  },
  "provisioningState": "Succeeded",
  "resourceGroup": "<ResourceGroupName>",
  "type": "Microsoft.Network/expressRouteCircuits/peerings/connections"
}
```

When this operation is complete, you'll have connectivity between your on-premises networks on both sides through your two ExpressRoute circuits.

Enable connectivity between ExpressRoute circuits in different Azure subscriptions

If the two circuits aren't in the same Azure subscription, you need authorization. In the following configuration, you generate authorization in circuit 2's subscription and pass the authorization key to circuit 1.

1. Generate an authorization key:

```
az network express-route auth create --circuit-name <Circuit2Name> -g <Circuit2ResourceGroupName> -n <AuthorizationName>
```

The CLI output looks like this:

```
{  
    "authorizationKey": "<authorizationKey>",  
    "authorizationUseStatus": "Available",  
    "etag": "W/\"cf15a2f-43a1-4361-9403-6a0be00746ed\"",  
    "id":  
        "/subscriptions/<SubscriptionID>/resourceGroups/<Circuit2ResourceGroupName>/providers/Microsoft.Network/  
expressRouteCircuits/<Circuit2Name>/authorizations/<AuthorizationName>",  
    "name": "<AuthorizationName>",  
    "provisioningState": "Succeeded",  
    "resourceGroup": "<Circuit2ResourceGroupName>",  
    "type": "Microsoft.Network/expressRouteCircuits/authorizations"  
}
```

2. Make a note of both the resource ID and the authorization key for circuit 2.
3. Run the following command against circuit 1, passing in circuit 2's resource ID and authorization key:

```
az network express-route peering connection create -g <ResourceGroupName> --circuit-name <Circuit1Name>  
--peering-name AzurePrivatePeering -n <ConnectionName> --peer-circuit <Circuit2ResourceID> --address-  
prefix <_.___.___.___.29> --authorization-key <authorizationKey>
```

When this operation is complete, you'll have connectivity between your on-premises networks on both sides through your two ExpressRoute circuits.

Get and verify the configuration

Use the following command to verify the configuration on the circuit where the configuration was made (circuit 1 in the preceding example):

```
az network express-route show -n <CircuitName> -g <ResourceGroupName>
```

In the CLI output, you'll see *CircuitConnectionStatus*. It tells you whether the connectivity between the two circuits is established ("Connected") or not established ("Disconnected").

Disable connectivity between your on-premises networks

To disable connectivity, run the following command against the circuit where the configuration was made (circuit 1 in the earlier example).

```
az network express-route peering connection delete -g <ResourceGroupName> --circuit-name <Circuit1Name> --  
peering-name AzurePrivatePeering -n <ConnectionName>
```

Use the `show` command to verify the status.

When this operation is complete, you'll no longer have connectivity between your on-premises networks through your ExpressRoute circuits.

Next steps

- [Learn more about ExpressRoute Global Reach](#)
- [Verify ExpressRoute connectivity](#)
- [Link an ExpressRoute circuit to a virtual network](#)

Configure a virtual network gateway for ExpressRoute using the Azure portal

11/13/2019 • 3 minutes to read • [Edit Online](#)

This article walks you through the steps to add a virtual network gateway for a pre-existing VNet. This article walks 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 that will be used in an ExpressRoute configuration. For more information about virtual network gateways and gateway configuration settings for ExpressRoute, see [About virtual network gateways for ExpressRoute](#).

Before beginning

The steps for this task use a VNet based on the values in the following configuration reference list. We use this list in our example steps. You can copy the list to use as a reference, replacing the values with your own.

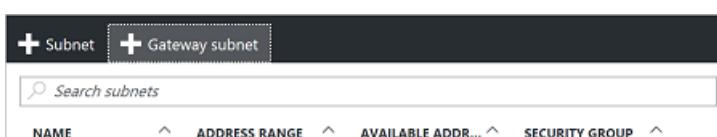
Configuration reference list

- Virtual Network Name = "TestVNet"
- Virtual Network address space = 192.168.0.0/16
- Subnet Name = "FrontEnd"
 - Subnet address space = "192.168.1.0/24"
- Resource Group = "TestRG"
- Location = "East US"
- Gateway Subnet name: "GatewaySubnet" You must always name a gateway subnet *GatewaySubnet*.
 - Gateway Subnet address space = "192.168.200.0/26"
- Gateway Name = "ERGW"
- Gateway Public IP Name = "MyERGVIP"
- Gateway type = "ExpressRoute" This type is required for an ExpressRoute configuration.

You can view a [Video](#) of these steps before beginning your configuration.

Create the gateway subnet

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



4. The **Name** for your subnet is automatically filled in with the value 'GatewaySubnet'. This value is required in order for Azure to recognize the subnet as the gateway subnet. Adjust the auto-filled **Address range** values to match your configuration requirements. We recommend creating a gateway subnet with a /27 or larger (/26, /25, etc.). Then, click **OK** to save the values and create the gateway subnet.



Create the virtual network gateway

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

The screenshot shows the 'Create virtual network gateway' blade with the following configuration:

- Name:** ERGW
- Gateway type:** ExpressRoute (selected)
- SKU:** Standard
- Virtual network:** TestVNet
- Public IP address:** (new) MyERGWVIP
- Subscription:** Windows Azure Internal Consumption
- Resource group:** TestRG
- Location:** East US

At the bottom, there is a 'Pin to dashboard' checkbox, a 'Create' button, and a note: "Provisioning a virtual network gateway may take up to 45 minutes."

3. **Name:** Name your gateway. This is not the same as naming a gateway subnet. It's the name of the gateway object you are creating.
4. **Gateway type:** Select **ExpressRoute**.

5. **SKU:** Select the gateway SKU from the dropdown.
6. **Location:** Adjust the **Location** field to point to the location where your virtual network is located. If the location is not pointing to the region where your virtual network resides, the virtual network doesn't appear in the 'Choose a virtual network' dropdown.
7. Choose the virtual network to which you want to add this gateway. Click **Virtual network** to open the **Choose a virtual network** blade. Select the VNet. If you don't see your VNet, make sure the **Location** field is pointing to the region in which your virtual network is located.
8. Choose a public IP address. Click **Public IP address** to open the **Choose public IP address** blade. Click **+Create New** to open the **Create public IP address blade**. Input a name for your public IP address. This blade creates a public IP address object to which a public IP address will be dynamically assigned. Click **OK** to save your changes to this blade.
9. **Subscription:** Verify that the correct subscription is selected.
10. **Resource group:** This setting is determined by the Virtual Network that you select.
11. Don't adjust the **Location** after you've specified the previous settings.
12. Verify the settings. If you want your gateway to appear on the dashboard, you can select **Pin to dashboard** at the bottom of the blade.
13. Click **Create** to begin creating the gateway. The settings are validated and the gateway deploys. Creating virtual network gateway can take up to 45 minutes to complete.

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 PowerShell

11/13/2019 • 4 minutes to read • [Edit Online](#)

This article helps you add, resize, and remove a virtual network (VNet) gateway for a pre-existing VNet. The steps for this configuration apply to VNets that were created using the Resource Manager deployment model for an ExpressRoute configuration. For more information, see [About virtual network gateways for ExpressRoute](#).

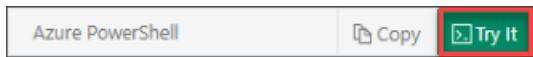
Before beginning

Working with PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Configuration reference list

The steps for this task use a VNet based on the values in the following configuration reference list. 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.

- Virtual Network Name = "TestVNet"
- Virtual Network address space = 192.168.0.0/16
- Resource Group = "TestRG"
- Subnet1 Name = "FrontEnd"
- Subnet1 address space = "192.168.1.0/24"
- 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.

If you are using the Azure Cloud Shell, you sign in to your Azure account automatically after clicking 'Try it'. To sign in locally, open your PowerShell console with elevated privileges and run the cmdlet to connect.

```
Connect-AzAccount
```

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

```
Get-AzSubscription
```

Specify the subscription that you want to use.

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

2. Declare your variables for this exercise. Be sure to edit the sample 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-AzVirtualNetwork -Name $VNetName -ResourceGroupName $RG
```

4. Add a gateway subnet to your Virtual Network. The gateway subnet must be named "GatewaySubnet". You should create a gateway subnet that is /27 or larger (/26, /25, etc.).

```
Add-AzVirtualNetworkSubnetConfig -Name GatewaySubnet -VirtualNetwork $vnet -AddressPrefix
192.168.200.0/26
```

5. Set the configuration.

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

6. Store the gateway subnet as a variable.

```
$subnet = Get-AzVirtualNetworkSubnetConfig -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-AzPublicIpAddress -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-AzVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress $pip
```

9. Create the gateway. In this step, the **-GatewayType** is especially important. You must use the value **ExpressRoute**. After running these cmdlets, the gateway can take 45 minutes or more to create.

```
New-AzVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG -Location $Location -IpConfigurations $ipconf -GatewayType Expressroute -GatewaySku Standard
```

Verify the gateway was created

Use the following commands to verify that the gateway has been created:

```
Get-AzVirtualNetworkGateway -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-AzVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG  
Resize-AzVirtualNetworkGateway -VirtualNetworkGateway $gw -GatewaySku HighPerformance
```

Remove a gateway

Use the following command to remove a gateway:

```
Remove-AzVirtualNetworkGateway -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 ExpressRoute and Site-to-Site coexisting connections using PowerShell

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

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

Configuring Site-to-Site VPN and ExpressRoute coexisting connections has several advantages:

- You can configure a Site-to-Site VPN as a secure failover path for ExpressRoute.
- Alternatively, you can use Site-to-Site VPNs to connect to sites that are not connected through ExpressRoute.

The steps to configure both scenarios are covered in this article. This article applies to the Resource Manager deployment model and uses PowerShell. You can also configure these scenarios using the Azure portal, although documentation is not yet available. You can configure either gateway first. Typically, you will incur no downtime when adding a new gateway or gateway connection.

NOTE

If you want to create a Site-to-Site VPN over an ExpressRoute circuit, please see [this article](#).

Limits and limitations

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

Configuration designs

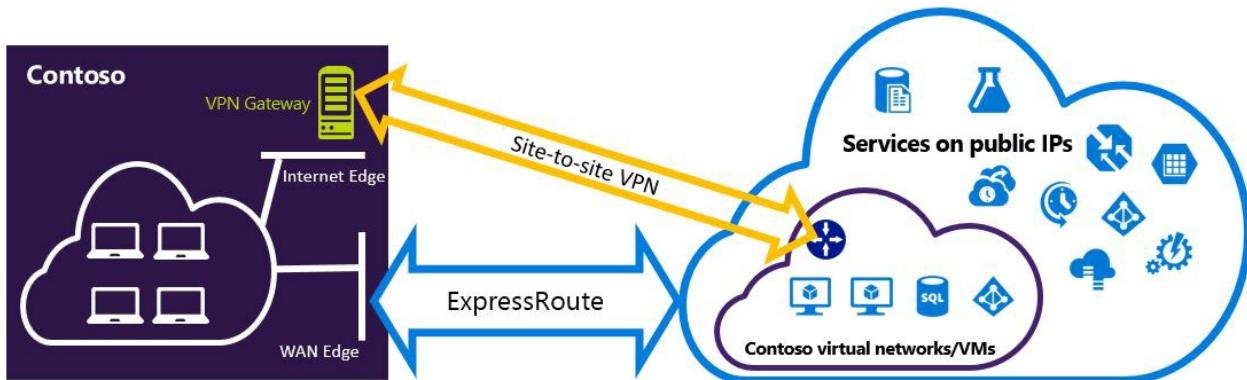
Configure a Site-to-Site VPN as a failover path for ExpressRoute

You can configure a Site-to-Site VPN connection as a backup for ExpressRoute. This connection applies only to virtual networks linked to the Azure private peering path. There is no VPN-based failover solution for services

accessible through Azure Microsoft peering. The ExpressRoute circuit is always the primary link. Data flows through the Site-to-Site VPN path only if the ExpressRoute circuit fails. To avoid asymmetrical routing, your local network configuration should also prefer the ExpressRoute circuit over the Site-to-Site VPN. You can prefer the ExpressRoute path by setting higher local preference for the routes received the ExpressRoute.

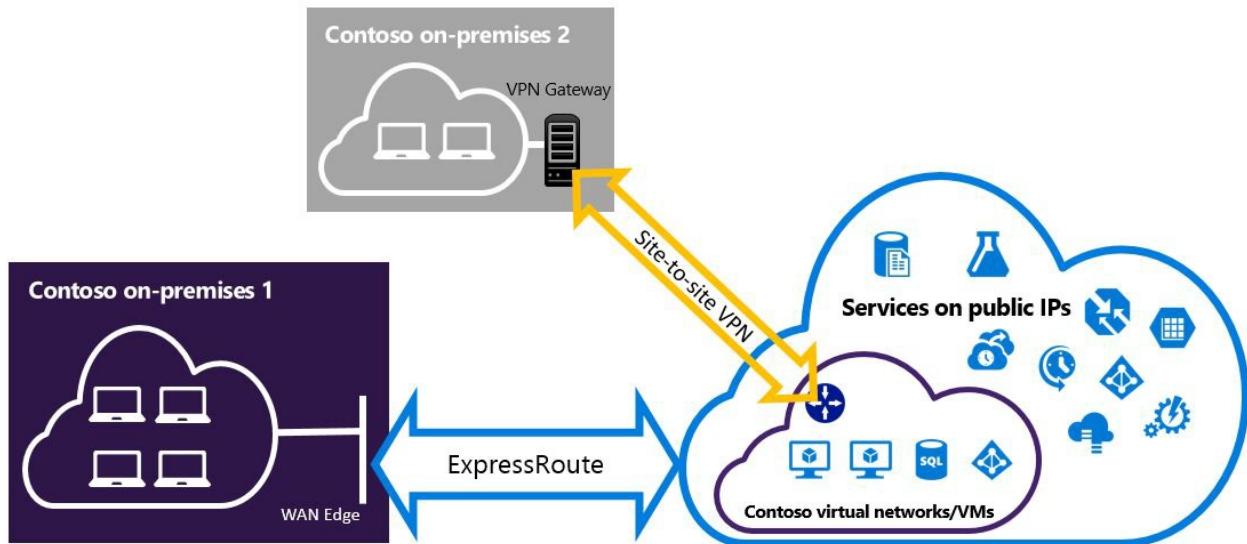
NOTE

While ExpressRoute circuit is preferred over Site-to-Site VPN when both routes are the same, Azure will use the longest prefix match to choose the route towards the packet's destination.



Configure a Site-to-Site VPN to connect to sites not connected through ExpressRoute

You can configure your network where some sites connect directly to Azure over Site-to-Site VPN, and some sites connect through ExpressRoute.



NOTE

You cannot configure a virtual network as a transit router.

Selecting the steps to use

There are two different sets of procedures to choose from. The configuration procedure that you select depends on whether you have an existing virtual network that you want to connect to, or you want to create a new virtual network.

- I don't have a VNet and need to create one.

If you don't already have a virtual network, this procedure walks you through creating a new virtual network using Resource Manager deployment model and creating new ExpressRoute and Site-to-Site VPN connections. To configure a virtual network, follow the steps in [To create a new virtual network and coexisting connections](#).

- I already have a Resource Manager deployment model VNet.

You may already have a virtual network in place with an existing Site-to-Site VPN connection or ExpressRoute connection. In this scenario if the gateway subnet mask is /28 or smaller (/28, /29, etc.), you have to delete the existing gateway. The [To configure coexisting connections for an already existing VNet](#) section walks you through deleting the gateway, and then creating new ExpressRoute and Site-to-Site VPN connections.

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

Before you begin

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

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

There are a few ways to launch the Cloud Shell:

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

To create a new virtual network and coexisting connections

This procedure walks you through creating a VNet and Site-to-Site and ExpressRoute connections that will coexist. The cmdlets that you use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.

1. Sign in and select your subscription.

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

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

```
Connect-AzAccount
```

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

```
Get-AzSubscription
```

Specify the subscription that you want to use.

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

2. Set variables.

```
$location = "Central US"  
$resgrp = New-AzResourceGroup -Name "ErVpnCoex" -Location $location  
$VNetASN = 65515
```

3. Create a virtual network including Gateway Subnet. For more information about creating a virtual network, see [Create a virtual network](#). For more information about creating subnets, see [Create a subnet](#)

IMPORTANT

The Gateway Subnet must be /27 or a shorter prefix (such as /26 or /25).

Create a new VNet.

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

Add subnets.

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

Save the VNet configuration.

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

4. Next, create your Site-to-Site VPN gateway. For more information about the VPN gateway configuration, see [Configure a VNet with a Site-to-Site connection](#). The `GatewaySku` is only supported for `VpnGw1`, `VpnGw2`, `VpnGw3`, `Standard`, and `HighPerformance` VPN gateways. ExpressRoute-VPN Gateway coexist configurations are not supported on the Basic SKU. The `VpnType` must be `RouteBased`.

```

$gwSubnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzPublicIpAddress -Name "VPNGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -
Location $location -AllocationMethod Dynamic
$gwConfig = New-AzVirtualNetworkGatewayIpConfig -Name "VPNGatewayIpConfig" -SubnetId $gwSubnet.Id -
PublicIpAddressId $gwIP.Id
New-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName -Location
$location -IpConfigurations $gwConfig -GatewayType "Vpn" -VpnType "RouteBased" -GatewaySku "VpnGw1"

```

Azure VPN gateway supports BGP routing protocol. You can specify ASN (AS Number) for that Virtual Network by adding the `-Asn` switch in the following command. Not specifying that parameter will default to AS number 65515.

```

$azureVpn = New-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
-Location $location -IpConfigurations $gwConfig -GatewayType "Vpn" -VpnType "RouteBased" -GatewaySku
"VpnGw1" -Asn $VNetASN

```

You can find the BGP peering IP and the AS number that Azure uses for the VPN gateway in `$azureVpn.BgpSettings.BgpPeeringAddress` and `$azureVpn.BgpSettings.Asn`. For more information, see [Configure BGP for Azure VPN gateway](#).

5. Create a local site VPN gateway entity. This command doesn't configure your on-premises VPN gateway. Rather, it allows you to provide the local gateway settings, such as the public IP and the on-premises address space, so that the Azure VPN gateway can connect to it.

If your local VPN device only supports static routing, you can configure the static routes in the following way:

```

$MyLocalNetworkAddress = @("10.100.0.0/16", "10.101.0.0/16", "10.102.0.0/16")
$localVpn = New-AzLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -GatewayIpAddress *<Public IP>* -AddressPrefix
$MyLocalNetworkAddress

```

If your local VPN device supports the BGP and you want to enable dynamic routing, you need to know the BGP peering IP and the AS number that your local VPN device uses.

```

$localVPNPublicIP = "<Public IP>"
$localBGPPeeringIP = "<Private IP for the BGP session>"
$localBGPASN = "<ASN>"
$localAddressPrefix = $localBGPPeeringIP + "/32"
$localVpn = New-AzLocalNetworkGateway -Name "LocalVPNGateway" -ResourceGroupName
$resgrp.ResourceGroupName -Location $location -GatewayIpAddress $localVPNPublicIP -AddressPrefix
$localAddressPrefix -BgpPeeringAddress $localBGPPeeringIP -Asn $localBGPASN

```

6. Configure your local VPN device to connect to the new Azure VPN gateway. For more information about VPN device configuration, see [VPN Device Configuration](#).
7. Link the Site-to-Site VPN gateway on Azure to the local gateway.

```

$azureVpn = Get-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
New-AzVirtualNetworkGatewayConnection -Name "VPNConnection" -ResourceGroupName $resgrp.ResourceGroupName
-Location $location -VirtualNetworkGateway1 $azureVpn -LocalNetworkGateway2 $localVpn -ConnectionType
IPsec -SharedKey <yourkey>

```

8. If you are connecting to an existing ExpressRoute circuit, skip steps 8 & 9 and, jump to step 10. Configure ExpressRoute circuits. For more information about configuring ExpressRoute circuit, see [create an ExpressRoute circuit](#).

9. Configure Azure private peering over the ExpressRoute circuit. For more information about configuring Azure private peering over the ExpressRoute circuit, see [configure peering](#)
10. Create an ExpressRoute gateway. For more information about the ExpressRoute gateway configuration, see [ExpressRoute gateway configuration](#). The GatewaySKU must be *Standard*, *HighPerformance*, or *UltraPerformance*.

```
$gwSubnet = Get-AzVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet
$gwIP = New-AzPublicIpAddress -Name "ERGatewayIP" -ResourceGroupName $resgrp.ResourceGroupName -Location $location -AllocationMethod Dynamic
$gwConfig = New-AzVirtualNetworkGatewayIpConfig -Name "ERGatewayIpConfig" -SubnetId $gwSubnet.Id - PublicIpAddressId $gwIP.Id
$gw = New-AzVirtualNetworkGateway -Name "ERGateway" -ResourceGroupName $resgrp.ResourceGroupName - Location $location -IpConfigurations $gwConfig -GatewayType "ExpressRoute" -GatewaySku Standard
```

11. Link the ExpressRoute gateway to the ExpressRoute circuit. After this step has been completed, the connection between your on-premises network and Azure, through ExpressRoute, is established. For more information about the link operation, see [Link VNets to ExpressRoute](#).

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

To configure coexisting connections for an already existing VNet

If you have a virtual network that has only one virtual network gateway (let's say, Site-to-Site VPN gateway) and you want to add another gateway of a different type (let's say, ExpressRoute gateway), check the gateway subnet size. If the gateway subnet is /27 or larger, you can skip the steps below and follow the steps in the previous section to add either a Site-to-Site VPN gateway or an ExpressRoute gateway. If the gateway subnet is /28 or /29, you have to first delete the virtual network gateway and increase the gateway subnet size. The steps in this section show you how to do that.

The cmdlets that you use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.

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

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

2. Delete Gateway Subnet.

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

3. Add a Gateway Subnet that is /27 or larger.

NOTE

If you don't have enough IP addresses left in your virtual network to increase the gateway subnet size, you need to add more IP address space.

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

Save the VNet configuration.

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

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

Set the variables.

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

Create the gateway.

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

Create the connection.

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

To add point-to-site configuration to the VPN gateway

You can follow the steps below to add Point-to-Site configuration to your VPN gateway in a coexistence setup. To upload the VPN root certificate, you must either install PowerShell locally to your computer, or use the Azure portal.

1. Add VPN Client address pool.

```
$azureVpn = Get-AzVirtualNetworkGateway -Name "VPNGateway" -ResourceGroupName $resgrp.ResourceGroupName
Set-AzVirtualNetworkGatewayVpnClientConfig -VirtualNetworkGateway $azureVpn -VpnClientAddressPool
"10.251.251.0/24"
```

2. Upload the VPN root certificate to Azure for your VPN gateway. In this example, it's assumed that the root certificate is stored in the local machine where the following PowerShell cmdlets are run and that you are running PowerShell locally. You can also upload the certificate using the Azure portal.

```
$p2sCertFullName = "RootErVpnCoexP2S.cer"
$p2sCertMatchName = "RootErVpnCoexP2S"
$p2sCertToUpload=get-childitem Cert:\CurrentUser\My | Where-Object {$_.Subject -match $p2sCertMatchName}
if ($p2sCertToUpload.count -eq 1){write-host "cert found"} else {write-host "cert not found" exit}
$p2sCertData = [System.Convert]::ToString($p2sCertToUpload.RawData)
Add-AzVpnClientRootCertificate -VpnClientRootCertificateName $p2sCertFullName -VirtualNetworkGatewayName
$azureVpn.Name -ResourceGroupName $resgrp.ResourceGroupName -PublicCertData $p2sCertData
```

For more information on Point-to-Site VPN, see [Configure a Point-to-Site connection](#).

Next steps

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

How to configure ExpressRoute Direct

2/5/2020 • 3 minutes to read • [Edit Online](#)

ExpressRoute Direct gives you the ability to connect directly into Microsoft's global network at peering locations strategically distributed across the world. For more information, see [About ExpressRoute Direct](#).

Create the resource

1. Sign in to Azure and select the subscription. The ExpressRoute Direct resource and ExpressRoute circuits must be in the same subscription.

```
Connect-AzAccount
```

```
Select-AzSubscription -Subscription "<SubscriptionID or SubscriptionName>"
```

2. Re-register your subscription to Microsoft.Network to access the expressrouteportslocation and expressrouteport APIs.

```
Register-AzResourceProvider -ProviderNameSpace "Microsoft.Network"
```

3. List all locations where ExpressRoute Direct is supported.

```
Get-AzExpressRoutePortsLocation
```

Example output

```

Name : Equinix-Ashburn-DC2
Id :
/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-Ashburn-
D
          C2
ProvisioningState : Succeeded
Address : 21715 Filigree Court, DC2, Building F, Ashburn, VA 20147
Contact : support@equinix.com
AvailableBandwidths : []

Name : Equinix-Dallas-DA3
Id :
/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-Dallas-
DA
          3
ProvisioningState : Succeeded
Address : 1950 N. Stemmons Freeway, Suite 1039A, DA3, Dallas, TX 75207
Contact : support@equinix.com
AvailableBandwidths : []

Name : Equinix-San-Jose-SV1
Id :
/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-San-
Jose-
          SV1
ProvisioningState : Succeeded
Address : 11 Great Oaks Blvd, SV1, San Jose, CA 95119
Contact : support@equinix.com
AvailableBandwidths : []

```

4. Determine if a location listed above has available bandwidth

```
Get-AzExpressRoutePortsLocation -LocationName "Equinix-San-Jose-SV1"
```

Example output

```

Name : Equinix-San-Jose-SV1
Id :
/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-San-
Jose-
          SV1
ProvisioningState : Succeeded
Address : 11 Great Oaks Blvd, SV1, San Jose, CA 95119
Contact : support@equinix.com
AvailableBandwidths : [
          {
            "OfferName": "100 Gbps",
            "ValueInGbps": 100
          }
        ]

```

5. Create an ExpressRoute Direct resource based on the location chosen above

ExpressRoute Direct supports both QinQ and Dot1Q encapsulation. If QinQ is selected, each ExpressRoute circuit will be dynamically assigned an S-Tag and will be unique throughout the ExpressRoute Direct resource. Each C-Tag on the circuit must be unique on the circuit, but not across the ExpressRoute Direct.

If Dot1Q encapsulation is selected, you must manage uniqueness of the C-Tag (VLAN) across the entire ExpressRoute Direct resource.

IMPORTANT

ExpressRoute Direct can only be one encapsulation type. Encapsulation cannot be changed after ExpressRoute Direct creation.

```
$ERDirect = New-AzExpressRoutePort -Name $Name -ResourceGroupName $ResourceGroupName -PeeringLocation  
$PeeringLocationName -BandwidthInGbps 100.0 -Encapsulation QinQ | Dot1Q -Location $AzureRegion
```

NOTE

The Encapsulation attribute could also be set to Dot1Q.

Example output:

```

Name : Contoso-Direct
ResourceGroupName : Contoso-Direct-rg
Location : westcentralus
Id : /subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.Network/exp
ressRoutePorts/Contoso-Direct
Etag : W/"<etagnumber> "
ResourceGuid : <number>
ProvisioningState : Succeeded
PeeringLocation : Equinix-Seattle-SE2
BandwidthInGbps : 100
ProvisionedBandwidthInGbps : 0
Encapsulation : QinQ
Mtu : 1500
EtherType : 0x8100
AllocationDate : Saturday, September 1, 2018
Links :
[ {
    "Name": "link1",
    "Etag": "W/"<etagnumber>\\"",
    "Id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.
Network/expressRoutePorts/Contoso-Direct/links/link1",
    "RouterName": "tst-09xgmr-cis-1",
    "InterfaceName": "HundredGigE2/2/2",
    "PatchPanelId": "PPID",
    "RackId": "RackID",
    "ConnectorType": "SC",
    "AdminState": "Disabled",
    "ProvisioningState": "Succeeded"
},
{
    "Name": "link2",
    "Etag": "W/"<etagnumber>\\"",
    "Id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.
Network/expressRoutePorts/Contoso-Direct/links/link2",
    "RouterName": "tst-09xgmr-cis-2",
    "InterfaceName": "HundredGigE2/2/2",
    "PatchPanelId": "PPID",
    "RackId": "RackID",
    "ConnectorType": "SC",
    "AdminState": "Disabled",
    "ProvisioningState": "Succeeded"
}
]
Circuits : []

```

Change Admin State of links

This process should be used to conduct a Layer 1 test, ensuring that each cross-connection is properly patched into each router for primary and secondary.

1. Get ExpressRoute Direct details.

```
$ERDirect = Get-AzExpressRoutePort -Name $Name -ResourceGroupName $ResourceGroupName
```

2. Set Link to Enabled. Repeat this step to set each link to enabled.

Links[0] is the primary port and Links[1] is the secondary port.

```

$ERDirect.Links[0].AdminState = "Enabled"
Set-AzExpressRoutePort -ExpressRoutePort $ERDirect
$ERDirect = Get-AzExpressRoutePort -Name $Name -ResourceGroupName $ResourceGroupName
$ERDirect.Links[1].AdminState = "Enabled"
Set-AzExpressRoutePort -ExpressRoutePort $ERDirect

```

Example output:

```

Name : Contoso-Direct
ResourceGroupName : Contoso-Direct-rg
Location : westcentralus
Id : /subscriptions/<number>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.Network/exp
ressRoutePorts/Contoso-Direct
Etag : W/"<etagnumber> "
ResourceGuid : <number>
ProvisioningState : Succeeded
PeeringLocation : Equinix-Seattle-SE2
BandwidthInGbps : 100
ProvisionedBandwidthInGbps : 0
Encapsulation : QinQ
Mtu : 1500
EtherType : 0x8100
AllocationDate : Saturday, September 1, 2018
Links : [
{
    "Name": "link1",
    "Etag": "W/"<etagnumber>\\"",
    "Id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.
Network/expressRoutePorts/Contoso-Direct/links/link1",
    "RouterName": "tst-09gmr-cis-1",
    "InterfaceName": "HundredGigE2/2/2",
    "PatchPanelId": "PPID",
    "RackId": "RackID",
    "ConnectorType": "SC",
    "AdminState": "Enabled",
    "ProvisioningState": "Succeeded"
},
{
    "Name": "link2",
    "Etag": "W/"<etagnumber>\\"",
    "Id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.
Network/expressRoutePorts/Contoso-Direct/links/link2",
    "RouterName": "tst-09gmr-cis-2",
    "InterfaceName": "HundredGigE2/2/2",
    "PatchPanelId": "PPID",
    "RackId": "RackID",
    "ConnectorType": "SC",
    "AdminState": "Enabled",
    "ProvisioningState": "Succeeded"
}
]
Circuits : []

```

Use the same procedure with `AdminState = "Disabled"` to turn down the ports.

Create a circuit

By default, you can create 10 circuits in the subscription where the ExpressRoute Direct resource is. This can be increased by support. You are responsible for tracking both Provisioned and Utilized Bandwidth. Provisioned bandwidth is the sum of bandwidth of all circuits on the ExpressRoute Direct resource and utilized bandwidth is

the physical usage of the underlying physical interfaces.

There are additional circuit bandwidths that can be utilized on ExpressRoute Direct only to support the scenarios outlined above. These are: 40Gbps and 100Gbps.

SkuTier can be Local, Standard or Premium.

SkuFamily must be MeteredData only as unlimited is not supported on ExpressRoute Direct.

Create a circuit on the ExpressRoute Direct resource.

```
New-AzExpressRouteCircuit -Name $Name -ResourceGroupName $ResourceGroupName -ExpressRoutePort $ERDirect -  
BandwidthInGbps 100.0 -Location $AzureRegion -SkuTier Premium -SkuFamily MeteredData
```

Other bandwidths include: 5.0, 10.0, and 40.0

Example output:

```
Name : ExpressRoute-Direct-ckt  
ResourceGroupName : Contoso-Direct-rg  
Location : westcentralus  
Id : /subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-  
rg/providers/Microsoft.Netwo  
rk/expressRouteCircuits/ExpressRoute-Direct-ckt  
Etag : W/"<etagnumber>"  
ProvisioningState : Succeeded  
Sku : {  
    "Name": "Premium_MeteredData",  
    "Tier": "Premium",  
    "Family": "MeteredData"  
}  
CircuitProvisioningState : Enabled  
ServiceProviderProvisioningState : Provisioned  
ServiceProviderNotes :  
ServiceProviderProperties : null  
ExpressRoutePort : {  
    "Id": "/subscriptions/<subscriptionID>n/resourceGroups/Contoso-Direct-  
rg/providers/Micros  
oft.Network/expressRoutePorts/Contoso-Direct"  
}  
BandwidthInGbps : 10  
Stag : 2  
ServiceKey : <number>  
Peerings : []  
Authorizations : []  
AllowClassicOperations : False  
GatewayManagerEtag
```

Next steps

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

Configure ExpressRoute Direct by using the Azure CLI

2/5/2020 • 4 minutes to read • [Edit Online](#)

You can use Azure ExpressRoute Direct to connect directly to the Microsoft global network at peering locations strategically distributed across the world. For more information, see [About ExpressRoute Direct Connect](#).

Create the resource

1. Sign in to Azure and select the subscription that contains ExpressRoute. The ExpressRoute Direct resource and your ExpressRoute circuits must be in the same subscription. In the Azure CLI, run the following commands:

```
az login
```

Check the subscriptions for the account:

```
az account list
```

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

```
az account set --subscription "<subscription ID>"
```

2. Re-register your subscription to Microsoft.Network to access the expressrouteportslocation and expressrouteport APIs

```
az provider register --namespace Microsoft.Network
```

3. List all locations where ExpressRoute Direct is supported:

```
az network express-route port location list
```

Example output

```
[
{
  "address": "21715 Filigree Court, DC2, Building F, Ashburn, VA 20147",
  "availableBandwidths": [],
  "contact": "support@equinix.com",
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-Ashburn-DC2",
  "location": null,
  "name": "Equinix-Ashburn-DC2",
  "provisioningState": "Succeeded",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePortsLocations"
},
{
  "address": "1950 N. Stemmons Freeway, Suite 1039A, DA3, Dallas, TX 75207",
  "availableBandwidths": [],
  "contact": "support@equinix.com",
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-Dallas-DA3",
  "location": null,
  "name": "Equinix-Dallas-DA3",
  "provisioningState": "Succeeded",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePortsLocations"
},
{
  "address": "111 8th Avenue, New York, NY 10011",
  "availableBandwidths": [],
  "contact": "support@equinix.com",
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-New-York-NY5",
  "location": null,
  "name": "Equinix-New-York-NY5",
  "provisioningState": "Succeeded",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePortsLocations"
},
{
  "address": "11 Great Oaks Blvd, SV1, San Jose, CA 95119",
  "availableBandwidths": [],
  "contact": "support@equinix.com",
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-San-Jose-SV1",
  "location": null,
  "name": "Equinix-San-Jose-SV1",
  "provisioningState": "Succeeded",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePortsLocations"
},
{
  "address": "2001 Sixth Ave., Suite 350, SE2, Seattle, WA 98121",
  "availableBandwidths": [],
  "contact": "support@equinix.com",
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-Seattle-SE2",
  "location": null,
  "name": "Equinix-Seattle-SE2",
  "provisioningState": "Succeeded",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePortsLocations"
}
]
```

4. Determine whether one of the locations listed in the preceding step has available bandwidth:

```
az network express-route port location show -l "Equinix-Ashburn-DC2"
```

Example output

```
{  
  "address": "21715 Filigree Court, DC2, Building F, Ashburn, VA 20147",  
  "availableBandwidths": [  
    {  
      "offerName": "100 Gbps",  
      "valueInGbps": 100  
    }  
  ],  
  "contact": "support@equinix.com",  
  "id": "/subscriptions/<subscriptionID>/providers/Microsoft.Network/expressRoutePortsLocations/Equinix-  
Ashburn-DC2",  
  "location": null,  
  "name": "Equinix-Ashburn-DC2",  
  "provisioningState": "Succeeded",  
  "tags": null,  
  "type": "Microsoft.Network/expressRoutePortsLocations"  
}
```

5. Create an ExpressRoute Direct resource that's based on the location you chose in the preceding steps.

ExpressRoute Direct supports both QinQ and Dot1Q encapsulation. If you select QinQ, each ExpressRoute circuit is dynamically assigned an S-Tag and is unique throughout the ExpressRoute Direct resource. Each C-Tag on the circuit must be unique on the circuit but not across the ExpressRoute Direct resource.

If you select Dot1Q encapsulation, you must manage uniqueness of the C-Tag (VLAN) across the entire ExpressRoute Direct resource.

IMPORTANT

ExpressRoute Direct can be only one encapsulation type. You can't change the encapsulation type after you create the ExpressRoute Direct resource.

```
az network express-route port create -n $name -g $RGName --bandwidth 100 gbps --encapsulation QinQ |  
Dot1Q --peering-location $PeeringLocationName -l $AzureRegion
```

NOTE

You also can set the **Encapsulation** attribute to **Dot1Q**.

Example output

```
{
  "allocationDate": "Wednesday, October 17, 2018",
  "bandwidthInGbps": 100,
  "circuits": null,
  "encapsulation": "Dot1Q",
  "etag": "W/\"<etagnumber>\\"",
  "etherType": "0x8100",
  "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct",
  "links": [
    {
      "adminState": "Disabled",
      "connectorType": "LC",
      "etag": "W/\"<etagnumber>\\"",
      "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct/links/link1",
      "interfaceName": "HundredGigE2/2/2",
      "name": "link1",
      "patchPanelId": "PPID",
      "provisioningState": "Succeeded",
      "rackId": "RackID",
      "resourceGroup": "Contoso-Direct-rg",
      "routerName": "tst-09xgmr-cis-1",
      "type": "Microsoft.Network/expressRoutePorts/links"
    },
    {
      "adminState": "Disabled",
      "connectorType": "LC",
      "etag": "W/\"<etagnumber>\\"",
      "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct/links/link2",
      "interfaceName": "HundredGigE2/2/2",
      "name": "link2",
      "patchPanelId": "PPID",
      "provisioningState": "Succeeded",
      "rackId": "RackID",
      "resourceGroup": "Contoso-Direct-rg",
      "routerName": "tst-09xgmr-cis-2",
      "type": "Microsoft.Network/expressRoutePorts/links"
    }
  ],
  "location": "westus",
  "mtu": "1500",
  "name": "Contoso-Direct",
  "peeringLocation": "Equinix-Ashburn-DC2",
  "provisionedBandwidthInGbps": 0.0,
  "provisioningState": "Succeeded",
  "resourceGroup": "Contoso-Direct-rg",
  "resourceGuid": "02ee21fe-4223-4942-a6bc-8d81daabc94f",
  "tags": null,
  "type": "Microsoft.Network/expressRoutePorts"
}
```

Change AdminState for links

Use this process to conduct a layer 1 test. Ensure that each cross-connection is properly patched into each router in the primary and secondary ports.

1. Set links to **Enabled**. Repeat this step to set each link to **Enabled**.

Links[0] is the primary port and Links[1] is the secondary port.

```
az network express-route port update -n Contoso-Direct -g Contoso-Direct-rg --set  
links[0].adminState="Enabled"
```

```
az network express-route port update -n Contoso-Direct -g Contoso-Direct-rg --set  
links[1].adminState="Enabled"
```

Example output

```
{  
    "allocationDate": "Wednesday, October 17, 2018",  
    "bandwidthInGbps": 100,  
    "circuits": null,  
    "encapsulation": "Dot1Q",  
    "etag": "W/\"<etagnumber>\\"",  
    "etherType": "0x8100",  
    "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-  
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct",  
    "links": [  
        {  
            "adminState": "Enabled",  
            "connectorType": "LC",  
            "etag": "W/\"<etagnumber>\\"",  
            "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-  
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct/links/link1",  
            "interfaceName": "HundredGigE2/2/2",  
            "name": "link1",  
            "patchPanelId": "PPID",  
            "provisioningState": "Succeeded",  
            "rackId": "RackID",  
            "resourceGroup": "Contoso-Direct-rg",  
            "routerName": "tst-09xgmr-cis-1",  
            "type": "Microsoft.Network/expressRoutePorts/links"  
        },  
        {  
            "adminState": "Enabled",  
            "connectorType": "LC",  
            "etag": "W/\"<etagnumber>\\"",  
            "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-  
rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct/links/link2",  
            "interfaceName": "HundredGigE2/2/2",  
            "name": "link2",  
            "patchPanelId": "PPID",  
            "provisioningState": "Succeeded",  
            "rackId": "RackID",  
            "resourceGroup": "Contoso-Direct-rg",  
            "routerName": "tst-09xgmr-cis-2",  
            "type": "Microsoft.Network/expressRoutePorts/links"  
        }  
    "location": "westus",  
    "mtu": "1500",  
    "name": "Contoso-Direct",  
    "peeringLocation": "Equinix-Ashburn-DC2",  
    "provisionedBandwidthInGbps": 0.0,  
    "provisioningState": "Succeeded",  
    "resourceGroup": "Contoso-Direct-rg",  
    "resourceGuid": "<resourceGUID>",  
    "tags": null,  
    "type": "Microsoft.Network/expressRoutePorts"  
}
```

Use the same procedure to down the ports by using `AdminState = "Disabled"`.

Create a circuit

By default, you can create 10 circuits in the subscription that contains the ExpressRoute Direct resource. Microsoft Support can increase the default limit. You're responsible for tracking provisioned and utilized bandwidth.

Provisioned bandwidth is the sum of the bandwidth of all the circuits on the ExpressRoute Direct resource. Utilized bandwidth is the physical usage of the underlying physical interfaces.

You can use additional circuit bandwidths on ExpressRoute Direct only to support the scenarios outlined here. The bandwidths are 40 Gbps and 100 Gbps.

SkuTier can be Local, Standard or Premium.

SkuFamily must be MeteredData only as unlimited is not supported on ExpressRoute Direct. Create a circuit on the ExpressRoute Direct resource:

```
az network express-route create --express-route-port "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct" -n "Contoso-Direct-ckt" -g "Contoso-Direct-rg" --sku-family MeteredData --sku-tier Standard --bandwidth 100 Gbps
```

Other bandwidths include 5 Gbps, 10 Gbps, and 40 Gbps.

Example output

```
{
  "allowClassicOperations": false,
  "allowGlobalReach": false,
  "authorizations": [],
  "bandwidthInGbps": 100.0,
  "circuitProvisioningState": "Enabled",
  "etag": "W/\"<etagnumber>\\"",
  "expressRoutePort": {
    "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-rg/providers/Microsoft.Network/expressRoutePorts/Contoso-Direct",
    "resourceGroup": "Contoso-Direct-rg"
  },
  "gatewayManagerEtag": "",
  "id": "/subscriptions/<subscriptionID>/resourceGroups/Contoso-Direct-rg/providers/Microsoft.Network/expressRouteCircuits/ERDirect-ckt-cli",
  "location": "westus",
  "name": "ERDirect-ckt-cli",
  "peerings": [],
  "provisioningState": "Succeeded",
  "resourceGroup": "Contoso-Direct-rg",
  "serviceKey": "<serviceKey>",
  "serviceProviderNotes": null,
  "serviceProviderProperties": null,
  "serviceProviderProvisioningState": "Provisioned",
  "sku": {
    "family": "MeteredData",
    "name": "Standard_MeteredData",
    "tier": "Standard"
  },
  "stag": null,
  "tags": null,
  "type": "Microsoft.Network/expressRouteCircuits"
}
```

Next steps

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

Configure MACsec on ExpressRoute Direct ports

11/14/2019 • 5 minutes to read • [Edit Online](#)

This article helps you configure MACsec to secure the connections between your edge routers and Microsoft's edge routers using PowerShell.

Before you begin

Before you start configuration, confirm the following:

- You understand [ExpressRoute Direct provisioning workflows](#).
- You've created an [ExpressRoute Direct port resource](#).
- If you want to run PowerShell locally, verify that the latest version of Azure PowerShell is installed on your computer.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Sign in and select the right subscription

To start the configuration, sign in to your Azure account and select the subscription that you want to use.

If you are using the Azure Cloud Shell, you sign in to your Azure account automatically after clicking 'Try it'. To sign in locally, open your PowerShell console with elevated privileges and run the cmdlet to connect.

```
Connect-AzAccount
```

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

```
Get-AzSubscription
```

Specify the subscription that you want to use.

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

1. Create Azure Key Vault, MACsec secrets, and user identity

1. Create a Key Vault instance to store MACsec secrets in a new resource group.

```
New-AzResourceGroup -Name "your_resource_group" -Location "resource_location"  
$keyVault = New-AzKeyVault -Name "your_key_vault_name" -ResourceGroupName "your_resource_group" -  
Location "resource_location" -EnableSoftDelete
```

If you already have a key vault or a resource group, you can reuse them. However, it is critical that you enable the **soft-delete feature** on your existing key vault. If soft-delete is not enabled, you can use the following commands to enable it:

```
($resource = Get-AzResource -ResourceId (Get-AzKeyVault -VaultName  
"your_existing_keyvault").ResourceId).Properties | Add-Member -MemberType "NoteProperty" -Name  
"enableSoftDelete" -Value "true"  
Set-AzResource -resourceid $resource.ResourceId -Properties $resource.Properties
```

2. Create a user identity.

```
$identity = New-AzUserAssignedIdentity -Name "identity_name" -Location "resource_location" -  
ResourceGroupName "your_resource_group"
```

If New-AzUserAssignedIdentity is not recognized as a valid PowerShell cmdlet, install the following module (in Administrator mode) and rerun the above command.

```
Install-Module -Name Az.ManagedServiceIdentity
```

3. Create a connectivity association key (CAK) and a connectivity association key name (CKN) and store them in the key vault.

```
$CAK = ConvertTo-SecureString "your_key" -AsPlainText -Force  
$CKN = ConvertTo-SecureString "your_key_name" -AsPlainText -Force  
$MACsecCAKSecret = Set-AzKeyVaultSecret -VaultName "your_key_vault_name" -Name "CAK_name" -SecretValue  
$CAK  
$MACsecCKNSecret = Set-AzKeyVaultSecret -VaultName "your_key_vault_name" -Name "CKN_name" -SecretValue  
$CKN
```

4. Assign the GET permission to the user identity.

```
Set-AzKeyVaultAccessPolicy -VaultName "your_key_vault_name" -PermissionsToSecrets get -ObjectId  
$identity.PrincipalId
```

Now this identity can get the secrets, for example CAK and CKN, from the key vault.

5. Set this user identity to be used by ExpressRoute.

```
$erIdentity = New-AzExpressRoutePortIdentity -UserAssignedIdentityId $identity.Id
```

2. Configure MACsec on ExpressRoute Direct ports

To enable MACsec

Each ExpressRoute Direct instance has two physical ports. You can choose to enable MACsec on both ports at the same time or enable MACsec on one port at a time. Doing it one port at time (by switching traffic to an active port while servicing the other port) can help minimize the interruption if your ExpressRoute Direct is already in service.

1. Set MACsec secrets and cipher and associate the user identity with the port so that the ExpressRoute management code can access the MACsec secrets if needed.

```
$erDirect = Get-AzExpressRoutePort -ResourceGroupName "your_resource_group" -Name "your_direct_port_name"  
$erDirect.Links[0].MacSecConfig.CknSecretIdentifier = $MacSecCKNSecret.Id  
$erDirect.Links[0].MacSecConfig.CakSecretIdentifier = $MacSecCAKSecret.Id  
$erDirect.Links[0].MacSecConfig.Cipher = "GcmAes256"  
$erDirect.Links[1].MacSecConfig.CknSecretIdentifier = $MacSecCKNSecret.Id  
$erDirect.Links[1].MacSecConfig.CakSecretIdentifier = $MacSecCAKSecret.Id  
$erDirect.Links[1].MacSecConfig.Cipher = "GcmAes256"  
$erDirect.identity = $erIdentity  
Set-AzExpressRoutePort -ExpressRoutePort $erDirect
```

2. (Optional) If the ports are in Administrative Down state you can run the following commands to bring up the ports.

```
$erDirect = Get-AzExpressRoutePort -ResourceGroupName "your_resource_group" -Name "your_direct_port_name"  
$erDirect.Links[0].AdminState = "Enabled"  
$erDirect.Links[1].AdminState = "Enabled"  
Set-AzExpressRoutePort -ExpressRoutePort $erDirect
```

At this point, MACsec is enabled on the ExpressRoute Direct ports on Microsoft side. If you haven't configured it on your edge devices, you can proceed to configure them with the same MACsec secrets and cipher.

To disable MACsec

If MACsec is no longer desired on your ExpressRoute Direct instance, you can run the following commands to disable it.

```
$erDirect = Get-AzExpressRoutePort -ResourceGroupName "your_resource_group" -Name "your_direct_port_name"  
$erDirect.Links[0].MacSecConfig.CknSecretIdentifier = $null  
$erDirect.Links[0].MacSecConfig.CakSecretIdentifier = $null  
$erDirect.Links[1].MacSecConfig.CknSecretIdentifier = $null  
$erDirect.Links[1].MacSecConfig.CakSecretIdentifier = $null  
$erDirect.identity = $null  
Set-AzExpressRoutePort -ExpressRoutePort $erDirect
```

At this point, MACsec is disabled on the ExpressRoute Direct ports on the Microsoft side.

Test connectivity

After you configure MACsec (including MACsec key update) on your ExpressRoute Direct ports, [check](#) if the BGP sessions of the circuits are up and running. If you don't have any circuit on the ports yet, please create one first and set up Azure Private Peering or Microsoft Peering of the circuit. If MACsec is misconfigured, including MACsec key mismatch, between your network devices and Microsoft's network devices, you won't see ARP resolution at layer 2

and BGP establishment at layer 3. If everything is configured properly, you should see the BGP routes advertised correctly in both directions and your application data flow accordingly over ExpressRoute.

Next steps

1. [Create an ExpressRoute circuit on ExpressRoute Direct](#)
2. [Link an ExpressRoute circuit to an Azure virtual network](#)
3. [Verify ExpressRoute connectivity](#)

Configure route filters for Microsoft peering: Azure portal

11/13/2019 • 5 minutes to read • [Edit Online](#)

Route filters are a way to consume a subset of supported services through Microsoft peering. The steps in this article help you configure and manage route filters for ExpressRoute circuits.

Office 365 services such as Exchange Online, SharePoint Online, and Skype for Business, and Azure services such as storage and SQL DB are accessible through Microsoft peering. When Microsoft peering is configured in an ExpressRoute circuit, all prefixes related to these services are advertised through the BGP sessions that are established. A BGP community value is attached to every prefix to identify the service that is offered through the prefix. For a list of the BGP community values and the services they map to, see [BGP communities](#).

If you require connectivity to all services, a large number of prefixes are advertised through BGP. This significantly increases the size of the route tables maintained by routers within your network. If you plan to consume only a subset of services offered through Microsoft peering, you can reduce the size of your route tables in two ways. You can:

- Filter out unwanted prefixes by applying route filters on BGP communities. This is a standard networking practice and is used commonly within many networks.
- Define route filters and apply them to your ExpressRoute circuit. A route filter is a new resource that lets you select the list of services you plan to consume through Microsoft peering. ExpressRoute routers only send the list of prefixes that belong to the services identified in the route filter.

About route filters

When Microsoft peering is configured on your ExpressRoute circuit, the Microsoft edge routers establish a pair of BGP sessions with the edge routers (yours or your connectivity provider's). No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter.

A route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. It is essentially a list of all the BGP community values you want to allow. Once a route filter resource is defined and attached to an ExpressRoute circuit, all prefixes that map to the BGP community values are advertised to your network.

To be able to attach route filters with Office 365 services on them, you must have authorization to consume Office 365 services through ExpressRoute. If you are not authorized to consume Office 365 services through ExpressRoute, the operation to attach route filters fails. For more information about the authorization process, see [Azure ExpressRoute for Office 365](#).

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit.

Workflow

To be able to successfully connect to services through Microsoft peering, you must complete the following configuration steps:

- You must have an active ExpressRoute circuit that has Microsoft peering provisioned. You can use the

following instructions to accomplish these tasks:

- [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.
- [Create Microsoft peering](#) if you manage the BGP session directly. Or, have your connectivity provider provision Microsoft peering for your circuit.
- You must create and configure a route filter.
 - Identify the services you wish to consume through Microsoft peering
 - Identify the list of BGP community values associated with the services
 - Create a rule to allow the prefix list matching the BGP community values
- You must attach the route filter to the ExpressRoute circuit.

Before you begin

Before you begin configuration, make sure you meet the following criteria:

- Review the [prerequisites](#) 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 before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state.
- You must have an active Microsoft peering. Follow instructions at [Create and modifying peering configuration](#)

Step 1: Get a list of prefixes and BGP community values

1. Get a list of BGP community values

BGP community values associated with services accessible through Microsoft peering is available in the [ExpressRoute routing requirements](#) page.

2. Make a list of the values that you want to use

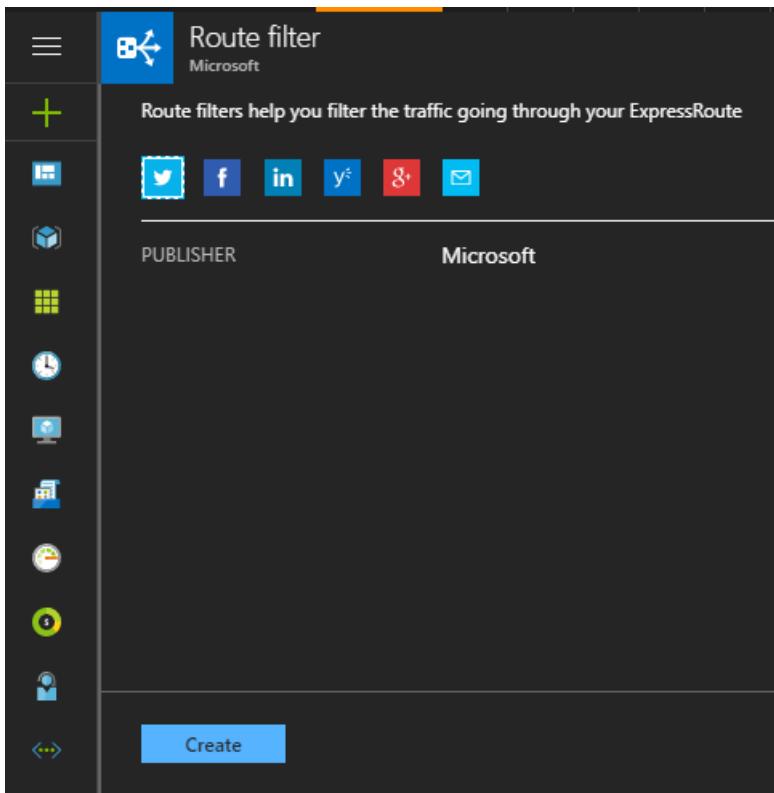
Make a list of [BGP community values](#) you want to use in the route filter.

Step 2: Create a route filter and a filter rule

A route filter can have only one rule, and the rule must be of type 'Allow'. This rule can have a list of BGP community values associated with it.

1. Create a route filter

You can create a route filter by selecting the option to create a new resource. Click **Create a resource** > **Networking** > **RouteFilter**, as shown in the following image:



You must place the route filter in a resource group.

Create route filter

Route filter must be created in the same location as the ExpressRoutes it will be associated with.

* Name: MyRouteFilter

* Subscription: ExpressRoute-Lab

Resource group:

Create new Use existing

Portal|Demo

* Location: East US

Pin to dashboard

Create [Automation options](#)

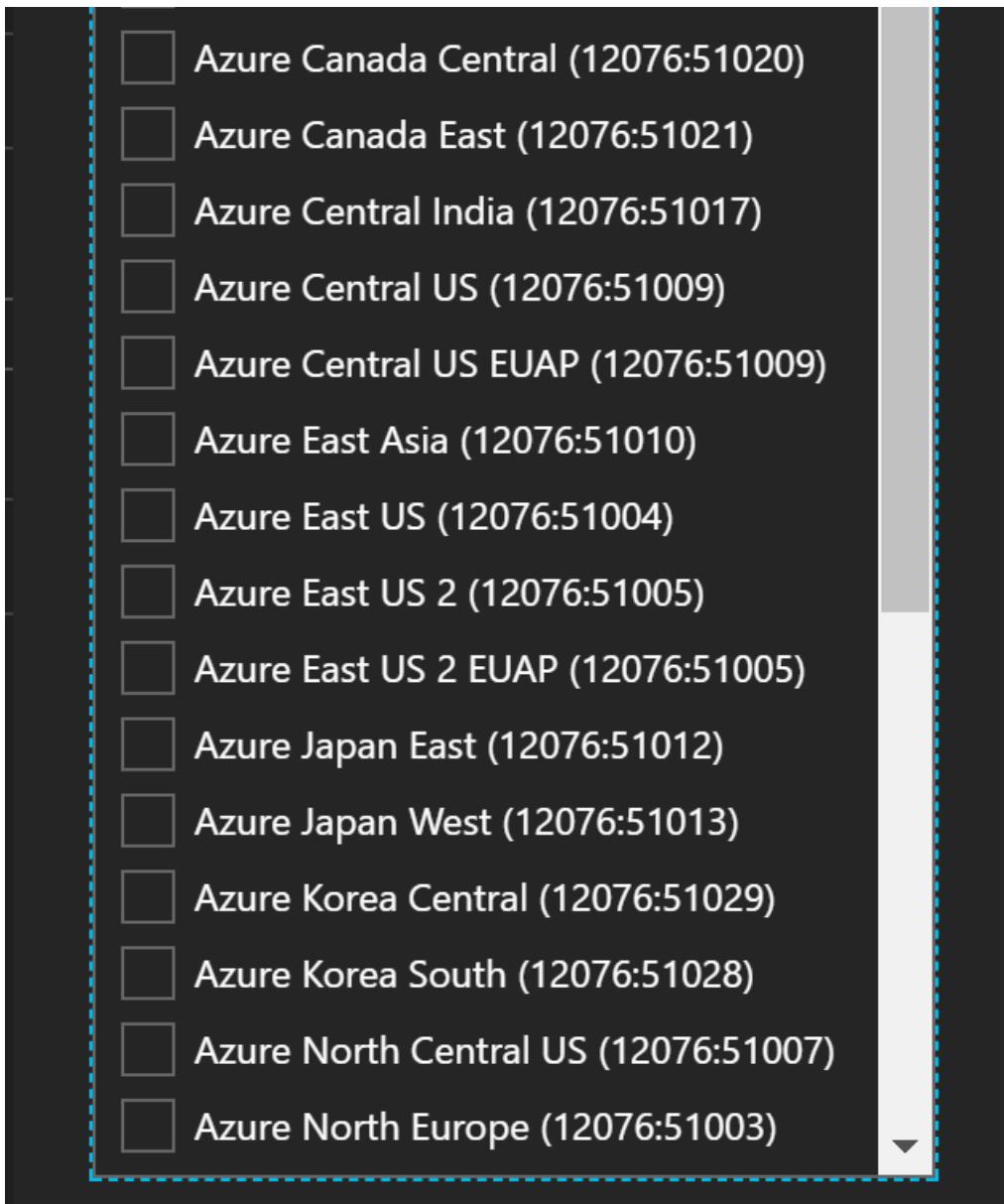
2. Create a filter rule

You can add and update rules by selecting the manage rule tab for your route filter.

The screenshot shows the Azure portal interface for managing a Route Filter named 'MyRouteFilter'. On the left, there's a sidebar with navigation links like Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, and New support request. The main area displays resource details: Resource group (Portal-Demo), Status (Succeeded), Location (East US), Subscription (ExpressRoute-Lab), and Subscription ID (4bfffbb15-d414-4874-a2e4-c548c6d45e2a). Below this, sections for 'Allowed service communities' and 'Circuits' are shown, both currently displaying 'No data'. A red box highlights the 'Manage rule' button at the top right of the main content area.

You can select the services you want to connect to from the drop-down list and save the rule when done.

The screenshot shows the 'Manage rule' dialog box. At the top, there's a title 'Manage rule' with a green icon, a close button (X), and two buttons: 'Save' (with a checkmark icon) and 'Discard'. Below the title, the rule name 'Rule1' is entered in a text input field. Under the heading 'Allowed service communities', a dropdown menu shows '2 selected'. Inside the dropdown, a list of services is displayed with checkboxes. The services listed are: Exchange (12076:5010), Other Office 365 Services (12076:5100) (which is checked), SharePoint Online (12076:5020) (which is checked), Skype For Business (12076:5030), CRM Online (12076:5040), Azure Australia East (12076:51015), Azure Australia Southeast (12076:510...), and Azure Brazil South (12076:51014).



Step 3: Attach the route filter to an ExpressRoute circuit

You can attach the route filter to a circuit by selecting the "add Circuit" button and selecting the ExpressRoute circuit from the drop-down list.

The screenshot shows the Azure portal interface for managing a route filter named 'MyRouteFilter'. The left sidebar has navigation links for Overview, Activity log, Access control (IAM), Tags, SETTINGS, Circuits, Locks, Automation script, SUPPORT + TROUBLESHOOTING, and New support request. The main content area has a search bar and a toolbar with Move, Delete, Manage rule, and Add circuit buttons (the latter is highlighted with a red box). Below the toolbar, there's an 'Essentials' section showing the resource group (Portal-Demo), status (Succeeded), location (East US), and subscription information (ExpressRoute-Lab, ID 4bfbb15-d414-4874-a2e4-c548c6d45e2a). The 'Allowed service communities' table lists Exchange and Other Office 365 Services with values 120765010 and 120765100 respectively. The 'Circuits' section at the bottom shows a table with columns NAME, CIRCUIT STATUS, PROVIDER STATUS, and PROVIDER, with a note 'No data'.

If the connectivity provider configures peering for your ExpressRoute circuit refresh the circuit from the

ExpressRoute circuit blade before you select the "add Circuit" button.

The screenshot shows the Azure portal's ExpressRoute circuit blade for the resource group 'NTT_SMC_Test'. The 'Overview' tab is selected. At the top right, there are buttons for 'Move', 'Delete', and 'Refresh', with 'Refresh' highlighted by a red box. The 'Essentials' section displays basic information: Provider 'NTT SmartConnect', Provider status 'Provisioned', Peering location 'Osaka', Bandwidth '50 Mbps', and Service key. Below this is a table showing circuit types and their status:

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET
Azure private	Provisioned	10.7.2.0/30	10.7.3.0/30
Azure public	Not provisioned	-	-
Microsoft	Provisioned	[redacted]	[redacted]

Common tasks

To get the properties of a route filter

You can view properties of a route filter when you open the resource in the portal.

The screenshot shows the Azure portal's Route Filter blade for the resource group 'USWest-ER-Demo-RG'. The 'Overview' tab is selected. At the top right, there are buttons for 'Move', 'Delete', 'Manage rule', and '+ Add circuit', with 'Manage rule' highlighted by a red box. The 'Essentials' section displays basic information: Resource group 'USWest-ER-Demo-RG', Status 'Succeeded', Location 'West US', Subscription 'ExpressRoute-Lab', and Subscription ID '4bfffbb15-d414-4874-a2e4-c548c6d45e2a'. Below this is a table showing allowed service communities:

NAME	VALUE
Exchange	12076:5100
Other Office 365 Services	12076:5100
SharePoint Online	12076:5020
Skype For Business	12076:5030
CRM Online	12076:5040

Below the communities table is a 'Circuits' section with a table:

NAME	CIRCUIT STATUS	PROVIDER STATUS	PROVIDER
ER-Demo-Ckt-SV	Enabled	Provisioned	Equinix

To update the properties of a route filter

You can update the list of BGP community values attached to a circuit by selecting the "Manage rule" button.

MyRouteFilter
Route filter

Search (Ctrl+ /)

Overview Activity log Access control (IAM) Tags

SETTINGS Circuits Locks Automation script

SUPPORT + TROUBLESHOOTING New support request

→ Move Delete Manage rule + Add circuit

Essentials ▾

Resource group (change)
Portal-Demo
Status
Succeeded
Location
East US
Subscription (change)
ExpressRoute-Lab
Subscription ID
4bfffbb15-d414-4874-a2e4-c548c6d45e2a

Communities filtered
0 communities
Circuits associated
0 circuits

Allowed service communities

Search communities

NAME	VALUE
No data	

Circuits

Search circuits

NAME	CIRCUIT STATUS	PROVIDER STATUS	PROVIDER
No data			

Manage rule

AllowSPO

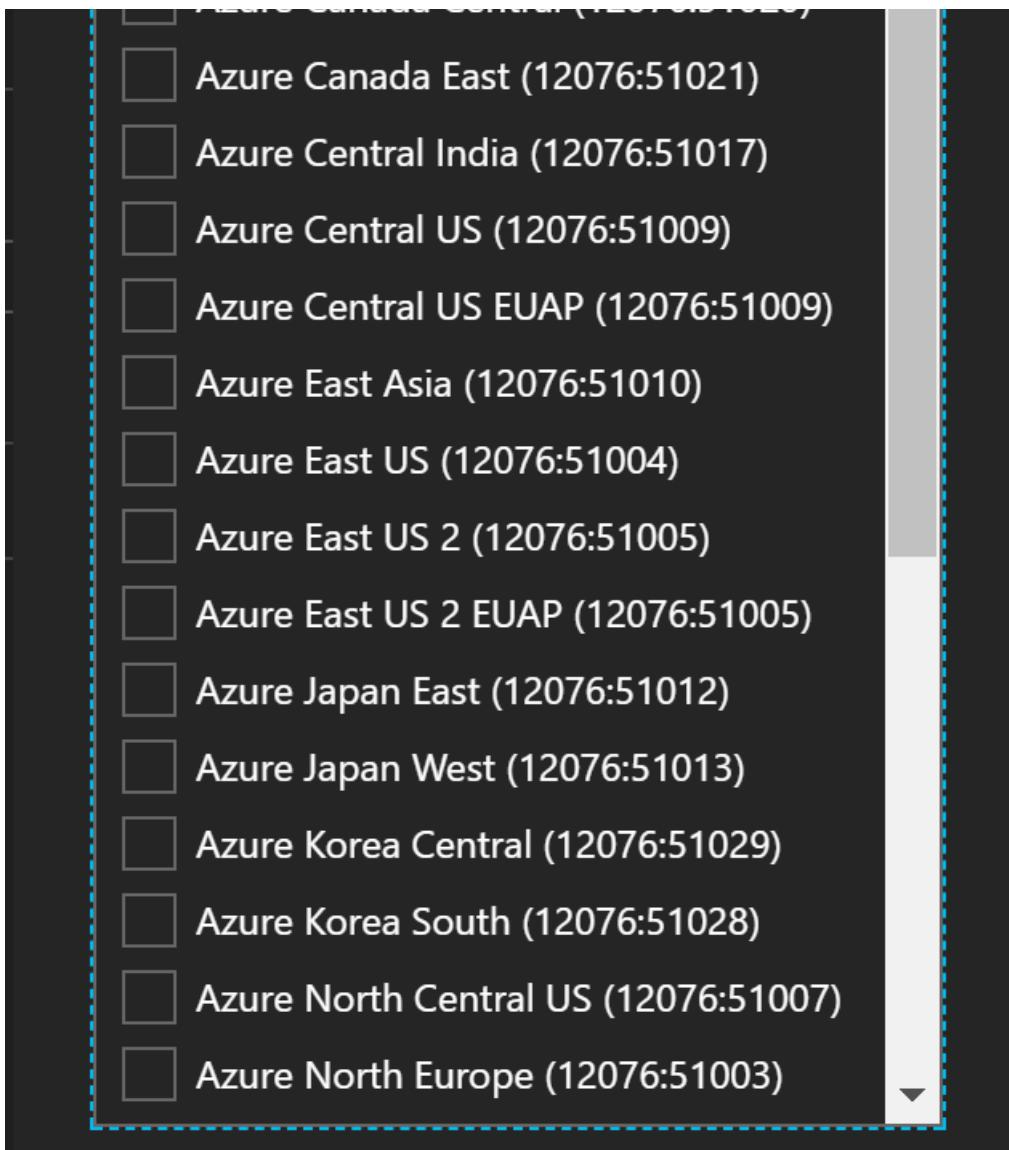
Save Discard

* Rule name

* Allowed service communities

2 selected

- Select all
- Exchange (12076:5010)
- Other Office 365 Services (12076:5100)
- SharePoint Online (12076:5020)
- Skype For Business (12076:5030)
- CRM Online (12076:5040)
- Azure Australia East (12076:51015)
- Azure Australia Southeast (12076:510...)
- Azure Brazil South (12076:51014)
- Azure Canada Central (12076:51020)



To detach a route filter from an ExpressRoute circuit

To detach a circuit from the route filter, right-click on the circuit and click on "disassociate".

The screenshot shows the Azure portal interface for managing a route filter named 'TestRouteFilter'. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, and New support request. The main content area displays the following details:

- Resource group:** USWest-EK-Demo-RG
- Status:** Succeeded
- Location:** West US
- Subscription:** ExpressRoute-Lab
- Subscription ID:** 4bfffbb15-d414-4874-a2e4-c548c6d45e2a
- Communities filtered:** 5 communities
- Circuits associated:** 1 circuits

Allowed service communities:

NAME	VALUE
Exchange	12076:5010
Other Office 365 Services	12076:5100
SharePoint Online	12076:5020
Skype For Business	12076:5030
CRM Online	12076:5040

Circuits:

NAME	CIRCUIT STATUS	PROVIDER STATUS	PROVIDER
ER-Demo-Ckt-SV	Enabled	Provisioned	Equinix

A red box highlights the 'Dissociate' button under the 'ER-Demo-Ckt-SV' row.

To delete a route filter

You can delete a route filter by selecting the delete button.

The screenshot shows the Azure portal interface for managing a route filter named 'TestRouteFilter'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Circuits, Locks, Automation script, and New support request. The main content area displays the 'Essentials' section with details about the resource group (USWest-ER-Demo-RG), status (Succeeded), location (West US), and subscription (ExpressRoute-Lab). It also lists the Subscription ID and the number of communities filtered (5) and circuits associated (1). Below this is the 'Allowed service communities' table, which includes rows for Exchange, Other Office 365 Services, SharePoint Online, Skype For Business, and CRM Online, each with a value of 120765010. At the bottom is the 'Circuits' table, showing one circuit named 'ER-Demo-Ckt-SV' with a provider status of 'Provisioned' and provider 'Equinix'. The 'Delete' button in the top navigation bar is highlighted with a red box.

Next Steps

- For more information about ExpressRoute, see the [ExpressRoute FAQ](#).
- For information about router configuration samples, see [Router configuration samples to set up and manage routing](#).

Configure route filters for Microsoft peering: PowerShell

11/13/2019 • 7 minutes to read • [Edit Online](#)

Route filters are a way to consume a subset of supported services through Microsoft peering. The steps in this article help you configure and manage route filters for ExpressRoute circuits.

Office 365 services such as Exchange Online, SharePoint Online, and Skype for Business, and Azure public services, such as storage and SQL DB are accessible through Microsoft peering. Azure public services are selectable on a per region basis and cannot be defined per public service.

When Microsoft peering is configured on an ExpressRoute circuit and a route filter is attached, all prefixes that are selected for these services are advertised through the BGP sessions that are established. A BGP community value is attached to every prefix to identify the service that is offered through the prefix. For a list of the BGP community values and the services they map to, see [BGP communities](#).

If you require connectivity to all services, a large number of prefixes are advertised through BGP. This significantly increases the size of the route tables maintained by routers within your network. If you plan to consume only a subset of services offered through Microsoft peering, you can reduce the size of your route tables in two ways. You can:

- Filter out unwanted prefixes by applying route filters on BGP communities. This is a standard networking practice and is used commonly within many networks.
- Define route filters and apply them to your ExpressRoute circuit. A route filter is a new resource that lets you select the list of services you plan to consume through Microsoft peering. ExpressRoute routers only send the list of prefixes that belong to the services identified in the route filter.

About route filters

When Microsoft peering is configured on your ExpressRoute circuit, the Microsoft network edge routers establish a pair of BGP sessions with the edge routers (yours or your connectivity provider's). No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter.

A route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. It is essentially an allow list of all the BGP community values. Once a route filter resource is defined and attached to an ExpressRoute circuit, all prefixes that map to the BGP community values are advertised to your network.

To be able to attach route filters with Office 365 services on them, you must have authorization to consume Office 365 services through ExpressRoute. If you are not authorized to consume Office 365 services through ExpressRoute, the operation to attach route filters fails. For more information about the authorization process, see [Azure ExpressRoute for Office 365](#).

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit.

Workflow

To be able to successfully connect to services through Microsoft peering, you must complete the following

configuration steps:

- You must have an active ExpressRoute circuit that has Microsoft peering provisioned. You can use the following instructions to accomplish these tasks:
 - [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.
 - [Create Microsoft peering](#) if you manage the BGP session directly. Or, have your connectivity provider provision Microsoft peering for your circuit.
- You must create and configure a route filter.
 - Identify the services you wish to consume through Microsoft peering
 - Identify the list of BGP community values associated with the services
 - Create a rule to allow the prefix list matching the BGP community values
- You must attach the route filter to the ExpressRoute circuit.

Before you begin

Before you begin configuration, make sure you meet the following criteria:

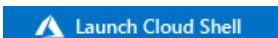
- Review the [prerequisites](#) and [workflows](#) before you begin configuration.
- You must have an active ExpressRoute circuit. Follow the instructions to [Create an ExpressRoute circuit](#) and have the circuit enabled by your connectivity provider before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state.
- You must have an active Microsoft peering. Follow the instructions in the [Create and modifying peering configuration](#) article.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Log in to your Azure account

Before beginning this configuration, you must log in to your Azure account. The cmdlet prompts you for the login credentials for your Azure account. After logging in, it downloads your account settings so they are available to Azure PowerShell.

Open your PowerShell console with elevated privileges, and connect to your account. Use the following example to help you connect. If you are using Azure Cloud Shell, you don't need to run this cmdlet, as you'll be automatically signed in.

```
Connect-AzAccount
```

If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzSubscription
```

Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

Step 1: Get a list of prefixes and BGP community values

1. Get a list of BGP community values

Use the following cmdlet to get the list of BGP community values associated with services accessible through Microsoft peering, and the list of prefixes associated with them:

```
Get-AzBgpServiceCommunity
```

2. Make a list of the values that you want to use

Make a list of BGP community values you want to use in the route filter.

Step 2: Create a route filter and a filter rule

A route filter can have only one rule, and the rule must be of type 'Allow'. This rule can have a list of BGP community values associated with it.

1. Create a route filter

First, create the route filter. The command 'New-AzRouteFilter' only creates a route filter resource. After you create the resource, you must then create a rule and attach it to the route filter object. Run the following command to create a route filter resource:

```
New-AzRouteFilter -Name "MyRouteFilter" -ResourceGroupName "MyResourceGroup" -Location "West US"
```

2. Create a filter rule

You can specify a set of BGP communities as a comma-separated list, as shown in the example. Run the following command to create a new rule:

```
$rule = New-AzRouteFilterRuleConfig -Name "Allow-EX0-D365" -Access Allow -RouteFilterRuleType Community -CommunityList 12076:5010,12076:5040
```

3. Add the rule to the route filter

Run the following command to add the filter rule to the route filter:

```
$routefilter = Get-AzRouteFilter -Name "RouteFilterName" -ResourceGroupName "ExpressRouteResourceGroupName"  
$routefilter.Rules.Add($rule)  
Set-AzRouteFilter -RouteFilter $routefilter
```

Step 3: Attach the route filter to an ExpressRoute circuit

Run the following command to attach the route filter to the ExpressRoute circuit, assuming you have only Microsoft peering:

```
$ckt = Get-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
$ckt.Peerings[0].RouteFilter = $routefilter  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Common tasks

To get the properties of a route filter

To get the properties of a route filter, use the following steps:

1. Run the following command to get the route filter resource:

```
$routefilter = Get-AzRouteFilter -Name "RouteFilterName" -ResourceGroupName  
"ExpressRouteResourceGroupName"
```

2. Get the route filter rules for the route-filter resource by running the following command:

```
$routefilter = Get-AzRouteFilter -Name "RouteFilterName" -ResourceGroupName  
"ExpressRouteResourceGroupName"  
$rule = $routefilter.Rules[0]
```

To update the properties of a route filter

If the route filter is already attached to a circuit, updates to the BGP community list automatically propagate appropriate prefix advertisement changes through the established BGP sessions. You can update the BGP community list of your route filter using the following command:

```
$routefilter = Get-AzRouteFilter -Name "RouteFilterName" -ResourceGroupName "ExpressRouteResourceGroupName"  
$routefilter.rules[0].Communities = "12076:5030", "12076:5040"  
Set-AzRouteFilter -RouteFilter $routefilter
```

To detach a route filter from an ExpressRoute circuit

Once a route filter is detached from the ExpressRoute circuit, no prefixes are advertised through the BGP session. You can detach a route filter from an ExpressRoute circuit using the following command:

```
$ckt.Peerings[0].RouteFilter = $null  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To delete a route filter

You can only delete a route filter if it is not attached to any circuit. Ensure that the route filter is not attached to

any circuit before attempting to delete it. You can delete a route filter using the following command:

```
Remove-AzRouteFilter -Name "MyRouteFilter" -ResourceGroupName "MyResourceGroup"
```

Next Steps

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

Configure route filters for Microsoft peering: Azure CLI

11/13/2019 • 6 minutes to read • [Edit Online](#)

Route filters are a way to consume a subset of supported services through Microsoft peering. The steps in this article help you configure and manage route filters for ExpressRoute circuits.

Office 365 services such as Exchange Online, SharePoint Online, and Skype for Business, are accessible through the Microsoft peering. When Microsoft peering is configured in an ExpressRoute circuit, all prefixes related to these services are advertised through the BGP sessions that are established. A BGP community value is attached to every prefix to identify the service that is offered through the prefix. For a list of the BGP community values and the services they map to, see [BGP communities](#).

If you require connectivity to all services, a large number of prefixes are advertised through BGP. This significantly increases the size of the route tables maintained by routers within your network. If you plan to consume only a subset of services offered through Microsoft peering, you can reduce the size of your route tables in two ways.

You can:

- Filter out unwanted prefixes by applying route filters on BGP communities. This is a standard networking practice and is used commonly within many networks.
- Define route filters and apply them to your ExpressRoute circuit. A route filter is a new resource that lets you select the list of services you plan to consume through Microsoft peering. ExpressRoute routers only send the list of prefixes that belong to the services identified in the route filter.

About route filters

When Microsoft peering is configured on your ExpressRoute circuit, the Microsoft edge routers establish a pair of BGP sessions with the edge routers (yours or your connectivity provider's). No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter.

A route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. It is essentially a white list of all the BGP community values. Once a route filter resource is defined and attached to an ExpressRoute circuit, all prefixes that map to the BGP community values are advertised to your network.

To be able to attach route filters with Office 365 services on them, you must have authorization to consume Office 365 services through ExpressRoute. If you are not authorized to consume Office 365 services through ExpressRoute, the operation to attach route filters fails. For more information about the authorization process, see [Azure ExpressRoute for Office 365](#).

IMPORTANT

Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through Microsoft peering, even if route filters are not defined. Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit.

Workflow

To be able to successfully connect to services through Microsoft peering, you must complete the following configuration steps:

- You must have an active ExpressRoute circuit that has Microsoft peering provisioned. You can use the

following instructions to accomplish these tasks:

- [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.
- [Create Microsoft peering](#) if you manage the BGP session directly. Or, have your connectivity provider provision Microsoft peering for your circuit.
- You must create and configure a route filter.
 - Identify the services you wish to consume through Microsoft peering
 - Identify the list of BGP community values associated with the services
 - Create a rule to allow the prefix list matching the BGP community values
- You must attach the route filter to the ExpressRoute circuit.

Before you begin

Before beginning, install the latest version of the CLI commands (2.0 or later). For information about installing the CLI commands, see [Install the Azure CLI](#) and [Get Started with Azure CLI](#).

- Review the [prerequisites](#) 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 before you proceed. The ExpressRoute circuit must be in a provisioned and enabled state.
- You must have an active Microsoft peering. Follow instructions at [Create and modifying peering configuration](#)

Sign in to your Azure account and select your subscription

To begin your configuration, sign in to your Azure account. If you are using the "Try It", you are signed in automatically and can skip the login step. Use the following examples to help you connect:

```
az login
```

Check the subscriptions for the account.

```
az account list
```

Select the subscription for which you want to create an ExpressRoute circuit.

```
az account set --subscription "<subscription ID>"
```

Step 1: Get a list of prefixes and BGP community values

1. Get a list of BGP community values

Use the following cmdlet to get the list of BGP community values associated with services accessible through Microsoft peering, and the list of prefixes associated with them:

```
az network route-filter rule list-service-communities
```

2. Make a list of the values that you want to use

Make a list of BGP community values you want to use in the route filter.

Step 2: Create a route filter and a filter rule

A route filter can have only one rule, and the rule must be of type 'Allow'. This rule can have a list of BGP community values associated with it.

1. Create a route filter

First, create the route filter. The command `az network route-filter create` only creates a route filter resource.

After you create the resource, you must then create a rule and attach it to the route filter object. Run the following command to create a route filter resource:

```
az network route-filter create -n MyRouteFilter -g MyResourceGroup
```

2. Create a filter rule

Run the following command to create a new rule:

```
az network route-filter rule create --filter-name MyRouteFilter -n CRM --communities 12076:5040 --access Allow -g MyResourceGroup
```

Step 3: Attach the route filter to an ExpressRoute circuit

Run the following command to attach the route filter to the ExpressRoute circuit:

```
az network express-route peering update --circuit-name MyCircuit -g ExpressRouteResourceGroupName --name MicrosoftPeering --route-filter MyRouteFilter
```

Common tasks

To get the properties of a route filter

To get the properties of a route filter, use the following command:

```
az network route-filter show -g ExpressRouteResourceGroupName --name MyRouteFilter
```

To update the properties of a route filter

If the route filter is already attached to a circuit, updates to the BGP community list automatically propagate appropriate prefix advertisement changes through the established BGP sessions. You can update the BGP community list of your route filter using the following command:

```
az network route-filter rule update --filter-name MyRouteFilter -n CRM -g ExpressRouteResourceGroupName --add communities '12076:5040' --add communities '12076:5010'
```

To detach a route filter from an ExpressRoute circuit

Once a route filter is detached from the ExpressRoute circuit, no prefixes are advertised through the BGP session. You can detach a route filter from an ExpressRoute circuit using the following command:

```
az network express-route peering update --circuit-name MyCircuit -g ExpressRouteResourceGroupName --name MicrosoftPeering --remove routeFilter
```

To delete a route filter

You can only delete a route filter if it is not attached to any circuit. Ensure that the route filter is not attached to any

circuit before attempting to delete it. You can delete a route filter using the following command:

```
az network route-filter delete -n MyRouteFilter -g MyResourceGroup
```

Next Steps

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

Reset ExpressRoute circuit peerings

1/14/2020 • 3 minutes to read • [Edit Online](#)

This article describes how to disable and enable peerings of an ExpressRoute circuit using PowerShell. When you disable a peering, the BGP session on both the primary connection and the secondary connection of your ExpressRoute circuit will be shut down. You will lose connectivity through this peering to Microsoft. When you enable a peering, the BGP session on both the primary connection and the secondary connection of your ExpressRoute circuit will be brought up. You will regain connectivity through this peering to Microsoft. You can enable and disable Microsoft Peering and Azure Private Peering on an ExpressRoute circuit independently. When you first configure the peerings on your ExpressRoute circuit, the peerings are enabled by default.

There are a couple scenarios where you may find it helpful resetting your ExpressRoute peerings.

- Test your disaster recovery design and implementation. For example, you have two ExpressRoute circuits. You can disable the peerings of one circuit and force your network traffic to fail over to the other circuit.
- Enable Bidirectional Forwarding Detection (BFD) on Azure Private Peering or Microsoft Peering of your ExpressRoute circuit. BFD is enabled by default on Azure Private Peering if your ExpressRoute circuit is created after August 1 2018 and on Microsoft Peering if your ExpressRoute circuit is created after January 10 2020. If your circuit was created before that, BFD wasn't enabled. You can enable BFD by disabling the peering and reenabling it.

Working with Azure PowerShell

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

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

There are a few ways to launch the Cloud Shell:

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

Reset a peering

1. If you are running PowerShell locally, open your PowerShell console with elevated privileges, and connect to

your account. Use the following example to help you connect:

```
Connect-AzAccount
```

2. If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzSubscription
```

3. Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

4. Run the following commands to retrieve your ExpressRoute circuit.

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

5. Identify the peering you want to disable or enable. *Peerings* is an array. In the following example, Peerings[0] is Azure Private Peering and Peerings[1] Microsoft Peering.

```
Name : ExpressRouteARMCircuit
ResourceGroupName : ExpressRouteResourceGroup
Location : westus
Id : /subscriptions/#####-####-####-####-#####
#/#/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/E
xpressRouteARMCircuit
Etag : W/"cd011bef-dc79-49eb-b4c6-81fb6ea5d178"
ProvisioningState : Succeeded
Sku :
{
    "Name": "Standard_MeteredData",
    "Tier": "Standard",
    "Family": "MeteredData"
}
CircuitProvisioningState : Enabled
ServiceProviderProvisioningState : Provisioned
ServiceProviderNotes :
ServiceProviderProperties :
{
    "ServiceProviderName": "Coresite",
    "PeeringLocation": "Los Angeles",
    "BandwidthInMbps": 50
}
ServiceKey : ######-###-###-###-#####
Peerings :
[
    {
        "Name": "AzurePrivatePeering",
        "Etag": "W/"cd011bef-dc79-49eb-b4c6-81fb6ea5d178"",
        "Id": "/subscriptions/#####-####-####-#####
#/#/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/E
xpressRouteARMCircuit/peerings/AzurePrivatePeering",
        "PeeringType": "AzurePrivatePeering",
        "State": "Enabled",
        "AzureASN": 12076,
        "PeerASN": 123,
        "PrimaryPeerAddressPrefix": "10.0.0.0/30",
        "SecondaryPeerAddressPrefix": "10.0.0.4/30",
        "PrimaryAzurePort": "",
        "SecondaryAzurePort": "",
        "VlanId": 789,
        "MicrosoftPeeringConfig": {
            "AdvertisedPublicPrefixes": []
        }
    }
]
```

```

        "AdvertisedCommunities": [],
        "AdvertisedPublicPrefixesState": "NotConfigured",
        "CustomerASN": 0,
        "LegacyMode": 0,
        "RoutingRegistryName": "NONE"
    },
    "ProvisioningState": "Succeeded",
    "GatewayManagerEtag": "",
    "LastModifiedBy": "Customer",
    "Connections": []
},
{
    "Name": "MicrosoftPeering",
    "Etag": "W/\"cd011bef-dc79-49eb-b4c6-81fb6ea5d178\"",
    "Id": "/subscriptions/#####-####-####-####-#####
#####/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/E
xpressRouteARMCircuit/peerings/MicrosoftPeering",
    "PeeringType": "MicrosoftPeering",
    "State": "Enabled",
    "AzureASN": 12076,
    "PeerASN": 123,
    "PrimaryPeerAddressPrefix": "3.0.0.0/30",
    "SecondaryPeerAddressPrefix": "3.0.0.4/30",
    "PrimaryAzurePort": "",
    "SecondaryAzurePort": "",
    "VlanId": 345,
    "MicrosoftPeeringConfig": {
        "AdvertisedPublicPrefixes": [
            "3.0.0.3/32"
        ],
        "AdvertisedCommunities": [],
        "AdvertisedPublicPrefixesState": "ValidationNeeded",
        "CustomerASN": 0,
        "LegacyMode": 0,
        "RoutingRegistryName": "NONE"
    },
    "ProvisioningState": "Succeeded",
    "GatewayManagerEtag": "",
    "LastModifiedBy": "Customer",
    "Connections": []
}
]
Authorizations : []
AllowClassicOperations : False
GatewayManagerEtag :

```

6. Run the following commands to change the state of the peering.

```
$ckt.Peering[0].State = "Disabled"
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The peering should be in a state you set.

Next steps

If you need help to troubleshoot an ExpressRoute problem, check out the following articles:

- [Verifying ExpressRoute connectivity](#)
- [Troubleshooting network performance](#)

Move a public peering to Microsoft peering

12/17/2019 • 5 minutes to read • [Edit Online](#)

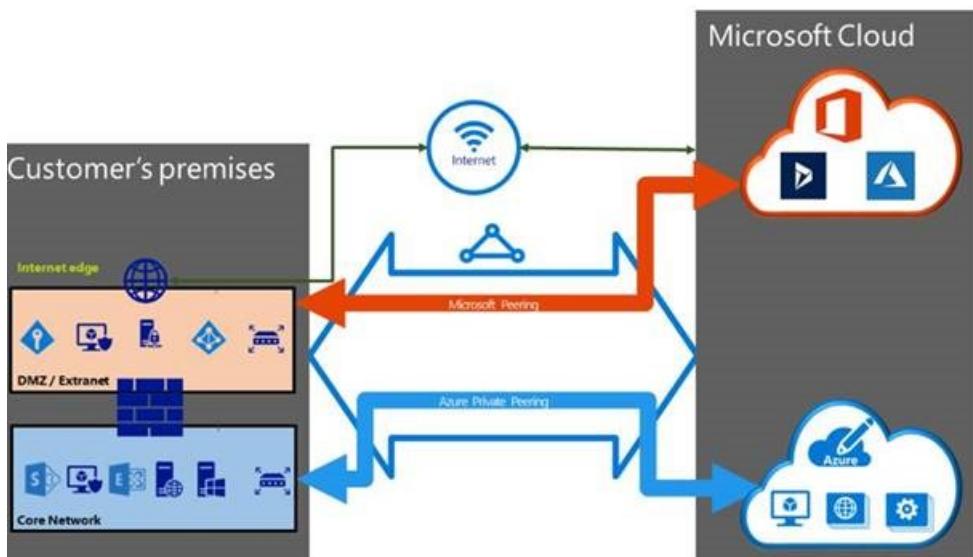
This article helps you move a public peering configuration to Microsoft peering with no downtime. ExpressRoute supports using Microsoft peering with route filters for Azure PaaS services, such as Azure storage and Azure SQL Database. You now need only one routing domain to access Microsoft PaaS and SaaS services. You can use route filters to selectively advertise the PaaS service prefixes for Azure regions you want to consume.

Azure public peering has 1 NAT IP address associated to each BGP session. Microsoft peering allows you to configure your own NAT allocations, as well as use route filters for selective prefix advertisements. Public Peering is a unidirectional service using which 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 can connect to all Azure services. We do not allow you to selectively pick services for which we advertise routes to. While Microsoft peering is a bi-directional connectivity where connection can be initiated from Microsoft Azure service along with your WAN. For more information about routing domains and peering, see [ExpressRoute circuits and routing domains](#).

Before you begin

To connect to Microsoft peering, you need to set up and manage NAT. Your connectivity provider may set up and manage the NAT as a managed service. If you are planning to access the Azure PaaS and Azure SaaS services on Microsoft peering, it's important to size the NAT IP pool correctly. For more information about NAT for ExpressRoute, see the [NAT requirements for Microsoft peering](#). When you connect to Microsoft through Azure ExpressRoute(Microsoft peering), 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.



WARNING

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

Refer to [Asymmetric routing with multiple network paths](#) for caveats of asymmetric routing before configuring

Microsoft peering.

- If you are using public peering and currently have IP Network rules for public IP addresses that are used to access [Azure Storage](#) or [Azure SQL Database](#), you need to make sure that the NAT IP pool configured with Microsoft peering is included in the list of public IP addresses for the Azure storage account or Azure SQL account.
- In order to move to Microsoft peering with no downtime, use the steps in this article in the order that they are presented.

1. Create Microsoft peering

If Microsoft peering has not been created, use any of the following articles to create Microsoft peering. If your connectivity provider offers managed layer 3 services, you can ask the connectivity provider to enable Microsoft peering for your circuit.

If the layer 3 is managed by you the following information is required 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. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A /30 subnet for the secondary link. This must be a valid public IPv4 prefix owned by you and registered in an RIR / IRR. From this subnet you will assign the first useable IP address to your router as Microsoft uses the second useable IP for its router.
- A valid VLAN ID to establish this peering on. Ensure that no other peering in the circuit uses the same VLAN ID. For both Primary and Secondary links you must use 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. If you plan to send a set of prefixes, you can send a comma-separated list. These prefixes must be registered to you in an RIR / IRR.
- Routing Registry Name: You can specify the RIR / IRR against which the AS number and prefixes are registered.
- **Optional** - 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.
- **Optional** - An MD5 hash if you choose to use one.

Detailed instructions to enable Microsoft peering can be found in the following articles:

- [Create Microsoft peering using Azure portal](#)
- [Create Microsoft peering using Azure Powershell](#)
- [Create Microsoft peering using Azure CLI](#)

2. Validate Microsoft peering is enabled

Verify that the Microsoft peering is enabled and the advertised public prefixes are in the configured state.

- [Azure portal](#)
- [Azure PowerShell](#)
- [Azure CLI](#)

3. Configure and attach a route filter to the circuit

By default, new Microsoft peering do not advertise any prefixes until a route filter is attached to the circuit. When you create a route filter rule, you can specify the list of service communities for Azure regions that you want to consume for Azure PaaS services. This provides you the flexibility to filter the routes as per your requirement, as shown in the following screenshot:

The screenshot shows the 'Manage rule' interface in the Azure portal. At the top, there is a 'Save' button and a 'Discard' button. Below that, a section labeled 'circuit' contains a 'Rule name' field with the value 'Rule_1' and a green checkmark icon. Another section labeled 'Allowed service communities' contains a dropdown menu with the value 'Azure Storage West US (12076:52006)' highlighted. A scrollable list of service communities follows, with 'Azure Storage West US (12076:52006)' checked and highlighted in blue.

Service Community
Azure Storage West India (12076:52018)
<input checked="" type="checkbox"/> Azure Storage West US (12076:52006)
Azure Storage West US 2 (12076:52026)
Azure SQL Australia Central (12076:53032)
Azure SQL Australia Central 2 (12076:53033)
Azure SQL Australia East (12076:53015)
Azure SQL Australia Southeast (12076:53016)
Azure SQL Brazil South (12076:53014)
Azure SQL Canada Central (12076:53020)
Azure SQL Canada East (12076:53021)
Azure SQL Central India (12076:53017)
Azure SQL Central US (12076:53009)

Configure route filters using any of the following articles:

- [Configure route filters for Microsoft peering using Azure portal](#)
- [Configure route filters for Microsoft peering using Azure PowerShell](#)
- [Configure route filters for Microsoft peering using Azure CLI](#)

4. Delete the public peering

After verifying that the Microsoft peering is configured and the prefixes you wish to consume are correctly advertised on Microsoft peering, you can then delete the public peering. To delete the public peering, use any of the following articles:

- [Delete Azure public peering using Azure PowerShell](#)
- [Delete Azure public peering using CLI](#)

5. View peerings

You can see a list of all ExpressRoute circuits and peerings in the Azure portal. For more information, see [View Microsoft peering details](#).

Next steps

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

Move ExpressRoute circuits from classic to Resource Manager deployment model using PowerShell

11/14/2019 • 4 minutes to read • [Edit Online](#)

To use an ExpressRoute circuit for both the classic and Resource Manager deployment models, you must move the circuit to the Resource Manager deployment model. The following sections help you move your circuit by using PowerShell.

Before you begin

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

- Verify that you have installed both the classic and Az Azure PowerShell modules locally on your computer. For more information, see [How to install and configure Azure PowerShell](#).
- Make sure that you have reviewed the [prerequisites](#), [routing requirements](#), and [workflows](#) before you begin configuration.
- Review the information that is provided under [Moving an ExpressRoute circuit from classic to Resource Manager](#). Make sure that you fully understand the limits and limitations.
- Verify that the circuit is fully operational in the classic deployment model.
- Ensure that you have a resource group that was created in the Resource Manager deployment model.

Move an ExpressRoute circuit

Step 1: Gather circuit details from the classic deployment model

Sign in to the Azure classic environment and gather the service key.

1. Sign in to your Azure account.

```
Add-AzureAccount
```

2. Select the appropriate Azure subscription.

```
Select-AzureSubscription "<Enter Subscription Name here>"
```

3. Import the PowerShell modules for Azure and ExpressRoute.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.1.1\Azure\Azure.psd1'  
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.1.1\ExpressRoute\ExpressRoute.psd1'
```

4. Use the cmdlet below to get the service keys for all of your ExpressRoute circuits. After retrieving the keys, copy the **service key** of the circuit that you want to move to the Resource Manager deployment model.

```
Get-AzureDedicatedCircuit
```

Step 2: Sign in and create a resource group

Sign in to the Resource Manager environment and create a new resource group.

1. Sign in to your Azure Resource Manager environment.

```
Connect-AzAccount
```

2. Select the appropriate Azure subscription.

```
Get-AzSubscription -SubscriptionName "<Enter Subscription Name here>" | Select-AzSubscription
```

3. Modify the snippet below to create a new resource group if you don't already have a resource group.

```
New-AzResourceGroup -Name "DemoRG" -Location "West US"
```

Step 3: Move the ExpressRoute circuit to the Resource Manager deployment model

You are now ready to move your ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model. Before proceeding, review the information provided in [Moving an ExpressRoute circuit from the classic to the Resource Manager deployment model](#).

To move your circuit, modify and run the following snippet:

```
Move-AzExpressRouteCircuit -Name "MyCircuit" -ResourceGroupName "DemoRG" -Location "West US" -ServiceKey "<Service-key>"
```

In classic mode, an ExpressRoute circuit does not have the concept of being tied to a region. However, in Resource Manager, every resource needs to be mapped to an Azure region. The region specified in the Move-AzExpressRouteCircuit cmdlet can technically be any region. For organizational purposes, you may want to choose a region that closely represents your peering location.

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.

Modify circuit access

To enable ExpressRoute circuit access for both deployment models

After moving your classic ExpressRoute circuit to the Resource Manager deployment model, you can enable access to both deployment models. Run the following cmdlets to enable access to both deployment models:

1. Get the circuit details.

```
$ckt = Get-AzExpressRouteCircuit -Name "DemoCkt" -ResourceGroupName "DemoRG"
```

2. Set "Allow Classic Operations" to TRUE.

```
$ckt.AllowClassicOperations = $true
```

3. Update the circuit. After this operation has finished successfully, you will be able to view the circuit in the classic deployment model.

```
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

4. Run the following cmdlet to get the details of the ExpressRoute circuit. You must be able to see the service key listed.

```
get-azurededicatedcircuit
```

5. You can now manage links to the ExpressRoute circuit using the classic deployment model commands for classic VNets, and the Resource Manager commands for Resource Manager VNets. The following articles help you 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](#)

To disable ExpressRoute circuit access to the classic deployment model

Run the following cmdlets to disable access to the classic deployment model.

1. Get details of the ExpressRoute circuit.

```
$ckt = Get-AzExpressRouteCircuit -Name "DemoCkt" -ResourceGroupName "DemoRG"
```

2. Set "Allow Classic Operations" to FALSE.

```
$ckt.AllowClassicOperations = $false
```

3. Update the circuit. After this operation has finished successfully, you will not be able to view the circuit in the classic deployment model.

```
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Next steps

- [Create and modify routing for your ExpressRoute circuit](#)
- [Link your virtual network to your ExpressRoute circuit](#)

Migrate ExpressRoute-associated virtual networks from classic to Resource Manager

2/6/2020 • 4 minutes to read • [Edit Online](#)

This article explains how to migrate ExpressRoute-associated virtual networks from the classic deployment model to the Azure Resource Manager deployment model after moving your ExpressRoute circuit.

Before you begin

NOTE

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

- Verify that you have the latest versions of the Azure PowerShell modules. For more information, see [How to install and configure Azure PowerShell](#). To install the PowerShell Service Management module (which is needed for the classic deployment model), see [Installing the Azure PowerShell Service Management Module](#).
- Make sure that you have reviewed the [prerequisites](#), [routing requirements](#), and [workflows](#) before you begin configuration.
- Review the information that is provided under [Moving an ExpressRoute circuit from classic to Resource Manager](#). Make sure that you fully understand the limits and limitations.
- Verify that the circuit is fully operational in the classic deployment model.
- Ensure that you have a resource group that was created in the Resource Manager deployment model.
- Review the following resource-migration documentation:
 - [Platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
 - [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
 - [FAQs: Platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
 - [Review most common migration errors and mitigations](#)

Supported and unsupported scenarios

- An ExpressRoute circuit can be moved from the classic to the Resource Manager environment without any downtime. You can move any ExpressRoute circuit from the classic to the Resource Manager environment with no downtime. Follow the instructions in [moving ExpressRoute circuits from the classic to the Resource Manager deployment model using PowerShell](#). This is a prerequisite to move resources connected to the virtual network.
- Virtual networks, gateways, and associated deployments within the virtual network that are attached to an ExpressRoute circuit in the same subscription can be migrated to the Resource Manager environment without any downtime. You can follow the steps described later to migrate resources such as virtual networks, gateways, and virtual machines deployed within the virtual network. You must ensure that the virtual networks are configured correctly before they are migrated.

- Virtual networks, gateways, and associated deployments within the virtual network that are not in the same subscription as the ExpressRoute circuit require some downtime to complete the migration. The last section of the document describes the steps to be followed to migrate resources.
- A virtual network with both ExpressRoute Gateway and VPN Gateway can't be migrated.
- ExpressRoute circuit cross-subscription migration is not supported. For more information, see [Microsoft.Network move support](#).

Move an ExpressRoute circuit from classic to Resource Manager

You must move an ExpressRoute circuit from the classic to the Resource Manager environment before you try to migrate resources that are attached to the ExpressRoute circuit. To accomplish this task, see the following articles:

- Review the information that is provided under [Moving an ExpressRoute circuit from classic to Resource Manager](#).
- [Move a circuit from classic to Resource Manager using Azure PowerShell](#).
- Use the Azure Service Management portal. You can follow the workflow to [create a new ExpressRoute circuit](#) and select the import option.

This operation does not involve downtime. You can continue to transfer data between your premises and Microsoft while the migration is in progress.

Migrate virtual networks, gateways, and associated deployments

The steps you follow to migrate depend on whether your resources are in the same subscription, different subscriptions, or both.

Migrate virtual networks, gateways, and associated deployments in the same subscription as the ExpressRoute circuit

This section describes the steps to be followed to migrate a virtual network, gateway, and associated deployments in the same subscription as the ExpressRoute circuit. No downtime is associated with this migration. You can continue to use all resources through the migration process. The management plane is locked while the migration is in progress.

- Ensure that the ExpressRoute circuit has been moved from the classic to the Resource Manager environment.
- Ensure that the virtual network has been prepared appropriately for the migration.
- Register your subscription for resource migration. To register your subscription for resource migration, use the following PowerShell snippet:

```
Select-AzSubscription -SubscriptionName <Your Subscription Name>
Register-AzResourceProvider -ProviderNamespace Microsoft.ClassicInfrastructureMigrate
Get-AzResourceProvider -ProviderNamespace Microsoft.ClassicInfrastructureMigrate
```

- Validate, prepare, and migrate. To move the virtual network, use the following PowerShell snippet:

```
Move-AzureVirtualNetwork -Validate -VirtualNetworkName $vnetName
Move-AzureVirtualNetwork -Prepare -VirtualNetworkName $vnetName
Move-AzureVirtualNetwork -Commit -VirtualNetworkName $vnetName
```

You can also abort migration by running the following PowerShell cmdlet:

```
Move-AzureVirtualNetwork -Abort $vnetName
```

Next steps

- Platform-supported migration of IaaS resources from classic to Azure Resource Manager
- Technical deep dive on platform-supported migration from classic to Azure Resource Manager
- FAQs: Platform-supported migration of IaaS resources from classic to Azure Resource Manager
- Review most common migration errors and mitigations

Create a zone-redundant virtual network gateway in Azure Availability Zones

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

You can deploy VPN and ExpressRoute gateways in Azure Availability Zones. This brings resiliency, scalability, and higher availability to virtual network gateways. Deploying gateways in Azure Availability Zones physically and logically separates gateways within a region, while protecting your on-premises network connectivity to Azure from zone-level failures. For information, see [About zone-redundant virtual network gateways](#) and [About Azure Availability Zones](#).

Before you begin

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

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

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

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

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

1. Declare your variables

Declare the variables that you want to use. Use the following sample, substituting the values for your own when necessary. If you close your PowerShell/Cloud Shell session at any point during the exercise, just copy and paste the values again to re-declare the variables. When specifying location, verify that the region you specify is supported. For more information, see the [FAQ](#).

```
$RG1      = "TestRG1"
$VNet1    = "VNet1"
$Location1 = "CentralUS"
$FESubnet1 = "FrontEnd"
$BESubnet1 = "Backend"
$GwSubnet1 = "GatewaySubnet"
$VNet1Prefix = "10.1.0.0/16"
$FEPrefix1  = "10.1.0.0/24"
$BEPrefix1  = "10.1.1.0/24"
$GwPrefix1  = "10.1.255.0/27"
$Gw1       = "VNet1GW"
$GwIP1     = "VNet1GWIP"
$GwIPConf1 = "gwipconf1"
```

2. Create the virtual network

Create a resource group.

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

Create a virtual network.

```
$fesub1 = New-AzVirtualNetworkSubnetConfig -Name $FESubnet1 -AddressPrefix $FEPrefix1
$besub1 = New-AzVirtualNetworkSubnetConfig -Name $BESubnet1 -AddressPrefix $BEPrefix1
$vnet = New-AzVirtualNetwork -Name $VNet1 -ResourceGroupName $RG1 -Location $Location1 -AddressPrefix
$VNet1Prefix -Subnet $fesub1,$besub1
```

3. Add the gateway subnet

The gateway subnet contains the reserved IP addresses that the virtual network gateway services use. Use the following examples to add and set a gateway subnet:

Add the gateway subnet.

```
$getvnet = Get-AzVirtualNetwork -ResourceGroupName $RG1 -Name VNet1
Add-AzVirtualNetworkSubnetConfig -Name 'GatewaySubnet' -AddressPrefix 10.1.255.0/27 -VirtualNetwork $getvnet
```

Set the gateway subnet configuration for the virtual network.

```
$getvnet | Set-AzVirtualNetwork
```

4. Request a public IP address

In this step, choose the instructions that apply to the gateway that you want to create. The selection of zones for deploying the gateways depends on the zones specified for the public IP address.

For zone-redundant gateways

Request a public IP address with a **Standard** PublicIpAddress SKU and do not specify any zone. In this case, the Standard public IP address created will be a zone-redundant public IP.

```
$pip1 = New-AzPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod Static -
Sku Standard
```

For zonal gateways

Request a public IP address with a **Standard** PublicIpAddress SKU. Specify the zone (1, 2 or 3). All gateway instances will be deployed in this zone.

```
$pip1 = New-AzPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod Static -Sku Standard -Zone 1
```

For regional gateways

Request a public IP address with a **Basic** PublicIpAddress SKU. In this case, the gateway is deployed as a regional gateway and does not have any zone-redundancy built into the gateway. The gateway instances are created in any zones, respectively.

```
$pip1 = New-AzPublicIpAddress -ResourceGroup $RG1 -Location $Location1 -Name $GwIP1 -AllocationMethod Dynamic -Sku Basic
```

5. Create the IP configuration

```
$getvnet = Get-AzVirtualNetwork -ResourceGroupName $RG1 -Name $VNet1  
$subnet = Get-AzVirtualNetworkSubnetConfig -Name $GwSubnet1 -VirtualNetwork $getvnet  
$gwipconf1 = New-AzVirtualNetworkGatewayIpConfig -Name $GwIPConf1 -Subnet $subnet -PublicIpAddress $pip1
```

6. Create the gateway

Create the virtual network gateway.

For ExpressRoute

```
New-AzVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations $GwIPConf1 -GatewayType ExpressRoute -GatewaySku ErGw1AZ
```

For VPN Gateway

```
New-AzVirtualNetworkGateway -ResourceGroup $RG1 -Location $Location1 -Name $Gw1 -IpConfigurations $GwIPConf1 -GatewayType Vpn -VpnType RouteBased -GatewaySku VpnGw1AZ
```

FAQ

What will change when I deploy these new SKUs?

From your perspective, you can deploy your gateways with zone-redundancy. This means that all instances of the gateways will be deployed across Azure Availability Zones, and each Availability Zone is a different fault and update domain. This makes your gateways more reliable, available, and resilient to zone failures.

Can I use the Azure portal?

Yes, you can use the Azure portal to deploy the new SKUs. However, you will see these new SKUs only in those Azure regions that have Azure Availability Zones.

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

See [Availability Zones](#) for the latest list of available regions.

Can I change/migrate/upgrade my existing virtual network gateways to zone-redundant or zonal gateways?

Migrating your existing virtual network gateways to zone-redundant or zonal gateways is currently not supported.

You can, however, delete your existing gateway and re-create a zone-redundant or zonal gateway.

Can I deploy both VPN and Express Route gateways in same virtual network?

Co-existence of both VPN and Express Route gateways in the same virtual network is supported. However, you should reserve a /27 IP address range for the gateway subnet.

Router configuration samples to set up and manage routing

11/13/2019 • 4 minutes to read • [Edit Online](#)

This page provides interface and routing configuration samples for Cisco IOS-XE and Juniper MX series routers when working with ExpressRoute. 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.

MTU and TCP MSS settings on router interfaces

- The MTU for the ExpressRoute interface is 1500, which is the typical default MTU for an Ethernet interface on a router. Unless your router has a different MTU by default, there is no need to specify a value on the router interface.
- Unlike an Azure VPN Gateway, the TCP MSS for an ExpressRoute circuit does not need to be specified.

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

11/13/2019 • 4 minutes to read • [Edit Online](#)

This page provides NAT configuration samples for Cisco ASA and Juniper SRX series routers when working with ExpressRoute. 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,
<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.

ExpressRoute monitoring, metrics, and alerts

12/11/2019 • 3 minutes to read • [Edit Online](#)

This article helps you understand ExpressRoute monitoring, metrics, and alerts using Azure Monitor. Azure Monitor is one stop shop for all metrics, alerting, diagnostic logs across all of Azure.

NOTE

Using **Classic Metrics** is not recommended.

ExpressRoute metrics

To view **Metrics**, navigate to the *Azure Monitor* page and click *Metrics*. To view **ExpressRoute** metrics, filter by Resource Type *ExpressRoute circuits*. To view **Global Reach** metrics, filter by Resource Type *ExpressRoute circuits* and select an ExpressRoute circuit resource that has Global Reach enabled. To view **ExpressRoute Direct** metrics, filter Resource Type by *ExpressRoute Ports*.

Once a metric is selected, the default aggregation will be applied. Optionally, you can apply splitting, which will show the metric with different dimensions.

Available Metrics

Metric	Category	Dimension(s)	Feature(s)
ARP Availability	Availability	<ul style="list-style-type: none">Peer (Primary/Secondary ExpressRoute router)Peering Type (Private/Public/Microsoft)	ExpressRoute
Bgp Availability	Availability	<ul style="list-style-type: none">Peer (Primary/Secondary ExpressRoute router)Peering Type	ExpressRoute
BitsInPerSecond	Traffic	<ul style="list-style-type: none">Peering Type (ExpressRoute)Link (ExpressRoute Direct)	<ul style="list-style-type: none">ExpressRouteExpressRoute Direct
BitsOutPerSecond	Traffic	<ul style="list-style-type: none">Peering Type (ExpressRoute)Link (ExpressRoute Direct)	<ul style="list-style-type: none">ExpressRouteExpressRoute Direct
GlobalReachBitsInPerSecond	Traffic	<ul style="list-style-type: none">Peered Circuit Skey (Service Key)	Global Reach
GlobalReachBitsOutPerSecond	Traffic	<ul style="list-style-type: none">Peered Circuit Skey (Service Key)	Global Reach
AdminState	Physical Connectivity	Link	ExpressRoute Direct
LineProtocol	Physical Connectivity	Link	ExpressRoute Direct

METRIC	CATEGORY	DIMENSION(S)	FEATURE(S)
RxLightLevel	Physical Connectivity	• Link • Lane	ExpressRoute Direct
TxLightLevel	Physical Connectivity	• Link • Lane	ExpressRoute Direct

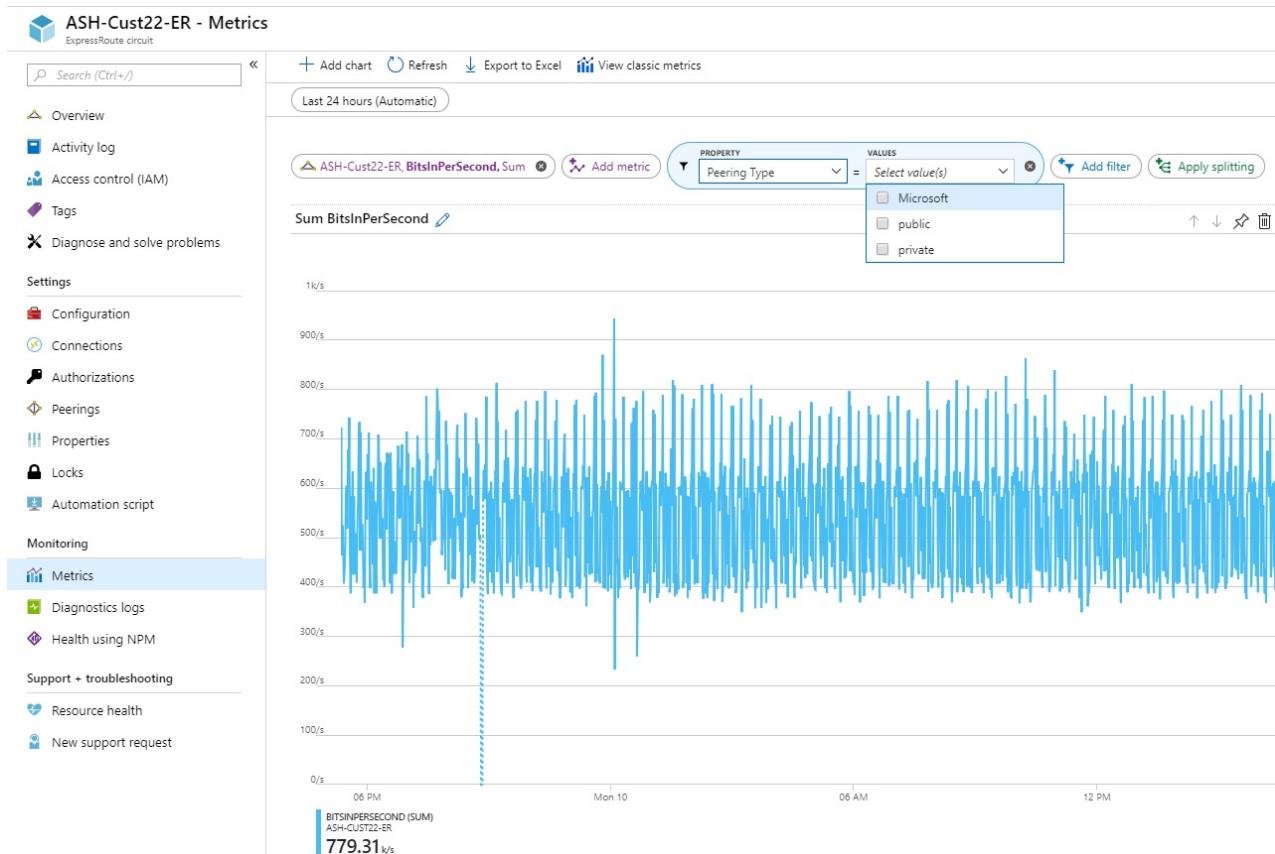
NOTE

Using `GlobalGlobalReachBitsInPerSecond` and `GlobalGlobalReachBitsOutPerSecond` will only be visible if at least one Global Reach connection is established.

Circuits metrics

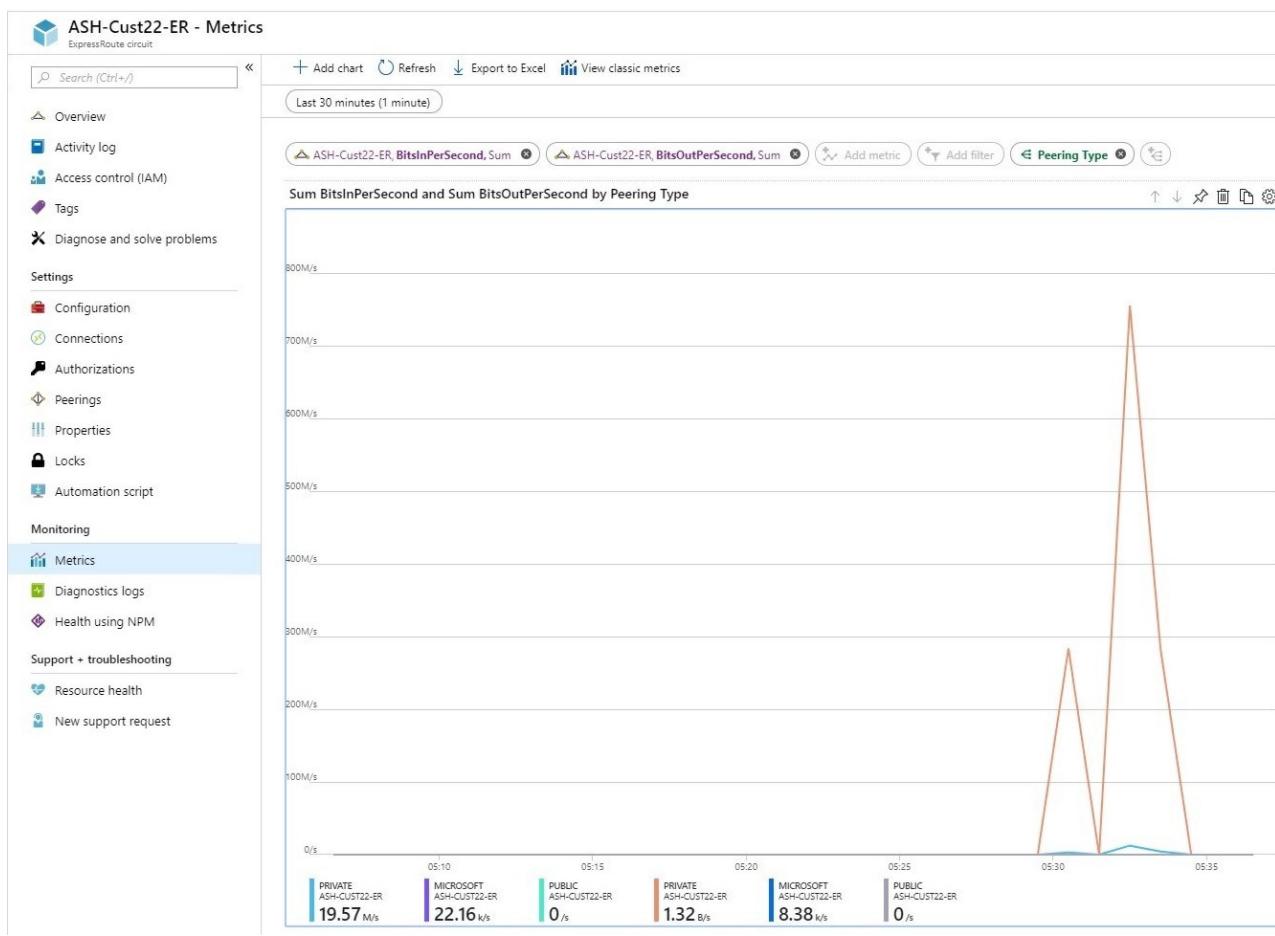
Bits In and Out - Metrics across all peerings

You can view metrics across all peerings on a given ExpressRoute circuit.



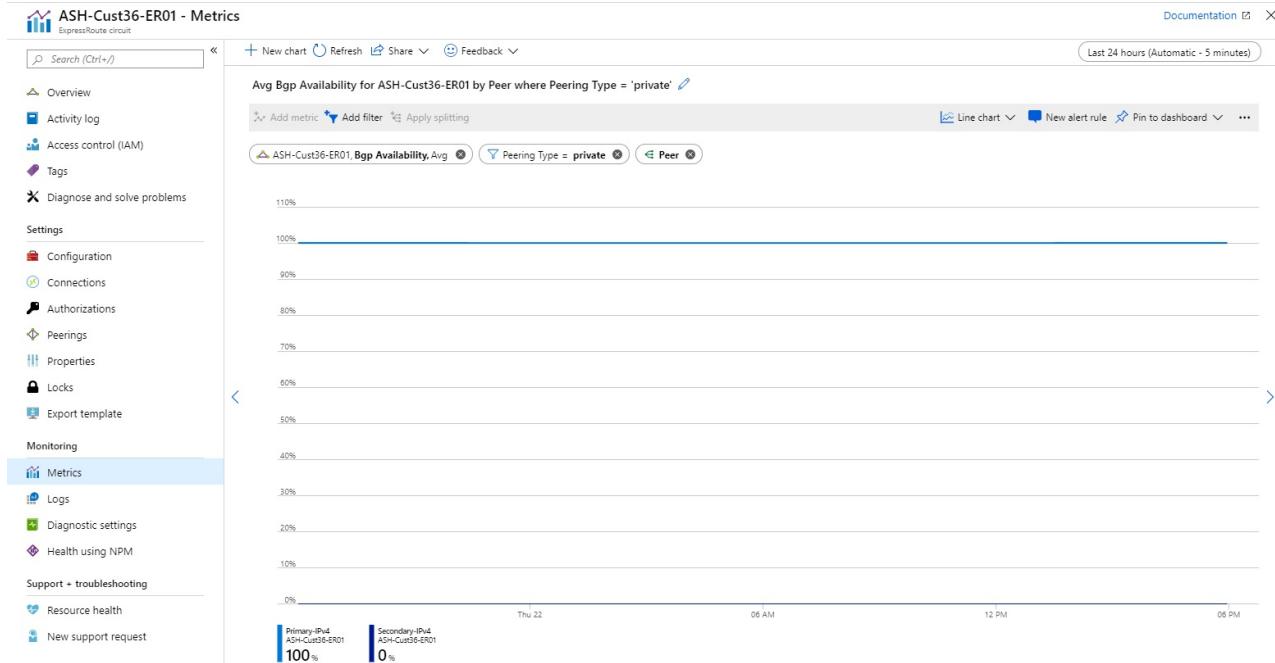
Bits In and Out - Metrics per peering

You can view metrics for private, public, and Microsoft peering in bits/second.



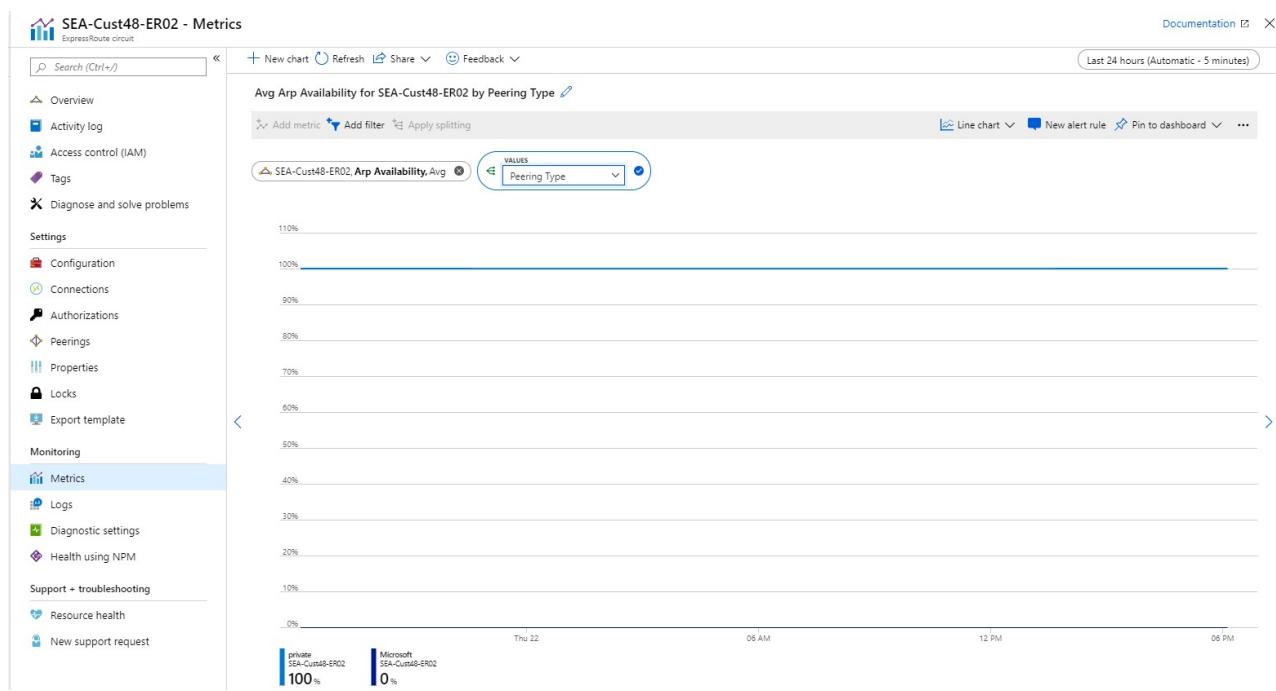
BGP Availability - Split by Peer

You can view near to real-time availability of BGP across peerings and peers (Primary and Secondary ExpressRoute routers). This dashboard shows the Primary BGP session up for private peering and the Second BGP session down for private peering.



ARP Availability - Split by Peering

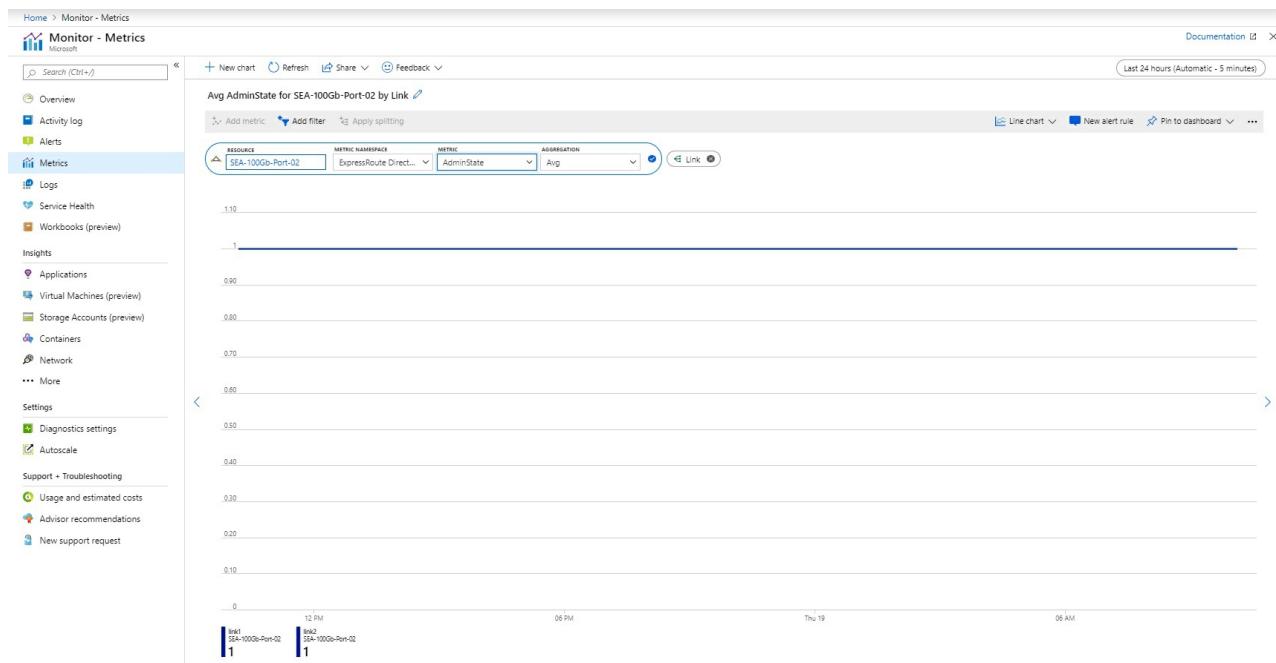
You can view near to real-time availability of ARP across peerings and peers (Primary and Secondary ExpressRoute routers). This dashboard shows the Private Peering ARP session up across both peers, but complete down for Microsoft peering across peerings. The default aggregation (Average) was utilized across both peers.



ExpressRoute Direct Metrics

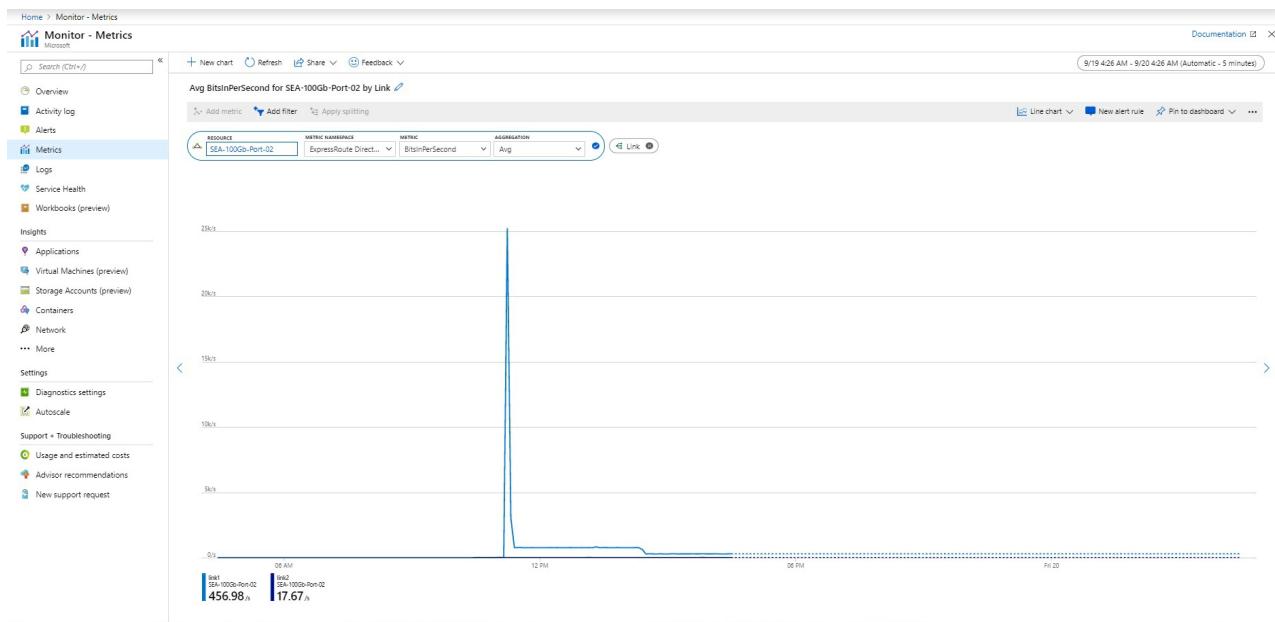
Admin State - Split by link

You can view the admin state for each link of the ExpressRoute Direct port pair.



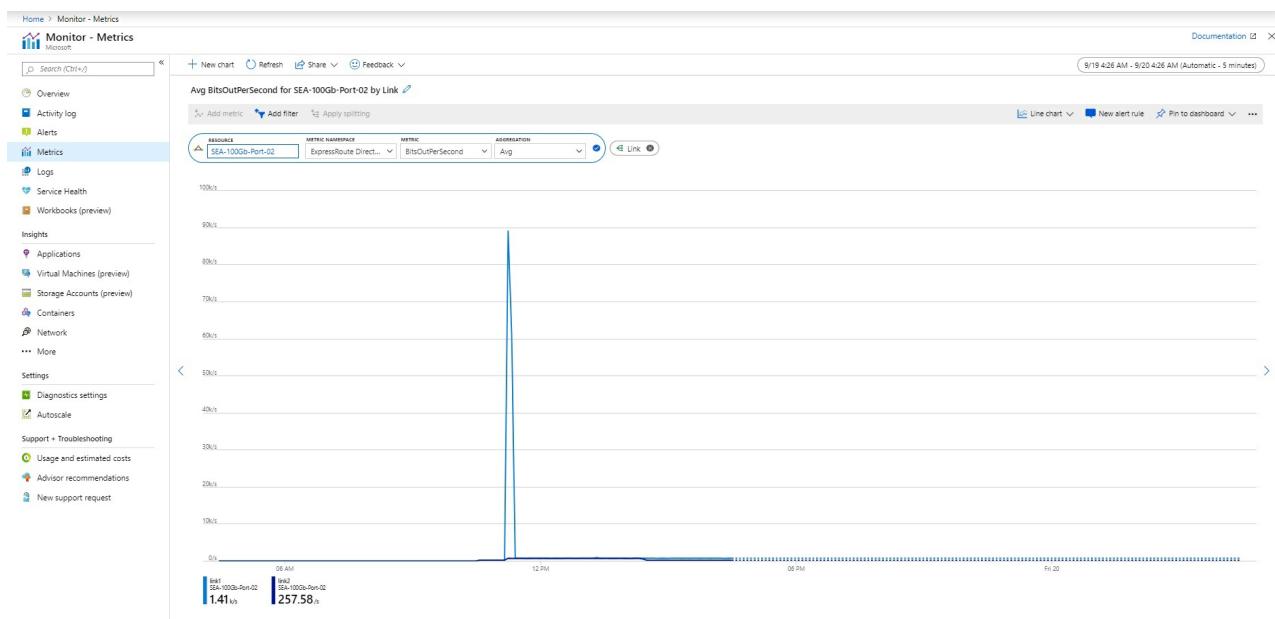
Bits In Per Second - Split by link

You can view the bits in per second across both links of the ExpressRoute Direct port pair.



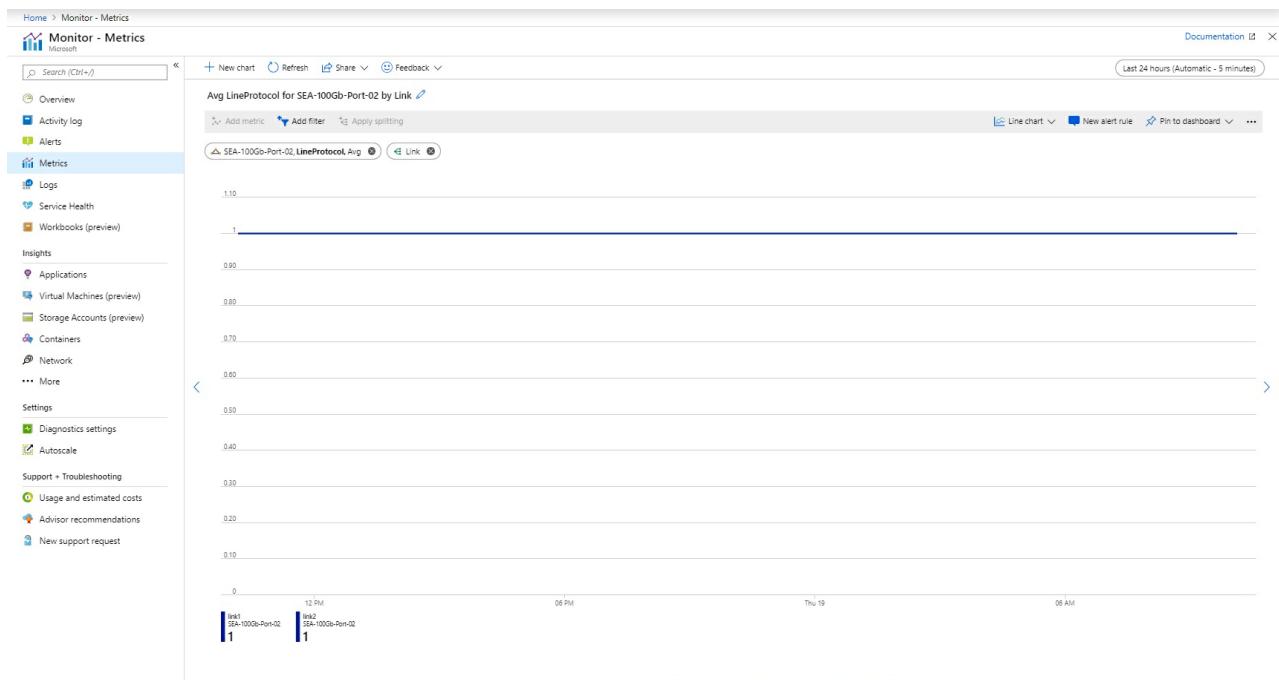
Bits Out Per Second - Split by link

You can also view the bits out per second across both links of the ExpressRoute Direct port pair.



Line Protocol - Split by link

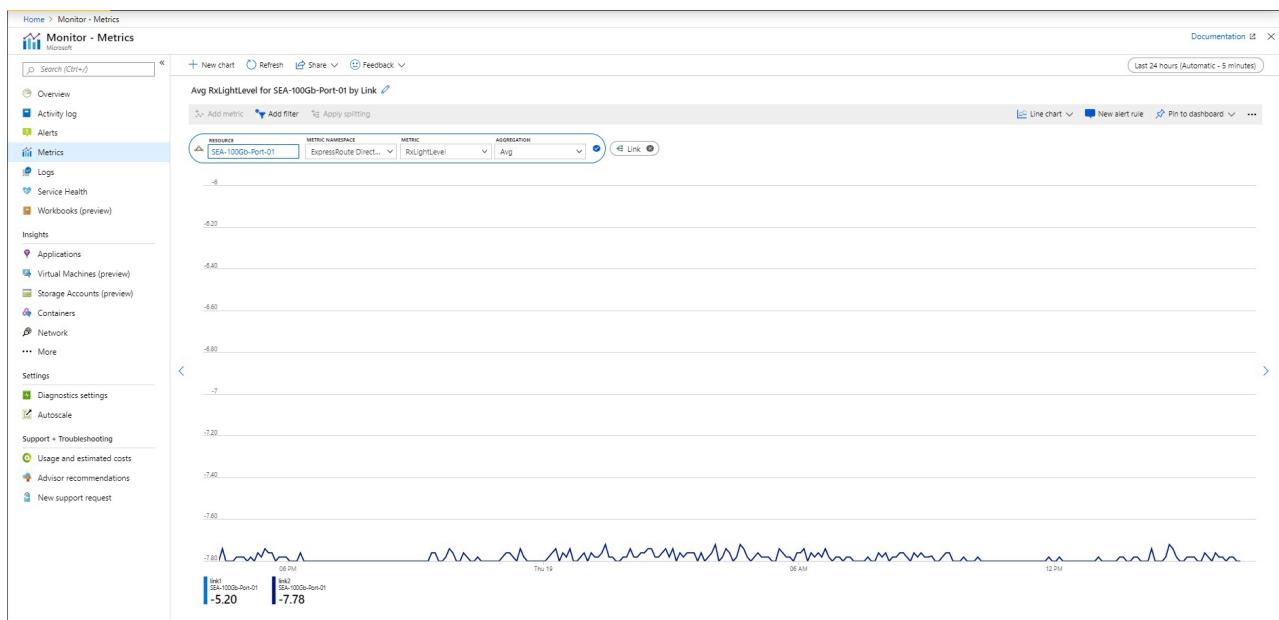
You can view the line protocol across each link of the ExpressRoute Direct port pair.



Rx Light Level - Split by link

You can view the Rx light level (the light level that the ExpressRoute Direct port is **receiving**) for each port.

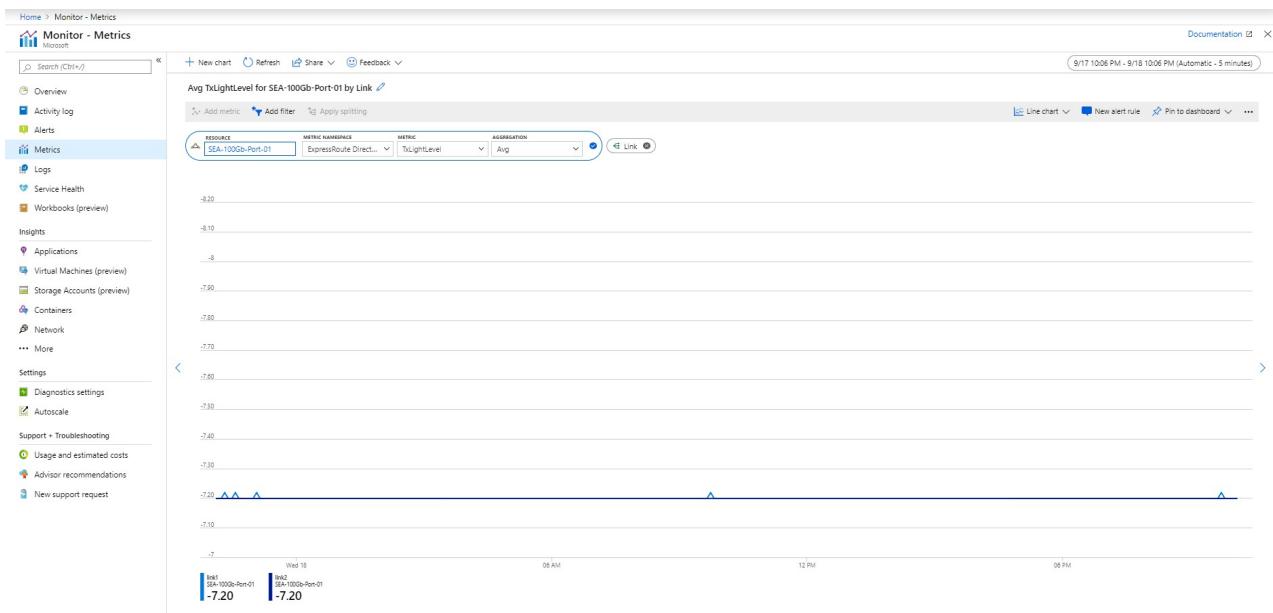
Healthy Rx light levels generally fall within a range of -10 to 0 dBm



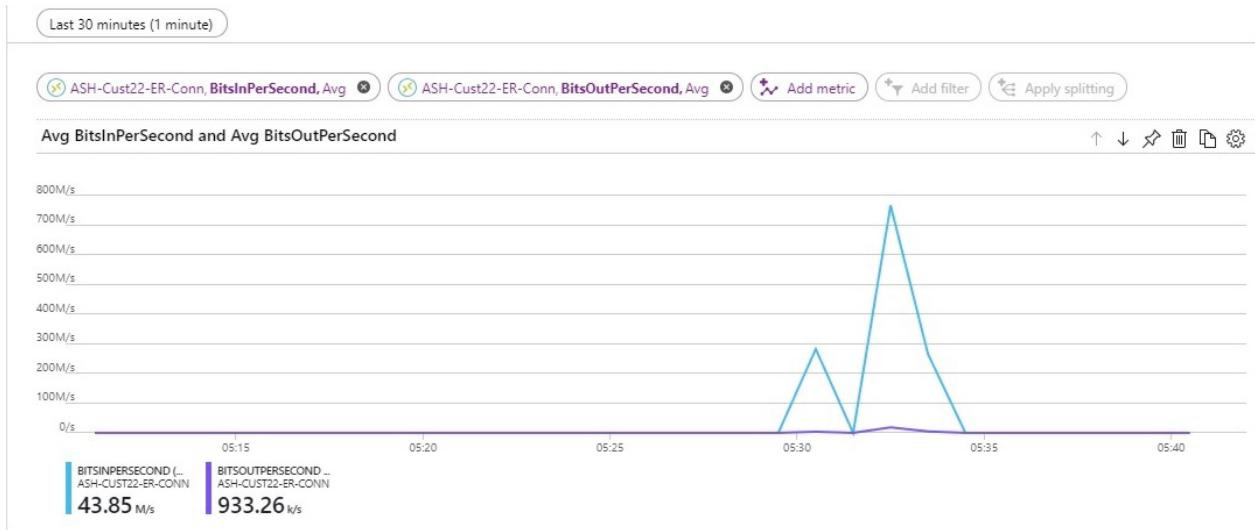
Tx Light Level - Split by link

You can view the Tx light level (the light level that the ExpressRoute Direct port is **transmitting**) for each port.

Healthy Tx light levels generally fall within a range of -10 to 0 dBm



ExpressRoute gateway connections in bits/seconds



Alerts for ExpressRoute gateway connections

1. In order to configure alerts, navigate to **Azure Monitor**, then click **Alerts**.

2. Click **+Select Target** and select the ExpressRoute gateway connection resource.

[Home](#) > [Monitor - Alerts](#) > [Create rule](#)

Create rule

[Rules management](#)

1. Define alert condition

Alert condition configuration requires 1) Target selection and 2) Alert criteria definition where signal(s) and alert logic is configured. Start by selecting a Target.

* Alert target: ASH-Cust22-ER-Conn
Target Hierarchy: ExpressRoute-Lab > ASH-Cust22

+ Select target

* Alert criteria: No criteria defined, click on 'Add criteria' to select a signal and define its logic

+ Add criteria

2. Define alert details

3. Define action group

Configure signal logic

BitsInSecond(Platform)

Show history: Over the last 6 hours | Select time series

0.000.000
0.000.000
0.000.000
0.000.000
0.000.000
0.000.000

0 12 PM 1 PM 2 PM 3 PM 4 PM 5 PM

i When multiple criteria are used, only selecting one value per dimension is supported.

This metric supports dimensions. Selecting the dimension values will help you filter to the right time series. If you do not select any value for a dimension, that dimension will be ignored.

DIMENSION NAME	DIMENSION VALUES
No results	

Alert logic

Condition: Greater than | Time Aggregation: Average | Threshold: 200000 count/second

Condition preview: Whenever the BitsInSecond is Greater than 200000 count/second

Evaluated based on:

Period: Over the last 5 minutes | Frequency: Every 1 Minute

Done

[Create alert rule](#)

3. Define the alert details.

[Home](#) > [Monitor - Alerts](#) > [Create rule](#)

Create rule

[Rules management](#)

1. Define alert condition

Alert condition configuration requires 1) Target selection and 2) Alert criteria definition where signal(s) and alert logic is configured. Start by selecting a Target.

* Alert target: ASH-Cust22-ER-Conn
Target Hierarchy: ExpressRoute-Lab > ASH-Cust22

+ Select target

* Alert criteria: Whenever the BitsInSecond is Greater than 200000 count/second

Monthly cost in USD (Estimated): \$ 0.10

Total: \$ 0.10

+ Add criteria

2. Define alert details

* Alert rule name: Connection traffic over

* Description: Generic description for customer reference

* Severity: Sev 3

Enable rule upon creation: Yes

i It can take up to 10 minutes for a metric alert rule to become active.

3. Define action group

Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more [here](#)

ACTION GROUP NAME	SUBSCRIPTION	ACTION GROUP TYPE	REMOVE
NPM Email ActionGroup	ExpressRoute-Lab	1 Email	

Select action group | + New action group

[Create alert rule](#)

4. Define and add the action group.



* Action group name	ER action
* Short name	ER
* Subscription	ExpressRoute-Lab
* Resource group	ASH-Cust22

Actions

ACTION NAME	ACTION TYPE	STATUS	DETAILS
Unique name for the act...			
Privacy Statement	Email/SMS/Push/Voice		
Pricing	Azure Function		
	LogicApp		
	Webhook		
	ITSM		
	Automation Runbook		

Alerts based on each peering

The screenshot shows the Azure portal interface for creating a new alert rule. On the left, there are three main sections: 'Define alert condition', 'Define alert details', and 'Define action group'. The 'Define alert condition' section is currently active, showing a target selection for 'ASH-Cust22-ER' under 'ExpressRoute-Lab'. Below it, there's a section for 'Alert criteria' which is currently empty. On the right, the 'Configure signal logic' section is displayed. It features a line chart titled 'Configure signal logic' showing a fluctuating signal over a five-hour period from 12 PM to 5 PM. Below the chart, a table lists 'DIMENSION NAME' and 'DIMENSION VALUES' for 'Peering Type' (private) and 'Alert logic' (Condition: Greater than, Time: public). The alert logic table also includes columns for 'Time' (Total) and 'Count/sec'. At the bottom of the right pane, there are 'Done' and 'Next Step' buttons.

Configure alerts for activity logs on circuits

In the **Alert Criteria**, you can select **Activity Log** for the Signal Type and select the Signal.

Create rule

Create rule
Rules management

1. Define alert condition

Alert condition configuration requires 1) Target selection and 2) Alert criteria definition where signal(s) and alert logic is configured. Start by selecting a Target.

Alert target: ASH-Cust22-ER

Target Hierarchy: ExpressRoute-Lab > ASH-Cust22

Select target

Alert criteria: No criteria defined, click on 'Add criteria' to select a signal and define its logic.

Add criteria

2. Define alert details

3. Define action group

Configure signal logic

Define your alert criteria by choosing a signal below and defining your alert condition on the next screen.

All signals (25)

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
All Administrative operations	Activity Log	Administrative
Gets ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Administrative
Create or Update ExpressRouteCircuit (expressR...	Activity Log	Administrative
Join Express Route Circuit (expressRouteCircuits)	Activity Log	Administrative
Deletes ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Administrative
All Security operations	Activity Log	Security
Gets ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Security
Create or Update ExpressRouteCircuit (expressR...	Activity Log	Security
Join Express Route Circuit (expressRouteCircuits)	Activity Log	Security
Deletes ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Security
All Recommendation operations	Activity Log	Recommendation
Gets ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Recommendation
Create or Update ExpressRouteCircuit (expressR...	Activity Log	Recommendation
Join Express Route Circuit (expressRouteCircuits)	Activity Log	Recommendation
Deletes ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Recommendation
All Policy operations	Activity Log	Policy
Gets ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Policy
Create or Update ExpressRouteCircuit (expressR...	Activity Log	Policy
Join Express Route Circuit (expressRouteCircuits)	Activity Log	Policy
Deletes ExpressRouteCircuit (expressRouteCircuits)	Activity Log	Policy
All Autoscale operations	Activity Log	Autoscale

Create alert rule

Done

Next steps

Configure your ExpressRoute connection.

- [Create and modify a circuit](#)
- [Create and modify peering configuration](#)
- [Link a VNet to an ExpressRoute circuit](#)

Configure Network Performance Monitor for ExpressRoute

11/13/2019 • 13 minutes to read • [Edit Online](#)

This article helps you configure a Network Performance Monitor extension to monitor ExpressRoute. Network Performance Monitor (NPM) is a cloud-based network monitoring solution that monitors connectivity between Azure cloud deployments and on-premises locations (Branch offices, etc.). NPM is part of Azure Monitor logs. NPM offers an extension for ExpressRoute that lets you monitor network performance over ExpressRoute circuits that are configured to use private peering or Microsoft peering. When you configure NPM for ExpressRoute, you can detect network issues to identify and eliminate. This service is also available for Azure Government Cloud.

NOTE

This article was recently updated to use the term Azure Monitor logs instead of Log Analytics. Log data is still stored in a Log Analytics workspace and is still collected and analyzed by the same Log Analytics service. We are updating the terminology to better reflect the role of [logs in Azure Monitor](#). See [Azure Monitor terminology changes](#) for details.

You can:

- Monitor loss and latency across various VNets and set alerts
- Monitor all paths (including redundant paths) on the network
- Troubleshoot transient and point-in-time network issues that are difficult to replicate
- Help determine a specific segment on the network that is responsible for degraded performance
- Get throughput per virtual network (If you have agents installed in each VNet)
- See the ExpressRoute system state from a previous point in time

Workflow

Monitoring agents are installed on multiple servers, both on-premises and in Azure. The agents communicate with each other, but do not send data, they send TCP handshake packets. The communication between the agents allows Azure to map the network topology and path the traffic could take.

1. Create an NPM Workspace. This is the same as a Log Analytics workspace.
2. Install and configure software agents. (If you only want to monitor over Microsoft Peering, you do not need to install and configure software agents.):
 - Install monitoring agents on the on-premises servers and the Azure VMs (for private peering).
 - Configure settings on the monitoring agent servers to allow the monitoring agents to communicate. (Open firewall ports, etc.)
3. Configure network security group (NSG) rules to allow the monitoring agent installed on Azure VMs to communicate with on-premises monitoring agents.
4. Set up monitoring: Auto-Discover and manage which networks are visible in NPM.

If you are already using Network Performance Monitor to monitor other objects or services, and you already have Workspace in one of the supported regions, you can skip Step 1 and Step 2, and begin your configuration with Step 3.

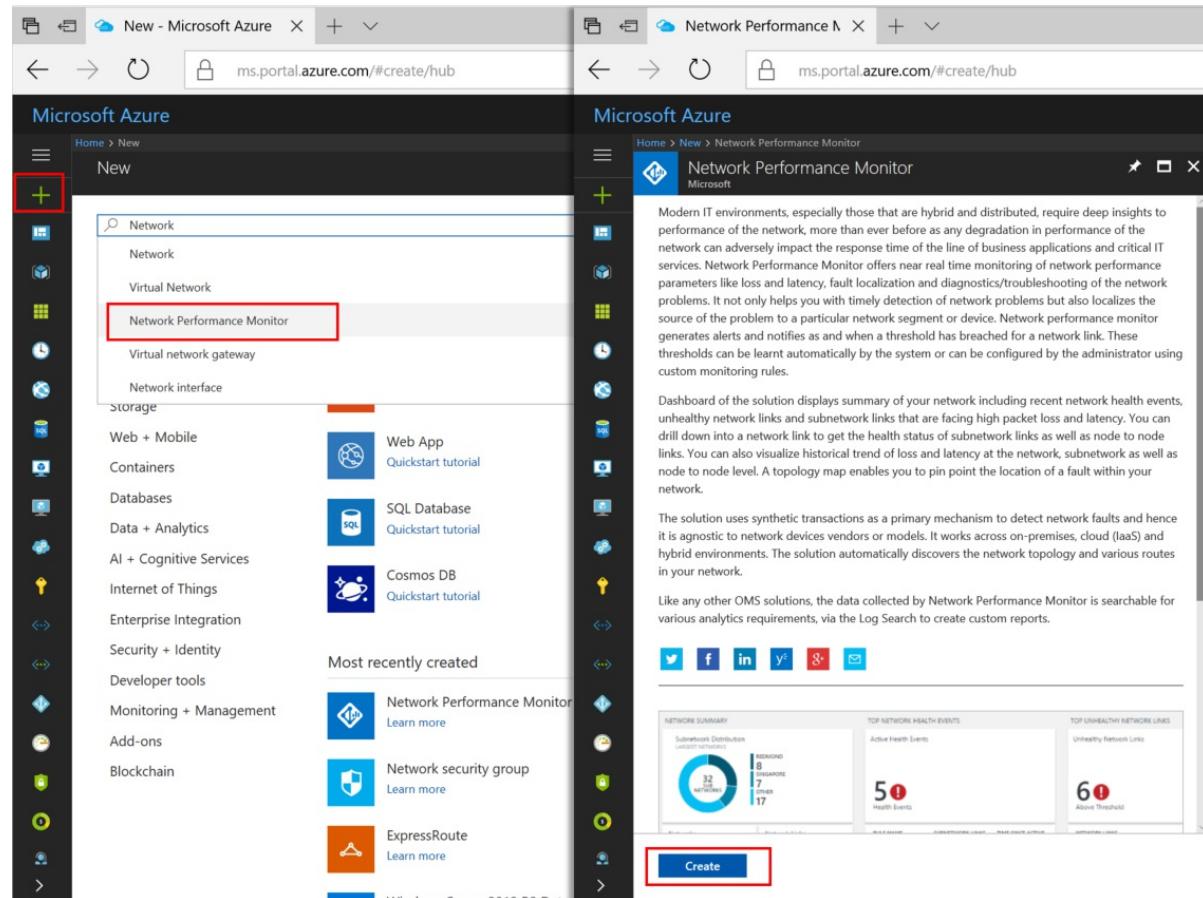
Step 1: Create a Workspace

Create a workspace in the subscription that has the VNets link to the ExpressRoute circuit(s).

1. In the [Azure portal](#), select the Subscription that has the VNets peered to your ExpressRoute circuit. Then, search the list of services in the **Marketplace** for 'Network Performance Monitor'. In the return, click to open the **Network Performance Monitor** page.

NOTE

You can create a new workspace, or use an existing workspace. If you want to use an existing workspace, you must make sure that the workspace has been migrated to the new query language. [More information...](#)



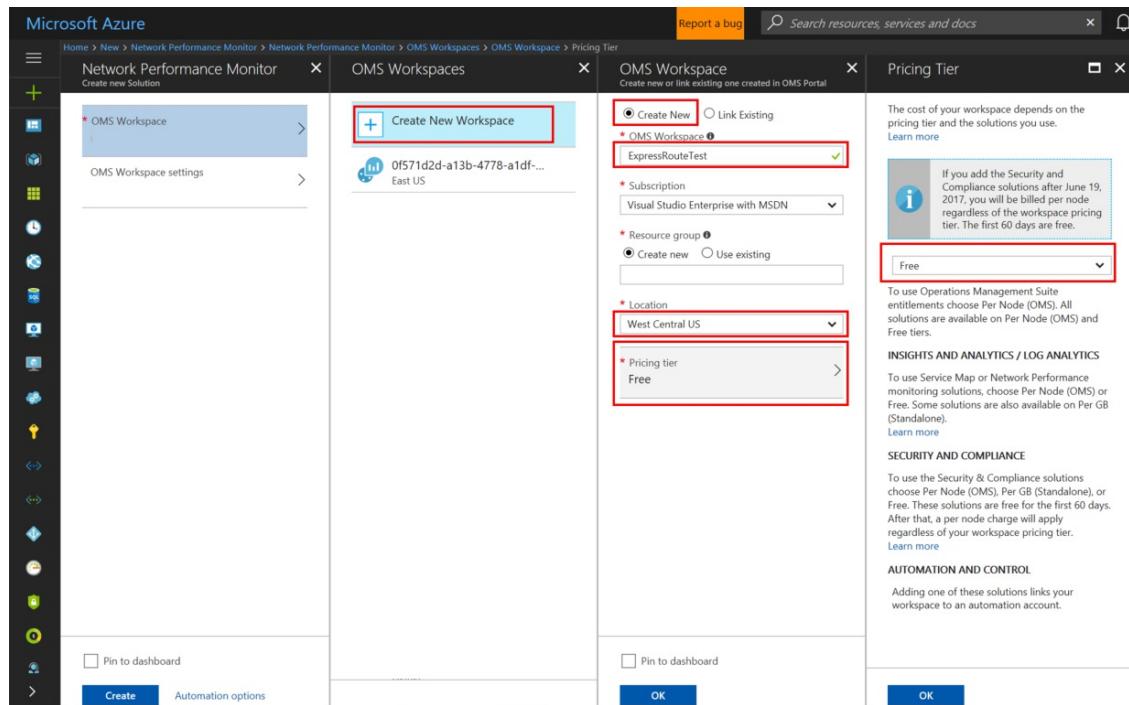
2. At the bottom of the main **Network Performance Monitor** page, click **Create** to open **Network Performance Monitor - Create new solution** page. Click **Log Analytics Workspace - select a workspace** to open the Workspaces page. Click **+ Create New Workspace** to open the Workspace page.

3. On the **Log Analytics workspace** page, select **Create New**, then configure the following settings:

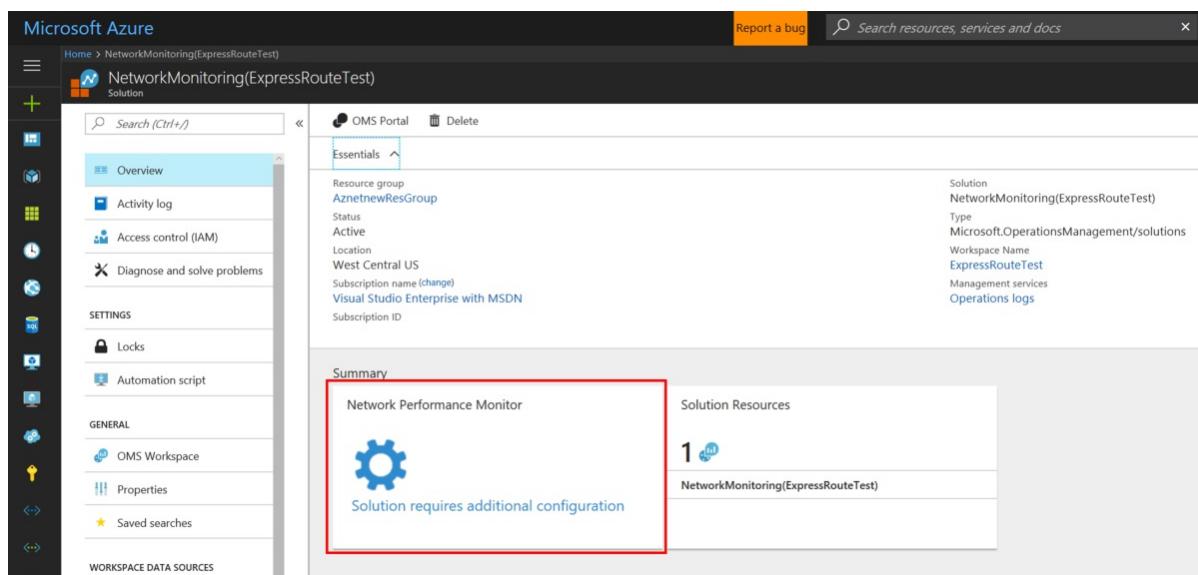
- Log Analytics Workspace - Type a name for your Workspace.
- Subscription - If you have multiple subscriptions, choose the one you want to associate with the new Workspace.
- Resource group - Create a resource group, or use an existing one.
- Location - This location is used to specify the location of the storage account that is used for the agent connection logs.
- Pricing tier - Select the pricing tier.

NOTE

The ExpressRoute circuit can be anywhere in the world. It doesn't have to be in the same region as the Workspace.



4. Click **OK** to save and deploy the settings template. Once the template validates, click **Create** to deploy the Workspace.
5. After the Workspace has been deployed, navigate to the **NetworkMonitoring(name)** resource that you created. Validate the settings, then click **Solution requires additional configuration**.



Step 2: Install and configure agents

2.1: Download the agent setup file

1. Go to the **Common Settings** tab of the **Network Performance Monitor Configuration** page for your resource. Click the agent that corresponds to your server's processor from the **Install Log Analytics Agents** section, and download the setup file.

2. Next, copy the **Workspace ID** and **Primary Key** to Notepad.
3. From the **Configure Log Analytics Agents for monitoring using TCP protocol** section, download the Powershell Script. The PowerShell script helps you open the relevant firewall port for the TCP transactions.

The screenshot shows the 'Network Performance Monitor Configuration' page. In the 'Common Settings' tab, under 'Setup OMS Agents', there's a section for '1. Install OMS Agents'. It includes a note: 'You'll need the Workspace ID and Key to install the agent.' Below this are two download buttons: 'Download Windows Agent (64 bit)' and 'Download Windows Agent (32 bit)'. To the right of these buttons are the 'Workspace ID' and 'Primary Key' fields, both of which are highlighted with red boxes. Other fields visible include 'Secondary Key' and 'NPM for Linux (Enroll for private preview)'. The 'OMS Gateway' section is also partially visible.

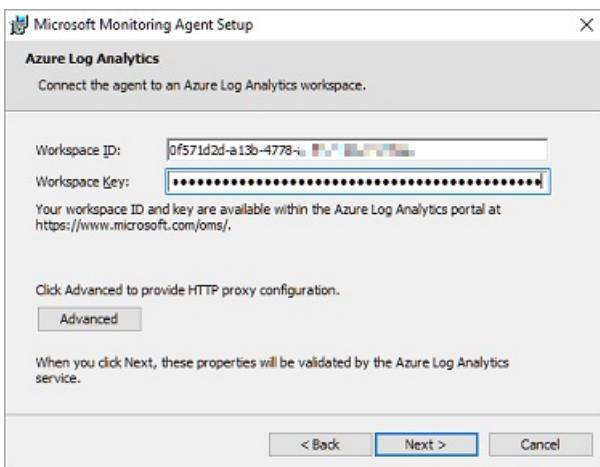
2.2: Install a monitoring agent on each monitoring server (on each VNET that you want to monitor)

We recommend that you install at least two agents on each side of the ExpressRoute connection for redundancy (for example, on-premises, Azure VNETs). The agent must be installed on a Windows Server (2008 SP1 or later). Monitoring ExpressRoute circuits using Windows Desktop OS and Linux OS is not supported. Use the following steps to install agents:

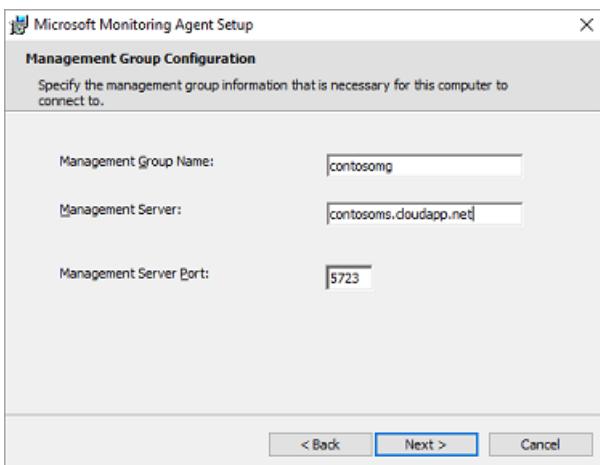
NOTE

Agents pushed by SCOM (includes MMA) may not be able to consistently detect their location if they are hosted in Azure. We recommend that you do not use these agents in Azure VNETs to monitor ExpressRoute.

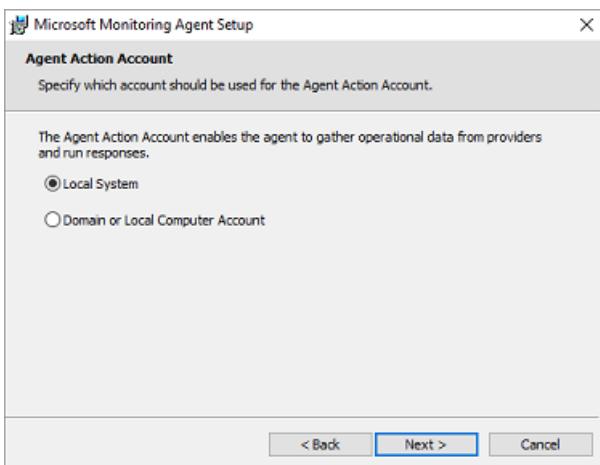
1. Run **Setup** to install the agent on each server that you want to use for monitoring ExpressRoute. The server you use for monitoring can either be a VM, or on-premises, and must have Internet access. You need to install at least one agent on-premises, and one agent on each network segment that you want to monitor in Azure.
2. On the **Welcome** page, click **Next**.
3. On the **License Terms** page, read the license, and then click **I Agree**.
4. On the **Destination Folder** page, change or keep the default installation folder, and then click **Next**.
5. On the **Agent Setup Options** page, you can choose to connect the agent to Azure Monitor logs or Operations Manager. Or, you can leave the choices blank if you want to configure the agent later. After making your selection(s), click **Next**.
 - If you chose to connect to **Azure Log Analytics**, paste the **Workspace ID** and **Workspace Key** (Primary Key) that you copied into Notepad in the previous section. Then, click **Next**.



- If you chose to connect to **Operations Manager**, on the **Management Group Configuration** page, type the **Management Group Name**, **Management Server**, and the **Management Server Port**. Then, click **Next**.



- On the **Agent Action Account** page, choose either the **Local System** account, or **Domain or Local Computer Account**. Then, click **Next**.



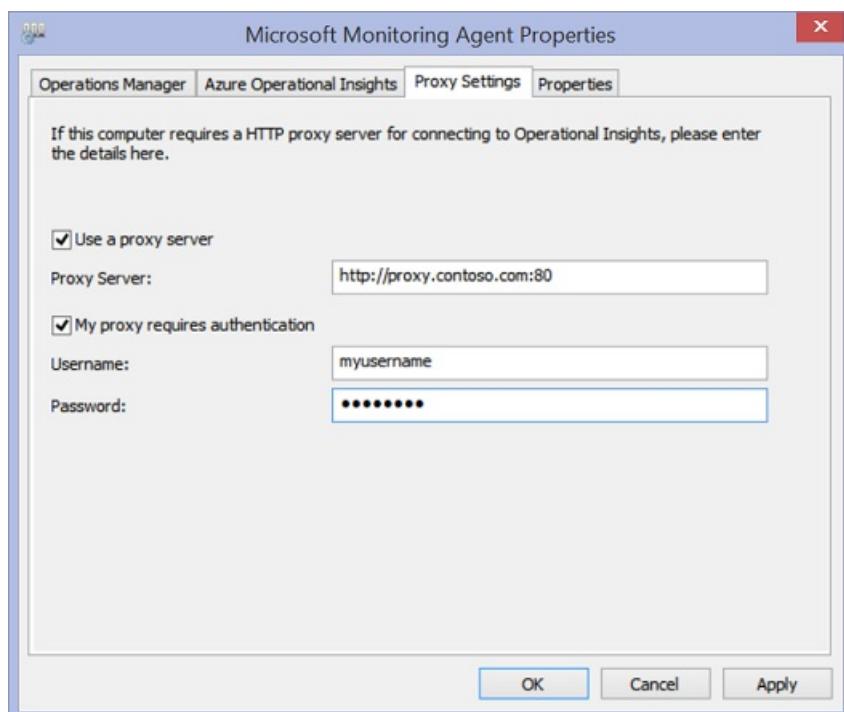
- On the **Ready to Install** page, review your choices, and then click **Install**.
- On the **Configuration completed successfully** page, click **Finish**.
- When complete, the Microsoft Monitoring Agent appears in the Control Panel. You can review your configuration there, and verify that the agent is connected to Azure Monitor logs. When connected, the agent displays a message stating: **The Microsoft Monitoring Agent has successfully connected to the Microsoft Operations Management Suite service.**
- Repeat this procedure for each VNET that you need to be monitored.

2.3: Configure proxy settings (optional)

If you are using a web proxy to access the Internet, use the following steps to configure proxy settings for the Microsoft Monitoring Agent. Perform these steps for each server. If you have many servers that you need to configure, you might find it easier to use a script to automate this process. If so, see [To configure proxy settings for the Microsoft Monitoring Agent using a script](#).

To configure proxy settings for the Microsoft Monitoring Agent using the Control Panel:

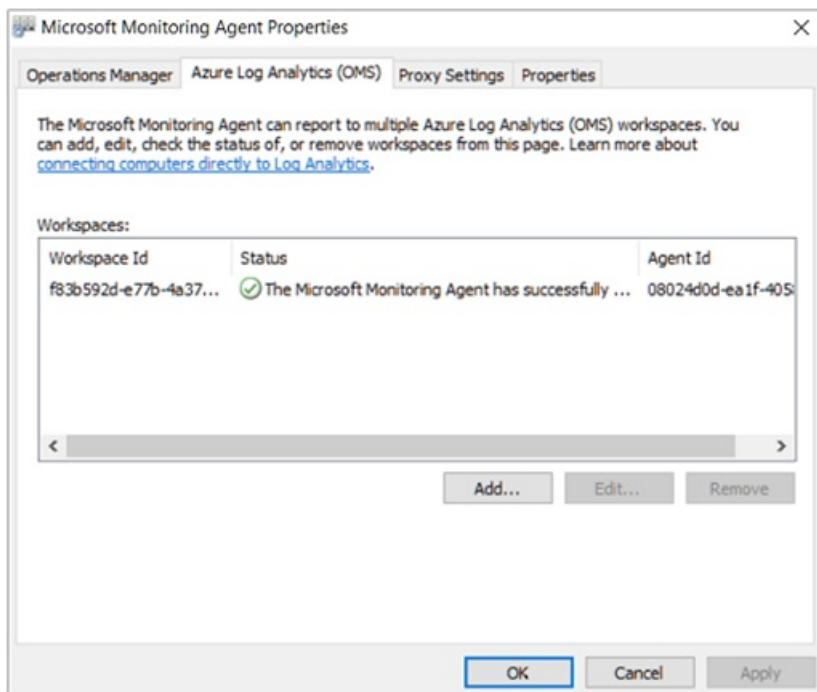
1. Open the **Control Panel**.
2. Open **Microsoft Monitoring Agent**.
3. Click the **Proxy Settings** tab.
4. Select **Use a proxy server** and type the URL and port number, if one is needed. If your proxy server requires authentication, type the username and password to access the proxy server.



2.4: Verify agent connectivity

You can easily verify whether your agents are communicating.

1. On a server with the monitoring agent, open the **Control Panel**.
2. Open the **Microsoft Monitoring Agent**.
3. Click the **Azure Log Analytics** tab.
4. In the **Status** column, you should see that the agent connected successfully to Azure Monitor logs.



2.5: Open the firewall ports on the monitoring agent servers

To use the TCP protocol, you must open firewall ports to ensure that the monitoring agents can communicate.

You can run a PowerShell script to create the registry keys that are required by the Network Performance Monitor. This script also creates the Windows Firewall rules to allow monitoring agents to create TCP connections with each other. The registry keys created by the script specify whether to log the debug logs, and the path for the logs file. It also defines the agent TCP port used for communication. The values for these keys are automatically set by the script. You should not manually change these keys.

Port 8084 is opened by default. You can use a custom port by providing the parameter 'portNumber' to the script. However, if you do so, you must specify the same port for all the servers on which you run the script.

NOTE

The 'EnableRules' PowerShell script configures Windows Firewall rules only on the server where the script is run. If you have a network firewall, you should make sure that it allows traffic destined for the TCP port being used by Network Performance Monitor.

On the agent servers, open a PowerShell window with administrative privileges. Run the [EnableRules](#) PowerShell script (which you downloaded earlier). Don't use any parameters.

A screenshot of a Windows PowerShell window titled 'Administrator: Windows PowerShell'. The command 'PS C:\> ./EnableRules.ps1' is typed into the prompt. The window has standard minimize, maximize, and close buttons at the top right.

Step 3: Configure network security group rules

To monitor agent servers that are in Azure, you must configure network security group (NSG) rules to allow TCP traffic on a port used by NPM for synthetic transactions. The default port is 8084. This allows a monitoring agent installed on an Azure VM to communicate with an on-premises monitoring agent.

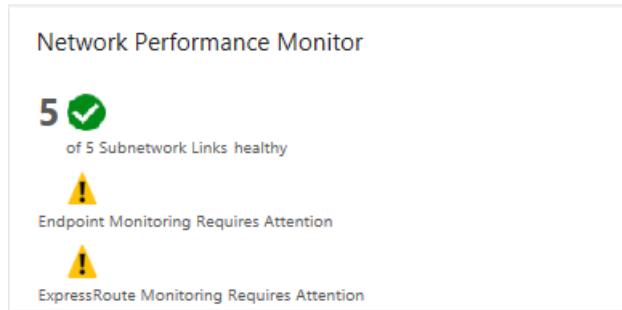
For more information about NSG, see [Network Security Groups](#).

NOTE

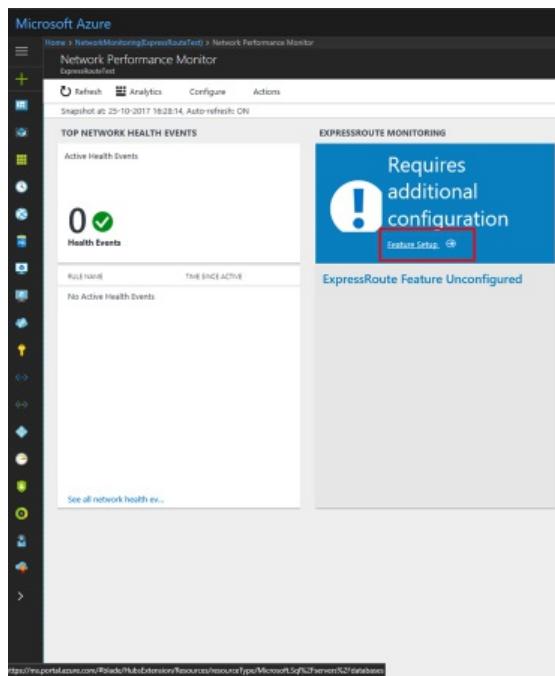
Make sure that you have installed the agents (both the on-premises server agent and the Azure server agent), and have run the PowerShell script before proceeding with this step.

Step 4: Discover peering connections

1. Navigate to the Network Performance Monitor overview tile by going to the **All Resources** page, then click on the whitelisted NPM Workspace.



2. Click the **Network Performance Monitor** overview tile to bring up the dashboard. The dashboard contains an ExpressRoute page, which shows that ExpressRoute is in an 'unconfigured state'. Click **Feature Setup** to open the Network Performance Monitor configuration page.



3. On the configuration page, navigate to the 'ExpressRoute Peerings' tab, located on the left side panel. Next, click **Discover Now**.

TCP SETUP	Discover Now
NETWORKS (3)	<input type="text" value="Search by ExpressRoute Circuit Name"/> X Filter
SUBNETWORKS (21)	
NODES (16) !	No Data
PERFORMANCE MONITOR (4)	
SERVICE ENDPOINT MONITOR (9)	
EXPRESSROUTE PEERINGS	

4. When discovery completes, you will see a list containing the following items:

- All of the Microsoft peering connections in the ExpressRoute circuit(s) that are associated with this subscription.
- All of the private peering connections that connect to the VNets associated with this subscription.

Step 5: Configure monitors

In this section, you configure the monitors. Follow the steps for the type of peering that you want to monitor:

private peering, or **Microsoft peering**.

Private peering

For private peering, when discovery completes, you see will rules for unique **Circuit Name** and **VNet Name**. Initially, these rules are disabled.

The screenshot shows the Azure portal interface for configuring a private peering monitor. The 'PEERING' tab is active, displaying the 'Private' type. The 'CIRCUIT NAME' is set to 'MSIT-CORP-WCUS-Ckt-5', 'PEERING TYPE' is 'Private', and 'VNET NAME' is 'ER-NPM-Test-CORP-WCUS-VN...'. In the 'HEALTH MONITORING' section, three checkboxes are checked: 'Monitor this Peering', 'Enable Health Monitoring for this peering', and 'Loss greater than equal to'. Each checkbox has a text input field for threshold values and a radio button for 'Auto Detect Sudden Changes'. Below the monitoring section, there are 'AZURE AGENTS' and 'ON-PREM AGENTS' sections, each with a 'Add Agents' button.

1. Check the **Monitor this peering** checkbox.
2. Select the checkbox **Enable Health Monitoring for this peering**.
3. Choose the monitoring conditions. You can set custom thresholds to generate health events by typing threshold values. Whenever the value of the condition goes above its selected threshold for the selected network/subnetwork pair, a health event is generated.
4. Click the ON-PREM AGENTS **Add Agents** button to add the on-premises servers from which you want to monitor the private peering connection. Make sure that you only choose agents that have connectivity to the Microsoft service endpoint that you specified in the section for Step 2. The on-premises agents must be able to reach the endpoint using the ExpressRoute connection.
5. Save the settings.
6. After enabling the rules and selecting the values and agents you want to monitor, there is a wait of approximately 30-60 minutes for the values to begin populating and the **ExpressRoute Monitoring** tiles to become available.

Microsoft peering

For Microsoft peering, click the Microsoft peering connection(s) that you want to monitor, and configure the settings.

1. Check the **Monitor this peering** checkbox.
2. (Optional) You can change the target Microsoft service endpoint. By default, NPM chooses a Microsoft service endpoint as the target. NPM monitors connectivity from your on-premises servers to this target endpoint through ExpressRoute.
 - To change this target endpoint, click the **(edit)** link under **Target:**, and select another Microsoft service target endpoint from the list of URLs.

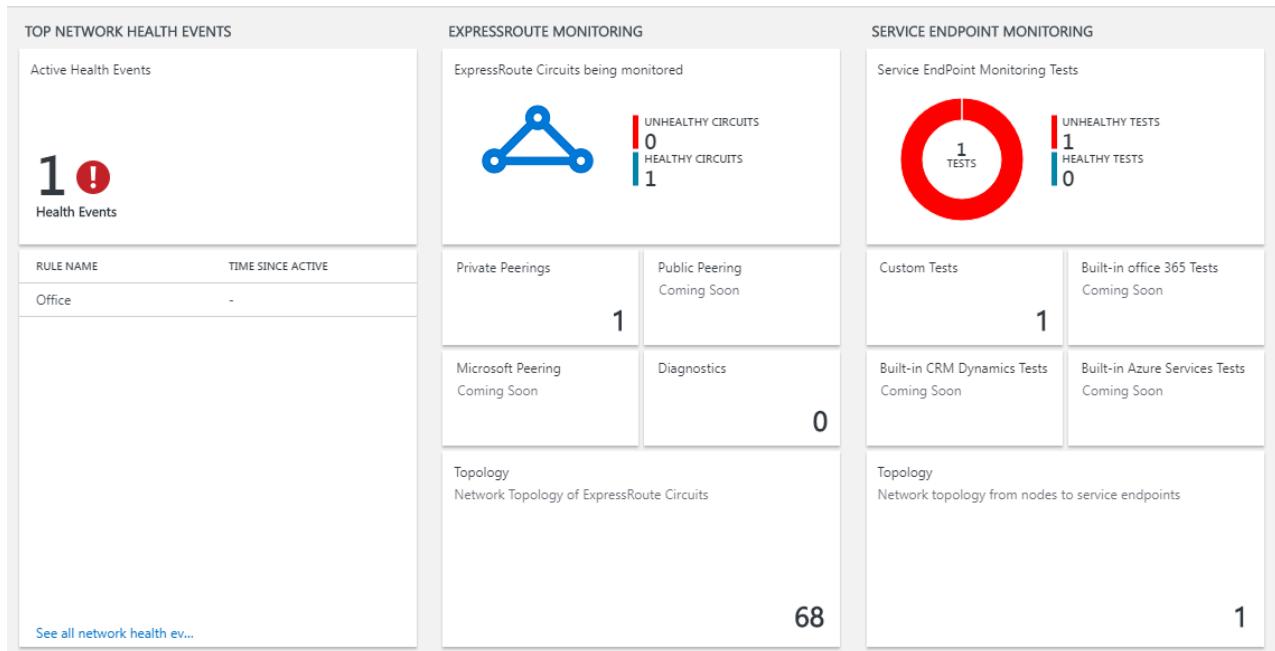
- You can use a custom URL or IP Address. This option is particularly relevant if you are using Microsoft peering to establish a connection to Azure PaaS services, such as Azure Storage, SQL databases, and Websites that are offered on public IP addresses. To do this, click the link (**Use custom URL or IP Address instead**) at the bottom of the URL list, then enter the public endpoint of your Azure PaaS service that is connected through the ExpressRoute Microsoft peering.

- If you are using these optional settings, make sure that only the Microsoft service endpoint is selected here. The endpoint must be connected to ExpressRoute and reachable by the on-premises agents.
- Select the checkbox **Enable Health Monitoring for this peering**.
 - Choose the monitoring conditions. You can set custom thresholds to generate health events by typing threshold values. Whenever the value of the condition goes above its selected threshold for the selected network/subnetwork pair, a health event is generated.
 - Click the ON-PREM AGENTS **Add Agents** button to add the on-premises servers from which you want to monitor the Microsoft peering connection. Make sure that you only choose agents that have connectivity to the Microsoft service endpoints that you specified in the section for Step 2. The on-premises agents must be able to reach the endpoint using the ExpressRoute connection.
 - Save the settings.
 - After enabling the rules and selecting the values and agents you want to monitor, there is a wait of approximately 30-60 minutes for the values to begin populating and the **ExpressRoute Monitoring** tiles to

become available.

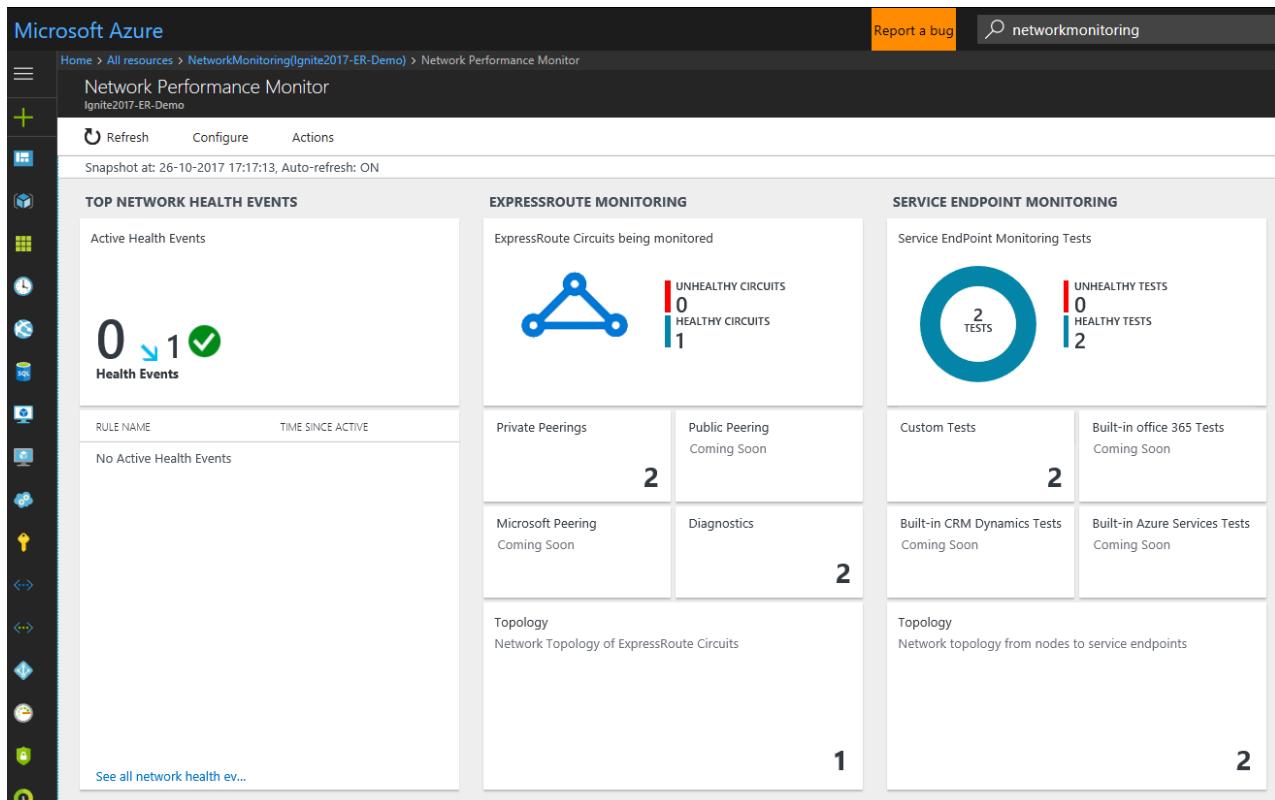
Step 6: View monitoring tiles

Once you see the monitoring tiles, your ExpressRoute circuits and connection resources are being monitored by NPM. You can click on Microsoft Peering tile to drill down on the health of Microsoft Peering connections.



Network Performance Monitor page

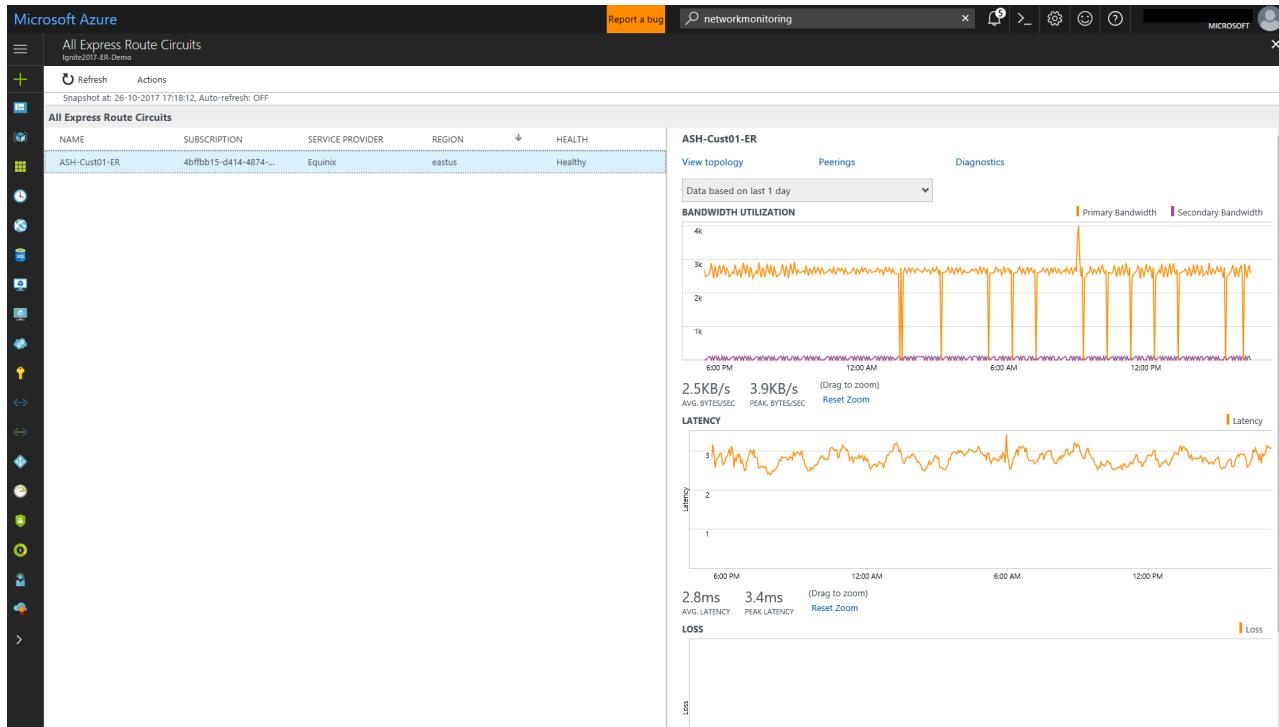
The NPM page contains a page for ExpressRoute that shows an overview of the health of ExpressRoute circuits and peerings.



List of circuits

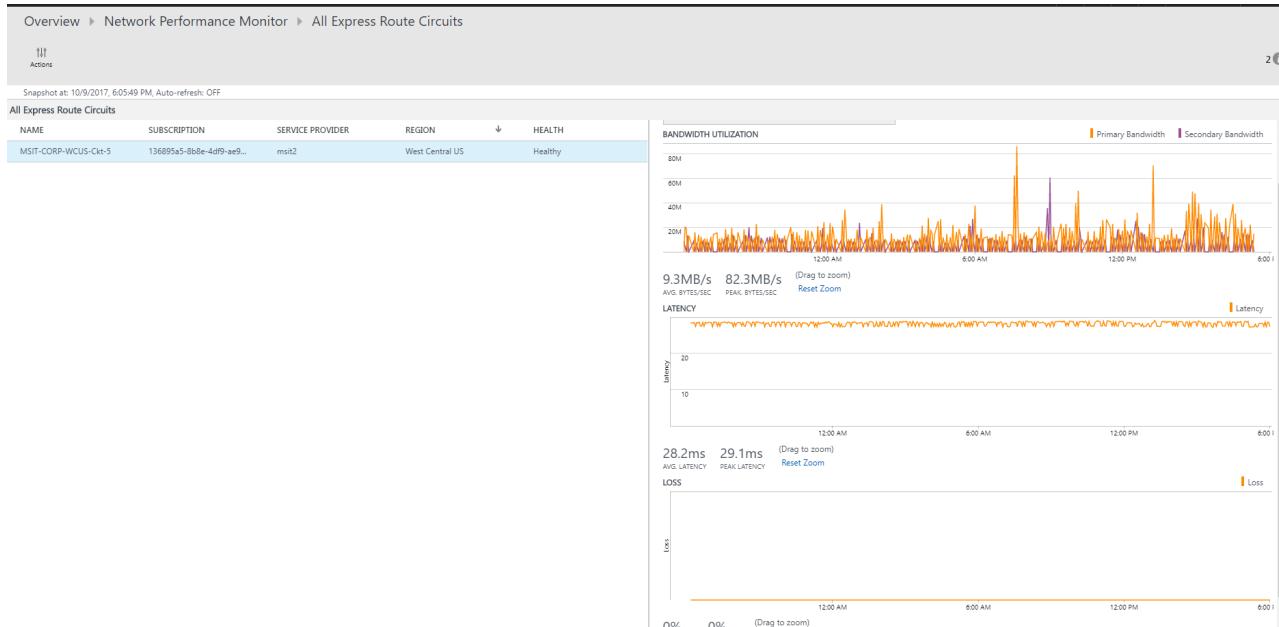
To view a list of all monitored ExpressRoute circuits, click the **ExpressRoute circuits** tile. You can select a circuit and view its health state, trend charts for packet loss, bandwidth utilization, and latency. The charts are interactive. You can select a custom time window for plotting the charts. You can drag the mouse over an area on the chart to

zoom in and see fine-grained data points.



Trend of Loss, Latency, and Throughput

The bandwidth, latency, and loss charts are interactive. You can zoom into any section of these charts, using mouse controls. You can also see the bandwidth, latency, and loss data for other intervals by clicking **Date/Time**, located below the Actions button on the upper left.



Peerings list

To view list of all connections to virtual networks over private peering, click the **Private Peerings** tile on the dashboard. Here, you can select a virtual network connection and view its health state, trend charts for packet loss, bandwidth utilization, and latency.

The screenshot shows the Microsoft Azure Network Performance Monitor interface. On the left, there's a sidebar with various icons. The main area displays 'All Express Route Peerings' for the connection 'ASHI-Cust01-ER'. It lists two entries: 'AzurePrivatePeer...' and 'AzurePrivatePeer...'. Below this, there's a table with columns: NAME, SUBSCRIPTI..., VNET, PEERING TYPE, LOCATION, and HEALTH. The table shows two rows, both marked as 'Healthy'. To the right, there are three charts: 'BANDWIDTH UTILIZATION' (Primary Bandwidth in orange, Secondary Bandwidth in purple), 'LATENCY' (Latency in ms from 1 to 3 ms), and 'LOSS' (Loss percentage from 0% to 1%).

Nodes view

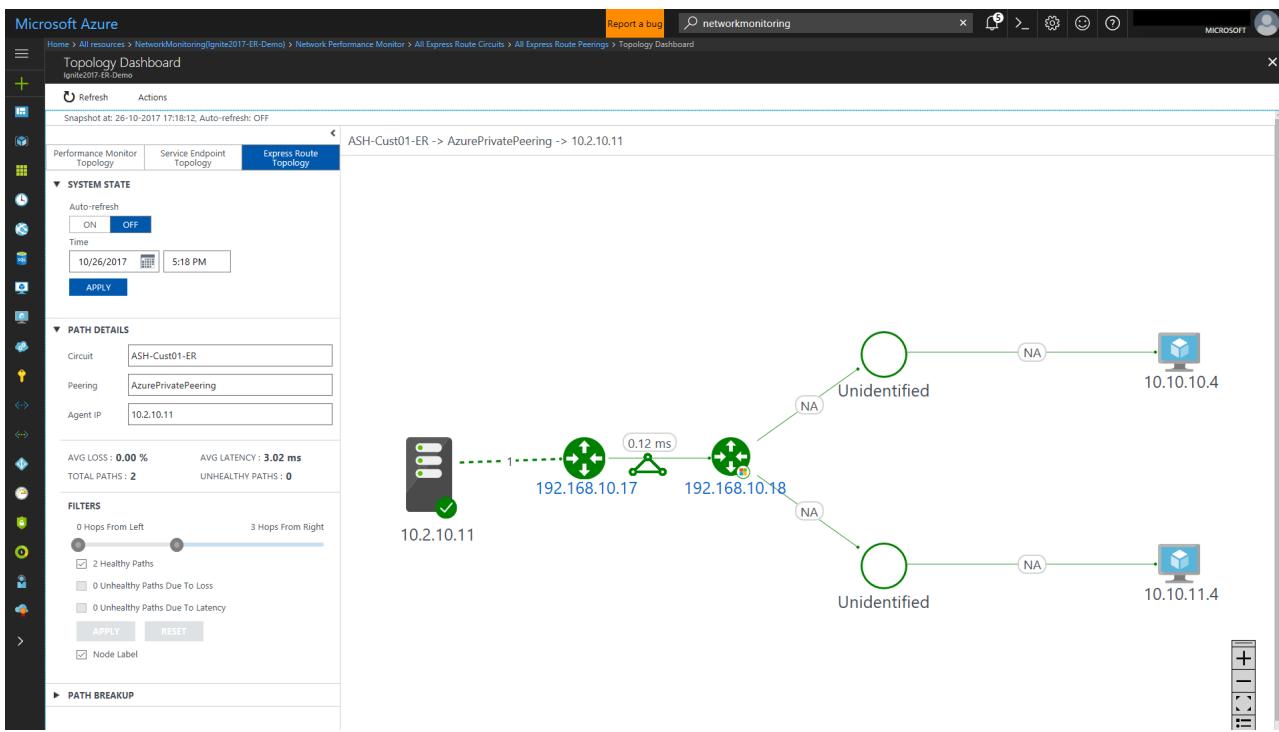
To view list of all the links between the on-premises nodes and Azure VMs/Microsoft service endpoints for the chosen ExpressRoute peering connection, click **View node links**. You can view the health status of each link, as well as the trend of loss and latency associated with them.

This screenshot shows the 'All ExpressRoute Node Links' section. It lists two links: 'LabVM02' to 'myVM02' and 'myVM02' to 'LabVM02'. Each entry shows the IP interfaces involved and their respective loss and latency values. To the right, there are two charts: 'LOSS' (Loss percentage from 0% to 1%) and 'LATENCY' (Latency in ms from 2 to 3 ms).

Circuit topology

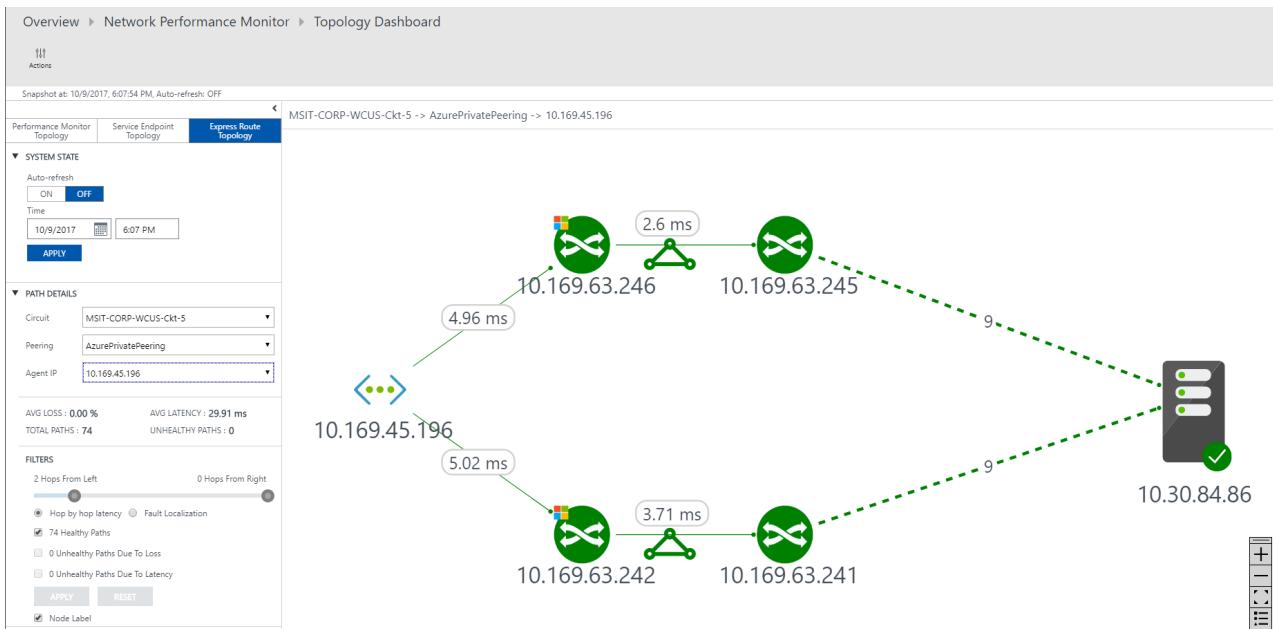
To view circuit topology, click the **Topology** tile. This takes you to the topology view of the selected circuit or peering. The topology diagram provides the latency for each segment on the network. Each layer 3 hop is represented by a node of the diagram. Clicking on a hop reveals more details about the hop.

You can increase the level of visibility to include on-premises hops by moving the slider bar below **Filters**. Moving the slider bar to the left or right, increases/decreases the number of hops in the topology graph. The latency across each segment is visible, which allows for faster isolation of high latency segments on your network.



Detailed Topology view of a circuit

This view shows VNet connections.



Modify an ExpressRoute circuit using PowerShell (classic)

12/8/2019 • 5 minutes to read • [Edit Online](#)

This article walks you through the steps to check the status, update, or delete and deprovision your ExpressRoute classic deployment model circuit. This article applies to the classic deployment model.

IMPORTANT

As of March 1, 2017, you can't create new ExpressRoute circuits in the classic deployment model.

- You can move an existing ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model without experiencing any connectivity down time. For more information, see [Move an existing circuit](#).
- You can connect to virtual networks in the classic deployment model by setting **allowClassicOperations** to TRUE.

Use the following links to create and manage ExpressRoute circuits in the Resource Manager deployment model:

- [Create and manage ExpressRoute circuits](#)
- [Configure routing \(peering\) for ExpressRoute circuits](#)

About Azure deployment models

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

Before you begin

Install the latest versions of the Azure Service Management (SM) PowerShell modules and the ExpressRoute module. You can't use the Azure CloudShell environment to run SM modules.

1. Use the instructions in the [Installing the Service Management module](#) article to install the Azure Service Management Module. If you have the Az or RM module already installed, be sure to use '-AllowClobber'.
2. Import the installed modules. When using the following example, adjust the path to reflect the location and version of your installed PowerShell modules.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\Azure.psd1'
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\ExpressRoute\ExpressRoute.psd1'
```

3. To sign in to your Azure account, open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect using the Service Management module:

```
Add-AzureAccount
```

Get the status of a 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 example:

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

Modify a circuit

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

You can do the following tasks with no downtime:

- Enable or disable an ExpressRoute premium add-on for your ExpressRoute circuit.
- Increase the bandwidth of your ExpressRoute circuit provided there is capacity available on the port. Downgrading the bandwidth of a circuit is not supported.
- Change the metering plan from Metered Data to Unlimited Data. 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.

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. As soon as the command has been successfully run, billing for the premium add-on capability begins.

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.

Considerations

- Make sure 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 fails, and you are billed the premium rates.
- You must unlink all virtual networks in other geopolitical regions. If you don't, your update request fails, and you are 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 drops and won't be reenabled until the number of advertised prefixes goes below 4,000.

To disable the premium add-on

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
```

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 may have to recreate the ExpressRoute circuit if there is inadequate capacity on the existing port. You cannot upgrade the circuit if there is no additional capacity available at that location.

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.

Resize a 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
```

Once your circuit has been sized up on the Microsoft side, you must contact your connectivity provider to update configurations on their side to match this change. Billing begins 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 must delete this circuit and create a new circuit of the size you need.

```
Set-AzureDedicatedCircuitProperties : InvalidOperationException : 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
```

Deprovision and delete a circuit

Considerations

- 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 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 stops billing for the circuit.

Delete a circuit

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

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


Create and modify peering for an ExpressRoute circuit (classic)

12/8/2019 • 10 minutes to read • [Edit Online](#)

This article walks you through the steps to create and manage peering/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. 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.

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

IMPORTANT

As of March 1, 2017, you can't create new ExpressRoute circuits in the classic deployment model.

- You can move an existing ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model without experiencing any connectivity down time. For more information, see [Move an existing circuit](#).
- You can connect to virtual networks in the classic deployment model by setting **allowClassicOperations** to TRUE.

Use the following links to create and manage ExpressRoute circuits in the Resource Manager deployment model:

- [Create and manage ExpressRoute circuits](#)
- [Configure routing \(peering\) for ExpressRoute circuits](#)

About Azure deployment models

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

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.

Download the latest PowerShell cmdlets

Install the latest versions of the Azure Service Management (SM) PowerShell modules and the ExpressRoute module. You can't use the Azure CloudShell environment to run SM modules.

1. Use the instructions in the [Installing the Service Management module](#) article to install the Azure Service Management Module. If you have the Az or RM module already installed, be sure to use '-AllowClobber'.
2. Import the installed modules. When using the following example, adjust the path to reflect the location

and version of your installed PowerShell modules.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\Azure.psd1'  
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\ExpressRoute\ExpressRoute.psd1'
```

3. To sign in to your Azure account, open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect using the Service Management module:

```
Add-AzureAccount
```

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. 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.

2. Check the ExpressRoute circuit to make sure it is provisioned.

Check to see if the ExpressRoute circuit is Provisioned and also Enabled.

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

Return:

```
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 isn't, work with your connectivity provider to get your circuit to the required state and status.

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

3. 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. Verify 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. Verify that you are not using 65515.
- An MD5 hash if you choose to use one. **Optional**.

You can use the following example 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
```

If you want to use an MD5 hash, use the following example to configure 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 -  
SharedKey "A1B2C3D4"
```

IMPORTANT

Verify that you specify your AS number as peering ASN, not customer ASN.

To view Azure private peering details

You can view configuration details using the following cmdlet:

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

Return:

```
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 following example, 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. You must make sure 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.

NOTE

Azure public peering is deprecated for new circuits.

To create Azure public peering

1. 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.

2. Check ExpressRoute circuit to verify that it is provisioned

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

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

Return:

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

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

ServiceProviderProvisioningState	:	Provisioned
Status	:	Enabled

3. Configure Azure public 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.
- A /30 subnet for the secondary link. This must be a valid public IPv4 prefix.
- A valid VLAN ID to establish this peering on. Verify 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. **Optional**.

IMPORTANT

Make sure that you specify your AS number as peering ASN and not customer ASN.

You can use the following example 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
```

If you want to use an MD5 hash, use the following example to configure your circuit:

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

To view Azure public peering details

To view configuration details, use the following cmdlet:

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

Return:

```
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. In this example, the VLAN ID of the circuit is being updated from 200 to 600.

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

Verify that the circuit shows as Provisioned and Enabled.

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. 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.

2. Check ExpressRoute circuit to verify that it is provisioned

Verify that the circuit shows as Provisioned and Enabled.

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

Return:

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

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

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

3. 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. Verify 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. **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. **Optional**.

Run the following cmdlet to configure Microsoft peering 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 view configuration details using the following cmdlet:

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

Return:

```
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 using PowerShell (classic)

12/8/2019 • 5 minutes to read • [Edit Online](#)

This article will help you link virtual networks (VNets) to Azure ExpressRoute circuits using PowerShell. A single VNet can be linked to up to four ExpressRoute circuits. Use the steps in this article to create a new link to each ExpressRoute circuit you are connecting to. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both. This article applies to virtual networks created using the classic deployment model.

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 enable the ExpressRoute premium add-on. Check the [FAQ](#) for more details about the premium add-on.

IMPORTANT

As of March 1, 2017, you can't create new ExpressRoute circuits in the classic deployment model.

- You can move an existing ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model without experiencing any connectivity down time. For more information, see [Move an existing circuit](#).
- You can connect to virtual networks in the classic deployment model by setting **allowClassicOperations** to TRUE.

Use the following links to create and manage ExpressRoute circuits in the Resource Manager deployment model:

- [Create and manage ExpressRoute circuits](#)
- [Configure routing \(peering\) for ExpressRoute circuits](#)

About Azure deployment models

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

Configuration prerequisites

- Review the [prerequisites](#), [routing requirements](#), and [workflows](#) before you begin configuration.
- You must have an active ExpressRoute circuit.
 - Follow the instructions to [create an ExpressRoute circuit](#) and have 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](#).

[Download the latest PowerShell cmdlets](#)

Install the latest versions of the Azure Service Management (SM) PowerShell modules and the ExpressRoute module. You can't use the Azure CloudShell environment to run SM modules.

1. Use the instructions in the [Installing the Service Management module](#) article to install the Azure Service Management Module. If you have the Az or RM module already installed, be sure to use '-AllowClobber'.
2. Import the installed modules. When using the following example, adjust the path to reflect the location and version of your installed PowerShell modules.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\Azure.psd1'  
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\ExpressRoute\ExpressRoute.psd1'
```

3. To sign in to your Azure account, open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect using the Service Management module:

```
Add-AzureAccount
```

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
```

Remove a virtual network link to a circuit

You can remove a virtual network link to an ExpressRoute circuit by using the following cmdlet. Make sure that the current subscription is selected for the given virtual network.

```
Remove-AzureDedicatedCircuitLink -ServiceKey "*****" -VNetName "MyVNet"
```

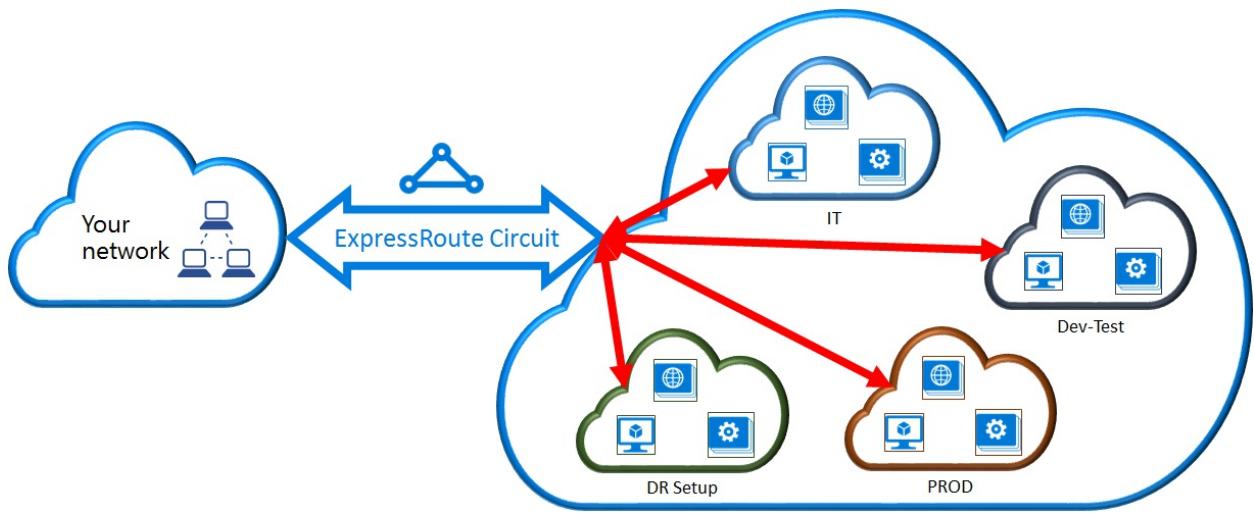
Connect a virtual network in a different subscription to a circuit

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'
```

Return:

```
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: "*****"
```

Return:

```
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
```

Return:

```
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
```

Return:

```
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'
```

Return:

```
State VnetName
-----
Provisioned SalesVNET1
```

Run this command in the newly linked subscription for the virtual network:

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

Next steps

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

Configure ExpressRoute and Site-to-Site coexisting connections (classic)

12/8/2019 • 9 minutes to read • [Edit Online](#)

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

IMPORTANT

As of March 1, 2017, you can't create new ExpressRoute circuits in the classic deployment model.

- You can move an existing ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model without experiencing any connectivity down time. For more information, see [Move an existing circuit](#).
- You can connect to virtual networks in the classic deployment model by setting **allowClassicOperations** to TRUE.

Use the following links to create and manage ExpressRoute circuits in the Resource Manager deployment model:

- [Create and manage ExpressRoute circuits](#)
- [Configure routing \(peering\) for ExpressRoute circuits](#)

About Azure deployment models

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

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.

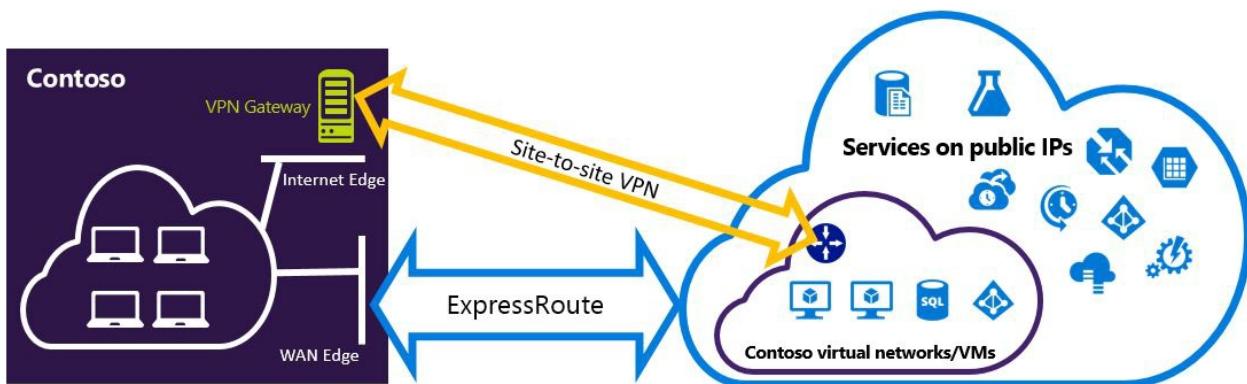
Configuration designs

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.

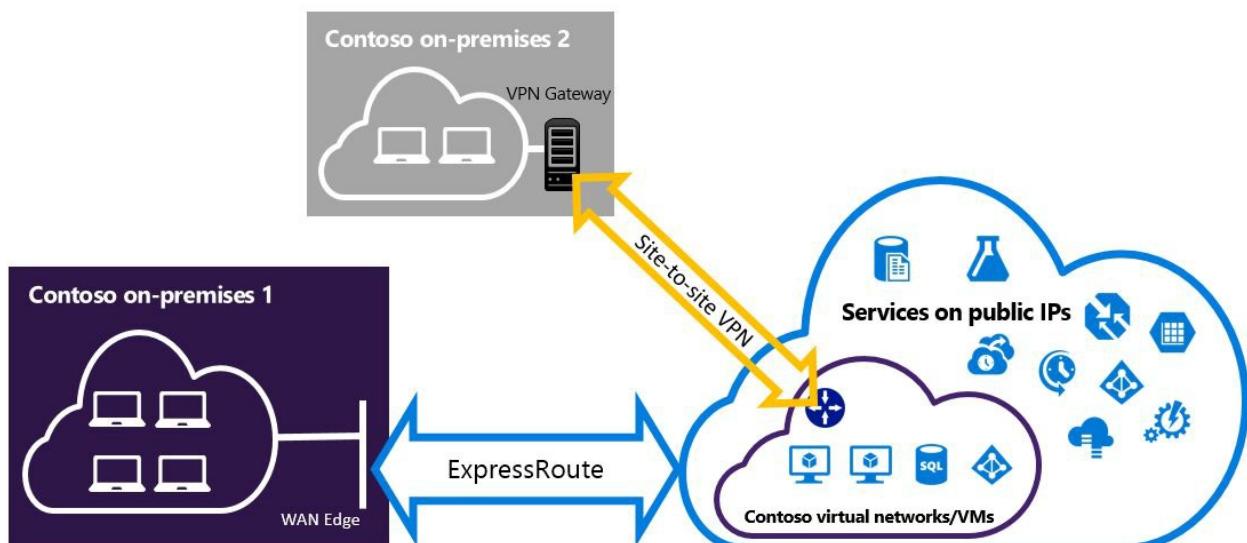
NOTE

While ExpressRoute circuit is preferred over Site-to-Site VPN when both routes are the same, Azure will use the longest prefix match to choose the route towards the packet's destination.



Configure a Site-to-Site VPN to connect to sites not connected through ExpressRoute

You can configure your network where some sites connect directly to Azure over Site-to-Site VPN, and some sites connect through ExpressRoute.



NOTE

You cannot configure a virtual network as a transit router.

Selecting the steps to use

There are two different sets of procedures to choose from in order to configure connections that can coexist. The configuration procedure that you select will depend on whether you have an existing virtual network that you want to connect to, or you want to create a new virtual network.

- I don't have a VNet and need to create one.

If you don't already have a virtual network, this procedure will walk you through creating a new virtual network using the classic deployment model and creating new ExpressRoute and Site-to-Site VPN connections. To configure, follow the steps in the article section [To create a new virtual network and coexisting connections](#).

- I already have a classic deployment model VNet.

You may already have a virtual network in place with an existing Site-to-Site VPN connection or ExpressRoute connection. The article section [To configure coexisting connections for an already existing VNet](#) will walk you through deleting the gateway, and then creating new ExpressRoute and Site-to-Site VPN connections. Note that when creating the new connections, the steps must be completed in a very specific order. Don't use the instructions in other articles to create your gateways and connections.

In this procedure, creating connections that can coexist will require you to delete your gateway, and then configure new gateways. This means you will have downtime for your cross-premises connections while you delete and recreate your gateway and connections, but you will not need to migrate any of your VMs or services to a new virtual network. Your VMs and services will still be able to communicate out through the load balancer while you configure your gateway if they are configured to do so.

Install PowerShell cmdlets

Install the latest versions of the Azure Service Management (SM) PowerShell modules and the ExpressRoute module. You can't use the Azure CloudShell environment to run SM modules.

1. Use the instructions in the [Installing the Service Management module](#) article to install the Azure Service Management Module. If you have the Az or RM module already installed, be sure to use '-AllowClobber'.
2. Import the installed modules. When using the following example, adjust the path to reflect the location and version of your installed PowerShell modules.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\Azure.psd1'  
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\ExpressRoute\ExpressRoute.psd1'
```

3. To sign in to your Azure account, open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect using the Service Management module:

```
Add-AzureAccount
```

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-7ae3a9fc01b
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 coexisting connections for an already existing VNet

If you have an existing virtual network, check the gateway subnet size. If the gateway subnet is /28 or /29, you must first delete the virtual network gateway and increase the gateway subnet size. The steps in this section will show you how to do that.

If the gateway subnet is /27 or larger and the virtual network is connected via ExpressRoute, you can skip the steps below and proceed to ["Step 6 - Create a Site-to-Site VPN gateway"](#) in the previous section.

NOTE

When you delete the existing gateway, your local premises will lose the connection to your virtual network while you are working on this configuration.

1. You'll need to install the latest version of the Azure Resource Manager PowerShell cmdlets. See [How to install and configure Azure PowerShell](#) for more information about installing the PowerShell cmdlets. Note that the cmdlets that you'll use for this configuration may be slightly different than what you might be familiar with. Be sure to use the cmdlets specified in these instructions.
2. Delete the existing ExpressRoute or Site-to-Site VPN gateway. Use the following cmdlet, replacing the values with your own.

```
Remove-AzureVNetGateway -VnetName MyAzureVNET
```

3. Export the virtual network schema. Use the following PowerShell cmdlet, replacing the values with your own.

```
Get-AzureVNetConfig -ExportToFile "C:\NetworkConfig.xml"
```

4. Edit the network configuration file schema so that the gateway subnet is /27 or a shorter prefix (such as /26 or /25). See the following example.

NOTE

If you don't have enough IP addresses left in your virtual network to increase the gateway subnet size, you need to add more IP address space. For more information about the configuration schema, see [Azure Virtual Network configuration schema](#).

```
<Subnet name="GatewaySubnet">
  <AddressPrefix>10.17.159.224/27</AddressPrefix>
</Subnet>
```

5. If your previous gateway was a Site-to-Site VPN, you must also change the connection type to **Dedicated**.

```
<Gateway>
  <ConnectionsToLocalNetwork>
    <LocalNetworkSiteRef name="MyLocalNetwork">
      <Connection type="Dedicated" />
    </LocalNetworkSiteRef>
  </ConnectionsToLocalNetwork>
</Gateway>
```

6. At this point, you'll have a VNet with no gateways. To create new gateways and complete your connections, you can proceed with [Step 4 - Create an ExpressRoute gateway](#), found in the preceding set of steps.

Next steps

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

Configure a virtual network gateway for ExpressRoute using PowerShell (classic)

12/8/2019 • 4 minutes to read • [Edit Online](#)

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 used in an ExpressRoute configuration.

IMPORTANT

As of March 1, 2017, you can't create new ExpressRoute circuits in the classic deployment model.

- You can move an existing ExpressRoute circuit from the classic deployment model to the Resource Manager deployment model without experiencing any connectivity down time. For more information, see [Move an existing circuit](#).
- You can connect to virtual networks in the classic deployment model by setting **allowClassicOperations** to TRUE.

Use the following links to create and manage ExpressRoute circuits in the Resource Manager deployment model:

- [Create and manage ExpressRoute circuits](#)
- [Configure routing \(peering\) for ExpressRoute circuits](#)

About Azure deployment models

Azure currently works with two deployment models: Resource Manager and classic. The two models are not completely compatible with each other. Before you begin, you need to know which model that you want to work in. For information about the deployment models, see [Understanding deployment models](#). If you are new to Azure, we recommend that you use the Resource Manager deployment model.

Before beginning

Verify that you have installed the Azure PowerShell cmdlets needed for this configuration.

Install the latest versions of the Azure Service Management (SM) PowerShell modules and the ExpressRoute module. You can't use the Azure CloudShell environment to run SM modules.

1. Use the instructions in the [Installing the Service Management module](#) article to install the Azure Service Management Module. If you have the Az or RM module already installed, be sure to use '-AllowClobber'.
2. Import the installed modules. When using the following example, adjust the path to reflect the location and version of your installed PowerShell modules.

```
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\Azure.psd1'
Import-Module 'C:\Program Files\WindowsPowerShell\Modules\Azure\5.3.0\ExpressRoute\ExpressRoute.psd1'
```

3. To sign in to your Azure account, open your PowerShell console with elevated rights and connect to your account. Use the following example to help you connect using the Service Management module:

```
Add-AzureAccount
```

NOTE

These examples do not apply to S2S/ExpressRoute coexist configurations. For more information about working with gateways in a coexist configuration, see [Configure coexisting connections](#).

Add a gateway

When you add a gateway to a virtual network using the classic resource model, you modify the network configuration file directly before creating the gateway. The values in the examples below must be present in the file to create a gateway. If your virtual network previously had a gateway associated to it, some of these values will already be present. Modify the file to reflect the values below.

Download the network configuration file

1. Download the network configuration file using the steps in [network configuration file](#) article. Open the file using a text editor.
2. Add a local network site to the file. You can use any valid address prefix. You can add any valid IP address for the VPN gateway. The address values in this section are not used for ExpressRoute operations, but are required for file validation. In the example, "branch1" is the name of the site. You may use a different name, but be sure to use the same value in the Gateway section of the file.

```
<VirtualNetworkConfiguration>
  <Dns />
  <LocalNetworkSites>
    <LocalNetworkSite name="branch1">
      <AddressSpace>
        <AddressPrefix>165.3.1.0/27</AddressPrefix>
      </AddressSpace>
      <VPNGatewayAddress>3.2.1.4</VPNGatewayAddress>
    </LocalNetworkSite>
```

3. Navigate to the VirtualNetworkSites and modify the fields.

- Verify that the Gateway Subnet exists for your virtual network. If it does not, you can add one at this time. The name must be "GatewaySubnet".
- Verify the Gateway section of the file exists. If it doesn't, add it. This is required to associate the virtual network with the local network site (which represents the network to which you are connecting).
- Verify that the connection type = Dedicated. This is required for ExpressRoute connections.

```
</LocalNetworkSites>
<VirtualNetworkSites>
    <VirtualNetworkSite name="myAzureVNET" Location="East US">
        <AddressSpace>
            <AddressPrefix>10.0.0.0/16</AddressPrefix>
        </AddressSpace>
        <Subnets>
            <Subnet name="default">
                <AddressPrefix>10.0.0.0/24</AddressPrefix>
            </Subnet>
            <Subnet name="GatewaySubnet">
                <AddressPrefix>10.0.1.0/27</AddressPrefix>
            </Subnet>
        </Subnets>
        <Gateway>
            <ConnectionsToLocalNetwork>
                <LocalNetworkSiteRef name="branch1">
                    <Connection type="Dedicated" />
                </LocalNetworkSiteRef>
            </ConnectionsToLocalNetwork>
        </Gateway>
    </VirtualNetworkSite>
</VirtualNetworkSites>
</VirtualNetworkConfiguration>
</NetworkConfiguration>
```

4. Save the file and upload it to Azure.

Create the gateway

Use the command below to create a gateway. Substitute any values for your own.

```
New-AzureVNetGateway -VNetName "MyAzureVNET" -GatewayType DynamicRouting -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-AzureVNetGateway
```

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-AzureVNetGateway -GatewayId <Gateway ID> -GatewaySKU HighPerformance
```

Remove a gateway

Use the command below to remove a gateway

```
Remove-AzureVnetGateway -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](#).

Create and manage ExpressRoute public peering

12/17/2019 • 9 minutes to read • [Edit Online](#)

This article helps you create and manage public peering routing configuration for an ExpressRoute circuit. You can also check the status, update, or delete and deprovision peerings. This article applies to Resource Manager circuits that were created before public peering was deprecated. If you have a previously existing circuit (created prior to public peering being deprecated), you can manage/configure public peering using [Azure PowerShell](#), [Azure CLI](#), and the [Azure portal](#).

NOTE

Public peering is deprecated. You cannot create public peering on new ExpressRoute circuits. If you have a new ExpressRoute circuit, instead, use [Microsoft peering](#) for your Azure services.

Connectivity

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. If your ExpressRoute circuit is enabled for Azure public peering, you can access the [public IP ranges used in Azure](#) over the circuit.

Once public peering is enabled, you can connect to most Azure services. We do not allow you to selectively pick services for which we advertise routes to.

- Services such as Azure Storage, SQL Databases, and Websites are offered on public IP addresses.
- Through the public peering routing domain, you can privately connect to services hosted on public IP addresses, including VIPs of your cloud services.
- 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.

Services

This section shows the services available over public peering. Because public peering is deprecated, there is no plan to add new or additional services to public peering. If you use public peering and the service you want to use is supported only over Microsoft peering, you must switch to Microsoft peering. See [Microsoft peering](#) for a list of supported services.

Supported:

- Power BI
- Most of the Azure services are supported. Check directly with the service that you want to use to verify support.

Not supported:

- CDN
- Azure Front Door
- Multi-factor Authentication Server (legacy)
- Traffic Manager

To validate availability for a specific service, you can check the documentation for that service to see if there is a reserved range published for that service. Then you may look up the IP ranges of the target service and compare

with the ranges listed in the [Azure IP Ranges and Service Tags – Public Cloud XML file](#). Alternatively, you can open a support ticket for the service in question for clarification.

Peering comparison

	PRIVATE PEERING	MICROSOFT PEERING	PUBLIC PEERING (DEPRECATED FOR NEW CIRCUITS)
Max. # prefixes supported per peering	4000 by default, 10,000 with ExpressRoute Premium	200	200
IP address ranges supported	Any valid IP address within your WAN.	Public IP addresses owned by you or your connectivity provider.	Public IP 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.
IP protocols supported	IPv4	IPv4, IPv6	IPv4
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

NOTE

Azure public peering has 1 NAT IP address associated to each BGP session. For greater than 2 NAT IP addresses, move to Microsoft peering. Microsoft peering allows you to configure your own NAT allocations, as well as use route filters for selective prefix advertisements. For more information, see [Move to Microsoft peering](#).

Custom route filters

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.

Azure PowerShell steps

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

There are a few ways to launch the Cloud Shell:

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

Azure PowerShell

 Copy

 Try It

Open Cloud Shell in your browser.

 Launch Cloud Shell

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



Because public peering is deprecated, you cannot configure public peering on a new ExpressRoute circuit.

1. Verify that you have an ExpressRoute circuit that is provisioned and also enabled. Use the following example:

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

The response is similar to the following example:

```
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 : []
```

2. Configure Azure public peering for the circuit. Make sure 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.
- Optional:
 - An MD5 hash if you choose to use one.

Run the following example to configure Azure public peering for your circuit

```
Add-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt -  
PeeringType AzurePublicPeering -PeerASN 100 -PrimaryPeerAddressPrefix "12.0.0.0/30" -  
SecondaryPeerAddressPrefix "12.0.0.4/30" -VlanId 100  
  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

If you choose to use an MD5 hash, use the following example:

```
Add-AzExpressRouteCircuitPeeringConfig -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-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

IMPORTANT

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

To get Azure public peering details

You can get configuration details using the following cmdlet:

```
$ckt = Get-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
Get-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -Circuit $ckt
```

To update Azure public peering configuration

You can update any part of the configuration using the following example. In this example, the VLAN ID of the circuit is being updated from 200 to 600.

```
Set-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt -PeeringType  
AzurePublicPeering -PeerASN 100 -PrimaryPeerAddressPrefix "123.0.0.0/30" -SecondaryPeerAddressPrefix  
"123.0.0.4/30" -VlanId 600  
  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

To delete Azure public peering

You can remove your peering configuration by running the following example:

```
Remove-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

Azure CLI steps

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

There are a few ways to launch the Cloud Shell:

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

Azure PowerShell  

Open Cloud Shell in your browser.



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



1. Check the ExpressRoute circuit to ensure it is provisioned and also enabled. Use the following example:

```
az network express-route list
```

The response is similar to the following example:

```
"allowClassicOperations": false,
"authorizations": [],
"circuitProvisioningState": "Enabled",
"etag": "W/\\"1262c492-ffef-4a63-95a8-a6002736b8c4\\\"",
"gatewayManagerEtag": null,
"id": "/subscriptions/81ab786c-56eb-4a4d-bb5f-
f60329772466/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits
/MyCircuit",
"location": "westus",
"name": "MyCircuit",
"peerings": [],
"provisioningState": "Succeeded",
"resourceGroup": "ExpressRouteResourceGroup",
"serviceKey": "1d05cf70-1db5-419f-ad86-1ca62c3c125b",
"serviceProviderNotes": null,
"serviceProviderProperties": {
    "bandwidthInMbps": 200,
    "peeringLocation": "Silicon Valley",
    "serviceProviderName": "Equinix"
},
"serviceProviderProvisioningState": "Provisioned",
"sku": {
    "family": "UnlimitedData",
    "name": "Standard_MeteredData",
    "tier": "Standard"
},
"tags": null,
"type": "Microsoft.Network/expressRouteCircuits"]
```

2. Configure Azure public peering for the circuit. Make sure 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.
- **Optional** - An MD5 hash if you choose to use one.

Run the following example to configure Azure public peering for your circuit:

```
az network express-route peering create --circuit-name MyCircuit --peer-asn 100 --primary-peer-subnet 12.0.0.0/30 -g ExpressRouteResourceGroup --secondary-peer-subnet 12.0.0.4/30 --vlan-id 200 --peering-type AzurePublicPeering
```

If you choose to use an MD5 hash, use the following example:

```
az network express-route peering create --circuit-name MyCircuit --peer-asn 100 --primary-peer-subnet 12.0.0.0/30 -g ExpressRouteResourceGroup --secondary-peer-subnet 12.0.0.4/30 --vlan-id 200 --peering-type AzurePublicPeering --SharedKey "A1B2C3D4"
```

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 example:

```
az network express-route peering show -g ExpressRouteResourceGroup --circuit-name MyCircuit --name AzurePublicPeering
```

The output is similar to the following example:

```
{  
    "azureAsn": 12076,  
    "etag": "W/\"2e97be83-a684-4f29-bf3c-96191e270666\"",  
    "gatewayManagerEtag": "18",  
    "id": "/subscriptions/9a0c2943-e0c2-4608-876c-e0ddffd1211b/resourceGroups/ExpressRouteResourceGroup/providers/Microsoft.Network/expressRouteCircuits/MyCircuit/peerings/AzurePublicPeering",  
    "lastModifiedBy": "Customer",  
    "microsoftPeeringConfig": null,  
    "name": "AzurePublicPeering",  
    "peerAsn": 7671,  
    "peeringType": "AzurePublicPeering",  
    "primaryAzurePort": "",  
    "primaryPeerAddressPrefix": "",  
    "provisioningState": "Succeeded",  
    "resourceGroup": "ExpressRouteResourceGroup",  
    "routeFilter": null,  
    "secondaryAzurePort": "",  
    "secondaryPeerAddressPrefix": "",  
    "sharedKey": null,  
    "state": "Enabled",  
    "stats": null,  
    "vlanId": 100  
}
```

To update Azure public peering configuration

You can update any part of the configuration using the following example. In this example, the VLAN ID of the circuit is being updated from 200 to 600.

```
az network express-route peering update --vlan-id 600 -g ExpressRouteResourceGroup --circuit-name MyCircuit --name AzurePublicPeering
```

To delete Azure public peering

You can remove your peering configuration by running the following example:

```
az network express-route peering delete -g ExpressRouteResourceGroup --circuit-name MyCircuit --name AzurePublicPeering
```

Azure portal steps

To configure peering, use the PowerShell or CLI steps contained in this article. To manage a peering, you can use the sections below. For reference, these steps look similar to managing a [Microsoft peering in the portal](#).

To view Azure public peering details

View the properties of Azure public peering by selecting the peering in the portal.

To update Azure public peering configuration

Select the row for peering, then modify the peering properties.

To delete Azure public peering

Remove your peering configuration by selecting the delete icon.

Next steps

Next step, [Link a virtual network 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](#).

2 minutes to read

Optimize ExpressRoute Routing

11/14/2019 • 7 minutes to read • [Edit Online](#)

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.

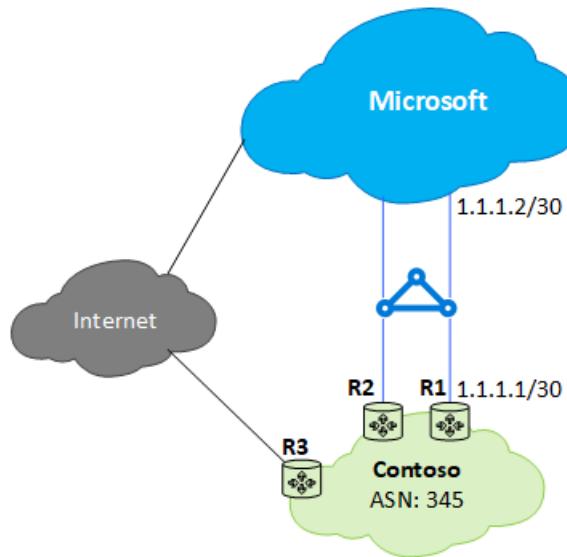
Path Selection on Microsoft and Public peerings

It's important to ensure that when utilizing Microsoft or Public peering that traffic flows over the desired path if you have one or more ExpressRoute circuits, as well as paths to the Internet via an Internet Exchange (IX) or Internet Service Provider (ISP). BGP utilizes a best path selection algorithm based on a number of factors including longest prefix match (LPM). To ensure that traffic destined for Azure via Microsoft or Public peering traverses the ExpressRoute path, customers must implement the *Local Preference* attribute to ensure that the path is always preferred on ExpressRoute.

NOTE

The default local preference is typically 100. Higher local preferences are more preferred.

Consider the following example scenario:



In the above example, to prefer ExpressRoute paths configure Local Preference as follows.

Cisco IOS-XE configuration from R1 perspective:

```
R1(config)#route-map prefer-ExR permit 10
R1(config-route-map)#set local-preference 150

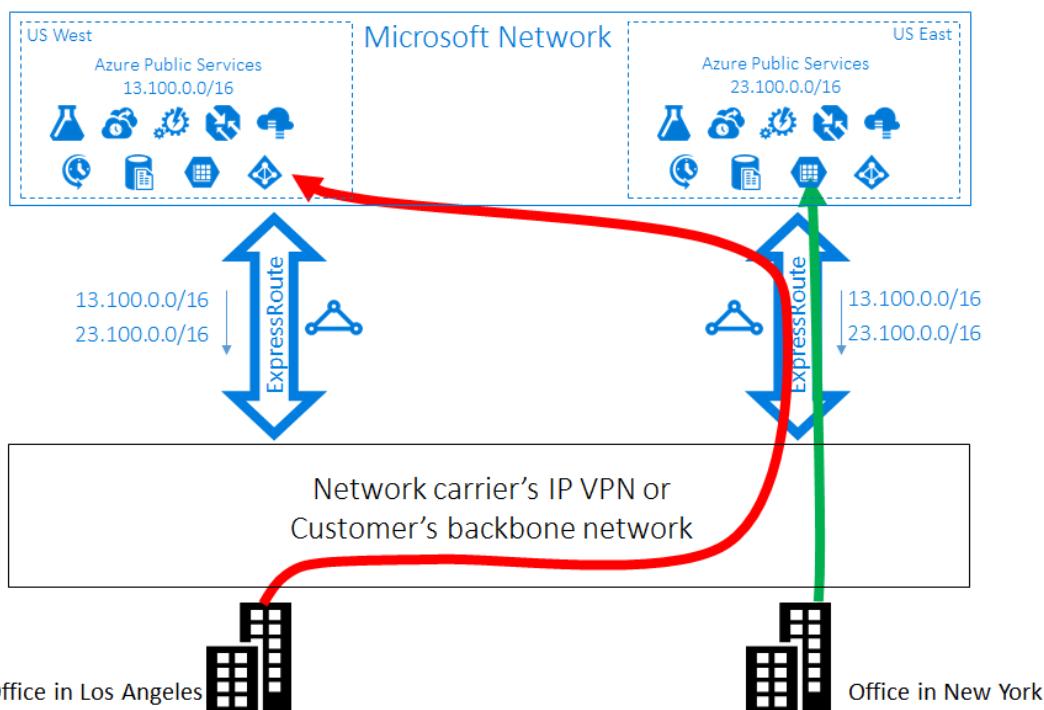
R1(config)#router BGP 345
R1(config-router)#neighbor 1.1.1.2 remote-as 12076
R1(config-router)#neighbor 1.1.1.2 activate
R1(config-router)#neighbor 1.1.1.2 route-map prefer-ExR in
```

Junos configuration from R1 perspective:

```
user@R1# set protocols bgp group ibgp type internal  
user@R1# set protocols bgp group ibgp local-preference 150
```

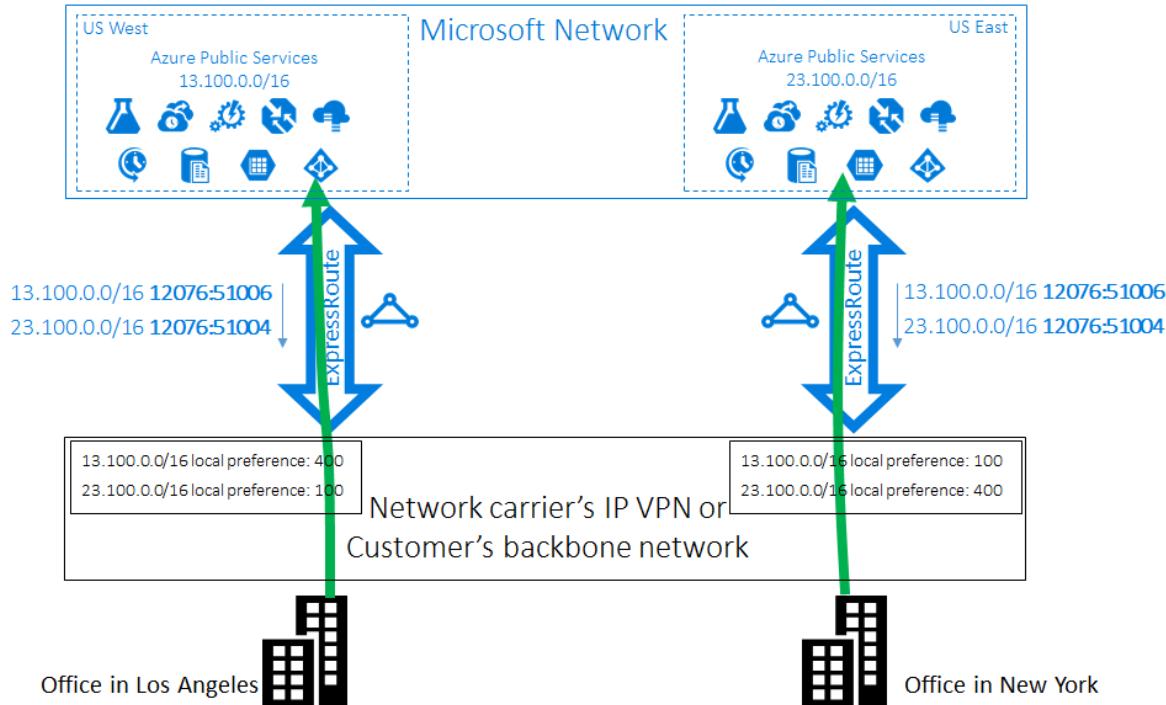
Suboptimal routing from customer to Microsoft

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 (for example, 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.

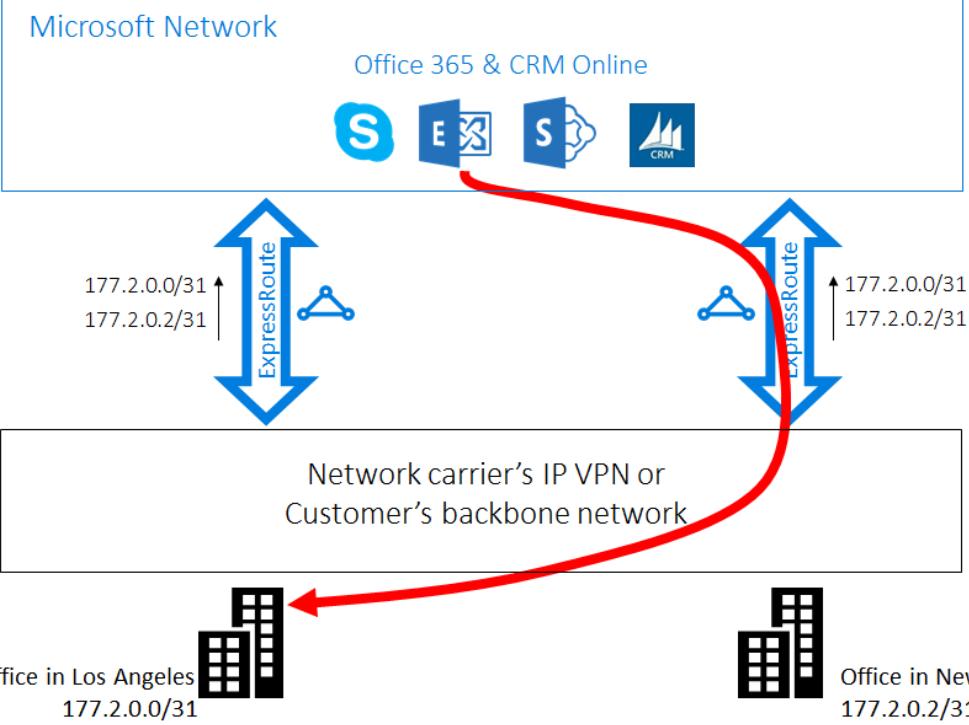


NOTE

The same technique, using Local Preference, can be applied to routing from customer to Azure Virtual Network. We don't tag BGP Community value to the prefixes advertised from Azure to your network. However, since you know which of your Virtual Network deployment is close to which of your office, you can configure your routers accordingly to prefer one ExpressRoute circuit to another.

Suboptimal routing from Microsoft to customer

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.



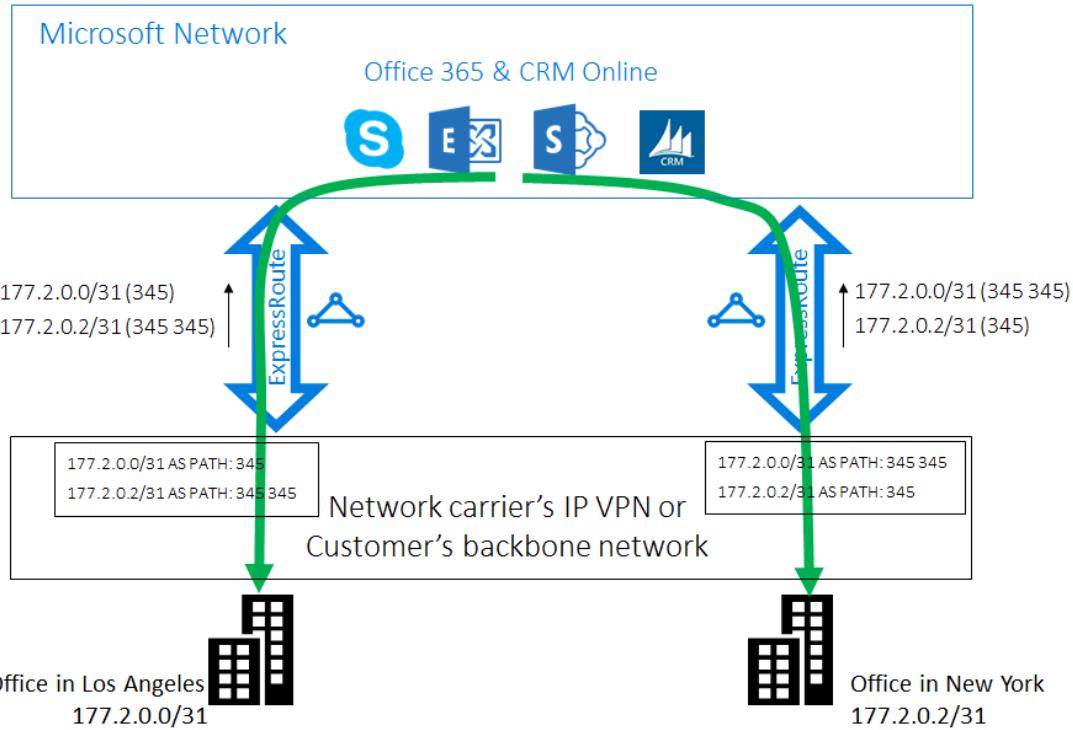
Solution: use AS PATH prepending

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 when peering using a private AS number. You need to peer with a public AS and append public AS numbers in the AS PATH to influence routing for Microsoft Peering.

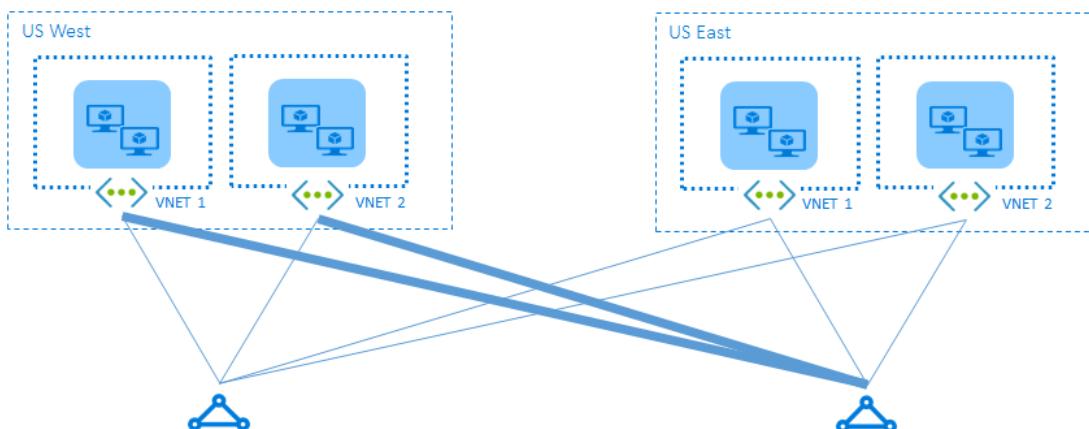


NOTE

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.

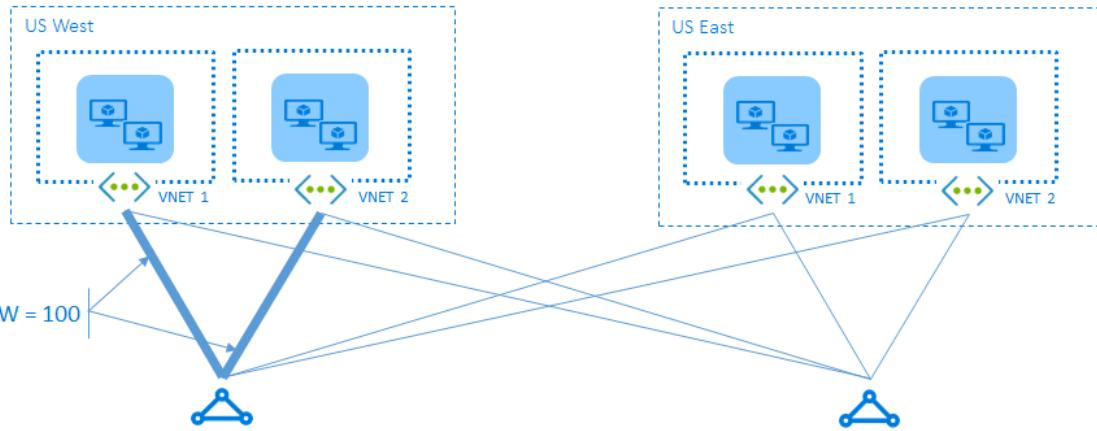
Suboptimal routing between virtual networks

With ExpressRoute, you can enable Virtual Network to Virtual Network (which is also known as "VNet") communication by linking them to an ExpressRoute circuit. When you link them to multiple ExpressRoute circuits, suboptimal routing can happen between the VNets. Let's consider an example. You have two ExpressRoute circuits, one in US West and one in US East. In each region, you have two VNets. Your web servers are deployed in one VNet and application servers in the other. For redundancy, you link the two VNets in each region to both the local ExpressRoute circuit and the remote ExpressRoute circuit. As can be seen below, from each VNet there are two paths to the other VNet. The VNets don't know which ExpressRoute circuit is local and which one is remote. Consequently as they do Equal-Cost-Multi-Path (ECMP) routing to load-balance inter-VNet traffic, some traffic flows will take the longer path and get routed at the remote ExpressRoute circuit.



Solution: assign a high weight to local connection

The solution is simple. Since you know where the VNets and the circuits are, you can tell us which path each VNet should prefer. Specifically for this example, you assign a higher weight to the local connection than to the remote connection (see the configuration example [here](#)). When a VNet receives the prefix of the other VNet on multiple connections it will prefer the connection with the highest weight to send traffic destined for that prefix.



NOTE

You can also influence routing from VNet to your on-premises network, if you have multiple ExpressRoute circuits, by configuring the weight of a connection instead of applying AS PATH prepending, a technique described in the second scenario above. For each prefix, we will always look at the connection weight before the AS Path length when deciding how to send traffic.

Asymmetric routing with multiple network paths

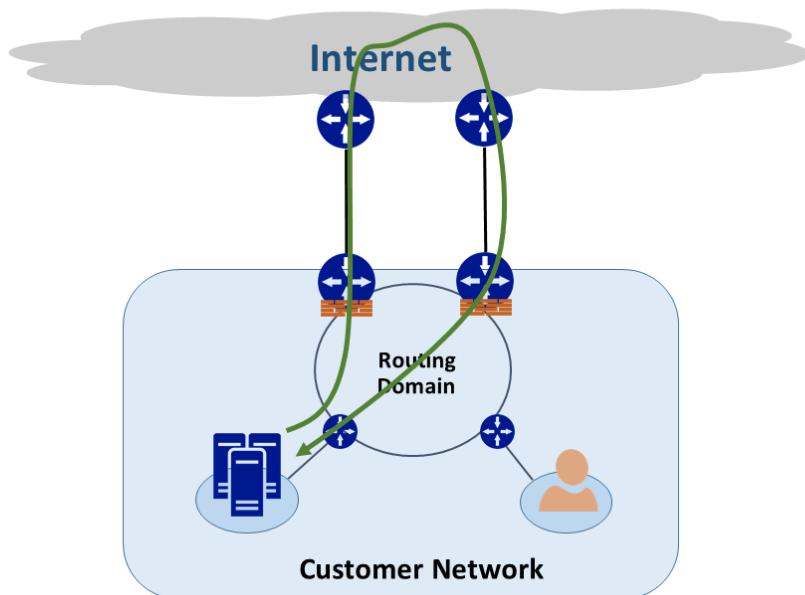
11/13/2019 • 6 minutes to read • [Edit Online](#)

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.

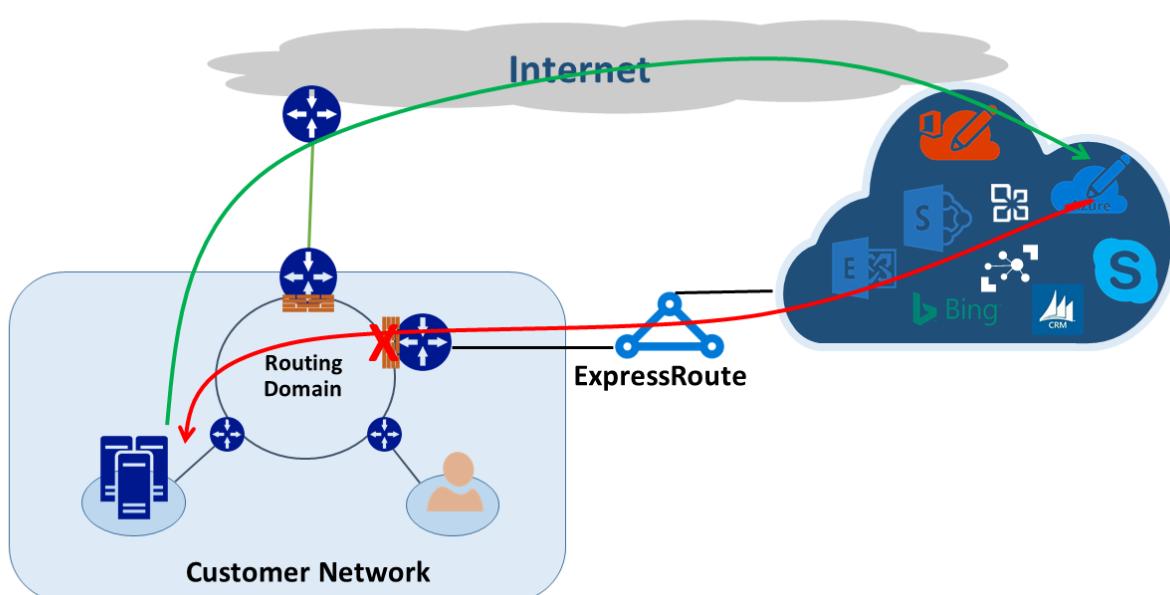
A firewall is a common example of a stateful device. A firewall allows or denies a packet to pass through its interfaces based on various fields such as protocol, TCP/UDP port, and URL headers. This level of packet inspection puts a heavy processing load on the device. To improve performance, the firewall 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 advertise 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).

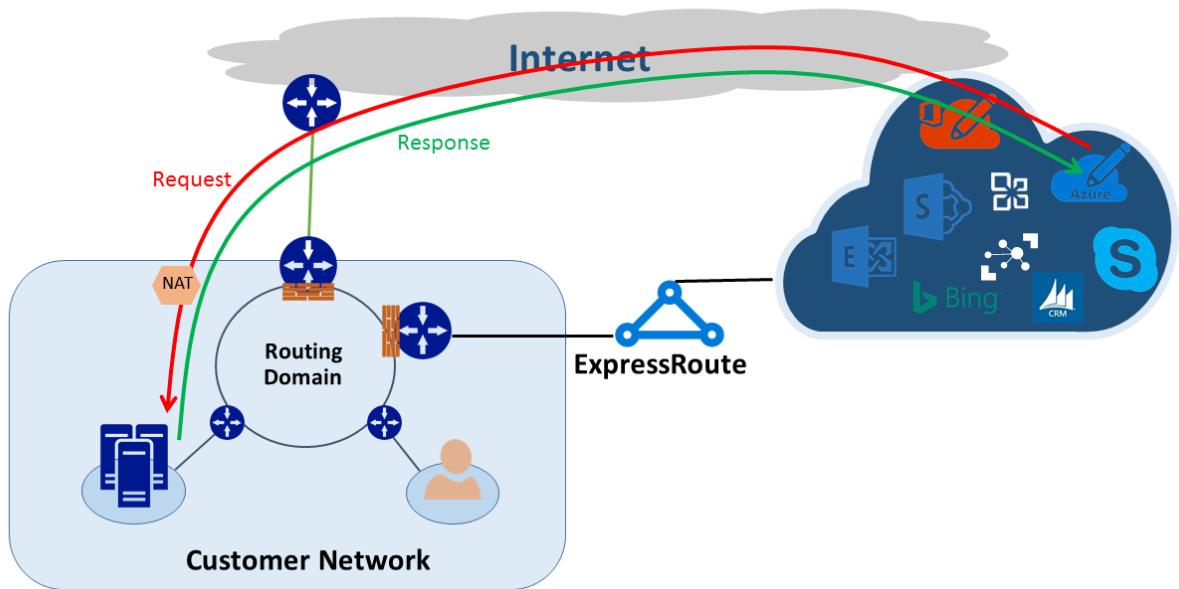
Routing

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.

ExpressRoute NAT requirements

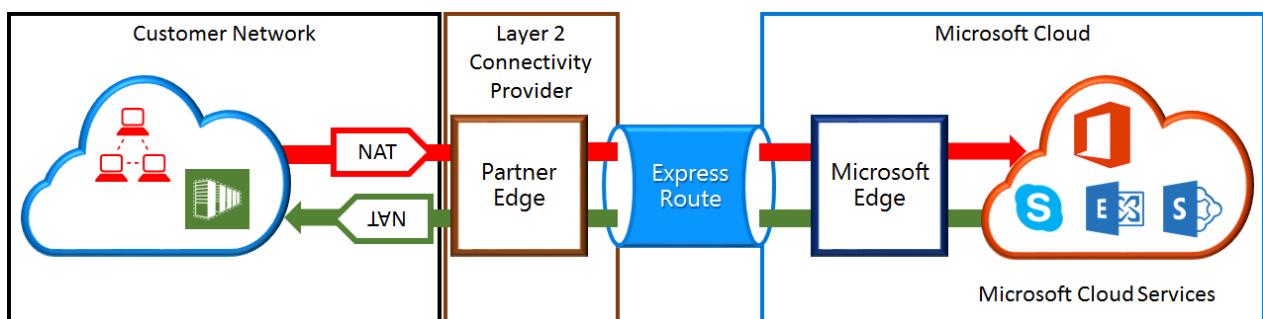
11/14/2019 • 4 minutes to read • [Edit Online](#)

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 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, and Skype for Business. 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 at your Internet edge to prevent [asymmetric routing](#). The figure below provides a high-level picture of how the NAT should be set up 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

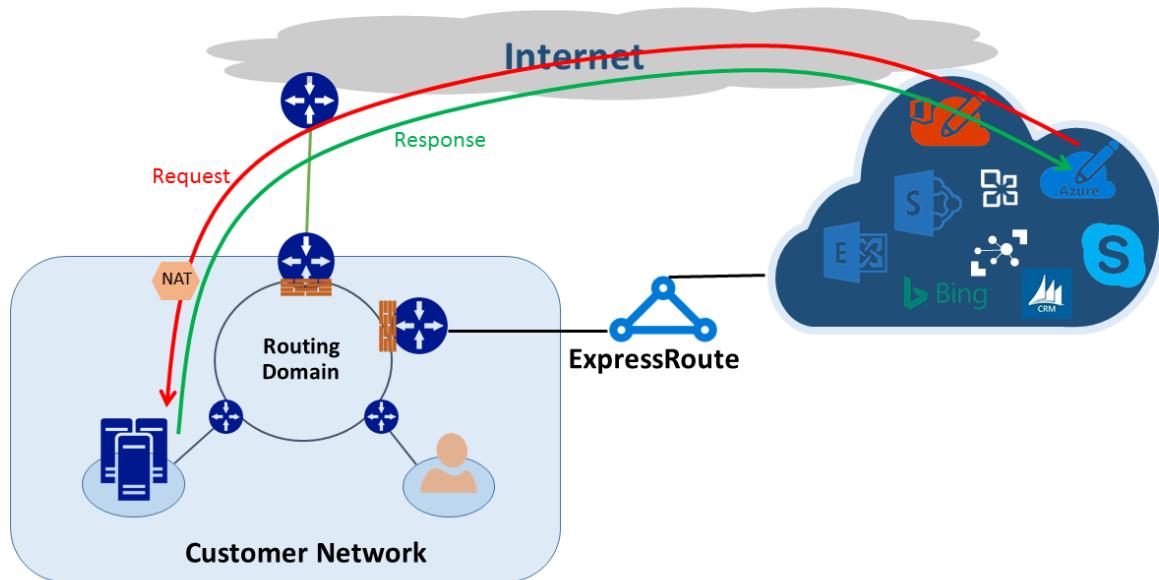
The NAT IP pool advertised to Microsoft must not be advertised to the Internet. This will break connectivity to other Microsoft services.

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 Microsoft traffic at the Internet edge for service endpoints within your network to prevent **asymmetric routing**. Requests **and replies** with a destination IP that match a route received via ExpressRoute will always be sent via ExpressRoute. Asymmetric routing exists if the request is received via the Internet with the reply sent via ExpressRoute. SNATing the incoming Microsoft traffic at the Internet edge forces reply traffic back to the Internet edge, resolving the problem.



NAT requirements for Azure public peering

NOTE

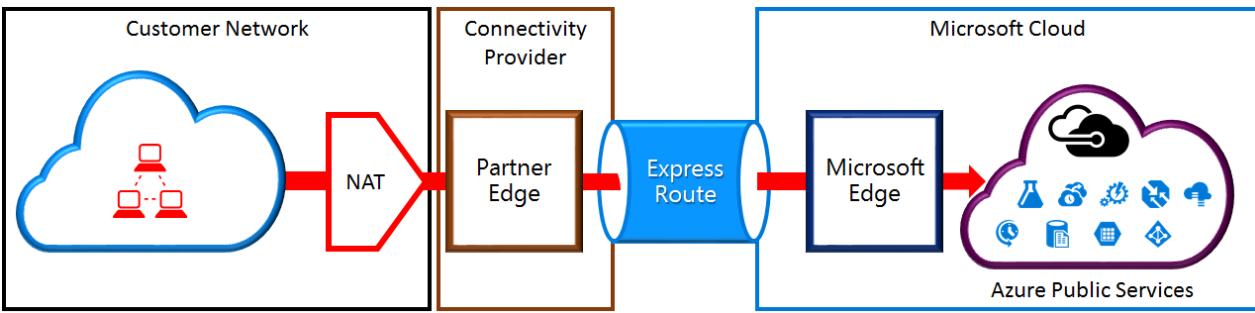
Azure public peering is not available for new circuits.

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 [ExpressRoute FAQ](#) and any services hosted by ISVs on Microsoft Azure.

IMPORTANT

Connectivity to Microsoft Azure services on public peering is always initiated from your network into the Microsoft network. Therefore, sessions cannot be initiated from Microsoft Azure services to your network over ExpressRoute. If attempted, packets sent to these advertised IPs will use the internet instead of ExpressRoute.

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.

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](#)

Verifying ExpressRoute connectivity

12/30/2019 • 10 minutes to read • [Edit Online](#)

This article helps you verify and troubleshoot ExpressRoute connectivity. ExpressRoute extends an on-premises network into the Microsoft cloud over a private connection that is commonly facilitated by a connectivity provider. ExpressRoute connectivity traditionally involves three distinct network zones, as follows:

- Customer Network
- Provider Network
- Microsoft Datacenter

NOTE

In the ExpressRoute direct connectivity model (offered at 10/100 Gbps bandwidth), customers can directly connect to Microsoft Enterprise Edge (MSEE) routers' port. Therefore, in the direct connectivity model, there are only customer and Microsoft network zones.

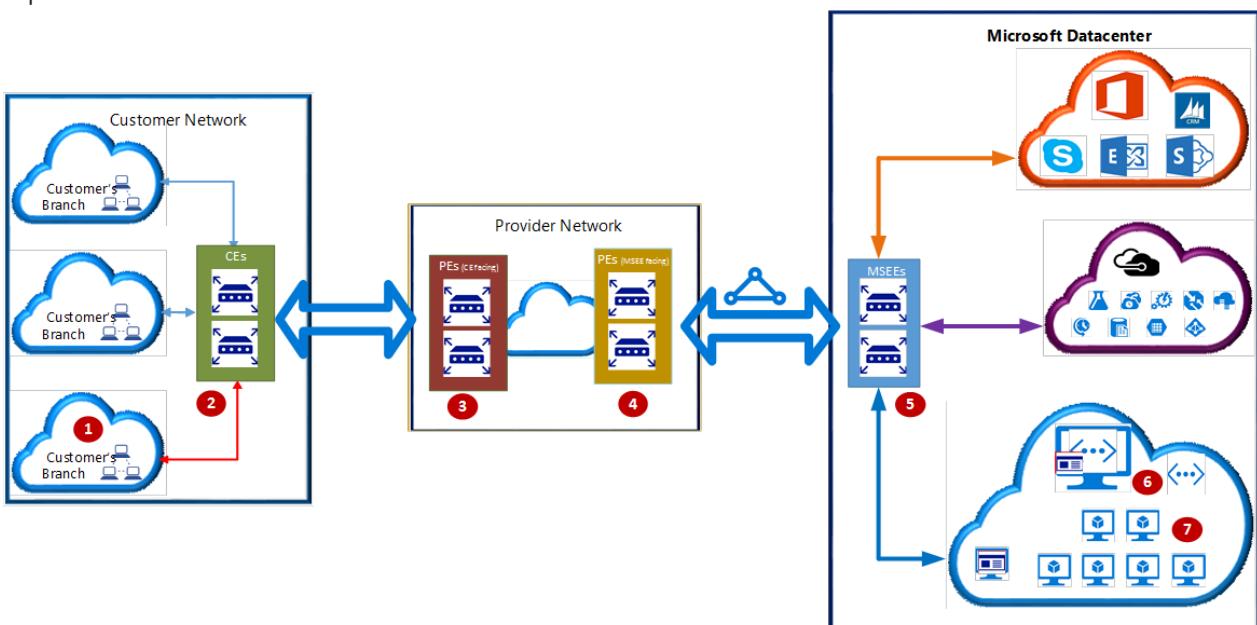
The purpose of this document is to help user to identify if and where a connectivity issue exists. Thereby, to help seek support from the appropriate team to resolve an issue. If Microsoft support is needed to resolve an issue, open a support ticket with [Microsoft Support](#).

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. These network points are referenced in this article at times 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) or routers (Layer 3 devices). In the direct connectivity model, there are no network points 3 and 4; instead CEs (2) are directly connected to MSEEs via dark fiber. 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. Referred to as PE-CEs in this document.
4. PEs (MSEE facing): Provider edge routers/switches that are facing MSEEs. Referred to as PE-MSEEs in this document.
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, Point-to-Point Ethernet, or direct connectivity models are used, CEs (2) establish BGP peering with MSEEs (5).

If the Any-to-any (IPVPN) connectivity model is used, PE-MSEEs (4) establish BGP peering with MSEEs (5). PE-MSEEs propagate the routes received from Microsoft back to the customer network via the IPVPN service provider network.

NOTE

For high availability, Microsoft establishes a fully redundant parallel connectivity between MSEEs (5) and PE-MSEEs (4) pairs. A fully redundant parallel network path is also encouraged between customer network and PE-CEs pair. For more information regarding high availability, see the article [Designing for high availability with ExpressRoute](#)

The following are the logical steps, in troubleshooting ExpressRoute circuit:

- [Verify circuit provisioning and state](#)
- [Validate Peering Configuration](#)
- [Validate ARP](#)
- [Validate BGP and routes on the MSEE](#)
- [Confirm the traffic flow](#)

Verify circuit provisioning and state

Provisioning an ExpressRoute circuit establishes a redundant Layer 2 connections between CEs/PE-MSEEs (2)/(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. 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, open the ExpressRoute circuit blade. In the  Overview section of the blade, the ExpressRoute essentials are listed as shown in the following screenshot:

The screenshot shows the Azure portal interface for an ExpressRoute circuit named 'ER-Demo-Ckt-SV'. At the top, there's a navigation bar with icons for Settings and Delete. Below it is a header section with 'Essentials' and three small icons. The main content area has two columns of information:

Resource group	Provider
USWest-ER-Demo-RG	Equinix
Circuit status	Provider status
Enabled	Provisioned
Location	Peering location
West US	Silicon Valley
Subscription name	Bandwidth
ExpressRoute-Demo	200 Mbps
Subscription ID	Service key

Below this, there's a link 'All settings →'. The 'Peerings' section follows, containing a table with three rows:

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	...
Azure private	Disabled	-	-	...
Azure public	Disabled	-	-	...
Microsoft	Disabled	-	-	...

At the bottom of the 'Peerings' section is a button 'Add a section +'

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

After configuring an ExpressRoute circuit, if the *Circuit status* is stuck in not enabled status, contact [Microsoft Support](#). On the other hand, if the *Provider status* is stuck in not provisioned status, contact your service provider.

Verification via PowerShell

To list all the ExpressRoute circuits in a Resource Group, use the following command:

```
Get-AzExpressRouteCircuit -ResourceGroupName "Test-ER-RG"
```

TIP

If you are looking for the name of a resource group, you can get it by listing all the resource groups in your subscription, using the command `Get-AzResourceGroup`

To select a particular ExpressRoute circuit in a Resource Group, use the following command:

```
Get-AzExpressRouteCircuit -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

After configuring an ExpressRoute circuit, if the *Circuit status* is stuck in not enabled status, contact [Microsoft Support](#). On the other hand, if the *Provider status* is stuck in not provisioned status, contact your service provider.

Validate Peering Configuration

After the service provider has completed the provisioning the ExpressRoute circuit, multiple eBGP based routing configurations can be created over the ExpressRoute circuit between CEs/MSEE-PEs (2)/(4) and MSEEs (5). Each ExpressRoute circuit can have: Azure private peering (traffic to private virtual networks in Azure), and/or Microsoft peering (traffic to public endpoints of PaaS and SaaS). 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

NOTE

In IPVPN connectivity model, service providers handle the responsibility of configuring the peerings (layer 3 services). In such a model, after the service provider has configured a peering and if the peering is blank in the portal, try refreshing the circuit configuration using the refresh button on the portal. This operation will pull the current routing configuration from your circuit.

In the Azure portal, status of an ExpressRoute circuit peering can be checked under the ExpressRoute circuit blade. In the  [Overview](#) section of the blade, the ExpressRoute peerings would be listed as shown in the following screenshot:

In the preceding example, as noted Azure private peering is provisioned, whereas Azure public and Microsoft peerings are not provisioned. A successfully provisioned peering context would also have the primary and secondary point-to-point subnets listed. The /30 subnets are used for the interface IP address of the MSEEs and CEs/PE-MSEEs. For the peerings that are provisioned, the listing also indicates who last modified the configuration.

NOTE

If enabling a peering fails, check if the primary and secondary subnets assigned match the configuration on the linked CE/PE-MSEE. 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 CE/PE-MSEE. If MD5 hashing is chosen, the shared key should be same on MSEE and PE-MSEE/CE pair. Previously configured shared key would not be displayed for security reasons. Should you need to change any of these configuration on an MSEE router, refer to [Create and modify routing for an ExpressRoute circuit](#).

NOTE

On a /30 subnet assigned for interface, Microsoft will pick the second usable IP address of the subnet for the MSEE interface. Therefore, ensure that the first usable IP address of the subnet has been assigned on the peered CE/PE-MSEE.

Verification via PowerShell

To get the Azure private peering configuration details, use the following commands:

```
$ckt = Get-AzExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzExpressRouteCircuitPeeringConfig -Name "AzurePrivatePeering" -ExpressRouteCircuit $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 : 123##
PrimaryPeerAddressPrefix : 172.16.0.0/30
SecondaryPeerAddressPrefix : 172.16.0.4/30
PrimaryAzurePort :
SecondaryAzurePort :
SharedKey :
VlanId : 200
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 CE/PE-MSEEs.

To get the Azure public peering configuration details, use the following commands:

```
$ckt = Get-AzExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" -ExpressRouteCircuit $ckt
```

To get the Microsoft peering configuration details, use the following commands:

```
$ckt = Get-AzExpressRouteCircuit -ResourceGroupName "Test-ER-RG" -Name "Test-ER-Ckt"
Get-AzExpressRouteCircuitPeeringConfig -Name "MicrosoftPeering" -ExpressRouteCircuit $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-AzExpressRouteCircuitPeeringConfig : Sequence contains no matching element
At line:1 char:1
+ Get-AzExpressRouteCircuitPeeringConfig -Name "AzurePublicPeering" ...
+ ~~~~~
+ CategoryInfo          : CloseError: (:) [Get-AzExpressRouteCircuitPeeringConfig], InvalidOperationException
+ FullyQualifiedErrorId : Microsoft.Azure.Commands.Network.GetAzureExpressRouteCircuitPeeringConfigCommand
```

NOTE

If enabling a peering fails, check if the primary and secondary subnets assigned match the configuration on the linked CE/PE-MSEE. 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 CE/PE-MSEE. If MD5 hashing is chosen, the shared key should be same on MSEE and PE-MSEE/CE pair. Previously configured shared key would not be displayed for security reasons. Should you need to change any of these configuration on an MSEE router, refer to [Create and modify routing for an ExpressRoute circuit](#).

NOTE

On a /30 subnet assigned for interface, Microsoft will pick the second usable IP address of the subnet for the MSEE interface. Therefore, ensure that the first usable IP address of the subnet has been assigned on the peered CE/PE-MSEE.

Validate ARP

The ARP table provides a mapping of the IP 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):

- Mapping of on-premises router interface ip address to the MAC address
- Mapping of ExpressRoute router interface ip address to the MAC address
- Age of the mapping ARP tables can help validate layer 2 configuration and troubleshooting basic layer 2 connectivity issues.

See [Getting ARP tables in the Resource Manager deployment model](#) document, for how to view the ARP table of an ExpressRoute peering, and for how to use the information to troubleshoot layer 2 connectivity issue.

Validate BGP and routes on the MSEE

To get the routing table from MSEE on the *Primary* path for the *Private* routing context, use the following command:

```
Get-AzExpressRouteCircuitRouteTable -DevicePath Primary -ExpressRouteCircuitName ***** -PeeringType AzurePrivatePeering -ResourceGroupName ***
```

An example response is:

```
Network : 10.1.0.0/16
NextHop : 10.17.17.141
LocPrf :
Weight : 0
Path : 65515

Network : 10.1.0.0/16
NextHop : 10.17.17.140*
LocPrf :
Weight : 0
Path : 65515

Network : 10.2.20.0/25
NextHop : 172.16.0.1
LocPrf :
Weight : 0
Path : 123##
```

NOTE

If the state of a eBGP peering between an MSEE and a CE/PE-MSEE is in Active or Idle, check if the primary and secondary peer subnets assigned match the configuration on the linked CE/PE-MSEE. 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 PE-MSEE/CE. If MD5 hashing is chosen, the shared key should be same on MSEE and CE/PE-MSEE pair. Should you need to change any of these configuration on an MSEE router, refer to [Create and modify routing for an ExpressRoute circuit](#).

NOTE

If certain destinations are not reachable over a peering, check the route table of the MSEEs for the corresponding 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 that are blocking the traffic.

The following example shows the response of the command for a peering that does not exist:

```
Get-AzExpressRouteCircuitRouteTable : The BGP Peering AzurePublicPeering with Service Key  
***** is not found.  
StatusCode: 400
```

Confirm the traffic flow

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-AzExpressRouteCircuitRouteTable : The BGP Peering AzurePublicPeering with Service Key  
***** is not found.  
StatusCode: 400
```

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](#)

Troubleshooting network performance

12/5/2019 • 15 minutes to read • [Edit Online](#)

Overview

Azure provides stable and fast ways to connect from your on-premises network to Azure. Methods like Site-to-Site VPN and ExpressRoute are successfully used by customers large and small to run their businesses in Azure. But what happens when performance doesn't meet your expectation or previous experience? This document can help standardize the way you test and baseline your specific environment.

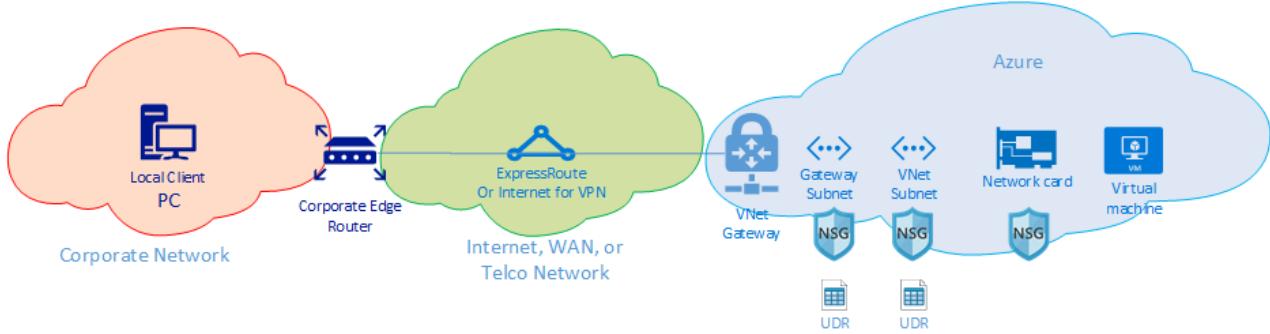
This document shows how you can easily and consistently test network latency and bandwidth between two hosts. This document also provides some advice on ways to look at the Azure network and help to isolate problem points. The PowerShell script and tools discussed require two hosts on the network (at either end of the link being tested). One host must be a Windows Server or Desktop, the other can be either Windows or Linux.

NOTE

The approach to troubleshooting, the tools, and methods used are personal preferences. This document describes the approach and tools I often take. Your approach will probably differ, there's nothing wrong with different approaches to problem solving. However, if you don't have an established approach, this document can get you started on the path to building your own methods, tools, and preferences to troubleshooting network issues.

Network components

Before digging into troubleshooting, let's discuss some common terms and components. This discussion ensures we're thinking about each component in the end-to-end chain that enables connectivity in Azure.



At the highest level, I describe three major network routing domains;

- the Azure network (blue cloud on the right)
- the Internet or WAN (green cloud in the center)
- the Corporate Network (peach cloud on the left)

Looking at the diagram from right to left, let's discuss briefly each component:

- **Virtual Machine** - The server may have multiple NICs, ensure any static routes, default routes, and Operating System settings are sending and receiving traffic the way you think it is. Also, each VM SKU has a bandwidth restriction. If you're using a smaller VM SKU, your traffic is limited by the bandwidth available to the NIC. I usually use a DS5v2 for testing (and then delete once done with testing to save money) to ensure adequate bandwidth at the VM.
- **NIC** - Ensure you know the private IP that is assigned to the NIC in question.

- **NIC NSG** - There may be specific NSGs applied at the NIC level, ensure the NSG rule-set is appropriate for the traffic you're trying to pass. For example, ensure ports 5201 for iPerf, 3389 for RDP, or 22 for SSH are open to allow test traffic to pass.
- **VNet Subnet** - The NIC is assigned to a specific subnet, ensure you know which one and the rules associated with that subnet.
- **Subnet NSG** - Just like the NIC, NSGs can be applied at the subnet as well. Ensure the NSG rule-set is appropriate for the traffic you're trying to pass. (for traffic inbound to the NIC the subnet NSG applies first, then the NIC NSG, conversely for traffic outbound from the VM the NIC NSG applies first then the Subnet NSG comes into play).
- **Subnet UDR** - User Defined Routes can direct traffic to an intermediate hop (like a firewall or load-balancer). Ensure you know if there is a UDR in place for your traffic and if so where it goes and what that next hop will do to your traffic. (for example, a firewall could pass some traffic and deny other traffic between the same two hosts).
- **Gateway subnet / NSG / UDR** - Just like the VM subnet, the gateway subnet can have NSGs and UDRs. Make sure you know if they are there and what effects they have on your traffic.
- **VNet Gateway (ExpressRoute)** - Once peering (ExpressRoute) or VPN is enabled, there aren't many settings that can affect how or if traffic routes. If you have multiple ExpressRoute circuits or VPN tunnels connected to the same VNet Gateway, you should be aware of the connection weight settings as this setting affects connection preference and affects the path your traffic takes.
- **Route Filter** (Not shown) - A route filter only applies to Microsoft Peering on ExpressRoute, but is critical to check if you're not seeing the routes you expect on Microsoft Peering.

At this point, you're on the WAN portion of the link. This routing domain can be your service provider, your corporate WAN, or the Internet. Many hops, technologies, and companies involved with these links can make it somewhat difficult to troubleshoot. Often, you work to rule out both Azure and your Corporate Networks first before jumping into this collection of companies and hops.

In the preceding diagram, on the far left is your corporate network. Depending on the size of your company, this routing domain can be a few network devices between you and the WAN or multiple layers of devices in a campus/enterprise network.

Given the complexities of these three different high-level network environments, it's often optimal to start at the edges and try to show where performance is good, and where it degrades. This approach can help identify the problem routing domain of the three and then focus your troubleshooting on that specific environment.

Tools

Most network issues can be analyzed and isolated using basic tools like ping and traceroute. It's rare that you need to go as deep as a packet analysis like Wireshark. To help with troubleshooting, the Azure Connectivity Toolkit (AzureCT) was developed to put some of these tools in an easy package. For performance testing, I like to use iPerf and PSPing. iPerf is a commonly used tool and runs on most operating systems. iPerf is good for basic performances tests and is fairly easy to use. PSPing is a ping tool developed by SysInternals. PSPing is an easy way to perform ICMP and TCP pings in one also easy to use command. Both of these tools are lightweight and are "installed" simply by coping the files to a directory on the host.

I've wrapped all of these tools and methods into a PowerShell module (AzureCT) that you can install and use.

AzureCT - the Azure Connectivity Toolkit

The AzureCT PowerShell module has two components [Availability Testing](#) and [Performance Testing](#). This document is only concerned with Performance testing, so lets focus on the two Link Performance commands in this PowerShell module.

There are three basic steps to use this toolkit for Performance testing. 1) Install the PowerShell module, 2) Install the supporting applications iPerf and PSPing 3) Run the performance test.

1. Installing the PowerShell Module

```
(new-object Net.WebClient).DownloadString("https://aka.ms/AzureCT") | Invoke-Expression
```

This command downloads the PowerShell module and installs it locally.

2. Install the supporting applications

```
Install-LinkPerformance
```

This AzureCT command installs iPerf and PSPing in a new directory "C:\ACTTools", it also opens the Windows Firewall ports to allow ICMP and port 5201 (iPerf) traffic.

3. Run the performance test

First, on the remote host you must install and run iPerf in server mode. Also ensure the remote host is listening on either 3389 (RDP for Windows) or 22 (SSH for Linux) and allowing traffic on port 5201 for iPerf. If the remote host is windows, install the AzureCT and run the Install-LinkPerformance command to set up iPerf and the firewall rules needed to start iPerf in server mode successfully.

Once the remote machine is ready, open PowerShell on the local machine and start the test:

```
Get-LinkPerformance -RemoteHost 10.0.0.1 -TestSeconds 10
```

This command runs a series of concurrent load and latency tests to help estimate the bandwidth capacity and latency of your network link.

4. Review the output of the tests

The PowerShell output format looks similar to:

Name	Bandwidth	Loss	P50
No Load	N/A	0%	1.87ms
1 Session	6.79 Gbits/sec	0%	0.92ms
6 Sessions	8.39 Gbits/sec	0%	1.94ms
16 Sessions	7.50 Gbits/sec	0%	4.34ms
16 Sessions with 1Mb window	7.33 Gbits/sec	0%	19.405ms
32 Sessions	7.17 Gbits/sec	0%	8.335ms

The detailed results of all the iPerf and PSPing tests are in individual text files in the AzureCT tools directory at "C:\ACTTools."

Troubleshooting

If the performance test is not giving you expected results, figuring out why should be a progressive step-by-step process. Given the number of components in the path, a systematic approach generally provides a faster path to resolution than jumping around and potentially needlessly doing the same testing multiple times.

NOTE

The scenario here is a performance issue, not a connectivity issue. The steps would be different if traffic wasn't passing at all.

First, challenge your assumptions. Is your expectation reasonable? For instance, if you have a 1-Gbps ExpressRoute circuit and 100 ms of latency it's unreasonable to expect the full 1 Gbps of traffic given the performance characteristics of TCP over high latency links. See the [References section](#) for more on performance assumptions.

Next, I recommend starting at the edges between routing domains and try to isolate the problem to a single major routing domain; the Corporate Network, the WAN, or the Azure Network. People often blame the "black box" in the path, while blaming the black box is easy to do, it may significantly delay resolution especially if the problem is actually in an area that you have the ability to make changes. Make sure you do your due diligence before handing off to your service provider or ISP.

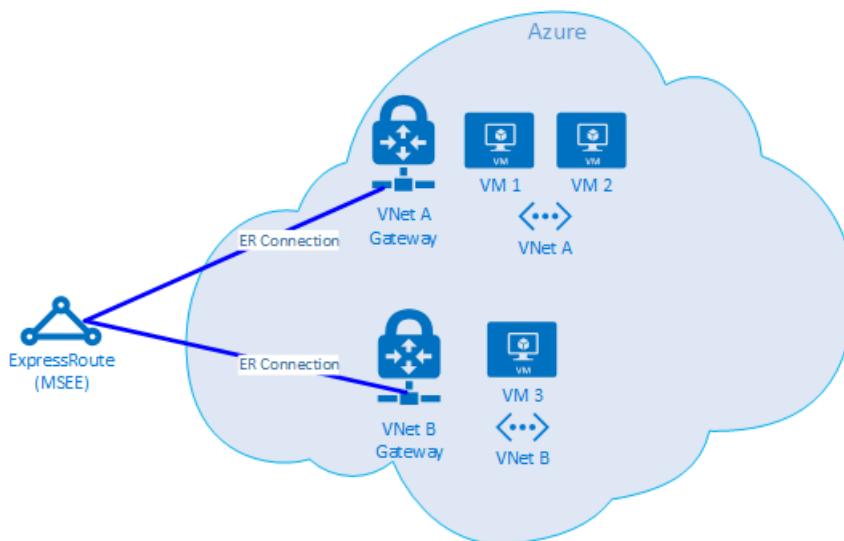
Once you've identified the major routing domain that appears to contain the problem, you should create a diagram of the area in question. Either on a whiteboard, notepad, or Visio as a diagram provides a concrete "battle map" to allow a methodical approach to further isolate the problem. You can plan testing points, and update the map as you clear areas or dig deeper as the testing progresses.

Now that you have a diagram, start to divide the network into segments and narrow the problem down. Find out where it works and where it doesn't. Keep moving your testing points to isolate down to the offending component.

Also, don't forget to look at other layers of the OSI model. It's easy to focus on the network and layers 1 - 3 (Physical, Data, and Network layers) but the problems can also be up at Layer 7 in the application layer. Keep an open mind and verify assumptions.

Advanced ExpressRoute troubleshooting

If you're not sure where the edge of the cloud actually is, isolating the Azure components can be a challenge. When ExpressRoute is used, the edge is a network component called the Microsoft Enterprise Edge (MSEE). **When using ExpressRoute**, the MSEE is the first point of contact into Microsoft's network, and the last hop leaving the Microsoft network. When you create a connection object between your VNet gateway and the ExpressRoute circuit, you're actually making a connection to the MSEE. Recognizing the MSEE as the first or last hop (depending on which direction you're going) is crucial to isolating Azure Network problems to either prove the issue is in Azure or further downstream in the WAN or the Corporate Network.



NOTE

Notice that the MSEE isn't in the Azure cloud. ExpressRoute is actually at the edge of the Microsoft network not actually in Azure. Once you're connected with ExpressRoute to an MSEE, you're connected to Microsoft's network, from there you can then go to any of the cloud services, like Office 365 (with Microsoft Peering) or Azure (with Private and/or Microsoft Peering).

If two VNets (VNets A and B in the diagram) are connected to the **same** ExpressRoute circuit, you can perform a series of tests to isolate the problem in Azure (or prove it's not in Azure)

Test plan

1. Run the Get-LinkPerformance test between VM1 and VM2. This test provides insight to if the problem is local or not. If this test produces acceptable latency and bandwidth results, you can mark the local VNet network as good.
2. Assuming the local VNet traffic is good, run the Get-LinkPerformance test between VM1 and VM3. This test exercises the connection through the Microsoft network down to the MSEE and back into Azure. If this test produces acceptable latency and bandwidth results, you can mark the Azure network as good.
3. If Azure is ruled out, you can perform a similar sequence of tests on your Corporate Network. If that also tests well, it's time to work with your service provider or ISP to diagnose your WAN connection. Example: Run this test between two branch offices, or between your desk and a data center server. Depending on what you're testing, find endpoints (servers, PCs, etc.) that can exercise that path.

IMPORTANT

It's critical that for each test you mark the time of day you run the test and record the results in a common location (I like OneNote or Excel). Each test run should have identical output so you can compare the resultant data across test runs and not have "holes" in the data. Consistency across multiple tests is the primary reason I use the AzureCT for troubleshooting. The magic isn't in the exact load scenarios I run, but instead the *magic* is the fact that I get a *consistent test and data output* from each and every test. Recording the time and having consistent data every single time is especially helpful if you later find that the issue is sporadic. Be diligent with your data collection up front and you'll avoid hours of retesting the same scenarios (I learned this hard way many years ago).

The problem is isolated, now what?

The more you can isolate the problem the easier it is to fix, however often you reach the point where you can't go deeper or further with your troubleshooting. Example: you see the link across your service provider taking hops through Europe, but your expected path is all in Asia. This point is when you should reach out for help. Who you ask is dependent on the routing domain you isolated the issue to, or even better if you are able to narrow it down to a specific component.

For corporate network issues, your internal IT department or service provider supporting your network (which may be the hardware manufacturer) may be able to help with device configuration or hardware repair.

For the WAN, sharing your testing results with your Service Provider or ISP may help get them started and avoid covering some of the same ground you've tested already. However, don't be offended if they want to verify your results themselves. "Trust but verify" is a good motto when troubleshooting based on other people's reported results.

With Azure, once you isolate the issue in as much detail as you're able, it's time to review the [Azure Network Documentation](#) and then if still needed [open a support ticket](#).

References

Latency/bandwidth expectations

TIP

Geographic latency (miles or kilometers) between the end points you're testing is by far the largest component of latency. While there is equipment latency (physical and virtual components, number of hops, etc.) involved, geography has proven to be the largest component of overall latency when dealing with WAN connections. It's also important to note that the distance is the distance of the fiber run not the straight-line or road map distance. This distance is incredibly hard to get with any accuracy. As a result, I generally use a city distance calculator on the internet and know that this method is a grossly inaccurate measure, but is enough to set a general expectation.

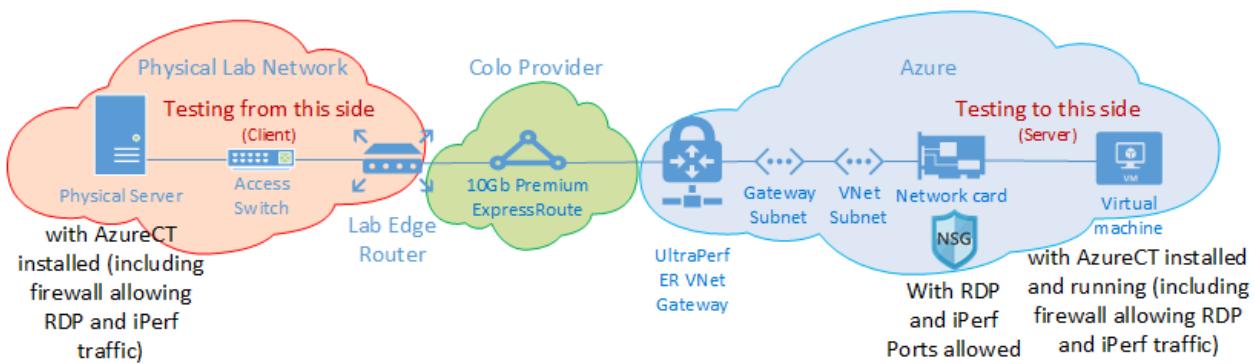
I've got an ExpressRoute setup in Seattle, Washington in the USA. The following table shows the latency and bandwidth I saw testing to various Azure locations. I've estimated the geographic distance between each end of the test.

Test setup:

- A physical server running Windows Server 2016 with a 10 Gbps NIC, connected to an ExpressRoute circuit.
- A 10Gbps Premium ExpressRoute circuit in the location identified with Private Peering enabled.
- An Azure VNet with an UltraPerformance gateway in the specified region.
- A DS5v2 VM running Windows Server 2016 on the VNet. The VM was non-domain joined, built from the default Azure image (no optimization or customization) with AzureCT installed.
- All testing was using the AzureCT Get-LinkPerformance command with a 5-minute load test for each of the six test runs. For example:

```
Get-LinkPerformance -RemoteHost 10.0.0.1 -TestSeconds 300
```

- The data flow for each test had the load flowing from the on-premises physical server (iPerf client in Seattle) up to the Azure VM (iPerf server in the listed Azure region).
- The "Latency" column data is from the No Load test (a TCP latency test without iPerf running).
- The "Max Bandwidth" column data is from the 16 TCP flow load test with a 1-Mb window size.



Latency/bandwidth results

IMPORTANT

These numbers are for general reference only. Many factors affect latency, and while these values are generally consistent over time, conditions within Azure or the Service Providers network can send traffic via different paths at any time, thus latency and bandwidth can be affected. Generally, the effects of these changes don't result in significant differences.

ExpressRoute Location	Azure Region	Estimated Distance (km)	Latency	1 Session Bandwidth	Maximum Bandwidth
Seattle	West US 2	191 km	5 ms	262.0 Mbits/sec	3.74 Gbits/sec
Seattle	West US	1,094 km	18 ms	82.3 Mbits/sec	3.70 Gbits/sec
Seattle	Central US	2,357 km	40 ms	38.8 Mbits/sec	2.55 Gbits/sec
Seattle	South Central US	2,877 km	51 ms	30.6 Mbits/sec	2.49 Gbits/sec
Seattle	North Central US	2,792 km	55 ms	27.7 Mbits/sec	2.19 Gbits/sec
Seattle	East US 2	3,769 km	73 ms	21.3 Mbits/sec	1.79 Gbits/sec
Seattle	East US	3,699 km	74 ms	21.1 Mbits/sec	1.78 Gbits/sec
Seattle	Japan East	7,705 km	106 ms	14.6 Mbits/sec	1.22 Gbits/sec
Seattle	UK South	7,708 km	146 ms	10.6 Mbits/sec	896 Mbits/sec
Seattle	West Europe	7,834 km	153 ms	10.2 Mbits/sec	761 Mbits/sec
Seattle	Australia East	12,484 km	165 ms	9.4 Mbits/sec	794 Mbits/sec
Seattle	Southeast Asia	12,989 km	170 ms	9.2 Mbits/sec	756 Mbits/sec
Seattle	Brazil South *	10,930 km	189 ms	8.2 Mbits/sec	699 Mbits/sec
Seattle	South India	12,918 km	202 ms	7.7 Mbits/sec	634 Mbits/sec

* The latency to Brazil is a good example where the straight-line distance significantly differs from the fiber run distance. I would expect that the latency would be in the neighborhood of 160 ms, but is actually 189 ms. This difference against my expectation could indicate a network issue somewhere, but most likely that the fiber run does not go to Brazil in a straight line and has an extra 1,000 km or so of travel to get to Brazil from Seattle.

Next steps

1. Download the Azure Connectivity Toolkit from GitHub at <https://aka.ms/AzCT>
2. Follow the instructions for [link performance testing](#)

Reset a failed ExpressRoute circuit

11/7/2019 • 2 minutes to read • [Edit Online](#)

When an operation on an ExpressRoute circuit does not complete successfully, the circuit may go into a 'failed' state. This article helps you reset a failed Azure ExpressRoute circuit.

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

```
Get-Module -ListAvailable Az
```

Reset a circuit

1. Install the latest version of the Azure Resource Manager PowerShell cmdlets. For more information, see [Install and configure Azure PowerShell](#).
2. Open your PowerShell console with elevated privileges, and connect to your account. Use the following example to help you connect:

```
Connect-AzAccount
```

3. If you have multiple Azure subscriptions, check the subscriptions for the account.

```
Get-AzSubscription
```

4. Specify the subscription that you want to use.

```
Select-AzSubscription -SubscriptionName "Replace_with_your_subscription_name"
```

5. Run the following commands to reset a circuit that is in a failed state:

```
$ckt = Get-AzExpressRouteCircuit -Name "ExpressRouteARMCircuit" -ResourceGroupName  
"ExpressRouteResourceGroup"  
  
Set-AzExpressRouteCircuit -ExpressRouteCircuit $ckt
```

The circuit should now be healthy. Open a support ticket with [Microsoft support](#) if the circuit is still in a failed state.

Next steps

Open a support ticket with [Microsoft support](#) if you are still experiencing issues.

Getting ARP tables in the Resource Manager deployment model

11/13/2019 • 6 minutes to read • [Edit Online](#)

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.

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

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).

NOTE

If layer 3 is provided by the service provider and the ARP tables are blank in the portal/output below, refresh the Circuit configuration using the refresh button on the portal. This operation will apply the right routing configuration on your circuit.

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-AzExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePrivatePeering -DevicePath Primary

# ARP table for Azure private peering - Secondary path
Get-AzExpressRouteCircuitARPTable -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-AzExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType
AzurePublicPeering -DevicePath Primary

# ARP table for Azure public peering - Secondary path
Get-AzExpressRouteCircuitARPTable -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-AzExpressRouteCircuitARPTable -ResourceGroupName $RG -ExpressRouteCircuitName $Name -PeeringType MicrosoftPeering -DevicePath Primary

# ARP table for Microsoft peering - Secondary path
Get-AzExpressRouteCircuitARPTable -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

If there are issues with the on-premises or connectivity provider you may see that either only one entry will appear in the ARP table or the on premises MAC address will show incomplete. 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

or

Age	InterfaceProperty	IpAddress	MacAddress
0	On-Prem	65.0.0.1	Incomplete
0	Microsoft	65.0.0.2	aaaa.bbbb.cccc

NOTE

Open a support request with your connectivity provider to debug such issues. 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.

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.

Getting ARP tables in the classic deployment model

11/13/2019 • 4 minutes to read • [Edit Online](#)

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	Interface	Property	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 side 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.

ExpressRoute FAQ

1/8/2020 • 26 minutes to read • [Edit Online](#)

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. They 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.

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 is 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 necessary?

Yes. ExpressRoute circuits are configured to allow you to burst up to two times the bandwidth limit you procured for no additional cost. Check with your service provider to see if they support this capability. This is not for a sustained period of time and is not guaranteed. If traffic flows through an ExpressRoute Gateway, the bandwidth for the sku is fixed and not burstable.

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

Yes. An ExpressRoute circuit, once set up, allows you to access services within a virtual network and other Azure services simultaneously. You connect to virtual networks over the private peering path, and to other services over the Microsoft peering path.

How are VNets advertised on ExpressRoute Private Peering?

The ExpressRoute gateway will advertise the *Address Space* of the Azure VNet, you can't include/exclude at the subnet level. It is always the VNet Address Space that is advertised. Also, if VNet Peering is used and the peered VNet has "Use Remote Gateway" enabled, the Address Space of the peered VNet will also

be advertised.

Can I filter routes coming from my on-premises network?

The only way to filter/include routes is on the on-premises edge router. User-defined Routes can be added in the VNet to affect specific routing, but this will be static and not part of the BGP advertisement.

Does ExpressRoute offer a Service Level Agreement (SLA)?

For information, see the [ExpressRoute SLA](#) page.

Supported services

ExpressRoute supports [three routing domains](#) for various types of services: private peering, Microsoft peering, and public peering (deprecated).

Private peering

Supported:

- Virtual networks, including all virtual machines and cloud services

Microsoft peering

If your ExpressRoute circuit is enabled for Azure Microsoft peering, you can access the [public IP address ranges](#) used in Azure over the circuit. Azure Microsoft peering will provide access to services currently hosted on Azure (with geo-restrictions depending on your circuit's SKU). To validate availability for a specific service, you can check the documentation for that service to see if there is a reserved range published for that service. Then, look up the IP ranges of the target service and compare with the ranges listed in the [Azure IP Ranges and Service Tags – Public Cloud XML file](#). Alternatively, you can open a support ticket for the service in question for clarification.

Supported:

- [Office 365](#)
- Power BI - Available via an Azure Regional Community, see [here](#) for how to find out the region of your Power BI tenant.
- Azure Active Directory
- [Windows Virtual Desktop](#)
- [Azure DevOps](#) (Azure Global Services community)
- Azure Public IP addresses for IaaS (Virtual Machines, Virtual Network Gateways, Load Balancers, etc)
- Most of the other Azure services are also supported. Please check directly with the service that you want to use to verify support.

Not supported:

- CDN
- Azure Front Door
- Multi-factor Authentication Server (legacy)
- Traffic Manager

Public peering

Public peering has been disabled on new ExpressRoute circuits. Azure services are now available on Microsoft peering. If you a circuit that was created prior to public peering being deprecated, you can choose to use Microsoft peering or public peering, depending on the services that you want.

For more information and configuration steps for public peering, see [ExpressRoute public peering](#).

Why I see 'Advertised public prefixes' status as 'Validation needed', while configuring Microsoft peering?

Microsoft verifies if the specified 'Advertised public prefixes' and 'Peer ASN' (or 'Customer ASN') are assigned to you in the Internet Routing Registry. If you are getting the public prefixes from another entity and if the assignment is not recorded with the routing registry, the automatic validation will not complete and will require manual validation. If the automatic validation fails, you will see the message 'Validation needed'.

If you see the message 'Validation needed', collect the document(s) that show the public prefixes are assigned to your organization by the entity that is listed as the owner of the prefixes in the routing registry and submit these documents for manual validation by opening a support ticket as shown below.

Home > Help + support > New support request

New support request

Basics Solutions Details Review + create

Create a new support request to get assistance with billing, subscription, technical (including advisory) or quota management issues. Complete the Basics tab by selecting the options that best describe your problem. Providing detailed, accurate information can help to solve your issues faster.

* Issue type: Technical

* Subscription: ExpressRoute-Lab (467fbb45-d411-4071-a2a1-e540c6d152a1) [Show more](#)

* Service: My services (selected) All services

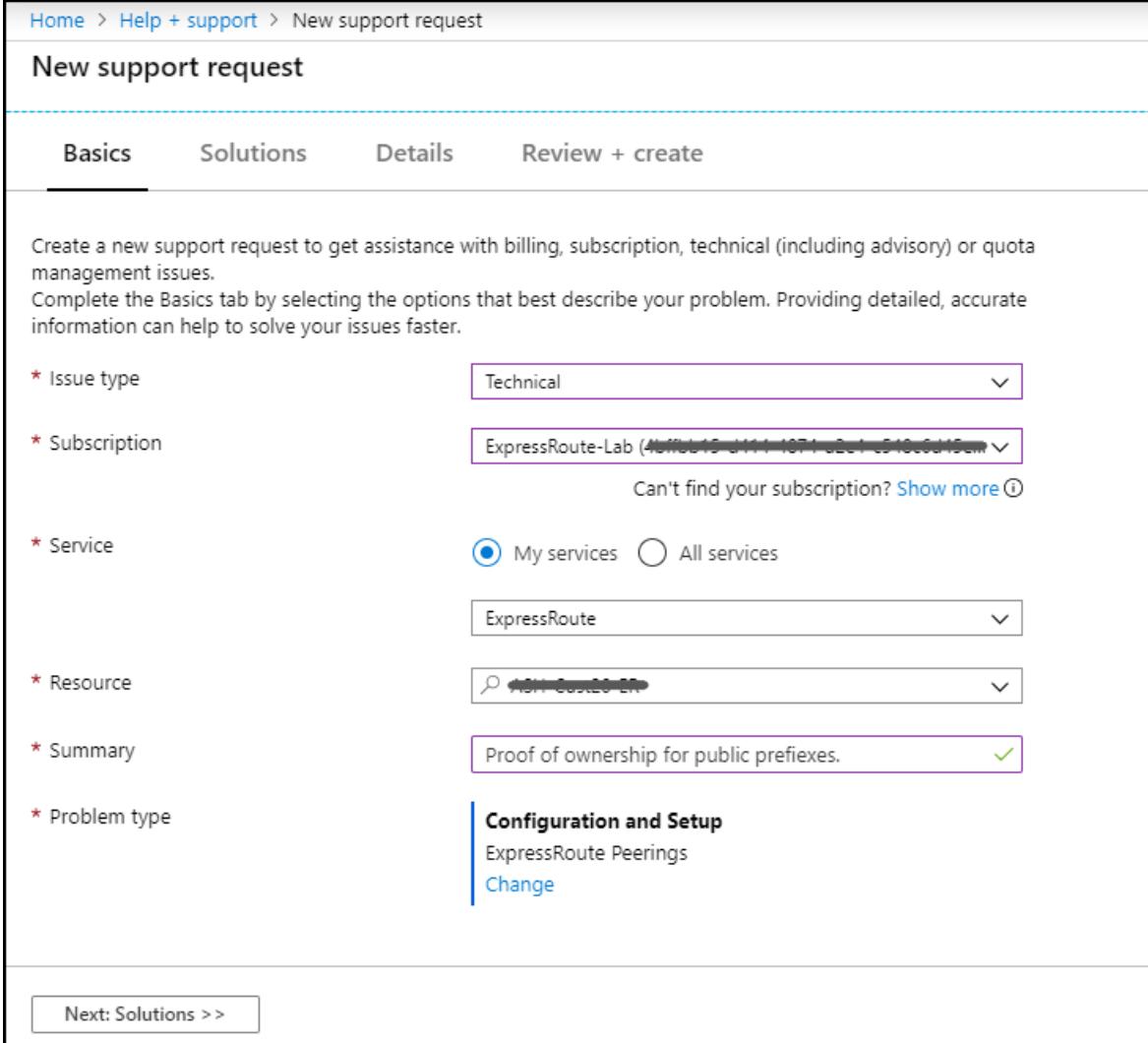
ExpressRoute

* Resource:

* Summary: Proof of ownership for public prefixes. [Change](#)

Configuration and Setup
ExpressRoute Peerings
[Change](#)

Next: Solutions >>



Is Dynamics 365 supported on ExpressRoute?

Dynamics 365 and Common Data Service (CDS) environments are hosted on Azure and therefore customers benefit from the underlying ExpressRoute support for Azure resources. You can connect to its service endpoints if your router filter includes the Azure regions your Dynamics 365/CDS environments are hosted in.

NOTE

ExpressRoute Premium is **not** required for Dynamics 365 connectivity via Azure ExpressRoute.

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, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps

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?

See [ExpressRoute prerequisites page](#) for requirements.

Are connections to ExpressRoute redundant?

Yes. Each ExpressRoute 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 and provide high availability of your ExpressRoute circuit. You can additionally create a circuit in a different peering location to achieve circuit-level resilience.

How do I implement redundancy on private peering?

Multiple ExpressRoute circuits from different peering locations can be connected to the same virtual network to provide high-availability in the case that a single circuit becomes unavailable. You can then [assign higher weights](#) to the local connection to favor prefer a specific circuit. It is strongly recommended that customers setup at least two ExpressRoute circuits to avoid single points of failure.

See [here](#) for designing for high availability and [here](#) for designing for disaster recovery.

How do I implement redundancy on Microsoft peering?

It is highly recommended when customers are using Microsoft peering to access Azure public services like Azure Storage or Azure SQL, as well as customers that are using Microsoft peering for Office 365 that they implement multiple circuits in different peering locations to avoid single points of failure.

Customers can either advertise the same prefix on both circuits and use [AS PATH prepending](#) or advertise different prefixes to determine path from on-premises.

See [here](#) for designing for high availability.

How do I ensure high availability on a virtual network connected to ExpressRoute?

You can achieve high availability by connecting ExpressRoute circuits in different peering locations (for example, Singapore, Singapore2) to your virtual network. If one ExpressRoute circuit goes down, connectivity will fail over to another ExpressRoute circuit. By default, traffic leaving your virtual network is routed based on Equal Cost Multi-path Routing (ECMP). You can use Connection Weight to prefer one circuit to another. For more information, see [Optimizing ExpressRoute Routing](#).

How do I ensure that my traffic destined for Azure Public services like Azure Storage and Azure SQL on Microsoft peering or public peering is preferred on the ExpressRoute path?

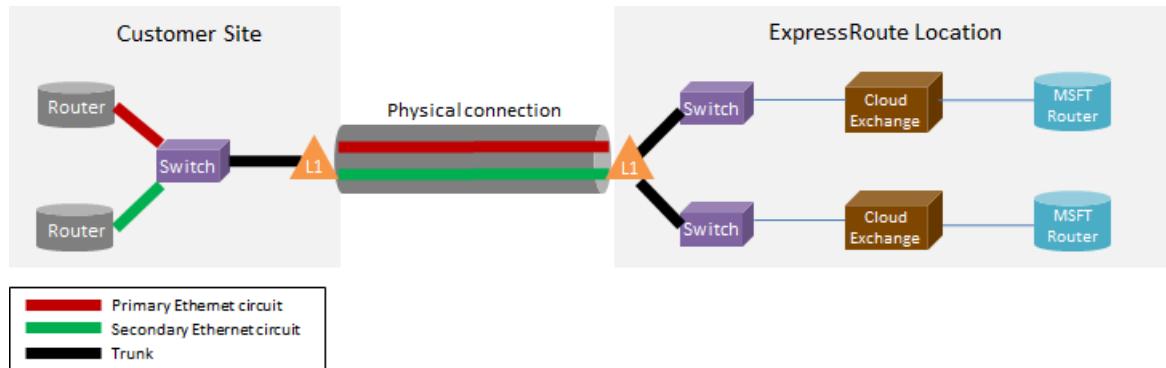
You must implement the *Local Preference* attribute on your router(s) to ensure that the path from on-premises to Azure is always preferred on your ExpressRoute circuit(s).

See additional details [here](#) on BGP path selection and common router configurations.

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?

If your service provider can establish two Ethernet virtual circuits over the physical connection, you only need one physical connection. The physical connection (for example, an optical fiber) is terminated on a layer 1 (L1) device (see the image). 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 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 is associated with one service provider only.

I see two ExpressRoute peering locations in the same metro, for example, Singapore and Singapore2. Which peering location should I choose to create my ExpressRoute circuit?

If your service provider offers ExpressRoute at both sites, you can work with your provider and pick either site to set up ExpressRoute.

Can I have multiple ExpressRoute circuits in the same metro? Can I link them to the same virtual network?

Yes. You can have multiple ExpressRoute circuits with the same or different service providers. If the metro has multiple ExpressRoute peering locations and the circuits are created at different peering locations, you can link them to the same virtual network. If the circuits are created at the same peering location, you can link up to 4 circuits to the same virtual network.

How do I connect my virtual networks to an ExpressRoute circuit

The basic steps are:

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

For more information, see [ExpressRoute workflows for circuit provisioning and circuit states](#).

Are there connectivity boundaries for my ExpressRoute circuit?

Yes. The [ExpressRoute partners and locations](#) article 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 have up to 10 virtual networks connections on a standard ExpressRoute circuit, and up to 100 on a [premium 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 link up to 10 virtual networks in the same subscription as the circuit or different subscriptions using a single ExpressRoute circuit. This limit can be increased by enabling the ExpressRoute premium feature.

For more information, see [Sharing an ExpressRoute circuit across multiple subscriptions](#).

I have multiple Azure subscriptions associated to different Azure Active Directory tenants or Enterprise Agreement enrollments. Can I connect virtual networks that are in separate tenants and enrollments to a single ExpressRoute circuit not in the same tenant or enrollment?

Yes. ExpressRoute authorizations can span subscription, tenant, and enrollment boundaries with no additional configuration required.

For more information, see [Sharing an ExpressRoute circuit across multiple subscriptions](#).

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

No. From a routing perspective, all virtual networks linked to the same ExpressRoute circuit are part of the same routing domain and are not isolated from each other. If you need route isolation, you 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 four ExpressRoute circuits in either the same or different peering 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 can 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.

If you advertise default routes, we force traffic to services offered over Microsoft 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 Microsoft peering path or over the Internet. If you've enabled a service endpoint for the service, the traffic to the service is not forced to your premises. The traffic remains within the Azure backbone network. To learn more about service endpoints, see [Virtual network service endpoints](#)

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. See [Configure ExpressRoute and site-to-site coexisting connections](#).

Why is there a public IP address associated with the ExpressRoute gateway on a virtual network?

The public IP address is used for internal management only, and does not constitute a security exposure of your virtual network.

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

Yes. We accept up to 4000 route prefixes for private peering and 200 for 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) for the Microsoft peering BGP session. We accept any prefix size (up to /32) on both the Microsoft and the private peering.

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. It is possible for you to configure different timers, and the BGP session parameters will be negotiated accordingly.

Can I change the bandwidth of an ExpressRoute circuit?

Yes, you can attempt to increase the bandwidth of your ExpressRoute circuit in the Azure portal, or by using PowerShell. If there is capacity available on the physical port on which your circuit was created, your change succeeds.

If your change fails, it means either there isn't enough capacity left on the current port and you need to create a new ExpressRoute circuit with the higher bandwidth, or that there is no additional capacity at that location, in which case you won't be able to increase the bandwidth.

You will also have to follow up with your connectivity provider to ensure that they update the throttles within their networks to support the bandwidth increase. You cannot, however, reduce the bandwidth of your ExpressRoute circuit. You have to create a new ExpressRoute circuit with lower bandwidth and delete the old circuit.

How do I change the bandwidth of an ExpressRoute circuit?

You can update the bandwidth of the ExpressRoute circuit using the REST API or PowerShell cmdlet.

ExpressRoute premium

What is ExpressRoute premium?

ExpressRoute premium is a collection of the following features:

- Increased routing table limit from 4000 routes to 10,000 routes for private peering.
- Increased number of VNets and ExpressRoute Global Reach connections that can be enabled on an ExpressRoute circuit (default is 10). For more information, see the [ExpressRoute Limits](#) table.
- Connectivity to Office 365
- Global connectivity over the Microsoft core network. You can now link a VNet in one geopolitical region with an ExpressRoute circuit in another region.

Examples:

- You can link a VNet created in Europe West to an ExpressRoute circuit created in Silicon Valley.
- On the Microsoft peering, prefixes from other geopolitical regions are advertised such that you can connect to, for example, SQL Azure in Europe West from a circuit in Silicon Valley.

How many VNets and ExpressRoute Global Reach connections can I enable on an ExpressRoute circuit if I enabled ExpressRoute premium?

The following tables show the ExpressRoute limits and the number of VNets and ExpressRoute Global

Reach connections per ExpressRoute circuit:

RESOURCE	DEFAULT/MAXIMUM LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription, with Azure Resource Manager	10
Maximum number of routes advertised to Azure private peering with ExpressRoute Standard	4,000
Maximum number of routes advertised to Azure private peering with ExpressRoute Premium add-on	10,000
Maximum number of routes advertised from Azure private peering from the VNet address space for an ExpressRoute connection	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Standard	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Premium add-on	200
Maximum number of ExpressRoute circuits linked to the same virtual network in the same peering location	4
Maximum number of ExpressRoute circuits linked to the same virtual network in different peering locations	4
Number of virtual network links allowed per ExpressRoute circuit	See the Number of virtual networks per ExpressRoute circuit table.

Number of virtual networks per ExpressRoute circuit

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

CIRCUIT SIZE	NUMBER OF VIRTUAL NETWORK LINKS FOR STANDARD	NUMBER OF VIRTUAL NETWORK LINKS WITH PREMIUM ADD-ON
40 Gbps*	10	100
100 Gbps*	10	100

*100 Gbps ExpressRoute Direct Only

NOTE

Global Reach connections count against the limit of virtual network connections per ExpressRoute Circuit. For example, a 10 Gbps Premium Circuit would allow for 5 Global Reach connections and 95 connections to the ExpressRoute Gateways or 95 Global Reach connections and 5 connections to the ExpressRoute Gateways or any other combination up to the limit of 100 connections for the circuit.

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 REST API / PowerShell cmdlet.

How do I disable ExpressRoute premium?

You can disable ExpressRoute premium by calling the REST API or PowerShell cmdlet. You must make sure that you have scaled your connectivity needs to meet the default limits before you disable ExpressRoute premium. If your utilization scales beyond the default limits, the request to disable ExpressRoute premium fails.

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

No. You can't pick the features. 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 Local

What is ExpressRoute Local?

ExpressRoute Local is a SKU of ExpressRoute circuit, in addition to the Standard SKU and the Premium SKU. A key feature of Local is that a Local circuit at an ExpressRoute peering location gives you access only to one or two Azure regions in or near the same metro. In contrast, a Standard circuit gives you access to all Azure regions in a geopolitical area and a Premium circuit to all Azure regions globally.

What are the benefits of ExpressRoute Local?

While you need to pay egress data transfer for your Standard or Premium ExpressRoute circuit, you don't pay egress data transfer separately for your ExpressRoute Local circuit. In other words, the price of ExpressRoute Local includes data transfer fees. ExpressRoute Local is a more economical solution if you have massive amount of data to transfer and you can bring your data over a private connection to an ExpressRoute peering location near your desired Azure regions.

What features are available and what are not on ExpressRoute Local?

Compared to a Standard ExpressRoute circuit, a Local circuit has the same set of features except:

- Scope of access to Azure regions as described above
- ExpressRoute Global Reach is not available on Local

ExpressRoute Local also has the same limits on resources (e.g. the number of VNets per circuit) as Standard.

Where is ExpressRoute Local available and which Azure regions is each peering location mapped to?

ExpressRoute Local is available at the peering locations where one or two Azure regions are close-by. It is not available at a peering location where there is no Azure region in that state or province or country. Please see the exact mappings on [the Locations page](#).

ExpressRoute for Office 365

Office 365 was created to be accessed securely and reliably via the Internet. Because of this, we recommend ExpressRoute for specific scenarios. For information about using ExpressRoute to access Office 365, visit [Azure ExpressRoute for Office 365](#).

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

1. Review the [ExpressRoute prerequisites page](#) to make sure you meet the requirements.
2. To ensure that your connectivity needs are met, review the list of service providers and locations in the [ExpressRoute partners and locations](#) article.
3. Plan your capacity requirements by reviewing [Network planning and performance tuning for Office 365](#).
4. Follow the steps listed in the workflows to set up connectivity [ExpressRoute workflows for circuit provisioning and circuit states](#).

IMPORTANT

Make sure that you have enabled ExpressRoute premium add-on when configuring connectivity to Office 365 services.

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

Yes. Your existing ExpressRoute circuit can be configured to support connectivity to Office 365 services. Make sure that you have sufficient capacity to connect to Office 365 services and that you have enabled premium add-on. [Network planning and performance tuning for Office 365](#) helps 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 cost?

Office 365 services require premium add-on to be enabled. See the [pricing details page](#) for costs.

What regions is ExpressRoute for Office 365 supported in?

See [ExpressRoute partners and locations](#) for information.

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. Please check with your organization's networking team if the network at your location is configured to connect to Office 365 services through ExpressRoute.

How can I plan for high availability for Office 365 network traffic on Azure ExpressRoute?

See the recommendation for [High availability and failover with Azure ExpressRoute](#)

Can I access Office 365 US Government Community (GCC) services over an Azure US Government ExpressRoute circuit?

Yes. Office 365 GCC service endpoints are reachable through the Azure US Government ExpressRoute. However, you first need to open a support ticket on the Azure portal to provide the prefixes you intend to advertise to Microsoft. Your connectivity to Office 365 GCC services will be established after the support ticket is resolved.

Route filters for Microsoft peering

I am turning on Microsoft peering for the first time, what routes will I see?

You will not see any routes. You have to attach a route filter to your circuit to start prefix advertisements. For instructions, see [Configure route filters for Microsoft peering](#).

I turned on Microsoft peering and now I am trying to select Exchange Online, but it is giving me an error that I am not authorized to do it.

When using route filters, any customer can turn on Microsoft peering. However, for consuming Office 365 services, you still need to get authorized by Office 365.

I enabled Microsoft peering prior to August 1, 2017, how can I take advantage of route filters?

Your existing circuit will continue advertising the prefixes for Office 365. If you want to add Azure public prefixes advertisements over the same Microsoft peering, you can create a route filter, select the services you need advertised (including the Office 365 service(s) you need), and attach the filter to your Microsoft peering. For instructions, see [Configure route filters for Microsoft peering](#).

I have Microsoft peering at one location, now I am trying to enable it at another location and I am not seeing any prefixes.

- Microsoft peering of ExpressRoute circuits that were configured prior to August 1, 2017 will have all service prefixes advertised through Microsoft peering, even if route filters are not defined.
- Microsoft peering of ExpressRoute circuits that are configured on or after August 1, 2017 will not have any prefixes advertised until a route filter is attached to the circuit. You will see no prefixes by default.

ExpressRoute Direct

What is ExpressRoute Direct?

ExpressRoute Direct provides customers with the ability to connect directly into Microsoft's global network at peering locations strategically distributed across the world. ExpressRoute Direct provides dual 100 or 10 Gbps connectivity, which supports Active/Active connectivity at scale.

How do customers connect to ExpressRoute Direct?

Customers will need to work with their local carriers and co-location providers to get connectivity to ExpressRoute routers to take advantage of ExpressRoute Direct.

What locations currently support ExpressRoute Direct?

Please check the availability on the [location page](#).

What is the SLA for ExpressRoute Direct?

ExpressRoute Direct will utilize the same [enterprise-grade of ExpressRoute](#).

What scenarios should customers consider with ExpressRoute Direct?

ExpressRoute Direct provides customers with direct 100 or 10 Gbps port pairs into the Microsoft global

backbone. The scenarios that will provide customers with the greatest benefits include: Massive data ingestion, physical isolation for regulated markets, and dedicated capacity for burst scenario, like rendering.

What is the billing model for ExpressRoute Direct?

ExpressRoute Direct will be billed for the port pair at a fixed amount. Standard circuits will be included at no additional hours and premium will have a slight add-on charge. Egress will be billed on a per circuit basis based on the zone of the peering location.

When does billing start for the ExpressRoute Direct port pairs?

ExpressRoute Direct's port pairs are billed 45 days into the creation of the ExpressRoute Direct resource or when 1 or both of the links are enabled, whichever comes first. The 45-day grace period is granted to allow customers to complete the cross-connection process with the colocation provider.

Global Reach

What is ExpressRoute Global Reach?

ExpressRoute Global Reach is an Azure service that connects your on-premises networks via the ExpressRoute service through Microsoft's global network. For example, if you have a private data center in California connected to ExpressRoute in Silicon Valley and another private data center in Texas connected to ExpressRoute in Dallas, with ExpressRoute Global Reach, you can connect your private data centers together through the two ExpressRoute connections and your cross data center traffic will traverse through Microsoft's network backbone.

How do I enable or disable ExpressRoute Global Reach?

You enable ExpressRoute Global Reach by connecting your ExpressRoute circuits together. You disable the feature by disconnecting the circuits. See the [configuration](#).

Do I need ExpressRoute Premium for ExpressRoute Global Reach?

If your ExpressRoute circuits are in the same geopolitical region, you don't need ExpressRoute Premium to connect them together. If two ExpressRoute circuits are in different geopolitical regions, you need ExpressRoute Premium for both circuits in order to enable connectivity between them.

How will I be charged for ExpressRoute Global Reach?

ExpressRoute enables connectivity from your on-premises network to Microsoft cloud services. ExpressRoute Global Reach enables connectivity between your own on-premises networks via your existing ExpressRoute circuits, leveraging Microsoft's global network. ExpressRoute Global Reach is billed separately from the existing ExpressRoute service. There is an Add-on fee for enabling this feature on each ExpressRoute circuit. Traffic between your on-premises networks enabled by ExpressRoute Global Reach will be billed for an egress rate at the source and for an ingress rate at the destination. The rates are based on the zone at which the circuits are located.

Where is ExpressRoute Global Reach supported?

ExpressRoute Global Reach is supported in [select countries/regions or places](#). The ExpressRoute circuits must be created at the peering locations in those countries/regions or places.

I have more than two on-premises networks, each connected to an ExpressRoute circuit. Can I enable ExpressRoute Global Reach to connect all of my on-premises networks together?

Yes, you can, as long as the circuits are in the supported countries/regions. You need to connect two ExpressRoute circuits at a time. To create a fully meshed network, you need to enumerate all circuit pairs and repeat the configuration.

Can I enable ExpressRoute Global Reach between two ExpressRoute circuits at the same peering location?

No. The two circuits must be from different peering locations. If a metro in a supported country/region has more than one ExpressRoute peering location, you can connect together the ExpressRoute circuits created at different peering locations in that metro.

If ExpressRoute Global Reach is enabled between circuit X and circuit Y, and between circuit Y and circuit Z, will my on-premises networks connected to circuit X and circuit Z talk to each other via Microsoft's network?

No. To enable connectivity between any two of your on-premises networks, you must connect the corresponding ExpressRoute circuits explicitly. In the above example, you must connect circuit X and circuit Z.

What is the network throughput I can expect between my on-premises networks after I enable ExpressRoute Global Reach?

The network throughput between your on-premises networks, enabled by ExpressRoute Global Reach, is capped by the smaller of the two ExpressRoute circuits. Premises-to-Azure traffic and premises-to-premises traffic share the same circuit and are subject to the same bandwidth cap.

With ExpressRoute Global Reach, what are the limits on the number of routes I can advertise and the number of routes I will receive?

The number of routes you can advertise to Microsoft on Azure private peering remains at 4000 on a Standard circuit or 10000 on a Premium circuit. The number of routes you will receive from Microsoft on Azure private peering will be the sum of the routes of your Azure virtual networks and the routes from your other on-premises networks connected via ExpressRoute Global Reach. Please make sure you set an appropriate maximum prefix limit on your on-premises router.

What is the SLA for ExpressRoute Global Reach?

ExpressRoute Global Reach will provide the same [availability SLA](#) as the regular ExpressRoute service.

Azure subscription and service limits, quotas, and constraints

2/25/2020 • 85 minutes to read • [Edit Online](#)

This document lists some of the most common Microsoft Azure limits, which are also sometimes called quotas.

To learn more about Azure pricing, see [Azure pricing overview](#). There, you can estimate your costs by using the [pricing calculator](#). You also can go to the pricing details page for a particular service, for example, [Windows VMs](#). For tips to help manage your costs, see [Prevent unexpected costs with Azure billing and cost management](#).

Managing limits

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

[Free Trial subscriptions](#) aren't eligible for limit or quota increases. If you have a [Free Trial subscription](#), you can upgrade to a [Pay-As-You-Go](#) subscription. For more information, see [Upgrade your Azure Free Trial subscription to a Pay-As-You-Go subscription](#) and the [Free Trial subscription FAQ](#).

Some limits are managed at a regional level.

Let's use vCPU quotas as an example. To request a quota increase with support for vCPUs, you must decide how many vCPUs you want to use in which regions. You then make a specific request for Azure resource group vCPU quotas for the amounts and regions that you want. If you need to use 30 vCPUs in West Europe to run your application there, you specifically request 30 vCPUs in West Europe. Your vCPU quota isn't increased in any other region--only West Europe has the 30-vCPU quota.

As a result, decide what your Azure resource group quotas must be for your workload in any one region. Then request that amount in each region into which you want to deploy. For help in how to determine your current quotas for specific regions, see [Resolve errors for resource quotas](#).

General limits

For limits on resource names, see [Naming rules and restrictions for Azure resources](#).

For information about Resource Manager API read and write limits, see [Throttling Resource Manager requests](#).

Subscription limits

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Subscriptions per Azure Active Directory tenant	Unlimited.	Unlimited.
Coadministrators per subscription	Unlimited.	Unlimited.
Resource groups per subscription	980	980

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Resource Manager API request size	4,194,304 bytes.	4,194,304 bytes.
Tags per subscription ¹	Unlimited.	Unlimited.
Unique tag calculations per subscription ¹	10,000	10,000
Subscription-level deployments per location	800 ²	800

¹You can apply an unlimited number of tags per subscription. The number of tags per resource or resource group is limited to 50. Resource Manager returns a [list of unique tag name and values](#) in the subscription only when the number of tags is 10,000 or less. You still can find a resource by tag when the number exceeds 10,000.

²If you reach the limit of 800 deployments, delete deployments from the history that are no longer needed. To delete subscription level deployments, use [Remove-AzDeployment](#) or [az deployment delete](#).

Resource group limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resources per resource group	N/A	Resources aren't limited by resource group. Instead, they're limited by resource type in a resource group. See next row.
Resources per resource group, per resource type	800	Some resource types can exceed the 800 limit. See Resources not limited to 800 instances per resource group .
Deployments per resource group in the deployment history	800 ¹	800
Resources per deployment	800	800
Management locks per unique scope	20	20
Number of tags per resource or resource group	50	50
Tag key length	512	512
Tag value length	256	256

¹If you reach the limit of 800 deployments per resource group, delete deployments from the history that are no longer needed. Deleting an entry from the deployment history doesn't affect the deployed resources. For more information, see [Resolve error when deployment count exceeds 800](#).

Template limits

VALUE	DEFAULT LIMIT	MAXIMUM LIMIT
Parameters	256	256

Value	Default Limit	Maximum Limit
Variables	256	256
Resources (including copy count)	800	800
Outputs	64	64
Template expression	24,576 chars	24,576 chars
Resources in exported templates	200	200
Template size	4 MB	4 MB
Parameter file size	64 KB	64 KB

You can exceed some template limits by using a nested template. For more information, see [Use linked templates when you deploy Azure resources](#). To reduce the number of parameters, variables, or outputs, you can combine several values into an object. For more information, see [Objects as parameters](#).

Active Directory limits

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

Category	Limits
Directories	A single user can belong to a maximum of 500 Azure AD directories as a member or a guest. A single user can create a maximum of 20 directories.
Domains	You can add no more than 900 managed domain names. If you set up all of your domains for federation with on-premises Active Directory, you can add no more than 450 domain names in each directory.
Resources	<ul style="list-style-type: none"> A maximum of 50,000 Azure AD resources can be created in a single directory by users of the Free edition of Azure Active Directory by default. If you have at least one verified domain, the default directory service quota in Azure AD is extended to 300,000 Azure AD resources. A non-admin user can create no more than 250 Azure AD resources. Both active resources and deleted resources that are available to restore count toward this quota. Only deleted Azure AD resources that were deleted fewer than 30 days ago are available to restore. Deleted Azure AD resources that are no longer available to restore count toward this quota at a value of one-quarter for 30 days. If you have developers who are likely to repeatedly exceed this quota in the course of their regular duties, you can create and assign a custom role with permission to create a limitless number of app registrations.

CATEGORY	LIMITS
Schema extensions	<ul style="list-style-type: none"> String-type extensions can have a maximum of 256 characters. Binary-type extensions are limited to 256 bytes. Only 100 extension values, across <i>all</i> types and <i>all</i> applications, can be written to any single Azure AD resource. Only User, Group, TenantDetail, Device, Application, and ServicePrincipal entities can be extended with string-type or binary-type single-valued attributes. Schema extensions are available only in the Graph API version 1.21 preview. The application must be granted write access to register an extension.
Applications	A maximum of 100 users can be owners of a single application.
Application Manifest	A maximum of 1200 entries can be added in the Application Manifest.

CATEGORY	LIMITS
Groups	<ul style="list-style-type: none"> A user can create a maximum of 250 groups in an Azure AD organization. An Azure AD organization can have a maximum of 5000 dynamic groups. A maximum of 100 users can be owners of a single group. Any number of Azure AD resources can be members of a single group. A user can be a member of any number of groups. The number of members in a group that you can synchronize from your on-premises Active Directory to Azure Active Directory by using Azure AD Connect is limited to 50,000 members. Nested Groups in Azure AD are not supported within all scenarios <p>At this time the following are the supported scenarios with nested groups.</p> <ul style="list-style-type: none"> One group can be added as a member of another group and you can achieve group nesting. Group membership claims (when an app is configured to receive group membership claims in the token, nested groups the signed-in user is a member of are included) Conditional access (when scoping a conditional access policy to a group) Restricting access to self-serve password reset Restricting which users can do Azure AD Join and device registration <p>The following scenarios DO NOT support nested groups:</p> <ul style="list-style-type: none"> App role assignment (assigning groups to an app is supported, but groups nested within the directly assigned group will not have access), both for access and for provisioning Group-based licensing (assigning a license automatically to all members of a group) Office 365 Groups.
Application Proxy	<ul style="list-style-type: none"> A maximum of 500 transactions per second per App Proxy application A maximum of 750 transactions per second for the Azure AD organization <p>A transaction is defined as a single http request and response for a unique resource. When throttled, clients will receive a 429 response (too many requests).</p>

CATEGORY	LIMITS
Access Panel	<ul style="list-style-type: none"> There's no limit to the number of applications that can be seen in the Access Panel per user. This applies to users assigned licenses for Azure AD Premium or the Enterprise Mobility Suite. A maximum of 10 app tiles can be seen in the Access Panel for each user. This limit applies to users who are assigned licenses for Azure AD Free license plan. Examples of app tiles include Box, Salesforce, or Dropbox. This limit doesn't apply to administrator accounts.
Reports	A maximum of 1,000 rows can be viewed or downloaded in any report. Any additional data is truncated.
Administrative units	An Azure AD resource can be a member of no more than 30 administrative units.
Admin roles and permissions	<ul style="list-style-type: none"> A group cannot be added as an owner. A group cannot be assigned to a role. Users' ability to read other users' directory information cannot be restricted outside of the Azure AD organization-wide switch to disable all non-admin users' access to all directory information (not recommended). More information on default permissions here. It may take up to 15 minutes or signing out/signing in before admin role membership additions and revocations take effect.

API Management limits

RESOURCE	LIMIT
Maximum number of scale units	10 per region ¹
Cache size	5 GiB per unit ²
Concurrent back-end connections ³ per HTTP authority	2,048 per unit ⁴
Maximum cached response size	2 MiB
Maximum policy document size	256 KiB ⁵
Maximum custom gateway domains per service instance ⁶	20
Maximum number of CA certificates per service instance	10
Maximum number of service instances per subscription ⁷	20
Maximum number of subscriptions per service instance ⁷	500
Maximum number of client certificates per service instance ⁷	50

RESOURCE	LIMIT
Maximum number of APIs per service instance ⁷	50
Maximum number of API operations per service instance ⁷	1,000
Maximum total request duration ⁷	30 seconds
Maximum buffered payload size ⁷	2 MiB
Maximum request URL size ⁸	4096 bytes

¹Scaling limits depend on the pricing tier. To see the pricing tiers and their scaling limits, see [API Management pricing](#).

²Per unit cache size depends on the pricing tier. To see the pricing tiers and their scaling limits, see [API Management pricing](#).

³Connections are pooled and reused unless explicitly closed by the back end.

⁴This limit is per unit of the Basic, Standard, and Premium tiers. The Developer tier is limited to 1,024. This limit doesn't apply to the Consumption tier.

⁵This limit applies to the Basic, Standard, and Premium tiers. In the Consumption tier, policy document size is limited to 4 KiB.

⁶This resource is available in the Premium tier only.

⁷This resource applies to the Consumption tier only.

⁸Applies to the Consumption tier only. Includes an up to 2048 bytes long query string.

App Service limits

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

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
Web, mobile, or API apps per Azure App Service plan ¹	10	100	Unlimited ²	Unlimited ²	Unlimited ²	Unlimited ²
App Service plan	10 per region	10 per resource group	100 per resource group	100 per resource group	100 per resource group	100 per resource group
Compute instance type	Shared	Shared	Dedicated ³	Dedicated ³	Dedicated ³	Dedicated ³
Scale out (maximum instances)	1 shared	1 shared	3 dedicated ³	10 dedicated ³	30 dedicated ³	100 dedicated ⁴
Storage ⁵	1 GB ⁵	1 GB ⁵	10 GB ⁵	50 GB ⁵	250 GB ⁵	1 TB ⁵
CPU time (5 minutes) ⁶	3 minutes	3 minutes	Unlimited, pay at standard rates			

Resource	Free	Shared	Basic	Standard	Premium (V2)	Isolated
CPU time (day) ⁶	60 minutes	240 minutes	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates
Memory (1 hour)	1,024 MB per App Service plan	1,024 MB per app	N/A	N/A	N/A	N/A
Bandwidth	165 MB	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply
Application architecture	32-bit	32-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit
Web sockets per instance ⁷	5	35	350	Unlimited	Unlimited	Unlimited
IP connections	600	600	Depends on instance size ⁸	Depends on instance size ⁸	Depends on instance size ⁸	16,000
Concurrent debugger connections per application	1	1	1	5	5	5
App Service Certificates per subscription ⁹	Not supported	Not supported	10	10	10	10
Custom domains per app	0 (azurewebsites.net subdomain only)	500	500	500	500	500
Custom domain SSL support	Not supported, wildcard certificate for *.azurewebsites.net available by default	Not supported, wildcard certificate for *.azurewebsites.net available by default	Unlimited SNI SSL connections	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included
Hybrid connections per plan			5	25	200	200
Integrated load balancer		X	X	X	X	X ¹⁰
Always On			X	X	X	X

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
Scheduled backups				Scheduled backups every 2 hours, a maximum of 12 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)
Autoscale				X	X	X
WebJobs ¹¹	X	X	X	X	X	X
Endpoint monitoring			X	X	X	X
Staging slots				5	20	20
SLA			99.95%	99.95%	99.95%	99.95%

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

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

³Dedicated instances can be of different sizes. For more information, see [App Service pricing](#).

⁴More are allowed upon request.

⁵The storage limit is the total content size across all apps in the same App service plan. The total content size of all apps across all App service plans in a single resource group and region cannot exceed 500GB.

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

⁷If you scale an app in the Basic tier to two instances, you have 350 concurrent connections for each of the two instances. For Standard tier and above, there are no theoretical limits to web sockets, but other factors can limit the number of web sockets. For example, maximum concurrent requests allowed (defined by `maxConcurrentRequestsPerCpu`) are: 7,500 per small VM, 15,000 per medium VM (7,500 x 2 cores), and 75,000 per large VM (18,750 x 4 cores).

⁸The maximum IP connections are per instance and depend on the instance size: 1,920 per B1/S1/P1V2 instance, 3,968 per B2/S2/P2V2 instance, 8,064 per B3/S3/P3V2 instance.

⁹The App Service Certificate quota limit per subscription can be increased via a support request to a maximum limit of 200.

¹⁰App Service Isolated SKUs can be internally load balanced (ILB) with Azure Load Balancer, so there's no public connectivity from the internet. As a result, some features of an ILB Isolated App Service must be used from machines that have direct access to the ILB network endpoint.

¹¹Run custom executables and/or scripts on demand, on a schedule, or continuously as a background task within your App Service instance. Always On is required for continuous WebJobs execution. There's no predefined limit on the number of WebJobs that can run in an App Service instance. There are practical limits that depend on what the application code is trying to do.

Automation limits

Process automation

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum number of new jobs that can be submitted every 30 seconds per Azure Automation account (nonscheduled jobs)	100	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum number of concurrent running jobs at the same instance of time per Automation account (nonscheduled jobs)	200	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum storage size of job metadata for a 30-day rolling period	10 GB (approximately 4 million jobs)	When this limit is reached, the subsequent requests to create a job fail.
Maximum job stream limit	1MB	A single stream cannot be larger than 1 MB.
Maximum number of modules that can be imported every 30 seconds per Automation account	5	
Maximum size of a module	100 MB	
Job run time, Free tier	500 minutes per subscription per calendar month	
Maximum amount of disk space allowed per sandbox ¹	1 GB	Applies to Azure sandboxes only.
Maximum amount of memory given to a sandbox ¹	400 MB	Applies to Azure sandboxes only.
Maximum number of network sockets allowed per sandbox ¹	1,000	Applies to Azure sandboxes only.
Maximum runtime allowed per runbook ¹	3 hours	Applies to Azure sandboxes only.
Maximum number of Automation accounts in a subscription	No limit	
Maximum number of Hybrid Worker Groups per Automation Account	4,000	
Maximum number of concurrent jobs that can be run on a single Hybrid Runbook Worker	50	
Maximum runbook job parameter size	512 kilobits	
Maximum runbook parameters	50	If you reach the 50-parameter limit, you can pass a JSON or XML string to a parameter and parse it with the runbook.

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum webhook payload size	512 kilobits	
Maximum days that job data is retained	30 days	
Maximum PowerShell workflow state size	5 MB	Applies to PowerShell workflow runbooks when checkpointing workflow.

¹A sandbox is a shared environment that can be used by multiple jobs. Jobs that use the same sandbox are bound by the resource limitations of the sandbox.

Change Tracking and Inventory

The following table shows the tracked item limits per machine for change tracking.

RESOURCE	LIMIT	NOTES
File	500	
Registry	250	
Windows software	250	Doesn't include software updates.
Linux packages	1,250	
Services	250	
Daemon	250	

Update Management

The following table shows the limits for Update Management.

RESOURCE	LIMIT	NOTES
Number of machines per update deployment	1000	

Azure Cache for Redis limits

RESOURCE	LIMIT
Cache size	1.2 TB
Databases	64
Maximum connected clients	40,000
Azure Cache for Redis replicas, for high availability	1
Shards in a premium cache with clustering	10

Azure Cache for Redis limits and sizes are different for each pricing tier. To see the pricing tiers and their

associated sizes, see [Azure Cache for Redis pricing](#).

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

Because configuration and management of Azure Cache for Redis instances is done by Microsoft, not all Redis commands are supported in Azure Cache for Redis. For more information, see [Redis commands not supported in Azure Cache for Redis](#).

Azure Cloud Services limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Web or worker roles per deployment ¹	25	25
Instance input endpoints per deployment	25	25
Input endpoints per deployment	25	25
Internal endpoints per deployment	25	25
Hosted service certificates per deployment	199	199

¹Each Azure Cloud Service with web or worker roles can have two deployments, one for production and one for staging. This limit refers to the number of distinct roles, that is, configuration. This limit doesn't refer to the number of instances per role, that is, scaling.

Azure Cognitive Search limits

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

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

Limits per subscription

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

Maximum service limits can be raised upon request. If you need more services within the same subscription, contact Azure Support.

RESOURCE	FREE ¹	BASIC	S1	S2	S3	S3 HD	L1	L2
Maximum services	1	16	16	8	6	6	6	6

Resource	Free	Basic	S1	S2	S3	S3 HD	L1	L2
Maximum scale in search units (SU) ²	N/A	3 SU	36 SU	36 SU	36 SU	36 SU	36 SU	36 SU

¹ Free is based on shared, not dedicated, resources. Scale-up is not supported on shared resources.

² Search units are billing units, allocated as either a *replica* or a *partition*. You need both resources for storage, indexing, and query operations. To learn more about SU computations, see [Scale resource levels for query and index workloads](#).

Limits per search service

Storage is constrained by disk space or by a hard limit on the *maximum number* of indexes, document, or other high-level resources, whichever comes first. The following table documents storage limits. For maximum limits on indexes, documents, and other objects, see [Limits by resource](#).

Resource	Free	Basic ¹	S1	S2	S3	S3 HD ²	L1	L2
Service level agreement (SLA) ³	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage per partition	50 MB	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Partitions per service	N/A	1	12	12	12	3	12	12
Partition size	N/A	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Replicas	N/A	3	12	12	12	12	12	12

¹ Basic has one fixed partition. At this tier, additional search units are used for allocating more replicas for increased query workloads.

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

³ Service level agreements are offered for billable services on dedicated resources. Free services and preview features have no SLA. For billable services, SLAs take effect when you provision sufficient redundancy for your service. Two or more replicas are required for query (read) SLAs. Three or more replicas are required for query and indexing (read-write) SLAs. The number of partitions isn't an SLA consideration.

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

Azure Cognitive Services limits

The following limits are for the number of Cognitive Services resources per Azure subscription. Each of the Cognitive Services may have additional limitations, for more information see [Azure Cognitive Services](#).

Type	Limit	Example
A mixture of Cognitive Services resources	Maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, 50 Speech Service resources in West US, and 50 Text Analytics resources in East US.
A single type of Cognitive Services resources.	Maximum of 100 resources per region, with a maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, and 100 Computer Vision resources in East US.

Azure Cosmos DB limits

For Azure Cosmos DB limits, see [Limits in Azure Cosmos DB](#).

Azure Data Explorer limits

The following table describes the maximum limits for Azure Data Explorer clusters.

Resource	Limit
Clusters per region per subscription	20
Instances per cluster	1000
Number of databases in a cluster	10,000
Number of attached database configurations in a cluster	70

The following table describes the limits on management operations performed on Azure Data Explorer clusters.

Scope	Operation	Limit
Cluster	read (for example, get a cluster)	500 per 5 minutes
Cluster	write (for example, create a database)	1000 per hour

Azure Database for MySQL

For Azure Database for MySQL limits, see [Limitations in Azure Database for MySQL](#).

Azure Database for PostgreSQL

For Azure Database for PostgreSQL limits, see [Limitations in Azure Database for PostgreSQL](#).

Azure Functions limits

Resource	Consumption Plan	Premium Plan	App Service Plan ¹
Scale out	Event driven	Event driven	Manual/autoscale

RESOURCE	CONSUMPTION PLAN	PREMIUM PLAN	APP SERVICE PLAN
Max instances	200	100	10-20
Default timeout duration (min)	5	30	30 ²
Max timeout duration (min)	10	unbounded ⁸	unbounded ³
Max outbound connections (per instance)	600 active (1200 total)	unbounded	unbounded
Max request size (MB) ⁴	100	100	100
Max query string length ⁴	4096	4096	4096
Max request URL length ⁴	8192	8192	8192
ACU per instance	100	210-840	100-840
Max memory (GB per instance)	1.5	3.5-14	1.75-14
Function apps per plan	100	100	unbounded ⁵
App Service plans	100 per region	100 per resource group	100 per resource group
Storage ⁶	1 GB	250 GB	50-1000 GB
Custom domains per app	500 ⁷	500	500
Custom domain SSL support	unbounded SNI SSL connection included	unbounded SNI SSL and 1 IP SSL connections included	unbounded SNI SSL and 1 IP SSL connections included

¹ For specific limits for the various App Service plan options, see the [App Service plan limits](#).

² By default, the timeout for the Functions 1.x runtime in an App Service plan is unbounded.

³ Requires the App Service plan be set to [Always On](#). Pay at standard [rates](#).

⁴ These limits are [set in the host](#).

⁵ The actual number of function apps that you can host depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.

⁶ The storage limit is the total content size in temporary storage across all apps in the same App Service plan. Consumption plan uses Azure Files for temporary storage.

⁷ When your function app is hosted in a [Consumption plan](#), only the CNAME option is supported. For function apps in a [Premium plan](#) or an [App Service plan](#), you can map a custom domain using either a CNAME or an A record.

⁸ Guaranteed for up to 60 minutes.

Azure Kubernetes Service limits

RESOURCE	DEFAULT LIMIT
Maximum clusters per subscription	100

RESOURCE	DEFAULT LIMIT
Maximum nodes per cluster with Virtual Machine Availability Sets and Basic Load Balancer SKU	100
Maximum nodes per cluster with Virtual Machine Scale Sets and Standard Load Balancer SKU	1000 (100 nodes per node pool)
Maximum pods per node: Basic networking with Kubenet	110
Maximum pods per node: Advanced networking with Azure Container Networking Interface	Azure CLI deployment: 30 ¹ Azure Resource Manager template: 30 ¹ Portal deployment: 30

¹When you deploy an Azure Kubernetes Service (AKS) cluster with the Azure CLI or a Resource Manager template, this value is configurable up to 250 pods per node. You can't configure maximum pods per node after you've already deployed an AKS cluster, or if you deploy a cluster by using the Azure portal.

Azure Machine Learning limits

The latest values for Azure Machine Learning Compute quotas can be found in the [Azure Machine Learning quota page](#)

Azure Maps limits

The following table shows the usage limit for the Azure Maps S0 pricing tier. Usage limit depends on the pricing tier.

RESOURCE	S0 PRICING TIER LIMIT
Maximum request rate per subscription	50 requests per second

The following table shows the data size limit for Azure Maps. The Azure Maps data service is available only at the S1 pricing tier.

RESOURCE	LIMIT
Maximum size of data	50 MB

For more information on the Azure Maps pricing tiers, see [Azure Maps pricing](#).

Azure Monitor limits

Alerts

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Metric alerts (classic)	100 active alert rules per subscription.	Call support.
Metric alerts	1000 active alert rules per subscription in Azure public, Azure China 21Vianet and Azure Government clouds.	Call support.
Activity log alerts	100 active alert rules per subscription.	Same as default.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Log alerts	512	Call support.
Action groups	2,000 action groups per subscription.	Call support.
Autoscale settings	100 per region per subscription.	Same as default.

Action groups

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure app push	10 Azure app actions per action group.	Call support.
Email	1,000 email actions in an action group. No more than 100 emails in an hour. Also see the rate limiting information .	Call support.
ITSM	10 ITSM actions in an action group.	Call support.
Logic app	10 logic app actions in an action group.	Call support.
Runbook	10 runbook actions in an action group.	Call support.
SMS	10 SMS actions in an action group. No more than 1 SMS message every 5 minutes. Also see the rate limiting information .	Call support.
Voice	10 voice actions in an action group. No more than 1 voice call every 5 minutes. Also see the rate limiting information .	Call support.
Webhook	10 webhook actions in an action group. Maximum number of webhook calls is 1500 per minute per subscription. Other limits are available at action-specific information .	Call support.

Log queries and language

LIMIT	DESCRIPTION
Query language	Azure Monitor uses the same Kusto query language as Azure Data Explorer. See Azure Monitor log query language differences for KQL language elements not supported in Azure Monitor.
Azure regions	Log queries can experience excessive overhead when data spans Log Analytics workspaces in multiple Azure regions. See Query limits for details.

LIMIT	DESCRIPTION
Cross resource queries	Maximum number of Application Insights resources and Log Analytics workspaces in a single query limited to 100. Cross-resource query is not supported in View Designer. Cross-resource query in log alerts is supported in the new scheduledQueryRules API. See Cross-resource query limits for details.
Query throttling	A user is limited to 200 queries per 30 seconds on any number of workspaces. This limit applies to programmatic queries or to queries initiated by visualization parts such as Azure dashboards and the Log Analytics workspace summary page.

Log Analytics workspaces

Data collection volume and retention

TIER	LIMIT PER DAY	DATA RETENTION	COMMENT
Current Per GB pricing tier (introduced April 2018)	No limit	30 - 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Free tiers (introduced April 2016)	500 MB	7 days	When your workspace reaches the 500 MB per day limit, data ingestion stops and resumes at the start of the next day. A day is based on UTC. Note that data collected by Azure Security Center is not included in this 500 MB per day limit and will continue to be collected above this limit.
Legacy Standalone Per GB tier (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Per Node (OMS) (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Standard tier	No limit	30 days	Retention can't be adjusted
Legacy Premium tier	No limit	365 days	Retention can't be adjusted

Number of workspaces per subscription.

Pricing tier	Workspace limit	Comments
Free tier	10	This limit can't be increased.
All other tiers	No limit	You're limited by the number of resources within a resource group and the number of resource groups per subscription.

Azure portal

Category	Limits	Comments
Maximum records returned by a log query	10,000	Reduce results using query scope, time range, and filters in the query.

Data Collector API

Category	Limits	Comments
Maximum size for a single post	30 MB	Split larger volumes into multiple posts.
Maximum size for field values	32 KB	Fields longer than 32 KB are truncated.

Search API

Category	Limits	Comments
Maximum records returned in a single query	500,000	
Maximum size of data returned	64,000,000 bytes (~61 MiB)	
Maximum query running time	10 minutes	See Timeouts for details.
Maximum request rate	200 requests per 30 seconds per AAD user or client IP address	See Rate limits for details.

General workspace limits

Category	Limits	Comments
Maximum columns in a table	500	
Maximum characters for column name	500	
Data export	Not currently available	Use Azure Function or Logic App to aggregate and export data.

Data ingestion volume rate

Azure Monitor is a high scale data service that serves thousands of customers sending terabytes of data each month at a growing pace. The default ingestion volume rate limit for data sent from Azure resources using [diagnostic settings](#) is approximately **6 GB/min** per workspace. This is an approximate value since the actual size can vary between data types depending on the log length and its compression ratio. This limit does not apply to

data that is sent from agents or [Data Collector API](#).

If you send data at a higher rate to a single workspace, some data is dropped, and an event is sent to the *Operation* table in your workspace every 6 hours while the threshold continues to be exceeded. If your ingestion volume continues to exceed the rate limit or you are expecting to reach it sometime soon, you can request an increase to your workspace by opening a support request.

To be notified on such an event in your workspace, create a [log alert rule](#) using the following query with alert logic base on number of results grater than zero.

```
Operation  
|where OperationCategory == "Ingestion"  
|where Detail startswith "The rate of data crossed the threshold"
```

NOTE

Depending on how long you've been using Log Analytics, you might have access to legacy pricing tiers. Learn more about [Log Analytics legacy pricing tiers](#).

Application Insights

There are some limits on the number of metrics and events per application, that is, per instrumentation key. Limits depend on the [pricing plan](#) that you choose.

RESOURCE	DEFAULT LIMIT	NOTE
Total data per day	100 GB	You can reduce data by setting a cap. If you need more data, you can increase the limit in the portal, up to 1,000 GB. For capacities greater than 1,000 GB, send email to AIDataCap@microsoft.com .
Throttling	32,000 events/second	The limit is measured over a minute.
Data retention	90 days	This resource is for Search , Analytics , and Metrics Explorer .
Availability multi-step test detailed results retention	90 days	This resource provides detailed results of each step.
Maximum event size	64,000,000 bytes	
Property and metric name length	150	See type schemas .
Property value string length	8,192	See type schemas .
Trace and exception message length	32,768	See type schemas .
Availability tests count per app	100	
Profiler data retention	5 days	
Profiler data sent per day	10 GB	

For more information, see [About pricing and quotas in Application Insights](#).

Azure Policy limits

There's a maximum count for each object type for Azure Policy. An entry of *Scope* means either the subscription or the [management group](#).

WHERE	WHAT	MAXIMUM COUNT
Scope	Policy definitions	500
Scope	Initiative definitions	100
Tenant	Initiative definitions	1,000
Scope	Policy or initiative assignments	100
Policy definition	Parameters	20
Initiative definition	Policies	100
Initiative definition	Parameters	100
Policy or initiative assignments	Exclusions (notScopes)	400
Policy rule	Nested conditionals	512

Azure SignalR Service limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure SignalR Service units per instance for Free tier	1	1
Azure SignalR Service units per instance for Standard tier	100	100
Azure SignalR Service units per subscription per region for Free tier	5	5
Total Azure SignalR Service unit counts per subscription per region	150	Unlimited
Connections per unit per day for Free tier	20	20
Connections per unit per day for Standard tier	1,000	1,000
Included messages per unit per day for Free tier	20,000	20,000

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Included messages per unit per day for Standard tier	1,000,000	1,000,000

To request an update to your subscription's default limits, open a support ticket.

Backup limits

For a summary of Azure Backup support settings and limitations, see [Azure Backup Support Matrices](#).

Batch limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Batch accounts per region per subscription	1-3	50
Dedicated cores per Batch account	90-900	Contact support
Low-priority cores per Batch account	10-100	Contact support
Active jobs and job schedules per Batch account (completed jobs have no limit)	100-300	1,000 ¹
Pools per Batch account	20-100	500 ¹

NOTE

Default limits vary depending on the type of subscription you use to create a Batch account. Cores quotas shown are for Batch accounts in Batch service mode. [View the quotas in your Batch account](#).

¹To request an increase beyond this limit, contact Azure Support.

Classic deployment model limits

If you use classic deployment model instead of the Azure Resource Manager deployment model, the following limits apply.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
vCPUs per subscription ¹	20	10,000
Coadministrators per subscription	200	200
Storage accounts per subscription ²	100	100
Cloud services per subscription	20	200
Local networks per subscription	10	500

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
DNS servers per subscription	9	100
Reserved IPs per subscription	20	100
Affinity groups per subscription	256	256
Subscription name length (characters)	64	64

¹Extra small instances count as one vCPU toward the vCPU limit despite using a partial CPU core.

²The storage account limit includes both Standard and Premium storage accounts.

Container Instances limits

RESOURCE	DEFAULT LIMIT
Standard sku container groups per region per subscription	100 ¹
Dedicated sku container groups per region per subscription	0 ¹
Number of containers per container group	60
Number of volumes per container group	20
Ports per IP	5
Container instance log size - running instance	4 MB
Container instance log size - stopped instance	16 KB or 1,000 lines
Container creates per hour	300 ¹
Container creates per 5 minutes	100 ¹
Container deletes per hour	300 ¹
Container deletes per 5 minutes	100 ¹

¹To request a limit increase, create an [Azure Support request](#).

Container Registry limits

The following table details the features and limits of the Basic, Standard, and Premium [service tiers](#).

RESOURCE	BASIC	STANDARD	PREMIUM
Storage ¹	10 GiB	100 GiB	500 GiB
Maximum image layer size	200 GiB	200 GiB	200 GiB
ReadOps per minute ^{2, 3}	1,000	3,000	10,000

RESOURCE	BASIC	STANDARD	PREMIUM
WriteOps per minute ^{2, 4}	100	500	2,000
Download bandwidth MBps ²	30	60	100
Upload bandwidth MBps ²	10	20	50
Webhooks	2	10	500
Geo-replication	N/A	N/A	Supported
Content trust	N/A	N/A	Supported
Virtual network access	N/A	N/A	Preview
Repository-scoped permissions	N/A	N/A	Preview
• Tokens	N/A	N/A	20,000
• Scope maps	N/A	N/A	20,000
• Repositories per scope map	N/A	N/A	500

¹The specified storage limits are the amount of *included* storage for each tier. You're charged an additional daily rate per GiB for image storage above these limits. For rate information, see [Azure Container Registry pricing](#).

²*ReadOps*, *WriteOps*, and *Bandwidth* are minimum estimates. Azure Container Registry strives to improve performance as usage requires.

³A [docker pull](#) translates to multiple read operations based on the number of layers in the image, plus the manifest retrieval.

⁴A [docker push](#) translates to multiple write operations, based on the number of layers that must be pushed. A [docker push](#) includes *ReadOps* to retrieve a manifest for an existing image.

Content Delivery Network limits

RESOURCE	DEFAULT LIMIT
Azure Content Delivery Network profiles	25
Content Delivery Network endpoints per profile	25
Custom domains per endpoint	25

A Content Delivery Network subscription can contain one or more Content Delivery Network profiles. A Content Delivery Network profile can contain one or more Content Delivery Network endpoints. You might want to use multiple profiles to organize your Content Delivery Network endpoints by internet domain, web application, or some other criteria.

Data Factory limits

Azure Data Factory is a multitenant service that has the following default limits in place to make sure customer subscriptions are protected from each other's workloads. To raise the limits up to the maximum for your subscription, contact support.

Version 2

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Data factories in an Azure subscription	50	Contact support .
Total number of entities, such as pipelines, data sets, triggers, linked services, and integration runtimes, within a data factory	5,000	Contact support .
Total CPU cores for Azure-SSIS Integration Runtimes under one subscription	256	Contact support .
Concurrent pipeline runs per data factory that's shared among all pipelines in the factory	10,000	Contact support .
Concurrent External activity runs per subscription per Azure Integration Runtime region External activities are managed on integration runtime but execute on linked services, including Databricks, stored procedure, HDInsights, Web, and others.	3000	Contact support .
Concurrent Pipeline activity runs per subscription per Azure Integration Runtime region Pipeline activities execute on integration runtime, including Lookup, GetMetadata, and Delete.	1000	Contact support .
Concurrent authoring operations per subscription per Azure Integration Runtime region Including test connection, browse folder list and table list, preview data.	200	Contact support .
Concurrent Data Integration Units ¹ consumption per subscription per Azure Integration Runtime region	Region group 1 ² : 6000 Region group 2 ² : 3000 Region group 3 ² : 1500	Contact support .
Maximum activities per pipeline, which includes inner activities for containers	40	40
Maximum number of linked integration runtimes that can be created against a single self-hosted integration runtime	100	Contact support .
Maximum parameters per pipeline	50	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
ForEach items	100,000	100,000
ForEach parallelism	20	50
Maximum queued runs per pipeline	100	100
Characters per expression	8,192	8,192
Minimum tumbling window trigger interval	15 min	15 min
Maximum timeout for pipeline activity runs	7 days	7 days
Bytes per object for pipeline objects ³	200 KB	200 KB
Bytes per object for dataset and linked service objects ³	100 KB	2,000 KB
Data Integration Units ¹ per copy activity run	256	Contact support.
Write API calls	1,200/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support.
Read API calls	12,500/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support.
Monitoring queries per minute	1,000	Contact support.
Entity CRUD operations per minute	50	Contact support.
Maximum time of data flow debug session	8 hrs	8 hrs
Concurrent number of data flows per factory	50	Contact support.
Concurrent number of data flow debug sessions per user per factory	3	3
Data Flow Azure IR TTL limit	4 hrs	Contact support.

¹ The data integration unit (DIU) is used in a cloud-to-cloud copy operation, learn more from [Data integration units \(version 2\)](#). For information on billing, see [Azure Data Factory pricing](#).

² [Azure Integration Runtime](#) is [globally available](#) to ensure data compliance, efficiency, and reduced network egress costs.

Region group	Regions
Region group 1	Central US, East US, East US2, North Europe, West Europe, West US, West US 2
Region group 2	Australia East, Australia Southeast, Brazil South, Central India, Japan East, Northcentral US, Southcentral US, Southeast Asia, West Central US
Region group 3	Canada Central, East Asia, France Central, Korea Central, UK South

³ Pipeline, data set, and linked service objects represent a logical grouping of your workload. Limits for these objects don't relate to the amount of data you can move and process with Azure Data Factory. Data Factory is designed to scale to handle petabytes of data.

Version 1

Resource	Default limit	Maximum limit
Pipelines within a data factory	2,500	Contact support .
Data sets within a data factory	5,000	Contact support .
Concurrent slices per data set	10	10
Bytes per object for pipeline objects ¹	200 KB	200 KB
Bytes per object for data set and linked service objects ¹	100 KB	2,000 KB
Azure HDInsight on-demand cluster cores within a subscription ²	60	Contact support .
Cloud data movement units per copy activity run ³	32	Contact support .
Retry count for pipeline activity runs	1,000	MaxInt (32 bit)

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

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

³ The cloud data movement unit (DMU) for version 1 is used in a cloud-to-cloud copy operation, learn more from [Cloud data movement units \(version 1\)](#). For information on billing, see [Azure Data Factory pricing](#).

Resource	Default lower limit	Minimum limit
Scheduling interval	15 minutes	15 minutes
Interval between retry attempts	1 second	1 second

RESOURCE	DEFAULT LOWER LIMIT	MINIMUM LIMIT
Retry timeout value	1 second	1 second

Web service call limits

Azure Resource Manager has limits for API calls. You can make API calls at a rate within the [Azure Resource Manager API limits](#).

Data Lake Analytics limits

Azure Data Lake Analytics makes the complex task of managing distributed infrastructure and complex code easy. It dynamically provisions resources, and you can use it to do analytics on exabytes of data. When the job completes, it winds down resources automatically. You pay only for the processing power that was used. As you increase or decrease the size of data stored or the amount of compute used, you don't have to rewrite code. To raise the default limits for your subscription, contact support.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of concurrent jobs	20	
Maximum number of analytics units (AUs) per account	250	Use any combination of up to a maximum of 250 AUs across 20 jobs. To increase this limit, contact Microsoft Support.
Maximum script size for job submission	3 MB	
Maximum number of Data Lake Analytics accounts per region per subscription	5	To increase this limit, contact Microsoft Support.

Data Lake Store limits

Azure Data Lake Storage Gen1 is an enterprise-wide hyper-scale repository for big data analytic workloads. You can use Data Lake Storage Gen1 to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. There's no limit to the amount of data you can store in a Data Lake Storage Gen1 account.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of Data Lake Storage Gen1 accounts, per subscription, per region	10	To request an increase for this limit, contact support.
Maximum number of access ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.
Maximum number of default ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.

Data Share limits

Azure Data Share enables organizations to simply and securely share data with their customers and partners.

RESOURCE	LIMIT
Maximum number of Data Share resources per Azure subscription	50
Maximum number of sent shares per Data Share resource	100
Maximum number of received shares per Data Share resource	100
Maximum number of invitations per sent share	100
Maximum number of share subscriptions per sent share	100
Maximum number of datasets per share	100
Maximum number of snapshot schedules per share	1

Database Migration Service Limits

Azure Database Migration Service is a fully managed service designed to enable seamless migrations from multiple database sources to Azure data platforms with minimal downtime.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of services per subscription, per region	2	To request an increase for this limit, contact support.

Event Grid limits

The following limits apply to Azure Event Grid system topics and custom topics, *not* event domains.

RESOURCE	LIMIT
Custom topics per Azure subscription	100
Event subscriptions per topic	500
Publish rate for a custom topic (ingress)	5,000 events per second per topic
Publish requests	250 per second
Event size	1 MB (charged in as multiple 64-KB events)

The following limits apply to event domains only.

RESOURCE	LIMIT
Topics per event domain	100,000
Event subscriptions per topic within a domain	500
Domain scope event subscriptions	50

RESOURCE	LIMIT
Publish rate for an event domain (ingress)	5,000 events per second
Publish requests	250 per second
Event Domains per Azure Subscription	100

Event Hubs limits

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

The following limits are common across basic, standard, and dedicated tiers.

LIMIT	SCOPE	NOTES	VALUE
Number of Event Hubs namespaces per subscription	Subscription	-	100
Number of event hubs per namespace	Namespace	Subsequent requests for creation of a new event hub are rejected.	10
Number of partitions per event hub	Entity	-	32
Maximum size of an event hub name	Entity	-	50 characters
Number of non-epoch receivers per consumer group	Entity	-	5
Maximum throughput units	Namespace	Exceeding the throughput unit limit causes your data to be throttled and generates a server busy exception . To request a larger number of throughput units for a Standard tier, file a support request . Additional throughput units are available in blocks of 20 on a committed purchase basis.	20
Number of authorization rules per namespace	Namespace	Subsequent requests for authorization rule creation are rejected.	12
Number of calls to the GetRuntimeInformation method	Entity	-	50 per second

LIMIT	SCOPE	NOTES	VALUE
Number of virtual network (VNet) and IP Config rules	Entity	-	128

Event Hubs Basic and Standard - quotas and limits

LIMIT	SCOPE	NOTES	BASIC	STANDARD
Maximum size of Event Hubs event	Entity		256 KB	1 MB
Number of consumer groups per event hub	Entity		1	20
Number of AMQP connections per namespace	Namespace	Subsequent requests for additional connections are rejected, and an exception is received by the calling code.	100	5,000
Maximum retention period of event data	Entity		1 day	1-7 days
Apache Kafka enabled namespace	Namespace	Event Hubs namespace streams applications using Kafka protocol	No	Yes
Capture	Entity	When enabled, micro-batches on the same stream	No	Yes

Event Hubs Dedicated - quotas and limits

The Event Hubs Dedicated offering is billed at a fixed monthly price, with a minimum of 4 hours of usage. The Dedicated tier offers all the features of the Standard plan, but with enterprise scale capacity and limits for customers with demanding workloads.

FEATURE	LIMITS
Bandwidth	20 CUs
Namespaces	50 per CU
Event Hubs	1000 per namespace
Ingress events	Included
Message Size	1 MB
Partitions	2000 per CU
Consumer groups	No limit per CU, 1000 per event hub

FEATURE	LIMITS
Brokered connections	100 K included
Message Retention	90 days, 10 TB included per CU
Capture	Included

Identity Manager limits

CATEGORY	LIMIT
User-assigned managed identities	<ul style="list-style-type: none"> When you create user-assigned managed identities, only alphanumeric characters (0-9, a-z, and A-Z) and the hyphen (-) are supported. For the assignment to a virtual machine or virtual machine scale set to work properly, the name is limited to 24 characters. If you use the managed identity virtual machine extension, the supported limit is 32 user-assigned managed identities. Without the managed identity virtual machine extension, the supported limit is 512 user-assigned identities.

IoT Central limits

IoT Central limits the number of applications you can deploy in a subscription to 10. If you need to increase this limit, contact [Microsoft support](#).

IoT Hub limits

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

RESOURCE	S1 STANDARD	S2 STANDARD	S3 STANDARD	F1 FREE
Messages/day	400,000	6,000,000	300,000,000	8,000
Maximum units	200	200	10	1

NOTE

If you anticipate using more than 200 units with an S1 or S2 tier hub or 10 units with an S3 tier hub, contact Microsoft Support.

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

RESOURCE	LIMIT
Maximum paid IoT hubs per Azure subscription	100
Maximum free IoT hubs per Azure subscription	1

RESOURCE	LIMIT
Maximum number of characters in a device ID	128
Maximum number of device identities returned in a single call	1,000
IoT Hub message maximum retention for device-to-cloud messages	7 days
Maximum size of device-to-cloud message	256 KB
Maximum size of device-to-cloud batch	AMQP and HTTP: 256 KB for the entire batch MQTT: 256 KB for each message
Maximum messages in device-to-cloud batch	500
Maximum size of cloud-to-device message	64 KB
Maximum TTL for cloud-to-device messages	2 days
Maximum delivery count for cloud-to-device messages	100
Maximum cloud-to-device queue depth per device	50
Maximum delivery count for feedback messages in response to a cloud-to-device message	100
Maximum TTL for feedback messages in response to a cloud-to-device message	2 days
Maximum size of device twin	8 KB for tags section, and 32 KB for desired and reported properties sections each
Maximum length of device twin string key	1 KB
Maximum length of device twin string value	4 KB
Maximum depth of object in device twin	10
Maximum size of direct method payload	128 KB
Job history maximum retention	30 days
Maximum concurrent jobs	10 (for S3), 5 for (S2), 1 (for S1)
Maximum additional endpoints	10 (for S1, S2, and S3)
Maximum message routing rules	100 (for S1, S2, and S3)
Maximum number of concurrently connected device streams	50 (for S1, S2, S3, and F1 only)

RESOURCE	LIMIT
Maximum device stream data transfer	300 MB per day (for S1, S2, S3, and F1 only)

NOTE

If you need more than 100 paid IoT hubs in an Azure subscription, contact Microsoft Support.

NOTE

Currently, the total number of devices plus modules that can be registered to a single IoT hub is capped at 1,000,000. If you want to increase this limit, contact [Microsoft Support](#).

IoT Hub throttles requests when the following quotas are exceeded.

THROTTLE	PER-HUB VALUE
Identity registry operations (create, retrieve, list, update, and delete), individual or bulk import/export	83.33/sec/unit (5,000/min/unit) (for S3). 1.67/sec/unit (100/min/unit) (for S1 and S2).
Device connections	6,000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Device-to-cloud sends	6,000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Cloud-to-device sends	83.33/sec/unit (5,000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S1 and S2).
Cloud-to-device receives	833.33/sec/unit (50,000/min/unit) (for S3), 16.67/sec/unit (1,000/min/unit) (for S1 and S2).
File upload operations	83.33 file upload initiations/sec/unit (5,000/min/unit) (for S3), 1.67 file upload initiations/sec/unit (100/min/unit) (for S1 and S2). 10,000 SAS URIs can be out for an Azure Storage account at one time. 10 SAS URIs/device can be out at one time.
Direct methods	24 MB/sec/unit (for S3), 480 KB/sec/unit (for S2), 160 KB/sec/unit (for S1). Based on 8-KB throttling meter size.
Device twin reads	500/sec/unit (for S3), Maximum of 100/sec or 10/sec/unit (for S2), 100/sec (for S1)
Device twin updates	250/sec/unit (for S3), Maximum of 50/sec or 5/sec/unit (for S2), 50/sec (for S1)
Jobs operations (create, update, list, and delete)	83.33/sec/unit (5,000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S2), 1.67/sec/unit (100/min/unit) (for S1).

THROTTLE	PER-HUB VALUE
Jobs per-device operation throughput	50/sec/unit (for S3), maximum of 10/sec or 1/sec/unit (for S2), 10/sec (for S1).
Device stream initiation rate	5 new streams/sec (for S1, S2, S3, and F1 only).

IoT Hub Device Provisioning Service limits

The following table lists the limits that apply to Azure IoT Hub Device Provisioning Service resources.

RESOURCE	LIMIT
Maximum device provisioning services per Azure subscription	10
Maximum number of enrollments	1,000,000
Maximum number of registrations	1,000,000
Maximum number of enrollment groups	100
Maximum number of CAs	25
Maximum number of linked IoT hubs	50
Maximum size of message	96 KB

NOTE

To increase the number of enrollments and registrations on your provisioning service, contact [Microsoft Support](#).

NOTE

Increasing the maximum number of CAs is not supported.

The Device Provisioning Service throttles requests when the following quotas are exceeded.

THROTTLE	PER-UNIT VALUE
Operations	200/min/service
Device registrations	200/min/service
Device polling operation	5/10 sec/device

Key Vault limits

Key transactions (maximum transactions allowed in 10 seconds, per vault per region¹):

KEY TYPE	HSM KEY CREATE KEY	HSM KEY ALL OTHER TRANSACTIONS	SOFTWARE KEY CREATE KEY	SOFTWARE KEY ALL OTHER TRANSACTIONS
RSA 2,048-bit	5	1,000	10	2,000
RSA 3,072-bit	5	250	10	500
RSA 4,096-bit	5	125	10	250
ECC P-256	5	1,000	10	2,000
ECC P-384	5	1,000	10	2,000
ECC P-521	5	1,000	10	2,000
ECC SECP256K1	5	1,000	10	2,000

NOTE

In the previous table, we see that for RSA 2,048-bit software keys, 2,000 GET transactions per 10 seconds are allowed. For RSA 2,048-bit HSM-keys, 1,000 GET transactions per 10 seconds are allowed.

The throttling thresholds are weighted, and enforcement is on their sum. For example, as shown in the previous table, when you perform GET operations on RSA HSM-keys, it's eight times more expensive to use 4,096-bit keys compared to 2,048-bit keys. That's because $1,000/125 = 8$.

In a given 10-second interval, an Azure Key Vault client can do *only one* of the following operations before it encounters a 429 throttling HTTP status code:

- 2,000 RSA 2,048-bit software-key GET transactions
- 1,000 RSA 2,048-bit HSM-key GET transactions
- 125 RSA 4,096-bit HSM-key GET transactions
- 124 RSA 4,096-bit HSM-key GET transactions and 8 RSA 2,048-bit HSM-key GET transactions

Secrets, managed storage account keys, and vault transactions:

TRANSACTIONS TYPE	MAXIMUM TRANSACTIONS ALLOWED IN 10 SECONDS, PER VAULT PER REGION ¹
All transactions	2,000

For information on how to handle throttling when these limits are exceeded, see [Azure Key Vault throttling guidance](#).

¹ A subscription-wide limit for all transaction types is five times per key vault limit. For example, HSM-other transactions per subscription are limited to 5,000 transactions in 10 seconds per subscription.

Media Services limits

NOTE

For resources that aren't fixed, open a support ticket to ask for an increase in the quotas. Don't create additional Azure Media Services accounts in an attempt to obtain higher limits.

RESOURCE	DEFAULT LIMIT
Azure Media Services accounts in a single subscription	25 (fixed)
Media reserved units per Media Services account	25 (S1) 10 (S2, S3) ¹
Jobs per Media Services account	50,000 ²
Chained tasks per job	30 (fixed)
Assets per Media Services account	1,000,000
Assets per task	50
Assets per job	100
Unique locators associated with an asset at one time	5 ⁴
Live channels per Media Services account	5
Programs in stopped state per channel	50
Programs in running state per channel	3
Streaming endpoints that are stopped or running per Media Services account	2
Streaming units per streaming endpoint	10
Storage accounts	1,000 ⁵ (fixed)
Policies	1,000,000 ⁶
File size	In some scenarios, there's a limit on the maximum file size supported for processing in Media Services. ⁷

¹If you change the type, for example, from S2 to S1, the maximum reserved unit limits are reset.

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

As of April 1, 2017, any job record in your account older than 90 days is automatically deleted, along with its associated task records. Automatic deletion occurs even if the total number of records is below the maximum quota. To archive the job and task information, use the code described in [Manage assets with the Media Services .NET SDK](#).

³When you make a request to list job entities, a maximum of 1,000 jobs is returned per request. To keep track of all submitted jobs, use the top or skip queries as described in [OData system query options](#).

⁴Locators aren't designed for managing per-user access control. To give different access rights to individual users, use digital rights management (DRM) solutions. For more information, see [Protect your content with Azure Media Services](#).

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

⁶There's a limit of 1,000,000 policies for different Media Services policies. An example is for the Locator policy or ContentKeyAuthorizationPolicy.

NOTE

If you always use the same days and access permissions, use the same policy ID. For information and an example, see [Manage assets with the Media Services .NET SDK](#).

⁷The maximum size supported for a single blob is currently up to 5 TB in Azure Blob Storage. Additional limits apply in Media Services based on the VM sizes that are used by the service. The size limit applies to the files that you upload and also the files that get generated as a result of Media Services processing (encoding or analyzing). If your source file is larger than 260-GB, your Job will likely fail.

The following table shows the limits on the media reserved units S1, S2, and S3. If your source file is larger than the limits defined in the table, your encoding job fails. If you encode 4K resolution sources of long duration, you're required to use S3 media reserved units to achieve the performance needed. If you have 4K content that's larger than the 260-GB limit on the S3 media reserved units, open a support ticket.

MEDIA RESERVED UNIT TYPE	MAXIMUM INPUT SIZE (GB)
S1	26
S2	60
S3	260

Mobile Services limits

TIER	FREE	BASIC	STANDARD
API calls	500,000	1.5 million per unit	15 million per unit
Active devices	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Unlimited units
Push notifications	Azure Notification Hubs Free tier included, up to 1 million pushes	Notification Hubs Basic tier included, up to 10 million pushes	Notification Hubs Standard tier included, up to 10 million pushes
Real-time messaging/ Web Sockets	Limited	350 per mobile service	Unlimited
Offline synchronizations	Limited	Included	Included
Scheduled jobs	Limited	Included	Included
Azure SQL Database (required) Standard rates apply for additional capacity	20 MB included	20 MB included	20 MB included
CPU capacity	60 minutes per day	Unlimited	Unlimited

TIER	FREE	BASIC	STANDARD
Outbound data transfer	165 MB per day (daily rollover)	Included	Included

For more information on limits and pricing, see [Azure Mobile Services pricing](#).

Multi-Factor Authentication limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of trusted IP addresses or ranges per subscription	0	50
Remember my devices, number of days	14	60
Maximum number of app passwords	0	No limit
Allow X attempts during MFA call	1	99
Two-way text message timeout seconds	60	600
Default one-time bypass seconds	300	1,800
Lock user account after X consecutive MFA denials	Not set	99
Reset account lockout counter after X minutes	Not set	9,999
Unlock account after X minutes	Not set	9,999

Networking limits

Networking limits - Azure Resource Manager The following limits apply only for networking resources managed through **Azure Resource Manager** per region per subscription. Learn how to [view your current resource usage against your subscription limits](#).

NOTE

We recently increased all default limits to their maximum limits. If there's no maximum limit column, the resource doesn't have adjustable limits. If you had these limits increased by support in the past and don't see updated limits in the following tables, [open an online customer support request at no charge](#)

RESOURCE	DEFAULT/MAXIMUM LIMIT
Virtual networks	1,000
Subnets per virtual network	3,000
Virtual network peerings per virtual network	500

RESOURCE	DEFAULT/MAXIMUM LIMIT
Virtual network gateways (VPN gateways) per virtual network	1
Virtual network gateways (ExpressRoute gateways) per virtual network	1
DNS servers per virtual network	20
Private IP addresses per virtual network	65,536
Private IP addresses per network interface	256
Private IP addresses per virtual machine	256
Public IP addresses per network interface	256
Public IP addresses per virtual machine	256
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000
Network interface cards	65,536
Network Security Groups	5,000
NSG rules per NSG	1,000
IP addresses and ranges specified for source or destination in a security group	4,000
Application security groups	3,000
Application security groups per IP configuration, per NIC	20
IP configurations per application security group	4,000
Application security groups that can be specified within all security rules of a network security group	100
User-defined route tables	200
User-defined routes per route table	400
Point-to-site root certificates per Azure VPN Gateway	20
Virtual network TAPs	100
Network interface TAP configurations per virtual network TAP	100

Public IP address limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP addresses ¹	10 for Basic.	Contact support.
Static Public IP addresses ¹	10 for Basic.	Contact support.
Standard Public IP addresses ¹	10	Contact support.
Public IP Prefixes	limited by number of Standard Public IPs in a subscription	Contact support.
Public IP prefix length	/28	Contact support.

¹Default limits for Public IP addresses vary by offer category type, such as Free Trial, Pay-As-You-Go, CSP. For example, the default for Enterprise Agreement subscriptions is 1000.

Load balancer limits

The following limits apply only for networking resources managed through Azure Resource Manager per region per subscription. Learn how to [view your current resource usage against your subscription limits](#).

Standard Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	1,500
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	600
Backend pool size	1,000 IP configurations, single virtual network
High-availability ports	1 per internal frontend
Outbound rules per Load Balancer	20

Basic Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	250
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	200
Backend pool size	300 IP configurations, single availability set
Availability sets per Load Balancer	150

The following limits apply only for networking resources managed through the classic deployment model per subscription. Learn how to [view your current resource usage against your subscription limits](#).

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual networks	100	100
Local network sites	20	50
DNS servers per virtual network	20	20
Private IP addresses per virtual network	4,096	4,096
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000, up to 1,000,000 for two or more NICs.	500,000, up to 1,000,000 for two or more NICs.
Network Security Groups (NSGs)	200	200
NSG rules per NSG	1,000	1,000
User-defined route tables	200	200
User-defined routes per route table	400	400
Public IP addresses (dynamic)	500	500
Reserved public IP addresses	500	500
Public VIP per deployment	5	Contact support
Private VIP (internal load balancing) per deployment	1	1
Endpoint access control lists (ACLs)	50	50

ExpressRoute limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription, with Azure Resource Manager	10
Maximum number of routes advertised to Azure private peering with ExpressRoute Standard	4,000
Maximum number of routes advertised to Azure private peering with ExpressRoute Premium add-on	10,000
Maximum number of routes advertised from Azure private peering from the VNet address space for an ExpressRoute connection	200

RESOURCE	DEFAULT/MAXIMUM LIMIT
Maximum number of routes advertised to Microsoft peering with ExpressRoute Standard	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Premium add-on	200
Maximum number of ExpressRoute circuits linked to the same virtual network in the same peering location	4
Maximum number of ExpressRoute circuits linked to the same virtual network in different peering locations	4
Number of virtual network links allowed per ExpressRoute circuit	See the Number of virtual networks per ExpressRoute circuit table .

Number of virtual networks per ExpressRoute circuit

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

*100 Gbps ExpressRoute Direct Only

NOTE

Global Reach connections count against the limit of virtual network connections per ExpressRoute Circuit. For example, a 10 Gbps Premium Circuit would allow for 5 Global Reach connections and 95 connections to the ExpressRoute Gateways or 95 Global Reach connections and 5 connections to the ExpressRoute Gateways or any other combination up to the limit of 100 connections for the circuit.

Virtual WAN limits

RESOURCE	LIMIT
Virtual WAN hubs per region	1
Virtual WAN hubs per virtual wan	Azure regions
VPN (branch) connections per hub	1,000
VNet connections per hub	500
Point-to-Site users per hub	10,000
Aggregate throughput per Virtual WAN VPN gateway	20 Gbps
Throughput per Virtual WAN VPN connection (2 tunnels)	2 Gbps with 1 Gbps/IPsec tunnel
Aggregate throughput per Virtual WAN ExpressRoute gateway	20 Gbps

Application Gateway limits

The following table applies to v1, v2, Standard, and WAF SKUs unless otherwise stated.

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Azure Application Gateway	1,000 per subscription	
Front-end IP configurations	2	1 public and 1 private
Front-end ports	100 ¹	
Back-end address pools	100 ¹	
Back-end servers per pool	1,200	
HTTP listeners	100 ¹	
HTTP load-balancing rules	100 ¹	
Back-end HTTP settings	100 ¹	
Instances per gateway	V1 SKU - 32 V2 SKU - 125	
SSL certificates	100 ¹	1 per HTTP listener
Maximum SSL certificate size	V1 SKU - 10 KB V2 SKU - 16 KB	
Authentication certificates	100	
Trusted root certificates	100	

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Request timeout minimum	1 second	
Request timeout maximum	24 hours	
Number of sites	100 ¹	1 per HTTP listener
URL maps per listener	1	
Maximum path-based rules per URL map	100	
Redirect configurations	100 ¹	
Concurrent WebSocket connections	Medium gateways 20k Large gateways 50k	
Maximum URL length	32KB	
Maximum header size for HTTP/2	4KB	
Maximum file upload size, Standard	2 GB	
Maximum file upload size WAF	V1 Medium WAF gateways, 100 MB V1 Large WAF gateways, 500 MB V2 WAF, 750 MB	
WAF body size limit, without files	128 KB	
Maximum WAF custom rules	100	
Maximum WAF exclusions	100	

¹ In case of WAF-enabled SKUs, we recommend that you limit the number of resources to 40 for optimal performance.

Network Watcher limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT	NOTE
Azure Network Watcher	1 per region	1 per region	Network Watcher is created to enable access to the service. Only one instance of Network Watcher is required per subscription per region.
Packet capture sessions	10,000 per region	10,000	Number of sessions only, not saved captures.

Private Link limits

The following limits apply to Azure private link:

RESOURCE	LIMIT
Number of private endpoints per virtual network	1000
Number of private endpoints per subscription	64000
Number of private link service per subscription	800
Number of IP Configurations on a private link service	8 (This number is for the NAT IP addresses used per PLS)
Number of private endpoints on the same private link service	1000

Traffic Manager limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Profiles per subscription	200
Endpoints per profile	200

Azure Bastion limits

RESOURCE	DEFAULT LIMIT
Concurrent RDP connections	25*
Concurrent SSH connections	More than 50**

*May vary due to other on-going RDP sessions or other on-going SSH sessions.

**May vary if there are existing RDP connections or usage from other on-going SSH sessions.

Azure DNS limits

Public DNS zones

RESOURCE	DEFAULT LIMIT
Public DNS Zones per subscription	250 ¹
Record sets per public DNS zone	10,000 ¹
Records per record set in public DNS zone	20
Number of Alias records for a single Azure resource	20
Private DNS zones per subscription	1000
Record sets per private DNS zone	25000
Records per record set for private DNS zones	20
Virtual Network Links per private DNS zone	1000

RESOURCE	DEFAULT LIMIT
Virtual Networks Links per private DNS zones with auto-registration enabled	100
Number of private DNS zones a virtual network can get linked to with auto-registration enabled	1
Number of private DNS zones a virtual network can get linked	1000
Number of DNS queries a virtual machine can send to Azure DNS resolver, per second	500 ²
Maximum number of DNS queries queued (pending response) per virtual machine	200 ²

¹If you need to increase these limits, contact Azure Support.

²These limits are applied to every individual virtual machine and not at the virtual network level. DNS queries exceeding these limits are dropped.

Azure Firewall limits

RESOURCE	DEFAULT LIMIT
Data throughput	30 Gbps ¹
Rules	10,000. All rule types combined.
Maximum DNAT rules	299
Minimum AzureFirewallSubnet size	/26
Port range in network and application rules	0-64,000. Work is in progress to relax this limitation.
Public IP addresses	100 maximum (Currently, SNAT ports are added only for the first five public IP addresses.)
Route table	<p>By default, AzureFirewallSubnet has a 0.0.0.0/0 route with the NextHopType value set to Internet.</p> <p>Azure Firewall must have direct Internet connectivity. If your AzureFirewallSubnet learns a default route to your on-premises network via BGP, you must override that with a 0.0.0.0/0 UDR with the NextHopType value set as Internet to maintain direct Internet connectivity. By default, Azure Firewall doesn't support forced tunneling to an on-premises network.</p> <p>However, if your configuration requires forced tunneling to an on-premises network, Microsoft will support it on a case by case basis. Contact Support so that we can review your case. If accepted, we'll allow your subscription and ensure the required firewall Internet connectivity is maintained.</p>

¹If you need to increase these limits, contact Azure Support.

Azure Front Door Service limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Azure Front Door Service resources per subscription	100
Front-end hosts, which includes custom domains per resource	100
Routing rules per resource	100
Back-end pools per resource	50
Back ends per back-end pool	100
Path patterns to match for a routing rule	25
Custom web application firewall rules per policy	10
Web application firewall policy per subscription	100
Web application firewall match conditions per custom rule	10
Web application firewall IP address ranges per match condition	600
Web application firewall string match values per match condition	10
Web application firewall string match value length	256
Web application firewall POST body parameter name length	256
Web application firewall HTTP header name length	256
Web application firewall cookie name length	256
Web application firewall HTTP request body size inspected	128 KB
Web application firewall custom response body length	2 KB

Timeout values

Client to Front Door

- Front Door has an idle TCP connection timeout of 61 seconds.

Front Door to application back-end

- If the response is a chunked response, a 200 is returned if or when the first chunk is received.
- After the HTTP request is forwarded to the back end, Front Door waits for 30 seconds for the first packet from the back end. Then it returns a 503 error to the client.
- After the first packet is received from the back end, Front Door waits for 30 seconds in an idle timeout. Then it returns a 503 error to the client.
- Front Door to the back-end TCP session timeout is 30 minutes.

Upload and download data limit

	WITH CHUNKED TRANSFER ENCODING (CTE)	WITHOUT HTTP CHUNKING
Download	There's no limit on the download size.	There's no limit on the download size.
Upload	There's no limit as long as each CTE upload is less than 2 GB.	The size can't be larger than 2 GB.

Other limits

- Maximum URL size - 8,192 bytes - Specifies maximum length of the raw URL (scheme + hostname + port + path + query string of the URL)
- Maximum Query String size - 4,096 bytes - Specifies the maximum length of the query string, in bytes.

Notification Hubs limits

TIER	FREE	BASIC	STANDARD
Included pushes	1 million	10 million	10 million
Active devices	500	200,000	10 million
Tag quota per installation or registration	60	60	60

For more information on limits and pricing, see [Notification Hubs pricing](#).

Role-based access control limits

RESOURCE	LIMIT
Role assignments for Azure resources per Azure subscription	2,000
Role assignments for Azure resources per management group	500
Custom roles for Azure resources per tenant	5,000
Custom roles for Azure resources per tenant (specialized clouds, such as Azure Government, Azure Germany, and Azure China 21Vianet)	2,000

Service Bus limits

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

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum number of Basic or Standard namespaces per Azure subscription	Namespace	Subsequent requests for additional Basic or Standard namespaces are rejected by the Azure portal.	100

Quota name	Scope	Notes	Value
Maximum number of Premium namespaces per Azure subscription	Namespace	Subsequent requests for additional Premium namespaces are rejected by the portal.	100
Queue or topic size	Entity	Defined upon creation of the queue or topic. Subsequent incoming messages are rejected, and an exception is received by the calling code.	1, 2, 3, 4 GB or 5 GB. In the Premium SKU, and the Standard SKU with partitioning enabled, the maximum queue or topic size is 80 GB.
Number of concurrent connections on a namespace	Namespace	Subsequent requests for additional connections are rejected, and an exception is received by the calling code. REST operations don't count toward concurrent TCP connections.	NetMessaging: 1,000. AMQP: 5,000.
Number of concurrent receive requests on a queue, topic, or subscription entity	Entity	Subsequent receive requests are rejected, and an exception is received by the calling code. This quota applies to the combined number of concurrent receive operations across all subscriptions on a topic.	5,000
Number of topics or queues per namespace	Namespace	Subsequent requests for creation of a new topic or queue on the namespace are rejected. As a result, if configured through the Azure portal , an error message is generated. If called from the management API, an exception is received by the calling code.	10,000 for the Basic or Standard tier. The total number of topics and queues in a namespace must be less than or equal to 10,000. For the Premium tier, 1,000 per messaging unit (MU). Maximum limit is 4,000.
Number of partitioned topics or queues per namespace	Namespace	Subsequent requests for creation of a new partitioned topic or queue on the namespace are rejected. As a result, if configured through the Azure portal , an error message is generated. If called from the management API, the exception QuotaExceededException is received by the calling code.	Basic and Standard tiers: 100. Partitioned entities aren't supported in the Premium tier . Each partitioned queue or topic counts toward the quota of 1,000 entities per namespace.
Maximum size of any messaging entity path: queue or topic	Entity	-	260 characters.

Quota Name	Scope	Notes	Value
Maximum size of any messaging entity name: namespace, subscription, or subscription rule	Entity	-	50 characters.
Maximum size of a message ID	Entity	-	128
Maximum size of a message session ID	Entity	-	128
Message size for a queue, topic, or subscription entity	Entity	<p>Incoming messages that exceed these quotas are rejected, and an exception is received by the calling code.</p> <p>Due to system overhead, this limit is less than these values.</p> <p>Maximum header size: 64 KB.</p> <p>Maximum number of header properties in property bag: byte/int.MaxValue.</p> <p>Maximum size of property in property bag: No explicit limit. Limited by maximum header size.</p>	<p>Maximum message size: 256 KB for Standard tier, 1 MB for Premium tier.</p>
Message property size for a queue, topic, or subscription entity	Entity	The exception SerializationException is generated.	<p>Maximum message property size for each property is 32,000. Cumulative size of all properties can't exceed 64,000. This limit applies to the entire header of the BrokeredMessage, which has both user properties and system properties, such as SequenceNumber, Label, and MessageId.</p>
Number of subscriptions per topic	Entity	Subsequent requests for creating additional subscriptions for the topic are rejected. As a result, if configured through the portal, an error message is shown. If called from the management API, an exception is received by the calling code.	2,000 per-topic for the Standard tier.
Number of SQL filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	2,000

Quota Name	Scope	Notes	Value
Number of correlation filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	100,000
Size of SQL filters or actions	Namespace	Subsequent requests for creation of additional filters are rejected, and an exception is received by the calling code.	Maximum length of filter condition string: 1,024 (1 K). Maximum length of rule action string: 1,024 (1 K). Maximum number of expressions per rule action: 32.
Number of SharedAccessAuthorizationRule rules per namespace, queue, or topic	Entity, namespace	Subsequent requests for creation of additional rules are rejected, and an exception is received by the calling code.	Maximum number of rules per entity type: 12. Rules that are configured on a Service Bus namespace apply to all types: queues, topics.
Number of messages per transaction	Transaction	Additional incoming messages are rejected, and an exception stating "Cannot send more than 100 messages in a single transaction" is received by the calling code.	100 For both Send() and SendAsync() operations.
Number of virtual network and IP filter rules	Namespace		128

Site Recovery limits

The following limits apply to Azure Site Recovery.

Limit Identifier	Default Limit
Number of vaults per subscription	500
Number of servers per Azure vault	250
Number of protection groups per Azure vault	No limit
Number of recovery plans per Azure vault	No limit
Number of servers per protection group	No limit
Number of servers per recovery plan	50

SQL Database limits

For SQL Database limits, see [SQL Database resource limits for single databases](#), [SQL Database resource limits for elastic pools and pooled databases](#), and [SQL Database resource limits for managed instances](#).

SQL Data Warehouse limits

For SQL Data Warehouse limits, see [SQL Data Warehouse resource limits](#).

Storage limits

The following table describes default limits for Azure general-purpose v1, v2, and Blob storage accounts. The *ingress* limit refers to all data from requests that are sent to a storage account. The *egress* limit refers to all data from responses that are received from a storage account.

RESOURCE	DEFAULT LIMIT
Number of storage accounts per region per subscription, including both standard and premium accounts	250
Maximum storage account capacity	2 PiB for US and Europe, and 500 TiB for all other regions (including the UK) ¹
Maximum number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate ¹ per storage account	20,000 requests per second
Maximum ingress ¹ per storage account (US, Europe regions)	25 Gbps
Maximum ingress ¹ per storage account (regions other than US and Europe)	5 Gbps if RA-GRS/GRS is enabled, 10 Gbps for LRS/ZRS ²
Maximum egress for general-purpose v2 and Blob storage accounts (all regions)	50 Gbps
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS ²
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS ²
Maximum number of virtual network rules per storage account	200
Maximum number of IP address rules per storage account	200

¹Azure Storage standard accounts support higher capacity limits and higher limits for ingress by request. To request an increase in account limits for ingress, contact [Azure Support](#). For more information, see [Announcing larger, higher scale storage accounts](#).

²If your storage account has read-access enabled with geo-redundant storage (RA-GRS) or geo-zone-redundant storage (RA-GZRS), then the egress targets for the secondary location are identical to those of the primary location. [Azure Storage replication](#) options include:

- [Locally redundant storage \(LRS\)](#)
- [Zone-redundant storage \(ZRS\)](#)

- [Geo-redundant storage \(GRS\)](#)
- [Read-access geo-redundant storage \(RA-GRS\)](#)
- [Geo-zone-redundant storage \(GZRS\)](#)
- [Read-access geo-zone-redundant storage \(RA-GZRS\)](#)

NOTE

Microsoft recommends that you use a general-purpose v2 storage account for most scenarios. You can easily upgrade a general-purpose v1 or an Azure Blob storage account to a general-purpose v2 account with no downtime and without the need to copy data. For more information, see [Upgrade to a general-purpose v2 storage account](#).

If the needs of your application exceed the scalability targets of a single storage account, you can build your application to use multiple storage accounts. You can then partition your data objects across those storage accounts. For information on volume pricing, see [Azure Storage pricing](#).

All storage accounts run on a flat network topology and support the scalability and performance targets outlined in this article, regardless of when they were created. For more information on the Azure Storage flat network architecture and on scalability, see [Microsoft Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency](#).

For more information on limits for standard storage accounts, see [Scalability targets for standard storage accounts](#).

Storage resource provider limits

The following limits apply only when you perform management operations by using Azure Resource Manager with Azure Storage.

RESOURCE	DEFAULT LIMIT
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	1200 per hour
Storage account management operations (list)	100 per 5 minutes

Azure Blob storage limits

RESOURCE	TARGET
Maximum size of single blob container	Same as maximum storage account capacity
Maximum number of blocks in a block blob or append blob	50,000 blocks
Maximum size of a block in a block blob	100 MiB
Maximum size of a block blob	50,000 X 100 MiB (approximately 4.75 TiB)
Maximum size of a block in an append blob	4 MiB
Maximum size of an append blob	50,000 x 4 MiB (approximately 195 GiB)
Maximum size of a page blob	8 TiB

RESOURCE	TARGET
Maximum number of stored access policies per blob container	5
Target request rate for a single blob	Up to 500 requests per second
Target throughput for a single page blob	Up to 60 MiB per second
Target throughput for a single block blob	Up to storage account ingress/egress limits ¹

¹ Throughput for a single blob depends on several factors, including, but not limited to: concurrency, request size, performance tier, speed of source for uploads, and destination for downloads. To take advantage of the performance enhancements of [high-throughput block blobs](#), upload larger blobs or blocks. Specifically, call the [Put Blob](#) or [Put Block](#) operation with a blob or block size that is greater than 4 MiB for standard storage accounts. For premium block blob or for Data Lake Storage Gen2 storage accounts, use a block or blob size that is greater than 256 KiB.

Azure Files limits

For more information on Azure Files limits, see [Azure Files scalability and performance targets](#).

RESOURCE	STANDARD FILE SHARES	PREMIUM FILE SHARES
Minimum size of a file share	No minimum; pay as you go	100 GiB; provisioned
Maximum size of a file share	100 TiB*, 5 TiB	100 TiB
Maximum size of a file in a file share	1 TiB	1 TiB
Maximum number of files in a file share	No limit	No limit
Maximum IOPS per share	10,000 IOPS*, 1,000 IOPS	100,000 IOPS
Maximum number of stored access policies per file share	5	5
Target throughput for a single file share	up to 300 MiB/sec*, Up to 60 MiB/sec ,	See premium file share ingress and egress values
Maximum egress for a single file share	See standard file share target throughput	Up to 6,204 MiB/s
Maximum ingress for a single file share	See standard file share target throughput	Up to 4,136 MiB/s
Maximum open handles per file	2,000 open handles	2,000 open handles
Maximum number of share snapshots	200 share snapshots	200 share snapshots
Maximum object (directories and files) name length	2,048 characters	2,048 characters
Maximum pathname component (in the path \A\B\C\D, each letter is a component)	255 characters	255 characters

* Available in most regions, see [Regional availability](#) for the details on available regions.

Azure File Sync limits

RESOURCE	TARGET	HARD LIMIT
Storage Sync Services per region	20 Storage Sync Services	Yes
Sync groups per Storage Sync Service	100 sync groups	Yes
Registered servers per Storage Sync Service	99 servers	Yes
Cloud endpoints per sync group	1 cloud endpoint	Yes
Server endpoints per sync group	50 server endpoints	No
Server endpoints per server	30 server endpoints	Yes
File system objects (directories and files) per sync group	100 million objects	No
Maximum number of file system objects (directories and files) in a directory	5 million objects	Yes
Maximum object (directories and files) security descriptor size	64 KiB	Yes
File size	100 GiB	No
Minimum file size for a file to be tiered	V9: Based on file system cluster size (double file system cluster size). For example, if the file system cluster size is 4kb, the minimum file size will be 8kb. V8 and older: 64 KiB	Yes

NOTE

An Azure File Sync endpoint can scale up to the size of an Azure file share. If the Azure file share size limit is reached, sync will not be able to operate.

Azure Queue storage limits

RESOURCE	TARGET
Maximum size of a single queue	500 TiB
Maximum size of a message in a queue	64 KiB
Maximum number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second, which assumes a 1-KiB message size
Target throughput for a single queue (1-KiB messages)	Up to 2,000 messages per second

Azure Table storage limits

RESOURCE	TARGET
Maximum size of a single table	500 TiB
Maximum size of a table entity	1 MiB
Maximum number of properties in a table entity	255, which includes three system properties: PartitionKey, RowKey, and Timestamp
Maximum total size of property values in an entity	1 MiB
Maximum total size of an individual property in an entity	Varies by property type. For more information, see Property Types in Understanding the Table Service Data Model .
Maximum number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second, which assumes a 1-KiB entity size
Target throughput for a single table partition (1 KiB-entities)	Up to 2,000 entities per second

Virtual machine disk limits

You can attach a number of data disks to an Azure virtual machine. Based on the scalability and performance targets for a VM's data disks, you can determine the number and type of disk that you need to meet your performance and capacity requirements.

IMPORTANT

For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks aren't highly utilized at the same time, the virtual machine can support a larger number of disks.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. There is no limit for the number of Managed Disks, snapshots and images per resource group.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Standard managed disks	50,000	50,000
Standard SSD managed disks	50,000	50,000
Premium managed disks	50,000	50,000
Standard_LRS snapshots	50,000	50,000
Standard_ZRS snapshots	50,000	50,000
Managed image	50,000	50,000

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

You can roughly calculate the number of highly utilized disks supported by a single Standard storage account based on the request rate limit. For example, for a Basic tier VM, the maximum number of highly utilized disks is about 66, which is 20,000/300 IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is 20,000/500 IOPS per disk.

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

For more information, see [Virtual machine sizes](#).

Managed virtual machine disks

Standard HDD managed disks

STANDARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000							
Throughput per disk	Up to 60 MiB/sec	Up to 300 MiB/sec	Up to 500 MiB/sec	Up to 500 MiB/sec							

Standard SSD managed disks

STANDARD SSD SIZE S	E1*	E2*	E3*	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 120	Up to 120	Up to 120	Up to 120	Up to 240	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000	Up to 6,000				
Throughput per disk	Up to 25 MiB/sec	Up to 50 MiB/sec	Up to 60 MiB/sec	Up to 400 MiB/sec	Up to 600 MiB/sec	Up to 750 MiB/sec	Up to 750 MiB/sec							

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

Premium SSD managed disks: Per-disk limits

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

Premium SSD managed disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: Per-disk limits

VM TIER	BASIC TIER VM	STANDARD TIER VM
Disk size	4,095 GB	4,095 GB
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum IOPS	66	40

Premium unmanaged virtual machine disks: Per-account limits

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

¹Ingress refers to all data from requests that are sent to a storage account. Egress refers to all data from responses that are received from a storage account.

Premium unmanaged virtual machine disks: Per-disk limits

PREMIUM STORAGE DISK TYPE	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec

PREMIUM STORAGE DISK TYPE	P10	P20	P30	P40	P50
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

StorSimple System limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of storage account credentials	64	
Maximum number of volume containers	64	
Maximum number of volumes	255	
Maximum number of schedules per bandwidth template	168	A schedule for every hour, every day of the week.
Maximum size of a tiered volume on physical devices	64 TB for StorSimple 8100 and StorSimple 8600	StorSimple 8100 and StorSimple 8600 are physical devices.
Maximum size of a tiered volume on virtual devices in Azure	30 TB for StorSimple 8010 64 TB for StorSimple 8020	StorSimple 8010 and StorSimple 8020 are virtual devices in Azure that use Standard storage and Premium storage, respectively.
Maximum size of a locally pinned volume on physical devices	9 TB for StorSimple 8100 24 TB for StorSimple 8600	StorSimple 8100 and StorSimple 8600 are physical devices.
Maximum number of iSCSI connections	512	
Maximum number of iSCSI connections from initiators	512	
Maximum number of access control records per device	64	
Maximum number of volumes per backup policy	24	

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of backups retained per backup policy	64	
Maximum number of schedules per backup policy	10	
Maximum number of snapshots of any type that can be retained per volume	256	This amount includes local snapshots and cloud snapshots.
Maximum number of snapshots that can be present in any device	10,000	
Maximum number of volumes that can be processed in parallel for backup, restore, or clone	16	<ul style="list-style-type: none"> If there are more than 16 volumes, they're processed sequentially as processing slots become available. New backups of a cloned or a restored tiered volume can't occur until the operation is finished. For a local volume, backups are allowed after the volume is online.
Restore and clone recover time for tiered volumes	<2 minutes	<ul style="list-style-type: none"> The volume is made available within 2 minutes of a restore or clone operation, regardless of the volume size. The volume performance might initially be slower than normal as most of the data and metadata still resides in the cloud. Performance might increase as data flows from the cloud to the StorSimple device. The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate might be affected by Internet bandwidth to the cloud. The restore or clone operation is complete when all the metadata is on the device. Backup operations can't be performed until the restore or clone operation is fully complete.

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

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

Stream Analytics limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of streaming units per subscription per region	500	To request an increase in streaming units for your subscription beyond 500, contact Microsoft Support .
Maximum number of inputs per job	60	There's a hard limit of 60 inputs per Azure Stream Analytics job.
Maximum number of outputs per job	60	There's a hard limit of 60 outputs per Stream Analytics job.
Maximum number of functions per job	60	There's a hard limit of 60 functions per Stream Analytics job.
Maximum number of streaming units per job	192	There's a hard limit of 192 streaming units per Stream Analytics job.
Maximum number of jobs per region	1,500	Each subscription can have up to 1,500 jobs per geographical region.
Reference data blob MB	300	Reference data blobs can't be larger than 300 MB each.

Virtual Machines limits

Virtual Machines limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual machines per cloud service ¹	50	50
Input endpoints per cloud service ²	150	150

¹Virtual machines created by using the classic deployment model instead of Azure Resource Manager are automatically stored in a cloud service. You can add more virtual machines to that cloud service for load balancing and availability.

²Input endpoints allow communications to a virtual machine from outside the virtual machine's cloud service. Virtual machines in the same cloud service or virtual network can automatically communicate with each other. For more information, see [How to set up endpoints to a virtual machine](#).

Virtual Machines limits - Azure Resource Manager

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT
VMs per subscription	25,000 ¹ per region.
VM total cores per subscription	20 ¹ per region. Contact support to increase limit.
Azure Spot VM total cores per subscription	20 ¹ per region. Contact support to increase limit.
VM per series, such as Dv2 and F, cores per subscription	20 ¹ per region. Contact support to increase limit.

RESOURCE	DEFAULT LIMIT
Availability sets per subscription	2,000 per region.
Virtual machines per availability set	200
Certificates per subscription	Unlimited ²

¹Default limits vary by offer category type, such as Free Trial and Pay-As-You-Go, and by series, such as Dv2, F, and G. For example, the default for Enterprise Agreement subscriptions is 350.

²With Azure Resource Manager, certificates are stored in the Azure Key Vault. The number of certificates is unlimited for a subscription. There's a 1-MB limit of certificates per deployment, which consists of either a single VM or an availability set.

NOTE

Virtual machine cores have a regional total limit. They also have a limit for regional per-size series, such as Dv2 and F. These limits are separately enforced. For example, consider a subscription with a US East total VM core limit of 30, an A series core limit of 30, and a D series core limit of 30. This subscription can deploy 30 A1 VMs, or 30 D1 VMs, or a combination of the two not to exceed a total of 30 cores. An example of a combination is 10 A1 VMs and 20 D1 VMs.

Shared Image Gallery limits

There are limits, per subscription, for deploying resources using Shared Image Galleries:

- 100 shared image galleries, per subscription, per region
- 1,000 image definitions, per subscription, per region
- 10,000 image versions, per subscription, per region

Virtual machine scale sets limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of VMs in a scale set	1,000	1,000
Maximum number of VMs based on a custom VM image in a scale set	600	600
Maximum number of scale sets in a region	2,000	2,000

See also

- [Understand Azure limits and increases](#)
- [Virtual machine and cloud service sizes for Azure](#)
- [Sizes for Azure Cloud Services](#)
- [Naming rules and restrictions for Azure resources](#)

ExpressRoute for Cloud Solution Providers (CSP)

7/12/2019 • 9 minutes to read • [Edit Online](#)

Microsoft provides hyper-scale services for traditional resellers and distributors (CSP) to be able to rapidly provision new services and solutions for your customers without the need to invest in developing these new services. To allow the Cloud Solution Provider (CSP) the ability to directly manage these new services, Microsoft provides programs and APIs that allow the CSP to manage Microsoft Azure resources on behalf of your customers. One of those resources is ExpressRoute. ExpressRoute allows the CSP to connect existing customer resources to Azure services. ExpressRoute is a high speed private communications link to services in Azure.

ExpressRoute is comprised of a pair of circuits for high availability that are attached to a single customer's subscription(s) and cannot be shared by multiple customers. Each circuit should be terminated in a different router to maintain the high availability.

NOTE

There are bandwidth and connection caps on ExpressRoute which means that large/complex implementations will require multiple ExpressRoute circuits for a single customer.

Microsoft Azure provides a growing number of services that you can offer to your customers. ExpressRoute helps you and your customers take advantage of these services by providing high speed low latency access to the Microsoft Azure environment.

Microsoft Azure management

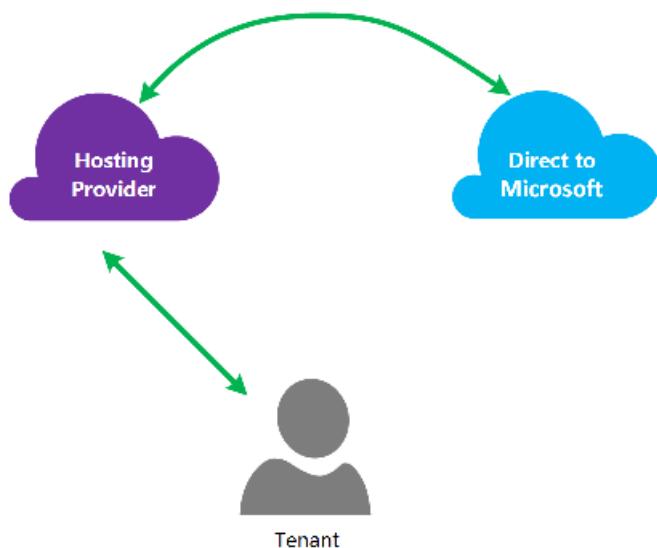
Microsoft provides CSPs with APIs to manage the Azure customer subscriptions by allowing programmatic integration with your own service management systems. Supported management capabilities can be found [here](#).

Microsoft Azure resource management

The contract you have with your customer will determine how the subscription will be managed. The CSP can directly manage the creation and maintenance of resources or the customer can maintain control of the Microsoft Azure subscription and create the Azure resources as they need. If your customer manages the creation of resources in their Microsoft Azure subscription they will use one of two models: "*Connect-Through*" model, or "*Direct-To*" model. These models are described in detail in the following sections.

Connect-through model

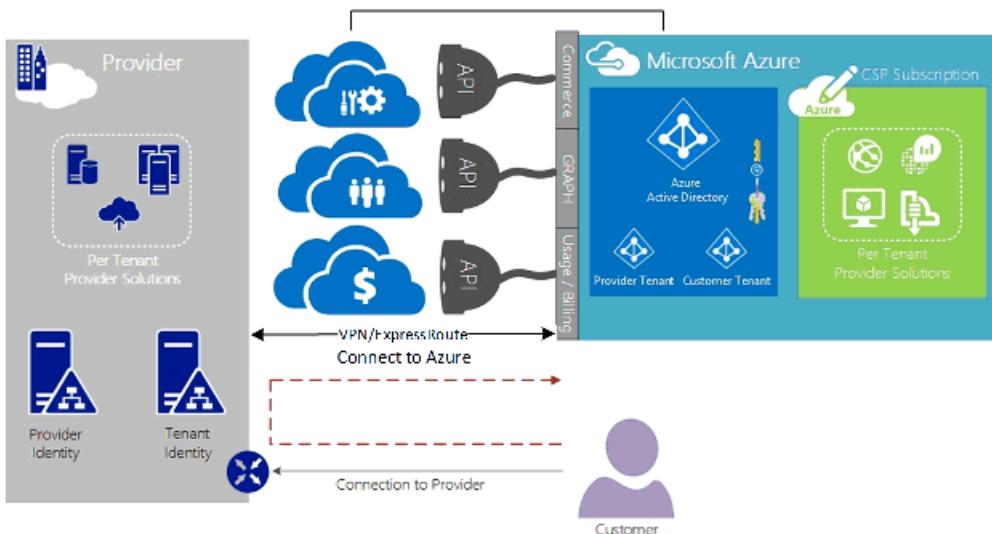
Connect-Through



In the connect-through model, the CSP creates a direct connection between your datacenter and your customer's Azure subscription. The direct connection is made using ExpressRoute, connecting your network with Azure. Then your customer connects to your network. This scenario requires that the customer passes through the CSP network to access Azure services.

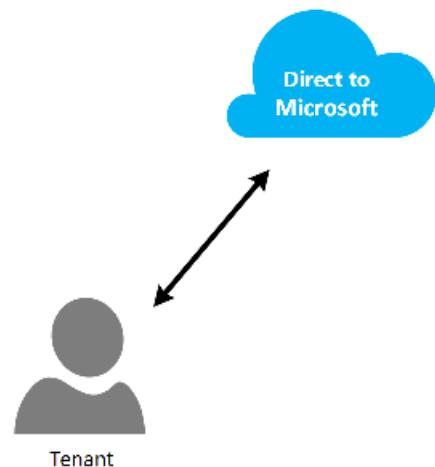
If your customer has other Azure subscriptions not managed by you, they would use the public Internet or their own private connection to connect to those services provisioned under the non CSP subscription.

For CSP managing Azure services, it is assumed that the CSP has a previously established customer identity store which would then be replicated into Azure Active Directory for management of their CSP subscription through Administrate-On-Behalf-Of (AOBO). Key drivers for this scenario include where a given partner or service provider has an established relationship with the customer, the customer is consuming provider services currently or the partner has a desire to provide a combination of provider-hosted and Azure-hosted solutions to provide flexibility and address customer challenges which cannot be satisfied by CSP alone. This model is illustrated in **Figure**, below.



Connect-to model

Connect-To

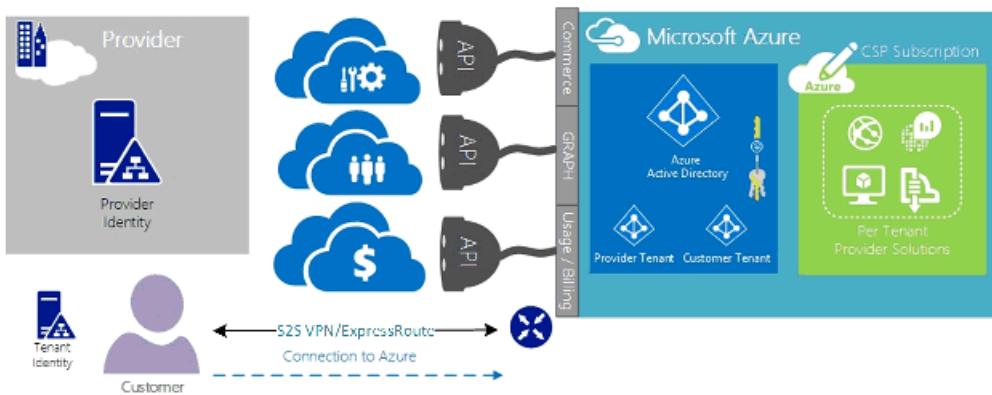


In the Connect-To model, the service provider creates a direct connection between their customer's datacenter and the CSP provisioned Azure subscription using ExpressRoute over the customer's (customer) network.

NOTE

For ExpressRoute the customer would need to create and maintain the ExpressRoute circuit.

This connectivity scenario requires that the customer connects directly through a customer network to access CSP-managed Azure subscription, using a direct network connection that is created, owned and managed either wholly or in part by the customer. For these customers it is assumed that the provider does not currently have a customer identity store established, and the provider would assist the customer in replicating their current identify store into Azure Active Directory for management of their subscription through AOOB. Key drivers for this scenario include where a given partner or service provider has an established relationship with the customer, the customer is consuming provider services currently, or the partner has a desire to provide services that are based solely on Azure-hosted solutions without the need for an existing provider datacenter or infrastructure.



The choice between these two option are based on your customer's needs and your current need to provide Azure services. The details of these models and the associated role-based access control, networking, and identity design patterns are covered in details in the following links:

- **Role Based Access Control (RBAC)** – RBAC is based on Azure Active Directory. For more information on

Azure RBAC see [here](#).

- **Networking** – Covers the various topics of networking in Microsoft Azure.
- **Azure Active Directory (Azure AD)** – Azure AD provides the identity management for Microsoft Azure and 3rd party SaaS applications. For more information about Azure AD see [here](#).

Network speeds

ExpressRoute supports network speeds from 50 Mb/s to 10Gb/s. This allows customers to purchase the amount of network bandwidth needed for their unique environment.

NOTE

Network bandwidth can be increased as needed without disrupting communications, but to reduce the network speed requires tearing down the circuit and recreating it at the lower network speed.

ExpressRoute supports the connection of multiple vNets to a single ExpressRoute circuit for better utilization of the higher-speed connections. A single ExpressRoute circuit can be shared among multiple Azure subscriptions owned by the same customer.

Configuring ExpressRoute

ExpressRoute can be configured to support three types of traffic ([routing domains](#)) over a single ExpressRoute circuit. This traffic is segregated into Microsoft peering, Azure public peering and private peering. You can choose one or all types of traffic to be sent over a single ExpressRoute circuit or use multiple ExpressRoute circuits depending on the size of the ExpressRoute circuit and isolation required by your customer. The security posture of your customer may not allow public traffic and private traffic to traverse over the same circuit.

Connect-through model

In a connect-through configuration the you will be responsible for all of the networking underpinnings to connect your customers datacenter resources to the subscriptions hosted in Azure. Each of your customer's that want to use Azure capabilities will need their own ExpressRoute connection, which will be managed by the You. The you will use the same methods the customer would use to procure the ExpressRoute circuit. The you will follow the same steps outlined in the article [ExpressRoute workflows](#) for circuit provisioning and circuit states. The you will then configure the Border Gateway Protocol (BGP) routes to control the traffic flowing between the on-premises network and Azure vNet.

Connect-to model

In a connect-to configuration, your customer already has an existing connection to Azure or will initiate a connection to the internet service provider linking ExpressRoute from your customer's own datacenter directly to Azure, instead of your datacenter. To begin the provisioning process, your customer will follow the steps as described in the Connect-Through model, above. Once the circuit has been established your customer will need to configure the on-premises routers to be able to access both your network and Azure vNets.

You can assist with setting up the connection and configuring the routes to allow the resources in your datacenter(s) to communicate with the client resources in your datacenter, or with the resources hosted in Azure.

ExpressRoute routing domains

ExpressRoute offers three routing domains: public, private, and Microsoft peering. Each of the routing domains are configured with identical routers in active-active configuration for high availability. For more details on ExpressRoute routing domains look [here](#).

You can define custom routes filters to allow only the route(s) you want to allow or need. For more information or to see how to make these changes see article: [Create and modify routing for an ExpressRoute circuit using](#)

[PowerShell](#) for more details about routing filters.

NOTE

For Microsoft and Public Peering connectivity must be though a public IP address owned by the customer or CSP and must adhere to all defined rules. For more information, see the [ExpressRoute Prerequisites](#) page.

Routing

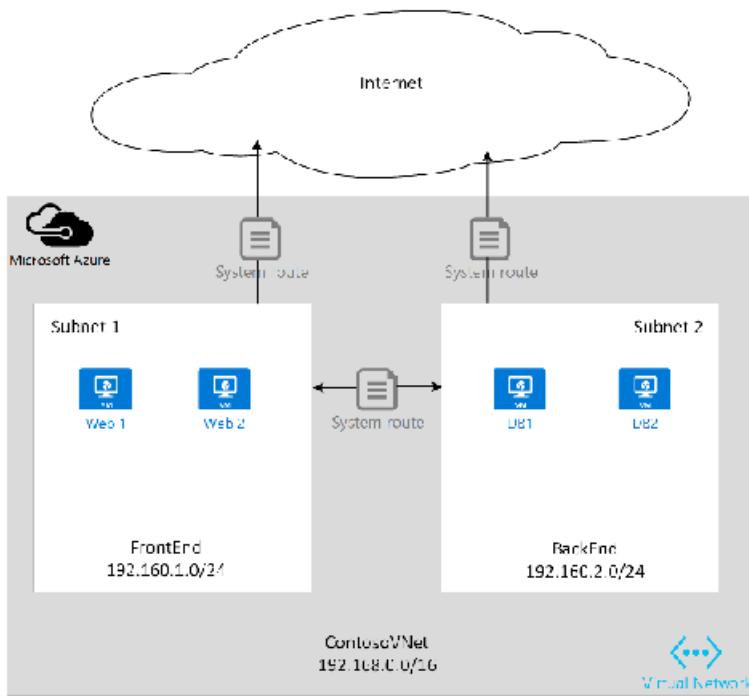
ExpressRoute connects to the Azure networks through the Azure Virtual Network Gateway. Network gateways provide routing for Azure virtual networks.

Creating Azure Virtual Networks also creates a default routing table for the vNet to direct traffic to/from the subnets of the vNet. If the default route table is insufficient for the solution custom routes can be created to route outgoing traffic to custom appliances or to block routes to specific subnets or external networks.

Default routing

The default route table includes the following routes:

- Routing within a subnet
- Subnet-to-subnet within the virtual network
- To the Internet
- Virtual network-to-virtual network using VPN gateway
- Virtual network-to-on-premises network using a VPN or ExpressRoute gateway



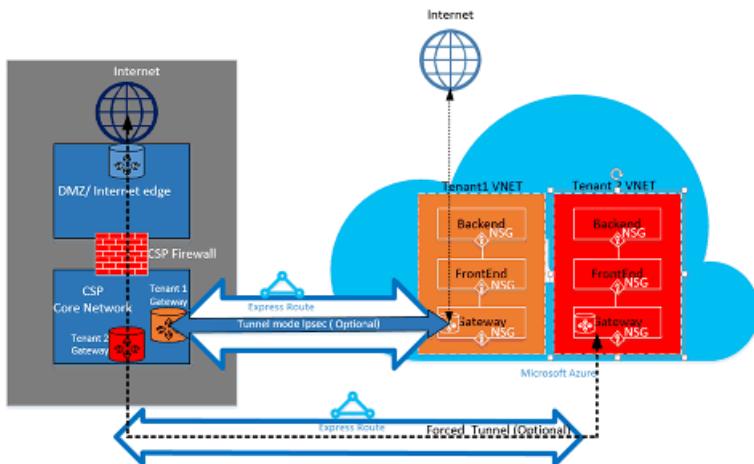
User-defined routing (UDR)

User-defined routes allow the control of traffic outbound from the assigned subnet to other subnets in the virtual network or over one of the other predefined gateways (ExpressRoute; internet or VPN). The default system routing table can be replaced with a user-defined routing table that replaces the default routing table with custom routes. With user-defined routing, customers can create specific routes to appliances such as firewalls or intrusion detection appliances, or block access to specific subnets from the subnet hosting the user-defined route. For an overview of User Defined Routes look [here](#).

Security

Depending on which model is in use, Connect-To or Connect-Through, your customer defines the security policies in their vNet or provides the security policy requirements to the CSP to define to their vNets. The following security criteria can be defined:

1. **Customer Isolation** — The Azure platform provides customer isolation by storing Customer ID and vNet info in a secure database, which is used to encapsulate each customer's traffic in a GRE tunnel.
2. **Network Security Group (NSG)** rules are for defining allowed traffic into and out of the subnets within vNets in Azure. By default, the NSG contain Block rules to block traffic from the Internet to the vNet and Allow rules for traffic within a vNet. For more information about Network Security Groups look [here](#).
3. **Force tunneling** — This is an option to redirect internet bound traffic originating in Azure to be redirected over the ExpressRoute connection to the on premises datacenter. For more information about Forced tunneling look [here](#).
4. **Encryption** — Even though the ExpressRoute circuits are dedicated to a specific customer, there is the possibility that the network provider could be breached, allowing an intruder to examine packet traffic. To address this potential, a customer or CSP can encrypt traffic over the connection by defining IPSec tunnel-mode policies for all traffic flowing between the on premises resources and Azure resources (refer to the optional Tunnel mode IPsec for Customer 1 in Figure 5: ExpressRoute Security, above). The second option would be to use a firewall appliance at each the end point of the ExpressRoute circuit. This will require additional 3rd party firewall VMs/Appliances to be installed on both ends to encrypt the traffic over the ExpressRoute circuit.



Next steps

The Cloud Solution Provider service provides you a way to increase your value to your customers without the need for expensive infrastructure and capability purchases, while maintaining your position as the primary outsourcing provider. Seamless integration with Microsoft Azure can be accomplished through the CSP API, allowing you to integrate management of Microsoft Azure within your existing management frameworks.

Additional Information can be found at the following links:

[Azure in Cloud Solution Provider program.](#)

[Get ready to transact as a Cloud Solution Provider.](#)

[Microsoft Cloud Solution Provider resources.](#)

ExpressRoute CrossConnections API development and integration

2/12/2020 • 3 minutes to read • [Edit Online](#)

The ExpressRoute Partner Resource Manager API allows ExpressRoute partners to manage the layer-2 and layer-3 configuration of customer ExpressRoute circuits. The ExpressRoute Partner Resource Manager API introduces a new resource type, **expressRouteCrossConnections**. Partners use this resource to manage customer ExpressRoute circuits.

Workflow

The expressRouteCrossConnections resource is a shadow resource to the ExpressRoute circuit. When an Azure customer creates an ExpressRoute circuit and selects a specific ExpressRoute partner, Microsoft creates an expressRouteCrossConnections resource in the partner's Azure ExpressRoute management subscription. In doing so, Microsoft defines a resource group to create the expressRouteCrossConnections resource in. The naming standard for the resource group is **CrossConnection-PeeringLocation**; where PeeringLocation = the ExpressRoute Location. For example, if a customer creates an ExpressRoute circuit in Denver, the CrossConnection will be created in the partner's Azure subscription in the following resource group: **CrossConnection-Denver**.

ExpressRoute partners manage layer-2 and layer-3 configuration by issuing REST operations against the expressRouteCrossConnections resource.

Benefits

Benefits of moving to the expressRouteCrossConnections resource:

- Any future enhancements for ExpressRoute partners will be made available on the ExpressRouteCrossConnection resource.
- Partners can apply [Role-Based Access Control](#) to the expressRouteCrossConnection resource. These controls can define permissions for which users accounts can modify the expressRouteCrossConnection resource and add/update/delete peering configurations.
- The expressRouteCrossConnection resource exposes APIs that can be helpful in troubleshooting ExpressRoute connections. This includes ARP table, BGP Route Table Summary, and BGP Route Table details. This capability is not supported by classic deployment APIs.
- Partners can also look up the advertised communities on Microsoft peering by using the *RouteFilter* resource.

API development and integration steps

To develop against the Partner API, ExpressRoute partners leverage a test customer and test partner setup. The test customer setup will be used to create ExpressRoute circuits in test peering locations that map to dummy devices and ports. The test partner setup is used to manage the ExpressRoute circuits created in the test peering location.

1. Enlist subscriptions

To request the test partner and test customer setup, enlist two Pay-As-You-Go Azure subscriptions to your ExpressRoute engineering contact:

- **ExpressRoute_API_Dev_Provider_Sub:** This subscription will be used to manage ExpressRoute circuits

created in test peering locations on dummy devices and ports.

- **ExpressRoute_API_Dev_Customer_Sub:** This subscription will be used to create ExpressRoute circuits in test peering locations that map to dummy devices and ports.

The test peering locations: dummy devices and ports are not exposed to production customers by default. In order to create ExpressRoute circuits that map to the test setup, a subscription feature flag needs to be enabled.

2. Register the Dev_Provider subscription to access the expressRouteCrossConnections API

In order to access the expressRouteCrossConnections API, the partner subscription needs to be enrolled in the **Microsoft.Network Resource Provider**. Follow the steps in the [Azure resource providers and types](#) article to complete the registration process.

3. Set up authentication for Azure Resource Manager REST API calls

Most Azure services require client code to authenticate with Resource Manager, using valid credentials, prior to calling service APIs. Authentication is coordinated between the various actors by Azure AD and provides the client with an access token as proof of authentication.

The authentication process involves two main steps:

1. [Register the client](#).
2. [Create the access request](#).

4. Provide Network Contributor permission to the client application

Once authentication has been successfully configured, you need to grant Network Contributor access to your client application, under the Dev_Provider_Sub. To grant permission, sign in to the [Azure portal](#) and complete the following steps:

1. Navigate to Subscriptions and select the Dev_Provider_Sub
2. Navigate to Access Control (IAM)
3. Add Role Assignment
4. Select the Network Contributor Role
5. Assign Access to Azure AD User, Group, or Service Principal
6. Select your client application
7. Save changes

5. Develop

Develop against the [expressRouteCrossConnections API](#).

REST API

See [ExpressRoute CrossConnections REST API](#) for REST API documentation.

Next steps

For more information on all ExpressRoute REST APIs, see [ExpressRoute REST APIs](#).