

AZ-700 Designing and Implementing Microsoft Azure Networking Solutions

Introduction to Azure Virtual Networks

Explore Azure Virtual Networks

Completed 100 XP

- 10 minutes

Azure Virtual Networks (VNets) are the fundamental building block of your private network in Azure. VNets enable you to build complex virtual networks that are similar to an on-premises network, with additional benefits of Azure infrastructure such as scale, availability, and isolation.

Each VNet you create has its own CIDR block and can be linked to other VNets and on-premises networks as long as the CIDR blocks don't overlap. You also have control of DNS server settings for VNets, and segmentation of the VNet into subnets.

Capabilities of Azure Virtual Networks

Azure VNets enable resources in Azure to securely communicate with each other, the internet, and on-premises networks.

- **Communication with the internet.** All resources in a VNet can communicate outbound to the internet, by default. You can communicate inbound to a resource by assigning a public IP address or a public Load Balancer. You can also use public IP or public Load Balancer to manage your outbound connections.
- **Communication between Azure resources.** There are three key mechanisms through which Azure resource can communicate: VNets, VNet service endpoints and VNet peering. Virtual Networks can connect not only VMs, but other Azure Resources, such as the App Service Environment, Azure Kubernetes Service, and Azure Virtual Machine Scale Sets. You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. When you create a VNet, your services and VMs within your VNet can communicate directly and securely with each other in the cloud.

- **Communication between on-premises resources.** Securely extend your data center. You can connect your on-premises computers and networks to a virtual network using any of the following options: Point-to-site virtual private network (VPN), Site-to-site VPN, Azure ExpressRoute.
- **Filtering network traffic.** You can filter network traffic between subnets using any combination of network security groups and network virtual appliances like firewalls, gateways, proxies, and Network Address Translation (NAT) services.
- **Routing network traffic.** Azure routes traffic between subnets, connected virtual networks, on-premises networks, and the Internet, by default. You can implement route tables or border gateway protocol (BGP) routes to override the default routes Azure creates.

Design considerations for Azure Virtual Networks

Address space and subnets

You can create multiple virtual networks per region per subscription. You can create multiple subnets within each virtual network.

Virtual Networks

When creating a VNet, it's recommended that you use the address ranges enumerated in RFC 1918, which have been set aside by the IETF for private, non-routable address spaces:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

In addition, you can't add the following address ranges:

- 224.0.0.0/4 (Multicast)
- 255.255.255.255/32 (Broadcast)
- 127.0.0.0/8 (Loopback)
- 169.254.0.0/16 (Link-local)
- 168.63.129.16/32 (Internal DNS)

Azure assigns resources in a virtual network a private IP address from the address space that you provision. For example, if you deploy a VM in a VNet with subnet address space 192.168.1.0/24, the VM will be assigned a private IP like 192.168.1.4. Azure reserves the

first four and last IP address for a total of 5 IP addresses within each subnet. These are x.x.x.0-x.x.x.3 and the last address of the subnet.

For example, the IP address range of 192.168.1.0/24 has the following reserved addresses:

- 192.168.1.0 : Network address
- 192.168.1.1 : Reserved by Azure for the default gateway
- 192.168.1.2, 192.168.1.3 : Reserved by Azure to map the Azure DNS IPs to the VNet space
- 192.168.1.255 : Network broadcast address.

When planning to implement virtual networks, you need to consider the following:

- Ensure non-overlapping address spaces. Make sure your VNet address space (CIDR block) doesn't overlap with your organization's other network ranges.
- Is any security isolation required?
- Do you need to mitigate any IP addressing limitations?
- Will there be connections between Azure VNets and on-premises networks?
- Is there any isolation required for administrative purposes?
- Are you using any Azure services that create their own VNets?

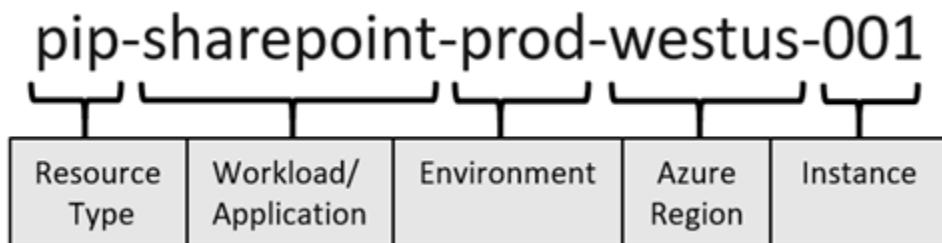
Subnets

A subnet is a range of IP address in the VNet. You can segment VNets into different size subnets, creating as many subnets as you require for organization and security within the subscription limit. You can then deploy Azure resources in a specific subnet. Just like in a traditional network, subnets allow you to segment your VNet address space into segments that are appropriate for the organization's internal network. This also improves address allocation efficiency. The smallest supported IPv4 subnet is /29, and the largest is /2 (using CIDR subnet definitions). IPv6 subnets must be exactly /64 in size. When planning to implement subnets, you need to consider the following:

- Each subnet must have a unique address range, specified in Classless Inter-Domain Routing (CIDR) format.
- Certain Azure services require their own subnet.
- Subnets can be used for traffic management. For example, you can create subnets to route traffic through a network virtual appliance.
- You can limit access to Azure resources to specific subnets with a virtual network service endpoint. You can create multiple subnets, and enable a service endpoint for some subnets, but not others.

Determine a naming convention

As part of your Azure network design, it's important to plan your naming convention for your resources. An effective naming convention composes resource names from important information about each resource. A well-chosen name helps you quickly identify the resource's type, its associated workload, its deployment environment, and the Azure region hosting it. For example, a public IP resource for a production SharePoint workload residing in the West US region might be pip-sharepoint-prod-westus-001



All Azure resource types have a scope that defines the level that resource names must be unique. A resource must have a unique name within its scope. There are four levels you can specify a scope: [management group](#), subscription, [resource group](#), and resource. Scopes are hierarchical, with each level of hierarchy making the scope more specific.

For example, a virtual network has a resource group scope, which means that there can be only one network named vnet-prod-westus-001 in each resource group. Other resource groups could have their own virtual network named vnet-prod-westus-001. Subnets are scoped to virtual networks, so each subnet within a virtual network must have a distinct name.

Understand Regions and Subscriptions

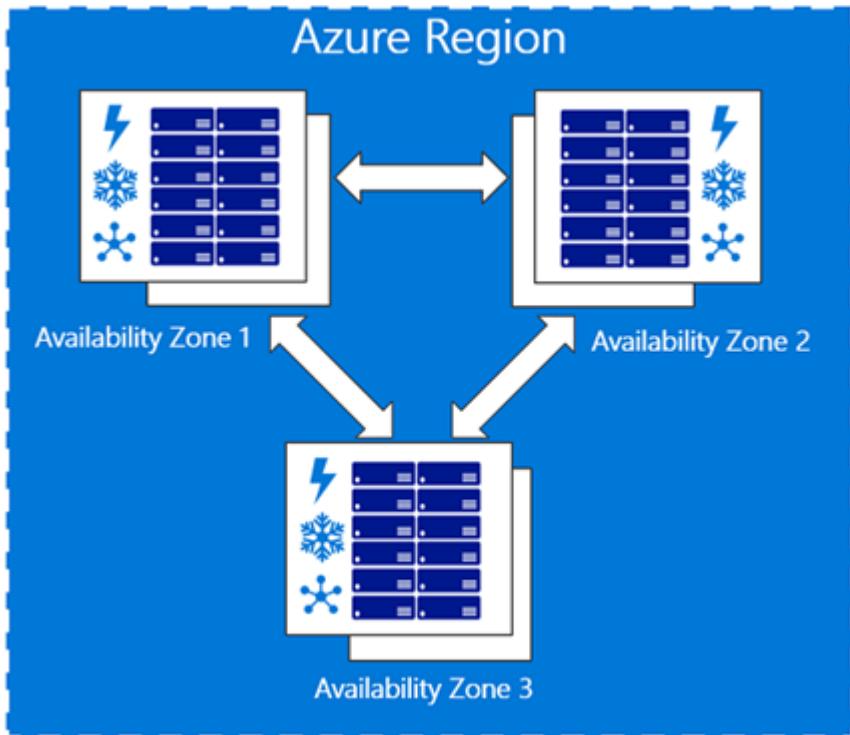
All Azure resources are created in an Azure region and subscription. A resource can only be created in a virtual network that exists in the same region and subscription as the resource. You can, however, connect virtual networks that exist in different subscriptions and regions. Azure regions are important to consider as you design your Azure network in relation to your infrastructure, data, applications, and end users.

You can deploy as many virtual networks as you need within each subscription, up to the subscription limit. Some larger organizations with global deployments have multiple virtual networks that are connected between regions, for example.



Azure Availability Zones

An Azure Availability Zone enables you to define unique physical locations within a region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. Designed to ensure high-availability of your Azure services, the physical separation of Availability Zones within a region protects applications and data from datacenter failures.



You should consider availability zones when designing your Azure network, and plan for services that support availability zones.

Azure services that support Availability Zones fall into three categories:

- Zonal services: Resources can be pinned to a specific zone. For example, virtual machines, managed disks, or standard IP addresses can be pinned to a specific zone, which allows for increased resilience by having one or more instances of resources spread across zones.
- Zone-redundant services: Resources are replicated or distributed across zones automatically. Azure replicates the data across three zones so that a zone failure doesn't impact its availability.
- Non-regional services: Services are always available from Azure geographies and are resilient to zone-wide outages as well as region-wide outages.

Configure public IP services

200 XP

- 7 minutes

Public networks like the Internet communicate by using public IP addresses. Private networks like your Azure Virtual Network use private IP addresses, which aren't routable

on public networks. To support a network that exists both in Azure and on-premises, you must configure IP addressing for both types of networks.

Public IP addresses enable Internet resources to communicate with Azure resources and enable Azure resources to communicate outbound with Internet and public-facing Azure services. A public IP address in Azure is dedicated to a specific resource, until it's unassigned by a network engineer. A resource without a public IP assigned can communicate outbound through network address translation services, where Azure dynamically assigns an available IP address that isn't dedicated to the resource.

As an example, public resources like web servers must be accessible from the internet. You want to ensure that you plan IP addresses that support these requirements.

In this unit, you'll learn about requirements for IP addressing when integrating an Azure network with on-premises networks, and you'll explore the constraints and limitations for public and private IP addresses in Azure. You also will look at the capabilities that are available in Azure to reassign IP addresses in your network.

Use dynamic and static public IP addresses

In Azure Resource Manager, a [public IP](#) address is a resource that has its own properties. Some of the resources you can associate a public IP address resource with:

- Virtual machine network interfaces
- Virtual machine scale sets
- Public Load Balancers
- Virtual Network Gateways (VPN/ER)
- NAT gateways
- Application Gateways
- Azure Firewall
- Bastion Host
- Route Server

Public IP addresses are created with an IPv4 or IPv6 address, which can be either static or dynamic.

A dynamic public IP address is an assigned address that can change over the lifespan of the Azure resource. The dynamic IP address is allocated when you create or start a VM. The IP address is released when you stop or delete the VM. In each Azure region, public IP addresses are assigned from a unique pool of addresses. The default allocation method is dynamic.

A static public IP address is an assigned address that won't change over the lifespan of the Azure resource. To ensure that the IP address for the resource remains the same, set the allocation method explicitly to static. In this case, an IP address is assigned immediately. It's released only when you delete the resource or change the IP allocation method to dynamic.

Choose the appropriate SKU for a public IP address

Public IP addresses are created with one of the following SKUs:

Public IP address	Standard	Basic
Allocation method	Static	For IPv4: Dynamic or Static; For IPv6: Dynamic.
Idle Timeout	Have an adjustable inbound originated flow idle timeout of 4-30 minutes, with a default of 4 minutes, and fixed outbound originated flow idle timeout of 4 minutes.	Have an adjustable inbound originated flow idle timeout of 4-30 minutes, with a default of 4 minutes, and fixed outbound originated flow idle timeout of 4 minutes.
Security	Secure by default model and be closed to inbound traffic when used as a frontend. Allow traffic with network security group (NSG) is required (for example, on the NIC of a virtual machine with a Standard SKU Public IP attached).	Open by default. Network security groups are recommended but optional for restricting inbound or outbound traffic
Availability zones	Supported. Standard IPs can be non-zonal, zonal, or zone-redundant. Zone redundant IPs can only be created in regions where 3 availability zones are live. IPs created before zones are live won't be zone redundant.	Not supported.
Routing preference	Supported to enable more granular control of how traffic is routed between Azure and the Internet.	Not supported.
Global tier	Supported via cross-region load balancers.	Not supported.

Create a public IP address prefix

Public IP prefixes are assigned from a pool of addresses in each Azure region. You create a public IP address prefix in an Azure region and subscription by specifying a name and prefix size. The prefix size is the number of addresses available for use. Public

IP address prefixes consist of IPv4 or IPv6 addresses. In regions with Availability Zones, Public IP address prefixes can be created as zone-redundant or associated with a specific availability zone. After the public IP prefix is created, you can create public IP addresses.

Custom IP address prefix (BYOIP)

Addresses from a custom IP address prefix can be used in the same way as Azure owned public IP address prefixes. Addresses from a custom IP address prefix can be associated to Azure resources, interact with internal/private IPs and virtual networks, and reach external destinations outbound from the Azure Wide Area Network.

Exercise: Design and implement a virtual network in Azure

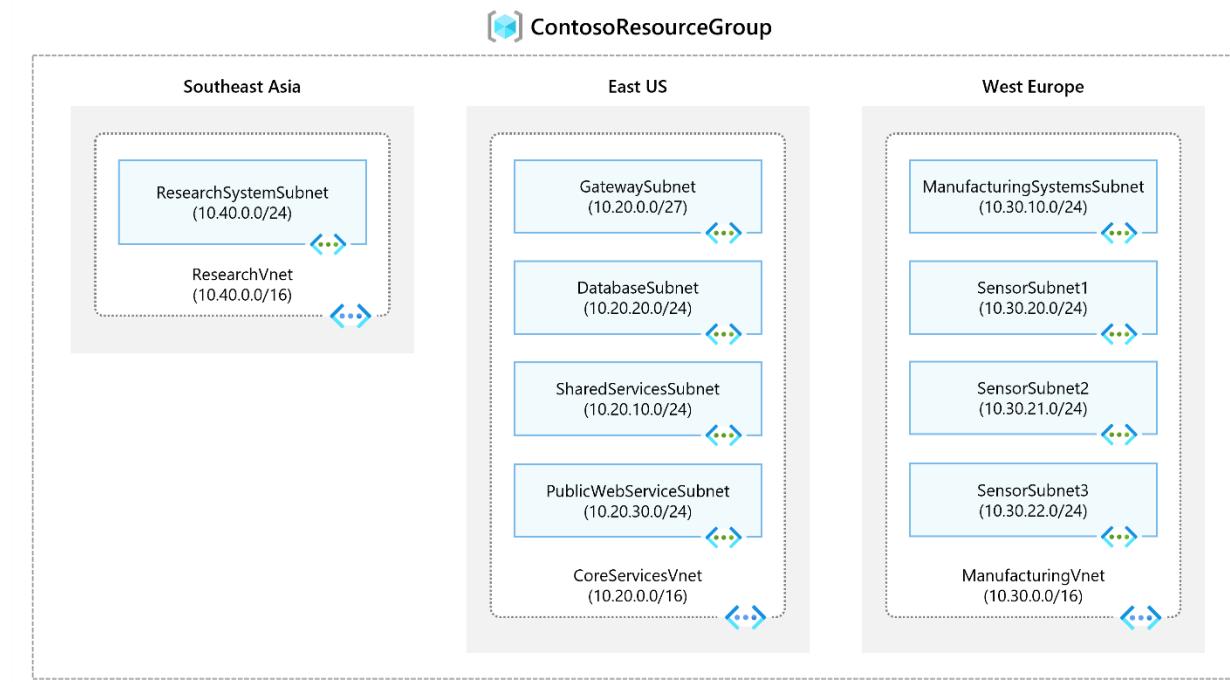
Completed 100 XP

- 7 minutes

Lab scenario

Contoso Ltd, a fictitious organization, is in the process of migrating infrastructure and applications to Azure. As network engineer, you will plan and implement three virtual networks and subnets to support resources in those virtual networks.

Architecture diagram



Objectives

- **Task 1:** Create the **Contoso** resource group.
- **Task 2:** Create the **CoreServicesVnet** virtual network and subnets.
- **Task 3:** Create the **ManufacturingVnet** virtual network and subnets.
- **Task 4:** Create the **ResearchVnet** virtual network and subnets.
- **Task 5:** Verify the creation of the virtual networks and subnets.

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

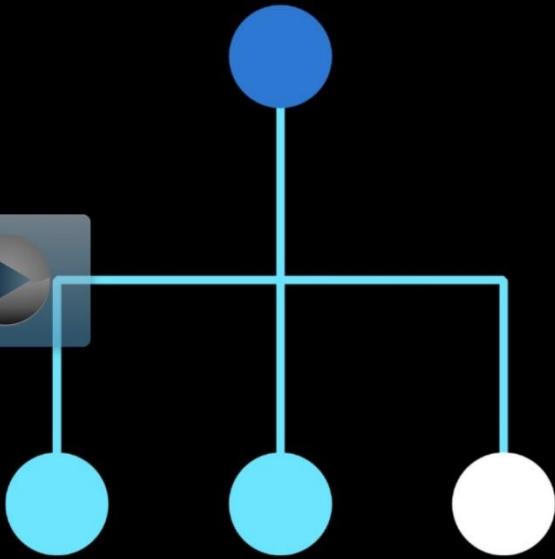
You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



Interactive Lab Simulation:

Design and implement a virtual network in Azure

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Design%20and%20implement%20a%20virtual%20network%20in%20Azure>

Design name resolution for your virtual network

200 XP

- 10 minutes

Depending on how you use Azure to host IaaS, PaaS, and hybrid solutions, you might need to allow the virtual machines (VMs), and other resources deployed in a virtual network to communicate with each other. Although you can enable communication by using IP addresses, it is much simpler to use names that can be easily remembered, and do not change.

DNS is split into two areas: Public, and Private DNS for resources accessible from your own internal networks.

Public DNS services

Public DNS services resolve names and IP addresses for resources and services accessible over the internet such as web servers. Azure DNS is a hosting service for DNS domain that provides name resolution by using Microsoft Azure infrastructure. DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers. Azure DNS uses anycast networking. Each DNS query is answered by the closest available DNS server to provide fast performance and high availability for your domain.

In Azure DNS, you can create address records manually within relevant zones. The records most frequently used will be:

- Host records: A/AAAA (IPv4/IPv6)
- Alias records: CNAME

Azure DNS provides a reliable, secure DNS service to manage and resolve domain names in a virtual network without needing to add a custom DNS solution.

A DNS zone hosts the DNS records for a domain. So, to start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone.

Considerations

- The name of the zone must be unique within the resource group, and the zone must not exist already.
- The same zone name can be reused in a different resource group or a different Azure subscription.
- Where multiple zones share the same name, each instance is assigned different name server addresses.
- Root/Parent domain is registered at the registrar and pointed to Azure NS.
- Child domains are registered in AzureDNS directly.

Note

You do not have to own a domain name to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to configure the domain.

Delegate DNS Domains

To delegate your domain to Azure DNS, you first need to know the name server names for your zone. Each time a DNS zone is created Azure DNS allocates name servers from

a pool. Once the Name Servers are assigned, Azure DNS automatically creates authoritative NS records in your zone.

Once the DNS zone is created, and you have the name servers, you need to update the parent domain. Each registrar has their own DNS management tools to change the name server records for a domain. In the registrar's DNS management page, edit the NS records and replace the NS records with the ones Azure DNS created.

Note

When delegating a domain to Azure DNS, you must use the name server names provided by Azure DNS. You should always use all four name server names, regardless of the name of your domain.

Child Domains

If you want to set up a separate child zone, you can delegate a subdomain in Azure DNS. For example, after configuring contoso.com in Azure DNS, you could configure a separate child zone for partners.contoso.com.

Setting up a subdomain follows the same process as typical delegation. The only difference is that NS records must be created in the parent zone contoso.com in Azure DNS, rather than in the domain registrar.

Note

The parent and child zones can be in the same or different resource group. Notice that the record set name in the parent zone matches the child zone name, in this case *partners*.

It's important to understand the difference between DNS record sets and individual DNS records. A record set is a collection of records in a zone that have the same name and are the same type.



A record set cannot contain two identical records. Empty record sets (with zero records) can be created, but do not appear on the Azure DNS name servers. Record sets of type CNAME can contain one record at most.

The **Add record set** page will change depending on the type of record you select. For an A record, you will need the TTL (Time to Live) and IP address. The time to live, or TTL, specifies how long each record is cached by clients before being requeried.

Add record set

azureadmininc.org

Name

helloworld ✓

.azureadmininc.org

Type

A ▼

Alias record set (i)

Yes No

TTL *

1

TTL unit

Hours ▼

IP address

0.0.0.0 ...

Private DNS services

Private DNS services resolve names and IP addresses for resources and services

When resources deployed in virtual networks need to resolve domain names to internal IP addresses, they can use one of the three methods:

- Azure DNS Private Zones
- Azure-provided name resolution
- Name resolution that uses your own DNS server

The type of name resolution you use depends on how your resources need to communicate with each other.

Your name resolution needs might go beyond the features provided by Azure. For example, you might need to use Microsoft Windows Server Active Directory domains, resolve DNS names between virtual networks. To cover these scenarios, Azure provides the ability for you to use your own DNS servers.

DNS servers within a virtual network can forward DNS queries to the recursive resolvers in Azure. This enables you to resolve host names within that virtual network. For example, a domain controller (DC) running in Azure can respond to DNS queries for its domains and forward all other queries to Azure. Forwarding queries allows VMs to see both your on-premises resources (via the DC) and Azure-provided host names (via the forwarder). Access to the recursive resolvers in Azure is provided via the virtual IP 168.63.129.16.

DNS forwarding also enables DNS resolution between virtual networks and allows your on-premises machines to resolve Azure-provided host names. In order to resolve a VM's host name, the DNS server VM must reside in the same virtual network and be configured to forward host name queries to Azure. Because the DNS suffix is different in each virtual network, you can use conditional forwarding rules to send DNS queries to the correct virtual network for resolution.

Azure provided DNS

Azure provides its own default internal DNS. It provides an internal DNS zone that always exists, supports automatic registration, requires no manual record creation, and is created when the VNet is created. And it's a free service. Azure provided name resolution provides only basic authoritative DNS capabilities. If you use this option, the DNS zone names and records will be automatically managed by Azure, and you will not be able to control the DNS zone names or the life cycle of DNS records.

Internal DNS defines a namespace as follows: .internal.cloudapp.net.

Any VM created in the VNet is registered in the internal DNS zone and gets a DNS domain name like myVM.internal.cloudapp.net. It's important to recognize that it's the Azure Resource name that is registered, not the name of the guest OS on the VM.

Limitations of Internal DNS

- Can't resolve across different VNets.

- Registers resource names, not guest OS names.
- Does not allow manual record creation.

Azure Private DNS Zones

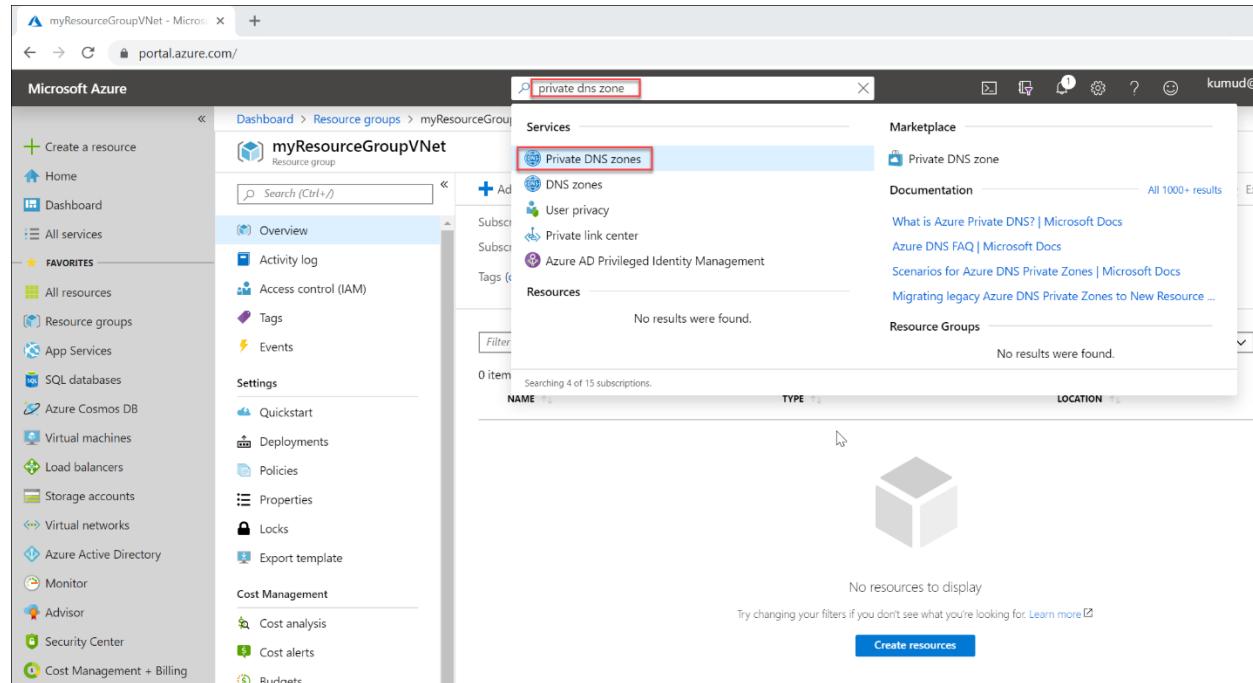
Private DNS zones in Azure are available to internal resources only. They are global in scope, so you can access them from any region, any subscription, any VNet, and any tenant. If you have permission to read the zone, you can use it for name resolution. Private DNS zones are highly resilient, being replicated to regions all throughout the world. They are not available to resources on the internet.

For scenarios which require more flexibility than Internal DNS allows, you can create your own private DNS zones. These zones enable you to:

- Configure a specific DNS name for a zone.
- Create records manually when necessary.
- Resolve names and IP addresses across different zones.
- Resolve names and IP addresses across different VNets.

Create a private DNS zone by using the portal

You can create a private DNS zone using the Azure portal, Azure PowerShell, or Azure CLI.



The screenshot shows the Microsoft Azure portal interface. The user has searched for 'private dns zone' in the search bar at the top right. The search results are displayed under the 'Services' section. The 'Private DNS zones' option is highlighted with a red box. Other options listed include 'DNS zones', 'User privacy', 'Private link center', and 'Azure AD Privileged Identity Management'. Below the services, there is a 'Resources' section with a message stating 'No results were found.' At the bottom of the page, there is a table header for 'NAME', 'TYPE', and 'LOCATION', followed by a note 'No resources to display' and a 'Create resources' button.

When the new DNS zone is deployed, you can manually create resource records, or use auto-registration, which will create resource records based on the Azure resource name.

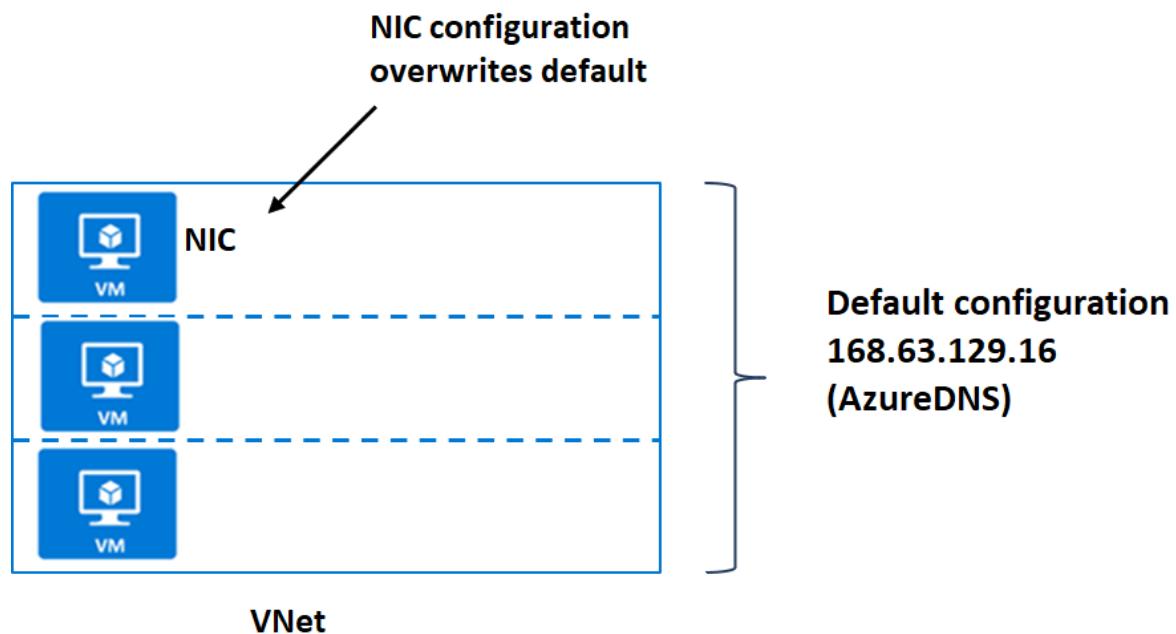
Private DNS zones support the full range of records including pointers, MX, SOA, service, and text records.

Link VNets to private DNS zones

In Azure, a VNet represents a group of 1 or more subnets, as defined by a CIDR range. Resources such as VMs are added to subnets.

At the VNet level, default DNS configuration is part of the DHCP assignments made by Azure, specifying the special address 168.63.129.16 to use Azure DNS services.

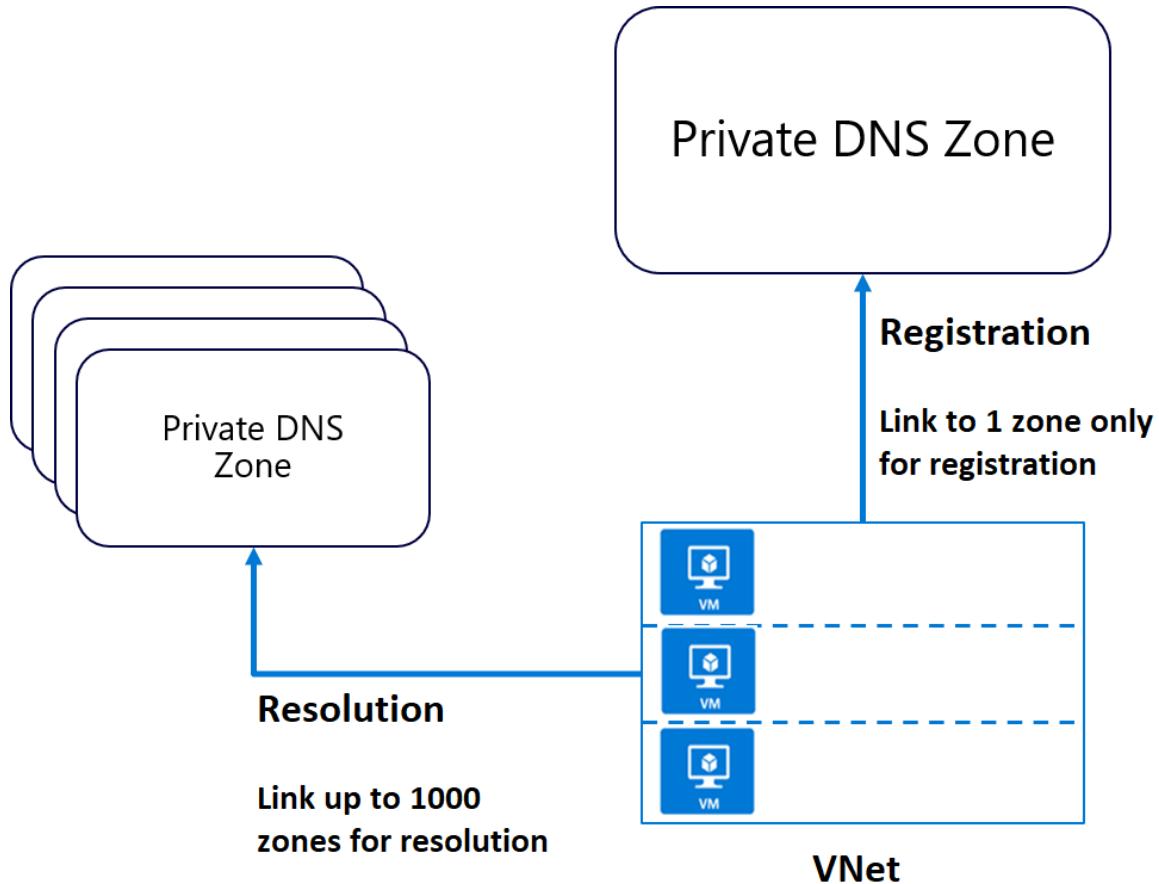
If necessary, you can override the default configuration by configuring an alternate DNS server at the VM NIC.



Two ways to link VNets to a private zone:

- **Registration:** Each VNet can link to one private DNS zone for registration. However, up to 100 VNets can link to the same private DNS zone for registration.

- **Resolution:** There may be many other private DNS zones for different namespaces. You can link a VNet to each of those zones for name resolution. Each VNet can link to up to 1000 private DNS Zones for name resolution.



Integrating on-premises DNS with Azure VNets

If you have an external DNS server, for example an on-premises server, you can use custom DNS configuration on your VNet to integrate the two.

Your external DNS can run on any DNS server: BIND on UNIX, Active Directory Domain Services DNS, and so on. If you want to use an external DNS server and not the default Azure DNS service, you must configure the desired DNS servers.

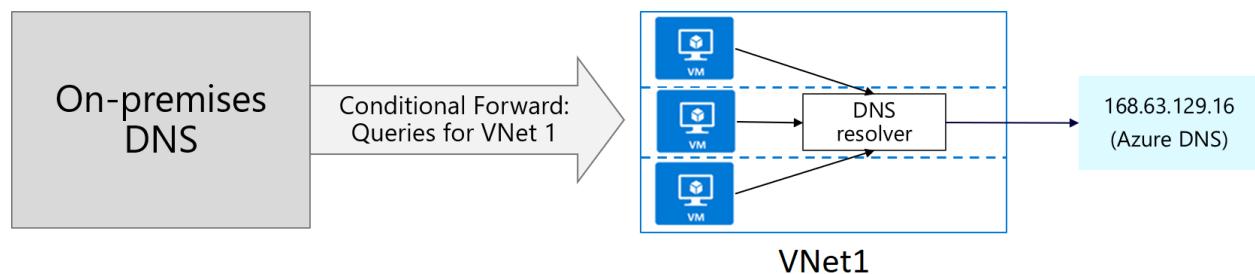
Organizations often use an internal Azure private DNS zone for auto registration, and then use a custom configuration to forward queries external zones from an external DNS server.

Forwarding takes two forms:

- Forwarding - specifies another DNS server (SOA for a zone) to resolve the query if the initial server cannot.
- Conditional forwarding - specifies a DNS server for a named zone, so that all queries for that zone are routed to the specified DNS server.

Note

If the DNS server is outside Azure, it doesn't have access to Azure DNS on 168.63.129.16. In this scenario, setup a DNS resolver inside your VNet, forward queries for to it, and then have it forward queries to 168.63.129.16 (Azure DNS). Essentially, you're using forwarding because 168.63.129.16 is not routable, and therefore not accessible to external clients.



Exercise: Configure domain name servers settings in Azure

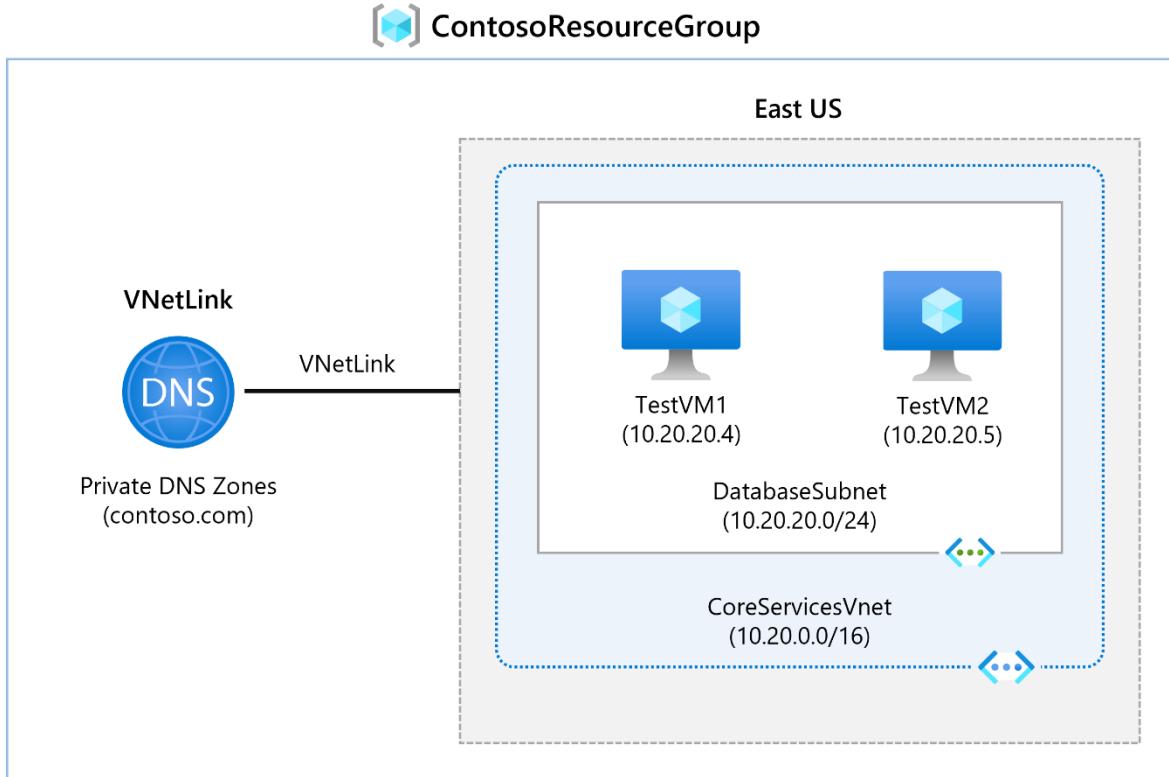
Completed 100 XP

- 7 minutes

Lab scenario

In this lab, you'll configure DNS name resolution for Contoso Ltd. You'll create a private DNS zone named contoso.com, link the VNets for registration and resolution, and then create two virtual machines and test the configuration.

Architecture diagram



Objectives

- **Task 1:** Create a private DNS Zone
- **Task 2:** Link subnet for auto registration
- **Task 3:** Create Virtual Machines to test the configuration
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 4:** Verify records are present in the DNS zone

Note

Select the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

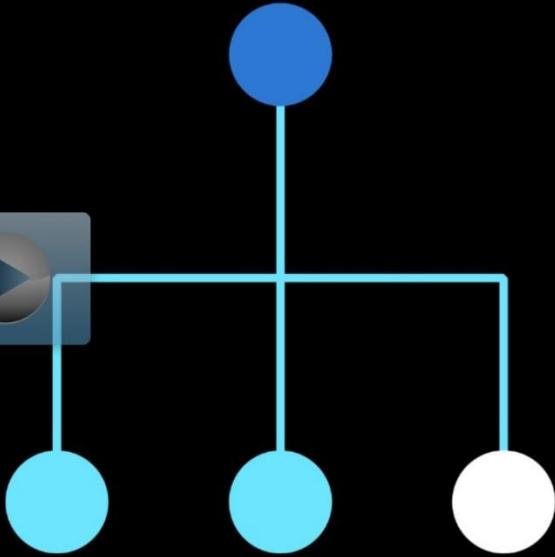
Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.

Interactive Lab Simulation:

Configure DNS settings in Azure

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Configure%20DNS%20settings%20in%20Azure>

Enable cross-virtual network connectivity with peering

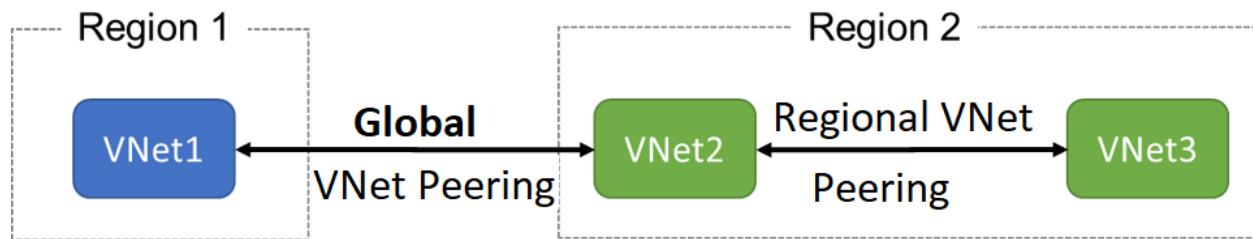
200 XP

- 6 minutes

Organizations with large scale operations will often need to create connections between different parts of their virtual network infrastructure. Virtual network peering enables you to seamlessly connect separate VNets with optimal network performance, whether they are in the same Azure region (VNet peering) or in different regions (Global VNet peering). Network traffic between peered virtual networks is private. The virtual networks appear as one for connectivity purposes. The traffic between virtual machines in peered virtual networks uses the Microsoft backbone infrastructure, and no public Internet, gateways, or encryption is required in the communication between the virtual networks.

Virtual network peering enables you to seamlessly connect two Azure virtual networks. Once peered, the virtual networks appear as one, for connectivity purposes. There are two types of VNet peering.

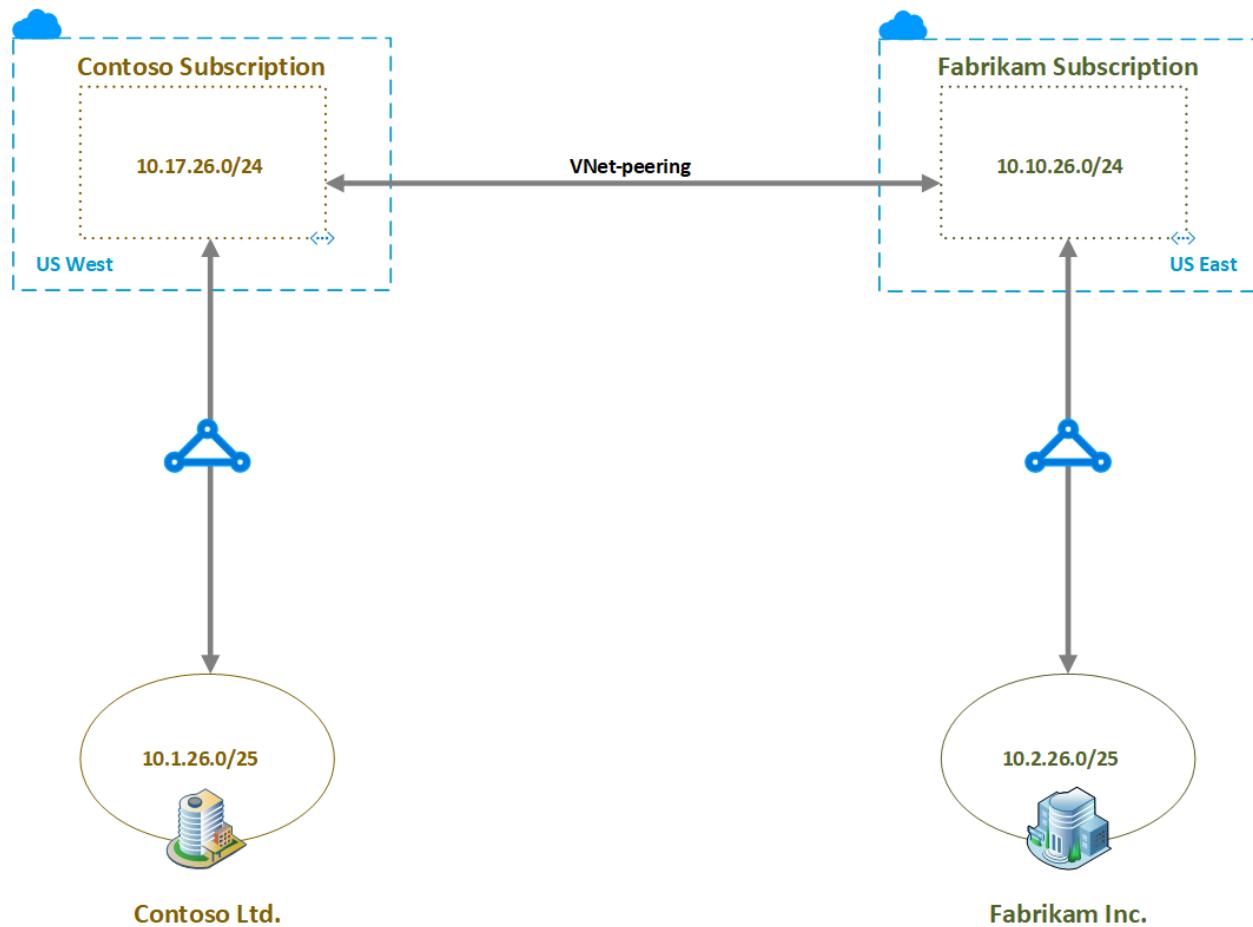
- **Regional VNet peering** connects Azure virtual networks in the same region.
- **Global VNet peering** connects Azure virtual networks in different regions. When creating a global peering, the peered virtual networks can exist in any Azure public cloud region or China cloud regions, but not in Government cloud regions. You can only peer virtual networks in the same region in Azure Government cloud regions.



The benefits of using virtual network peering, whether local or global, include:

- A low-latency, high-bandwidth connection between resources in different virtual networks.
- The ability to apply network security groups in either virtual network to block access to other virtual networks or subnets.
- The ability to transfer data between virtual networks across Azure subscriptions, Microsoft Entra tenants, deployment models, and Azure regions.
- The ability to peer virtual networks created through the Azure Resource Manager.
- The ability to peer a virtual network created through Resource Manager to one created through the classic deployment model.
- No downtime to resources in either virtual network is required when creating the peering, or after the peering is created.

The following diagram shows a scenario where resources on the Contoso VNet and resources on the Fabrikam VNet need to communicate. The Contoso subscription in the US West region, is connected to the Fabrikam subscription in the US East region.



The routing tables show the routes known to the resources in each subscription. The following routing table shows the routes known to Contoso, with the final entry being the Global VNet peering entry to the Fabrikam 10.10.26.0/24 subnet.

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

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

Scope						
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 routing table shows the routes known to Fabrikam. Again, the final entry is the Global VNet peering entry, this time to the Contoso 10.17.26.0/24 subnet.

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	-

Configure VNet Peering

Here are the steps to configure VNet peering. Notice you will need two virtual networks. To test the peering, you will need a virtual machine in each network. Initially, the VMs will not be able to communicate, but after configuration the communication will work. The step that is new is configuring the peering of the virtual networks.

1. Create two virtual networks.
2. **Peer the virtual networks.**
3. Create virtual machines in each virtual network.
4. Test the communication between the virtual machines.

To configure the peering use the **Add peering** page. There are only a few optional configuration parameters to consider.

This virtual network

Peering link name *

Traffic to remote virtual network ⓘ

- Allow (default)
 Block all traffic to the remote virtual network

Traffic forwarded from remote virtual network ⓘ

- Allow (default)
 Block traffic that originates from outside this virtual network

Virtual network gateway or Route Server ⓘ

- Use this virtual network's gateway or Route Server
 Use the remote virtual network's gateway or Route Server
 None (default)

Remote virtual network

Peering link name *

Note

When you add a peering on one virtual network, the second virtual network configuration is automatically added.

Gateway Transit and Connectivity

When virtual networks are peered, you configure a VPN gateway in the peered virtual network as a transit point. In this case, a peered virtual network uses the remote gateway to gain access to other resources. A virtual network can have only one gateway. Gateway transit is supported for both VNet Peering and Global VNet Peering.

When you Allow Gateway Transit the virtual network can communicate to resources outside the peering. For example, the subnet gateway could:

- Use a site-to-site VPN to connect to an on-premises network.
- Use a VNet-to-VNet connection to another virtual network.
- Use a point-to-site VPN to connect to a client.

In these scenarios, gateway transit allows peered virtual networks to share the gateway and get access to resources. This means you do not need to deploy a VPN gateway in the peer virtual network.

Note

Network security groups can be applied in either virtual network to block access to other virtual networks or subnets. When configuring virtual network peering, you can either open or close the network security group rules between the virtual networks.

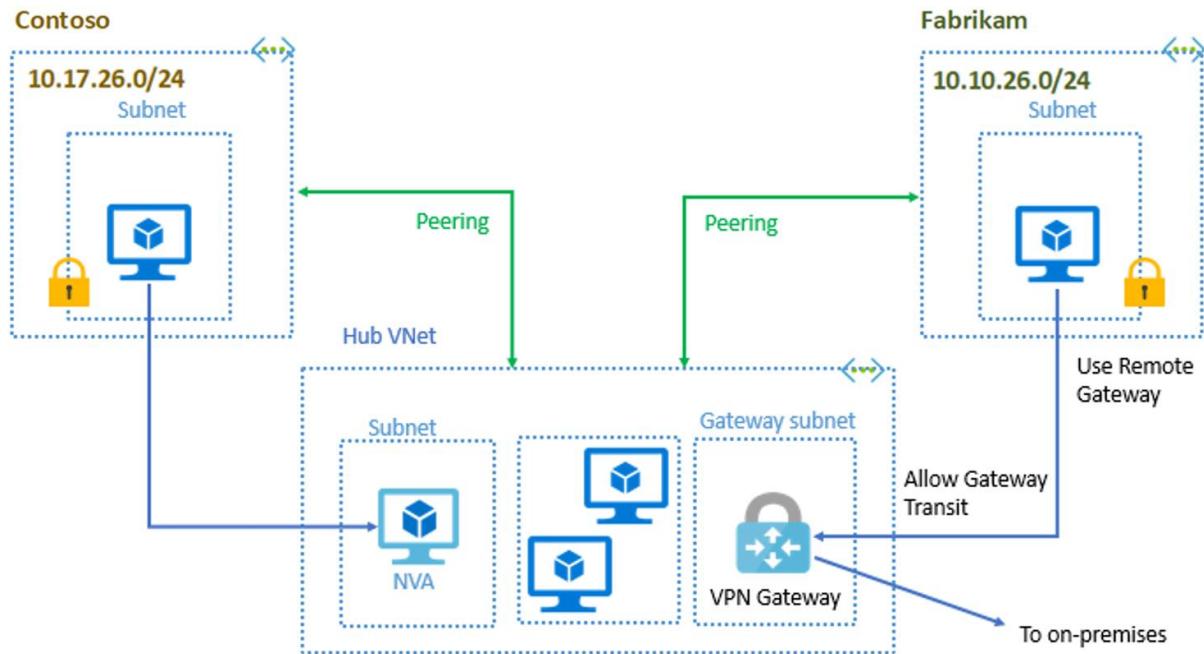
Use service chaining to direct traffic to a gateway

Suppose you want to direct traffic from the Contoso VNet to a specific network virtual appliance (NVA). Create user-defined routes to direct traffic from the Contoso VNet to the NVA in the Fabrikam VNet. This technique is known as service chaining.

To enable service chaining, add user-defined routes pointing to virtual machines in the peered virtual network as the next hop IP address. User-defined routes can also point to virtual network gateways.

Azure virtual networks can be deployed in a hub-and-spoke topology, with the hub VNet acting as a central point of connectivity to all the spoke VNets. The hub virtual network hosts infrastructure components such as an NVA, virtual machines and a VPN gateway. All the spoke virtual networks peer with the hub virtual network. Traffic flows through network virtual appliances or VPN gateways in the hub virtual network. The benefits of using a hub and spoke configuration include cost savings, overcoming subscription limits, and workload isolation.

The following diagram shows a scenario in which hub VNet hosts a VPN gateway that manages traffic to the on-premises network, enabling controlled communication between the on-premises network and the peered Azure VNets.



Exercise: Connect two Azure virtual networks using global virtual network peering

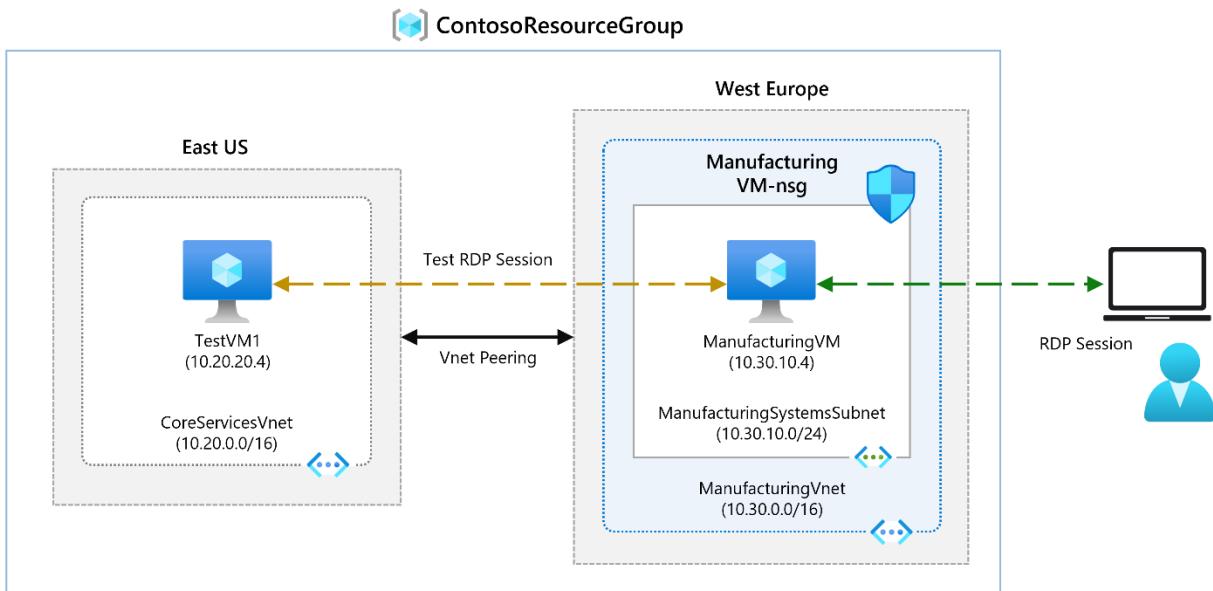
Completed 100 XP

- 5 minutes

Lab scenario

In this lab, you will configure connectivity between the CoreServicesVnet and the ManufacturingVnet by adding peerings to allow traffic flow.

Architecture diagram



Objectives

- **Task 1:** Create a Virtual Machine to test the configuration
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 2:** Connect to the **Test VMs** using RDP
- **Task 3:** Test the connection between the VMs
- **Task 4:** Create VNet peerings between **CoreServicesVnet** and **ManufacturingVnet**
- **Task 5:** Test the connection between the VMs

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

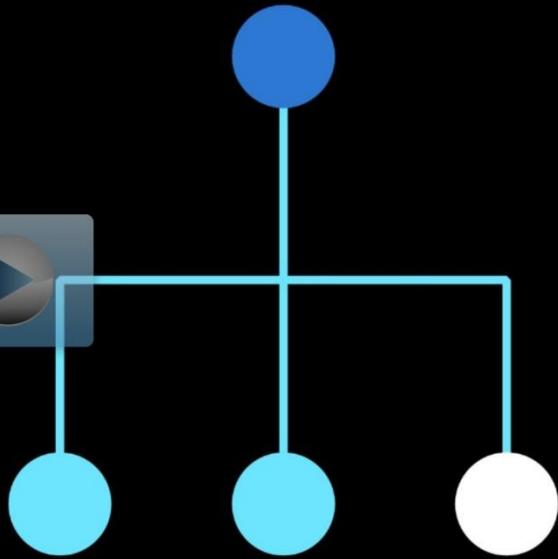
You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



Interactive Lab Simulation:

Connect two Azure virtual networks using global virtual network peering

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Connect%20two%20Azure%20virtual%20networks%20using%20global%20virtual%20network%20peering>

Implement virtual network traffic routing

200 XP

- 16 minutes

Azure automatically creates a route table for each subnet within an Azure virtual network and adds system default routes to the table. You can override some of Azure's system routes with [custom routes](#), and add additional custom routes to route tables. Azure routes outbound traffic from a subnet based on the routes in a subnet's route table.

System routes

Azure automatically creates system routes and assigns the routes to each subnet in a virtual network. You can't create or remove system routes, but you can override some

system routes with custom routes. Azure creates default system routes for each subnet, and adds additional optional default routes to specific subnets, or every subnet, when you use specific Azure capabilities.

Default routes

Each route contains an address prefix and next hop type. When traffic leaving a subnet is sent to an IP address within the address prefix of a route, the route that contains the prefix is the route Azure uses. Whenever a virtual network is created, Azure automatically creates the following default system routes for each subnet within the virtual network:

Source	Address prefixes	Next hop type
Default	Unique to the virtual network	Virtual network
Default	0.0.0.0/0	Internet
Default	10.0.0.0/8	None
Default	192.168.0.0/16	None
Default	100.64.0.0/10	None

In routing terms, a hop is a waypoint on the overall route. Therefore, the next hop is the next waypoint that the traffic is directed to on its journey to its ultimate destination. The next hop types listed in the previous table represent how Azure routes traffic destined for the address prefix listed. The next hop types are defined as follows:

- **Virtual network:** Routes traffic between address ranges within the address space of a virtual network. Azure creates a route with an address prefix that corresponds to each address range defined within the address space of a virtual network. Azure automatically routes traffic between subnets using the routes created for each address range.
- **Internet:** Routes traffic specified by the address prefix to the Internet. The system default route specifies the 0.0.0.0/0 address prefix. Azure routes traffic for any address not specified by an address range within a virtual network to the Internet, unless the destination address is for an Azure service. Azure routes any traffic destined for its service directly to the service

over the backbone network, rather than routing the traffic to the Internet. You can override Azure's default system route for the 0.0.0.0/0 address prefix with a custom route.

- **None:** Traffic routed to the None next hop type is dropped, rather than routed outside the subnet. Azure automatically creates default routes for the following address prefixes:
 - 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16: Reserved for private use in RFC 1918.
 - 100.64.0.0/10: Reserved in RFC 6598.

If you assign any of the previous address ranges within the address space of a virtual network, Azure automatically changes the next hop type for the route from None to Virtual network. If you assign an address range to the address space of a virtual network that includes, but isn't the same as, one of the four reserved address prefixes, Azure removes the route for the prefix and adds a route for the address prefix you added, with Virtual network as the next hop type.

Optional default routes

Azure adds default system routes for any Azure capabilities that you enable. Depending on the capability, Azure adds optional default routes to either specific subnets within the virtual network, or to all subnets within a virtual network. The additional system routes and next hop types that Azure may add when you enable different capabilities are:

Source	Address prefixes	Next hop type	Subnet within virtual network that route is added to
Default	Unique to the virtual network, for example: 10.1.0.0/16	VNet peering	All
Virtual network gateway	Prefixes advertised from on-premises via BGP, or configured in the local network gateway	Virtual network gateway	All
Default	Multiple	VirtualNetworkServiceEndpoint	Only the subnet a service endpoint is enabled for

Virtual network (VNet) peering: When you create a virtual network peering between two virtual networks, a route is added for each address range within the address space of each virtual network.

- **Virtual network gateway:** When you add a virtual network gateway to a virtual network, Azure adds one or more routes with Virtual network gateway as the next hop type. The source is listed as virtual network gateway because the gateway adds the routes to the subnet.
 - If your on-premises network gateway exchanges border gateway protocol ([BGP](#)) routes with an Azure virtual network gateway, a route is added for each route propagated from the on-premises network gateway. There are limits to the number of routes you can propagate to an Azure virtual network gateway, so you should summarize on-premises routes to the largest address ranges possible. For more information on the number of routes you can propagate, see [Networking limits](#).

VirtualNetworkServiceEndpoint: Azure adds the public IP addresses for certain services to the route table when you enable a service endpoint to the service. Service endpoints are enabled for individual subnets within a virtual network, so the route is only added to the route table of a subnet a service endpoint is enabled for. The public IP addresses of Azure services change periodically, and Azure manages the updates to the routing tables when necessary.

The **VNet peering** and **VirtualNetworkServiceEndpoint** next hop types are only added to route tables of subnets within virtual networks created through the Azure Resource Manager deployment model. The next hop types are not added to route tables that are associated to virtual network subnets created through the classic deployment model.

Custom routes

To control the way network traffic is routed more precisely, you can override the default routes that Azure creates by using your own user-defined routes (UDR). This technique can be useful when you want to ensure that traffic between two subnets passes through a firewall appliance, or if you want to ensure that no traffic from a VNet could be routed to the internet.

User-defined routes

You can create custom, or user-defined(static), routes in Azure to override Azure's default system routes, or to add additional routes to a subnet's route table.

In Azure, each subnet can have zero or one associated route table. When you create a route table and associate it to a subnet, the routes within it are combined with, or override, the default routes Azure adds to a subnet.

You can specify the following next hop types when creating a user-defined route:

Virtual appliance: A virtual appliance is a virtual machine that typically runs a network application, such as a firewall. When you create a route with the virtual appliance hop type, you also specify a next hop IP address. The IP address can be:

- The private IP address of a network interface attached to a virtual machine.
- The private IP address of an Azure internal load balancer.

Virtual network gateway: Specify when you want traffic destined for specific address prefixes routed to a virtual network gateway. The virtual network gateway must be created with type **VPN**.

None: Specify when you want to drop traffic to an address prefix, rather than forwarding the traffic to a destination.

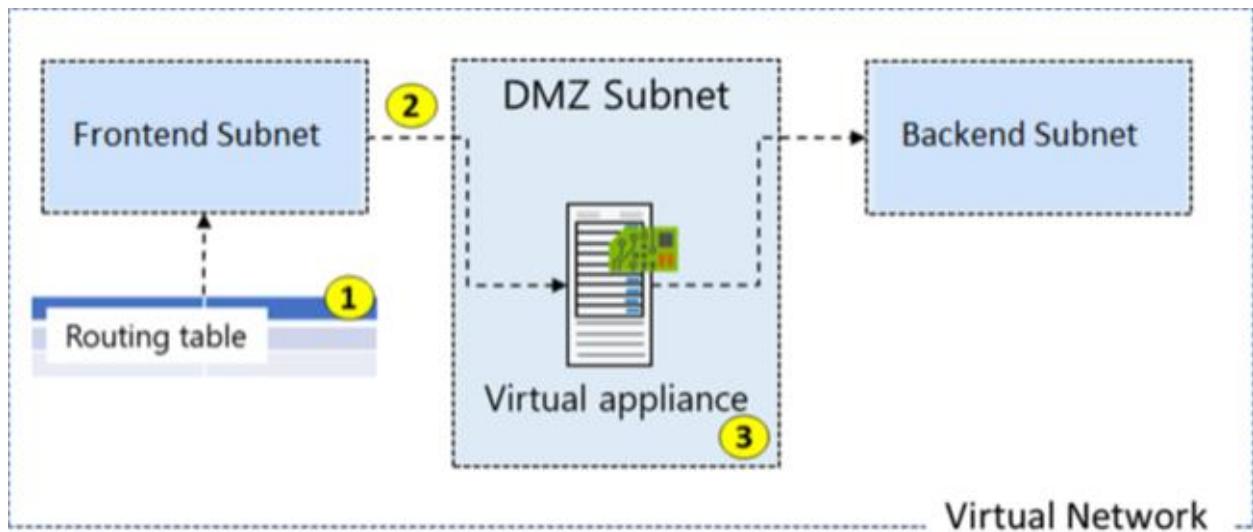
Virtual network: Specify when you want to override the default routing within a virtual network.

Internet: Specify when you want to explicitly route traffic destined to an address prefix to the Internet, or if you want traffic destined for Azure services with public IP addresses kept within the Azure backbone network.

Configure User-defined routes

Here is an example where you have a virtual network that includes three subnets.

- The subnets are Frontend, DMZ, and Backend. In the DMZ subnet, there is a network virtual appliance (NVA). NVAs are VMs that help with network functions like routing and firewall optimization.
- You want to ensure all traffic from the Frontend subnet goes through the NVA to the Backend subnet.



Create a Routing Table

Creating a routing table is straightforward. You provide **Name**, **Subscription**, **Resource Group**, and **Location**. You also decide to use **Virtual network gateway route propagation**.

Create Route table

X

Subscription * ⓘ

Visual Studio Enterprise

Resource group * ⓘ

(New) myRGWest

[Create new](#)

Instance details

Region * ⓘ

West US

Name * ⓘ

myRouteTablePublic

Propagate gateway routes * ⓘ

Yes

No

Routes are automatically added to the route table for all subnets with Virtual network gateway propagation enabled. When you are using ExpressRoute, propagation ensures all subnets get the routing information.

Create a Custom Route

For our example,

- The new route is named *ToPrivateSubnet*.
- The Private subnet is at 10.0.1.0/24.
- The route uses a virtual appliance. Notice the other choices for *Next hop type*: virtual network gateway, virtual network, internet, and none.
- The virtual appliance is located at 10.0.2.4.

Add route

myRouteTablePublic

Route name *

ToPrivateSubnet ✓

Address prefix * ⓘ

10.0.1.0/24 ✓

Next hop type ⓘ

Virtual network gateway ^

Virtual network gateway

Virtual network

Internet

Virtual appliance

None

In summary, this route applies to any address prefixes in 10.0.1.0/24 (private subnet). Traffic headed to these addresses will be sent to the virtual appliance with a 10.0.2.4 address.

Associate the Route Table

The last step in our example is to associate the Public subnet with the new routing table. Each subnet can have zero or one route table associated to it.

Public

myVirtualNetwork

X

Name

Public



Subnet address range * ⓘ

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

Add IPv6 address space ⓘ

NAT gateway ⓘ

None



Network security group

None



Route table

myRouteTablePublic



SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected



SUBNET DELEGATION

Delegate subnet to a service ⓘ

None



NETWORK POLICY FOR PRIVATE ENDPOINTS

The network policy affects all private endpoints in this subnet. To use network security groups, application security groups, or user defined routes to control traffic going to a private endpoint, set the private endpoint network policy to enabled. [Learn more](#)

Private endpoint network policy

Disabled



Save

Cancel

Note

By default, using system routes traffic would go directly to the private subnet. However, with a user-defined route you can force the traffic through the virtual appliance.

Note

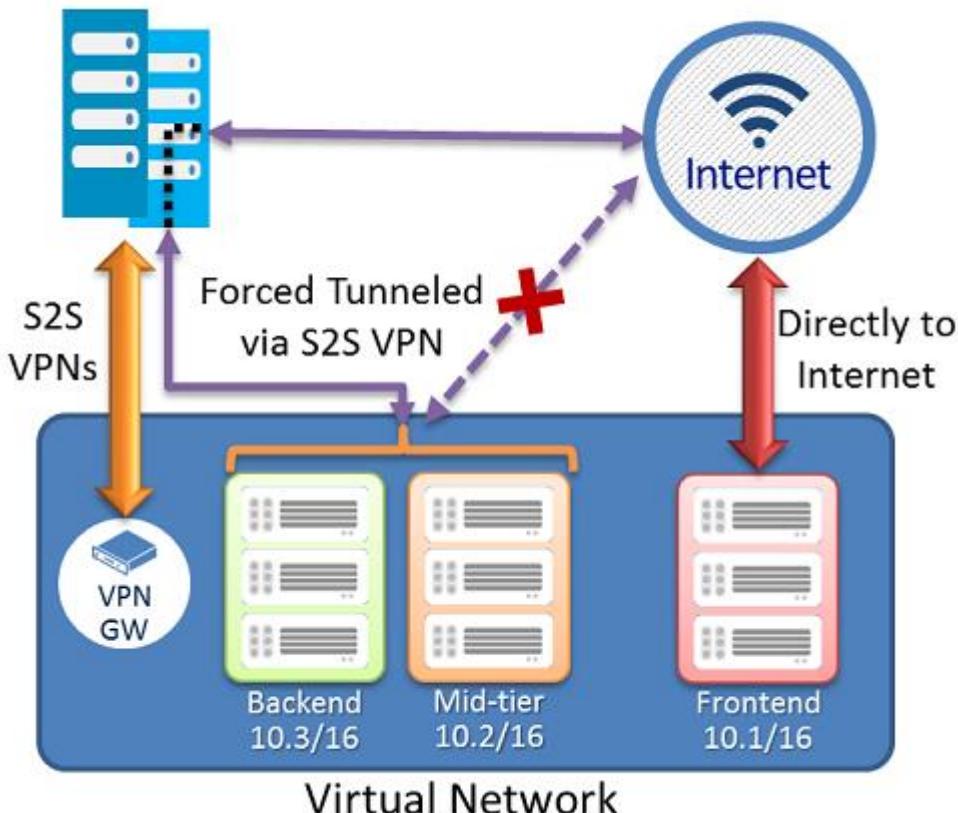
In this example, the virtual appliance shouldn't have a public IP address and IP forwarding should be enabled.

Secure a VNet by using forced tunneling

Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. This is a critical security requirement for most enterprise IT policies. If you don't configure forced tunneling, Internet-bound traffic from your VMs in Azure always traverses from the Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic. Unauthorized Internet access can potentially lead to information disclosure or other types of security breaches. Forced tunneling can be configured by using Azure PowerShell. It can't be configured using the Azure portal.

In the following example, the Frontend subnet is not force tunneled. The workloads in the Frontend subnet can continue to accept and respond to customer requests from the Internet directly. The Mid-tier and Backend subnets are forced tunneled. Any outbound connections from these two subnets to the Internet will be forced or redirected back to an on-premises site via one of the Site-to-site (S2S) VPN tunnels.

On-premises



Configure forced tunneling

Forced tunneling in Azure is configured using virtual network custom user-defined routes.

- Each virtual network subnet has a built-in, system routing table. The system routing table has the following three groups of routes:
 - Local VNet routes: Route directly to the destination VMs in the same virtual network.
 - On-premises routes: Route to the Azure VPN gateway.
 - Default route: Route directly to the Internet. Packets destined to the private IP addresses not covered by the previous two routes are dropped.
- To configure forced tunneling, you must:
 - Create a routing table.
 - Add a user-defined default route to the VPN Gateway.
 - Associate the routing table to the appropriate VNet subnet(s).

- Forced tunneling must be associated with a VNet that has a route-based VPN gateway.
 - You must set a default site connection among the cross-premises local sites connected to the virtual network.
 - The on-premises VPN device must be configured using 0.0.0.0/0 as traffic selectors.

Using forced tunneling allows you to restrict and inspect Internet access from your VMs and cloud services in Azure, while continuing to enable your multi-tier service architecture the Internet access it requires.

Configure Azure Route Server

Azure Route Server simplifies dynamic routing between your network virtual appliance (NVA) and your virtual network. It allows you to exchange routing information directly through Border Gateway Protocol (BGP) routing protocol between any NVA that supports the BGP routing protocol and the Azure Software Defined Network (SDN) in the Azure Virtual Network (VNET) without the need to manually configure or maintain route tables. Azure Route Server is a fully managed service and is configured with high availability.

Azure Route Server simplifies configuration, management, and deployment of your NVA in your virtual network.

- You no longer need to manually update the routing table on your NVA whenever your virtual network addresses are updated.
- You no longer need to update User-Defined Routes manually whenever your NVA announces new routes or withdraw old ones.
- You can peer multiple instances of your NVA with Azure Route Server. You can configure the BGP attributes in your NVA and, depending on your design (e.g., active-active for performance or active-passive for resiliency), let Azure Route Server know which NVA instance is active or which one is passive.
- The interface between NVA and Azure Route Server is based on a common standard protocol. As long as your NVA supports BGP, you can peer it with Azure Route Server.
- You can deploy Azure Route Server in any of your new or existing virtual network.
- Learn how to deploy [Azure Route Server](#)

Diagnose a routing problem

Imagine your attempts to connect to a specific virtual machine (VM) in your Azure virtual network fail persistently. You can diagnose a routing problem by viewing the routes that are effective for a network interface in a VM. The effective routes for all network interfaces in a subnet are the combination of routes you create, Azure's default routes, and any routes propagated from your on-premises network through an Azure VPN gateway via the border gateway protocol (BGP).

You can view the effective routes for each network interface by using the Azure portal, Azure PowerShell, or Azure CLI. The following steps show examples of each technique. In each case, output is only returned if the VM is in the running state. If there are multiple network interfaces attached to the VM, you can review the effective routes for each network interface. Since each network interface can be in a different subnet, each network interface can have different effective routes.

View effective routes in Azure portal

1. Log into the Azure portal with an Azure account that has the [necessary permissions](#).
2. In the search box, enter the name of the VM that you want to investigate.
3. Select the VM from the search results.
4. Under **Settings**, select **Networking**, and navigate to the network interface resource by selecting its name.

- Under **Support + troubleshooting**, select **Effective routes**. The effective routes for a network interface named **myVMNic1** are shown, in the following image:

Source	State	Address Prefixes	Next Hop Type	Next Hop Type IP Address	User Defined Route Name
Default	Active	10.1.1.0/24	Virtual network	-	-
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	-	-

View effective routes by using Azure PowerShell

You can view the effective routes for a network interface with the `Get-AzEffectiveRouteTable` command. The following example gets the effective routes for a

network interface named myVMNic1, that is in a resource group named myResourceGroup:

PowerShellCopy

```
Get-AzEffectiveRouteTable  
-NetworkInterfaceName myVMNic1  
-ResourceGroupName myResourceGroup
```

Resolve the routing issue

Steps you might take to resolve the routing problem might include:

1. Add a custom route to override a default route. Learn how to [add a custom route](#).
2. Change or remove a custom route that causes traffic to be routed to an undesired location. Learn how to [change or delete](#) a custom route.
3. Ensure that the route table is associated to the correct subnet (the one that contains the network interface). Learn how to [associate a route table to a subnet](#).
4. Ensure that devices such as Azure VPN gateway or network virtual appliances you've deployed are operating as intended.

Configure internet access with Azure Virtual NAT

200 XP

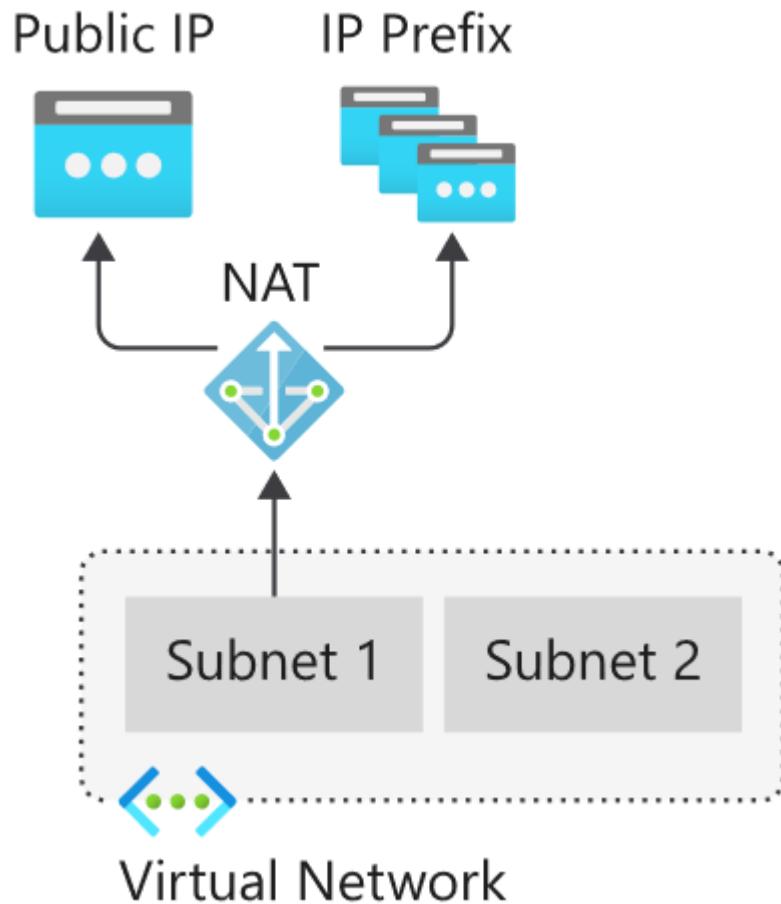
- 6 minutes

Globally, IPv4 address ranges are in very short supply, and can be an expensive way to grant access to Internet resources. Network Address Translation (NAT) arose out of this need for internal resources on a private network to share routable IPv4 addresses to gain access to external resources on a public network. Rather than purchasing an IPv4 address for each resource that requires internet access, you can use a NAT service to map outgoing requests from internal resources to an external IP address, so that communication can take place.

NAT services provide mappings for a single IP address, a range of IP addresses defined by an IP Prefix, and a range of ports associated with an IP address. NAT is compatible with standard SKU public IP address resources or public IP prefix resources or a

combination of both. You can use a public IP prefix directly or distribute the public IP addresses of the prefix across multiple NAT gateway resources. NAT will map all traffic to the range of IP addresses of the prefix. NAT allows flows to be created from the virtual network to the Internet. Return traffic from the Internet is only allowed in response to an active flow.

The following diagram shows outbound traffic flow from Subnet 1 through the NAT gateway to be mapped to a Public IP address or a Public IP prefix.



You define the NAT configuration for each subnet within a VNet to enable outbound connectivity by specifying which NAT gateway resource to use. After NAT is configured, all UDP and TCP outbound flows from any virtual machine instance will use NAT for internet connectivity. No further configuration is necessary, and you don't need to create any user-defined routes. NAT takes precedence over other outbound scenarios and replaces the default Internet destination of a subnet.

Support dynamic workloads by scaling NAT

With NAT, you don't need to do extensive pre-planning or pre-allocate addresses because NAT scales to support dynamic workloads. By using port network address translation (PNAT or PAT), NAT provides up to 64,000 concurrent flows for UDP and TCP respectively, for each attached public IP address. NAT can support up to 16 public IP addresses.

How to deploy NAT

Configuring and using NAT gateway is a straightforward process:

NAT gateway resource:

1. Create regional or zonal (zone-isolated) NAT gateway resource,
2. Assign IP addresses,
3. If necessary, modify TCP idle timeout (optional).

Virtual network:

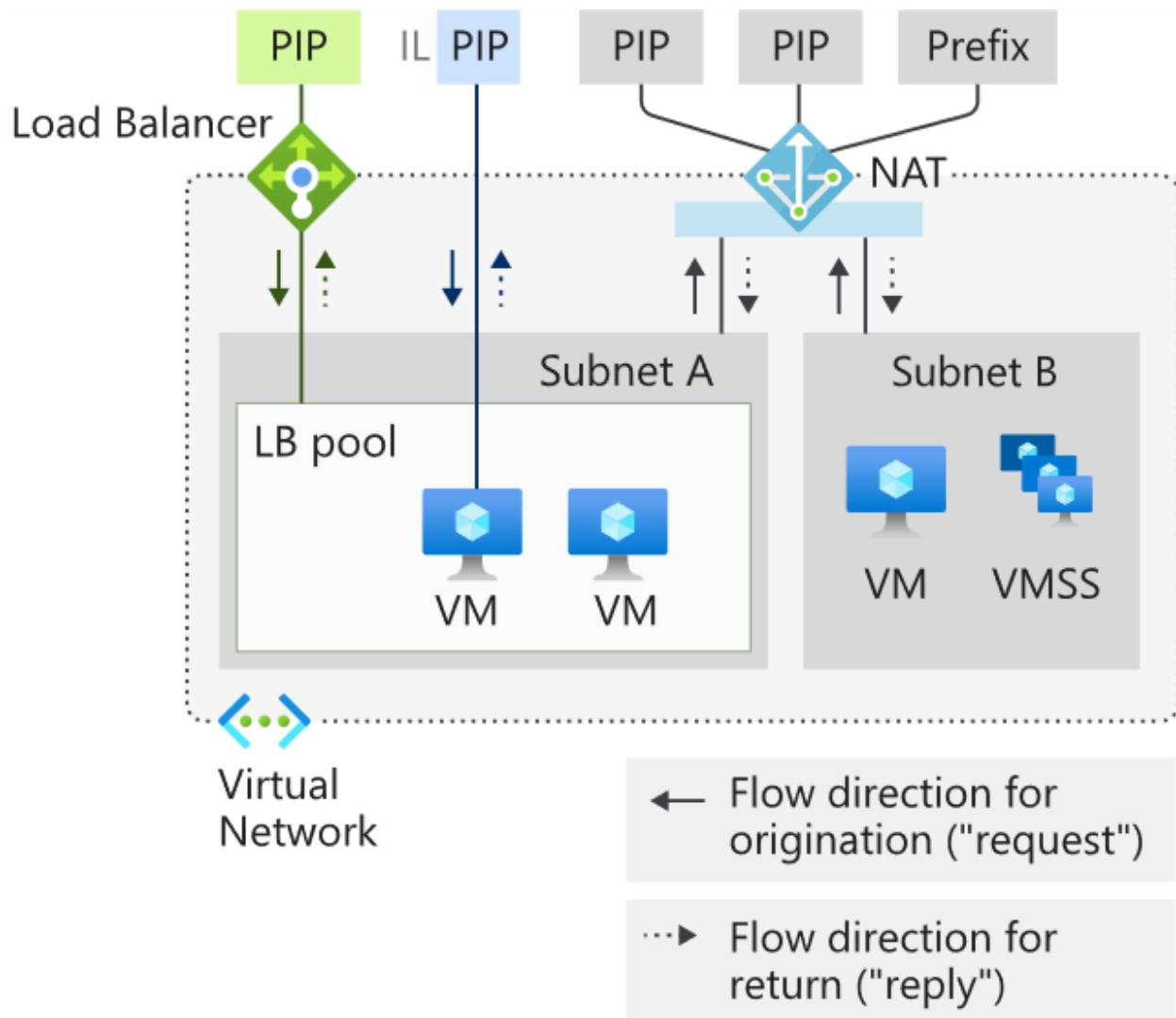
- Configure virtual network subnet to use a NAT gateway.
- User-defined routes are not necessary.

Coexistence of inbound and outbound

NAT is compatible with the following standard SKU resources:

- Load balancer
- Public IP address
- Public IP prefix

NAT and compatible Standard SKU features are aware of the direction the flow was started. Inbound and outbound scenarios can coexist. These scenarios will receive the correct network address translations because these features are aware of the flow direction. When used together with NAT, these resources provide inbound Internet connectivity to your subnet(s).



Limitations of NAT

- NAT is compatible with standard SKU public IP, public IP prefix, and load balancer resources. Basic resources (for example basic load balancer) and any products derived from them aren't compatible with NAT. Basic resources must be placed on a subnet not configured with NAT.
- IPv4 address family is supported. NAT doesn't interact with IPv6 address family. NAT can't be deployed on a subnet with an IPv6 prefix.
- NAT can't span multiple virtual networks.
- IP fragmentation isn't supported.

Summary

Completed 100 XP

- 1 minute

As your organization moves to Azure, you must design a secure virtual networking environment that provides connectivity and name resolution for both virtual and on-premises resources. Users must be able to access the resources they need smoothly and securely, regardless of where they're accessing the network from.

In this module you saw a broad overview of some of the most crucial aspects of designing and planning an Azure virtual network, including planning VNets, subnets and micro-segmentation, assigning appropriate IP addresses to resources and configuring DNS name resolution.

Design and implement hybrid networking

Design and implement Azure VPN Gateway

200 XP

- 24 minutes

A virtual private network (VPN) provides a secure encrypted connection across another network. VPNs typically are deployed to connect two or more trusted private networks to one another over an untrusted network such as the internet. Traffic is encrypted while traveling over the untrusted network to prevent a third party from eavesdropping on the network communication.

One option for connecting an on-premises network to an Azure Virtual Network is a VPN connection.

Here, we'll look at Azure VPN Gateway, which provides an endpoint for incoming connections to an Azure Virtual Network.

Azure VPN Gateways

An Azure VPN gateway is a specific type of virtual network gateway that is used to send and receive encrypted traffic between an Azure virtual network and an on-premises location over the public Internet. Azure VPN gateways can also be used to connect separate Azure virtual networks using an encrypted tunnel across the Microsoft network backbone.

Note

A virtual network gateway is composed of two or more special VMs that are deployed to a specific subnet called the gateway subnet. Virtual network gateway VMs host routing tables and run specific gateway services. These VMs that constitute the gateway are created when you create the virtual network gateway and are managed automatically by Azure and do not require administrative attention.

Creating a virtual network gateway can take some time to complete, so it's vital that you plan appropriately. When you create a virtual network gateway, the provisioning process generates the gateway VMs and deploys them to the gateway subnet. These VMs will have the settings that you configure on the gateway.

Now, let's look at the factors you need to consider for planning your VPN gateway deployment.

Plan a VPN gateway

When you're planning a VPN gateway, there are three architectures to consider:

- Point to site over the internet
- Site to site over the internet
- Site to site over a dedicated network, such as Azure ExpressRoute

Planning factors

Factors that you need to cover during your planning process include:

- Throughput - Mbps or Gbps
- Backbone - Internet or private?
- Availability of a public (static) IP address
- VPN device compatibility
- Multiple client connections or a site-to-site link?

- VPN gateway type
- Azure VPN Gateway SKU

Choose the appropriate Gateway SKU and Generation

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. Select the SKU that satisfies your requirements based on the types of workloads, throughputs, features, and SLAs. The table below shows the available SKUs and what S2S and P2S configurations they support.

VPN Gateway Generation	SKU	S2S/VNet-to-VNet Tunnels	P2S SSTP Connections	P2S IKEv2/OpenVPN Connections	Aggregate Throughput Benchmark	BGP	Zone-redundant
Generation1	Basic	Max: 10	Max: 128	Not Supported	100 Mbps	Not Supported	No
Generation1	VpnGw1	Max: 30*	Max: 128	Max: 250	650 Mbps	Supported	No
Generation1	VpnGw2	Max: 30*	Max: 128	Max: 500	1 Gbps	Supported	No
Generation1	VpnGw3	Max: 30*	Max: 128	Max: 1000	1.25 Gbps	Supported	No
Generation1	VpnGw1AZ	Max: 30*	Max: 128	Max: 250	650 Mbps	Supported	Yes
Generation1	VpnGw2AZ	Max: 30*	Max: 128	Max: 500	1 Gbps	Supported	Yes
Generation1	VpnGw3AZ	Max: 30*	Max: 128	Max: 1000	1.25 Gbps	Supported	Yes
Generation2	VpnGw2	Max: 30*	Max: 128	Max: 500	1.25 Gbps	Supported	No
Generation2	VpnGw3	Max: 30*	Max: 128	Max: 1000	2.5 Gbps	Supported	No
Generation2	VpnGw4	Max: 30*	Max: 128	Max: 3000	5 Gbps	Supported	No
Generation2	VpnGw5	Max: 30*	Max: 128	Max: 10000	10 Gbps	Supported	No
Generation2	VpnGw2AZ	Max: 30*	Max: 128	Max: 500	1.25 Gbps	Supported	Yes
Generation2	VpnGw3AZ	Max: 30*	Max: 128	Max: 1000	2.5 Gbps	Supported	Yes
Generation2	VpnGw4AZ	Max: 100*	Max: 128	Max: 5000	5 Gbps	Supported	Yes
Generation2	VpnGw5AZ	Max: 100*	Max: 128	Max: 10000	10 Gbps	Supported	Yes

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

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

VPN Gateway types

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

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

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

PolicyBased

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

- Policy based VPNs which support IKEv1 protocols can be used with Basic Gateway SKUs only.
- You can have only 1 tunnel when using a PolicyBased VPN.
- You can only use PolicyBased VPNs for S2S connections, and only for certain configurations. Most VPN Gateway configurations require a RouteBased VPN.

RouteBased

RouteBased VPNs were previously called dynamic routing gateways in the classic deployment model. RouteBased VPNs use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then

encrypt or decrypt the packets in and out of the tunnels. The policy (or traffic selector) for RouteBased VPNs are configured as any-to-any (or wild cards). The value for a RouteBased VPN type is RouteBased.

VPN Gateway configuration requirements

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

Features	PolicyBased Basic VPN Gateway	RouteBased Basic VPN Gateway	RouteBased Standard VPN Gateway	RouteBased High Performance VPN Gateway
Site-to-Site connectivity (S2S)	PolicyBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration	RouteBased VPN configuration
Point-to-Site connectivity (P2S)	Not supported	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)	Supported (Can coexist with S2S)
Authentication method	Pre-shared key	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity	Pre-shared key for S2S connectivity, Certificates for P2S connectivity
Maximum number of S2S connections	1	10	10	30
Maximum number of P2S connections	Not supported	128	128	128
Active routing support (BGP) (*)	Not supported	Not supported	Supported	Supported

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

Create the VPN Gateway

The VPN gateway settings that you chose are critical to creating a successful

Create virtual network gateway

Instance details

Name *

Region *

Gateway type * VPN ExpressRoute

VPN type * Route-based Policy-based

SKU *

Generation

VIRTUAL NETWORK

Virtual network *

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

Enable active-active mode * Enabled Disabled

Configure BGP ASN * Enabled Disabled

connection.

- Gateway type. VPN or ExpressRoute.
- VPN Type. Route based or Policy based. Most VPN types are Route-based. The type of VPN you choose depends on the make and model of your VPN device, and the kind of VPN connection you intend to create. Typical route-based gateway scenarios include point-to-site, inter-virtual network, or multiple site-to-site connections. Route-based is also selected when you coexist with an ExpressRoute gateway or if you need to use IKEv2. Policy-based gateways support only IKEv1.
- SKU. Use the drop-down to select a gateway SKU. Your choice will affect the number of tunnels you can have and the aggregate throughput benchmark. The benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- Generation. Generation1 or Generation2. You cannot change generations or SKUs across generations. Basic and VpnGw1 SKUs are only supported in Generation1. VpnGw4 and VpnGw5 SKUs are only supported in Generation2.

- Virtual Networks. The virtual network that will be able to send and receive traffic through the virtual network gateway. A virtual network cannot be associated with more than one gateway.

Note

You can view the IP address assigned to the gateway. The gateway should appear as a connected device.

Gateway subnet

VPN Gateways require a gateway subnet. You can create a Gateway subnet before you create a VPN gateway, or you can create it during the creation of the VPN Gateway. The gateway subnet contains the IP addresses that the virtual network gateway VMs and services use. When you create your virtual network gateway, gateway VMs are deployed to the gateway subnet and configured with the required VPN gateway settings. Never deploy anything else (for example, additional VMs) to the gateway subnet. The gateway subnet must be named `GatewaySubnet` to work properly. Naming the gateway subnet `GatewaySubnet` tells Azure that this is the subnet to deploy the virtual network gateway VMs and services to.

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.

Create the Local Network Gateway

The local network gateway typically refers to the on-premises location. You give the site a name by which Azure can refer to it, then specify the IP address or FQDN of the on-premises VPN device for the connection. You also specify the IP address prefixes that will be routed through the VPN gateway to the VPN device. The address prefixes you specify are the prefixes located in the on-premises network.

Create local network gateway

Name *

VNet1LocalNet



Endpoint ⓘ

IP address FQDN

IP address * ⓘ

33.2.1.5



Address space ⓘ

192.168.3.0/24

...

Add additional address range

...

Configure BGP settings

IP Address. The public IP address of the local gateway.

Address Space. One or more IP address ranges (in CIDR notation) that define your local network's address space. If you plan to use this local network gateway in a BGP-enabled connection, then the minimum prefix you need to declare is the host address of your BGP Peer IP address on your VPN device.

Configure the on-premises VPN device

There is a validated list of standard VPN devices that work well with the VPN gateway. This list was created in partnership with device manufacturers like Cisco, Juniper, Ubiquiti, and Barracuda Networks.

When your device is not listed in the validated VPN devices table, the device may still work. Contact your device manufacturer for support and configuration instructions.

To configure your VPN device, you will need:

A shared key. The same shared key that you specify when creating the VPN connection.

The public IP address of your VPN gateway. The IP address can be new or existing.

Note

Depending on the VPN device that you have, you may be able to [download a VPN device configuration script](#).

Create the VPN Connection

Once your VPN gateways are created, you can create the connection between them. If your VNets are in the same subscription, you can use the portal.

The screenshot shows two overlapping windows from the Azure portal. The left window is titled 'Add connection' and has 'vng01' in the top right corner. It contains fields for 'Name' (Azure-to-OnPrem), 'Connection type' (Site-to-site (IPsec)), 'Virtual network gateway' (vng01), 'Local network gateway' (Azure-to-OnPrem), and 'Shared key (PSK)' (abc123). The right window is titled 'Choose local network gateway' and lists 'Create new' and 'Azure-to-OnPrem NetworkRG'.

Name. Enter a name for your connection.

Connection type. Select Site-to-Site (IPSec) from the drop-down.

Shared key (PSK). In this field, enter a shared key for your connection. You can generate or create this key yourself. In a site-to-site connection, the key you use is the same for your on-premises device and your virtual network gateway connection.

Verify the VPN Connection

After you have configured all the Site-to-Site components, it is time to verify that everything is working. You can verify the connections either in the portal, or by using PowerShell.

High availability options for VPN connections

To provide better availability for your VPN connections, there are a few options available:

- VPN Gateway redundancy (Active-standby)
- Multiple on-premises VPN devices
- Active-active Azure VPN gateway
- Combination of both

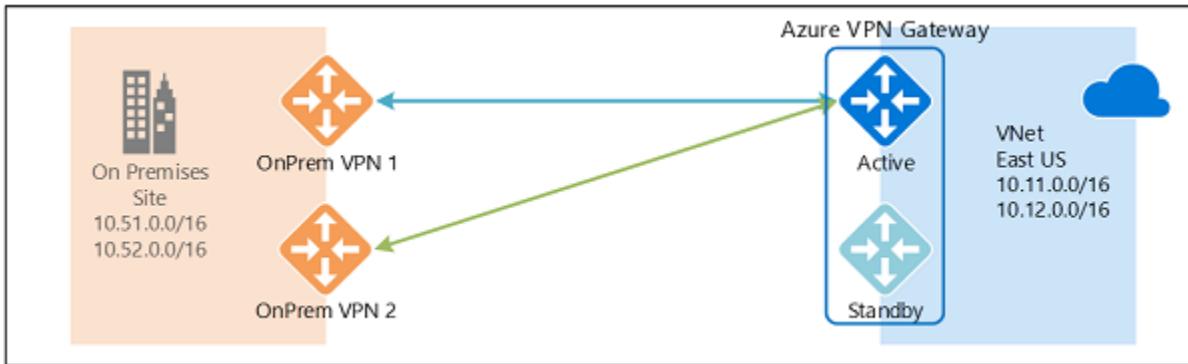
VPN Gateway redundancy

Every Azure VPN gateway consists of two instances in an active-standby configuration. For any planned maintenance or unplanned disruption that happens to the active instance, the standby instance would take over (failover) automatically and resume the S2S VPN or VNet-to-VNet connections. The switch over will cause a brief interruption. For planned maintenance, the connectivity should be restored within 10 to 15 seconds. For unplanned issues, the connection recovery will be longer, about 1 to 3 minutes in the worst case. For P2S VPN client connections to the gateway, the P2S connections will be disconnected, and the users will need to reconnect from the client machines.



Multiple on-premises VPN devices

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



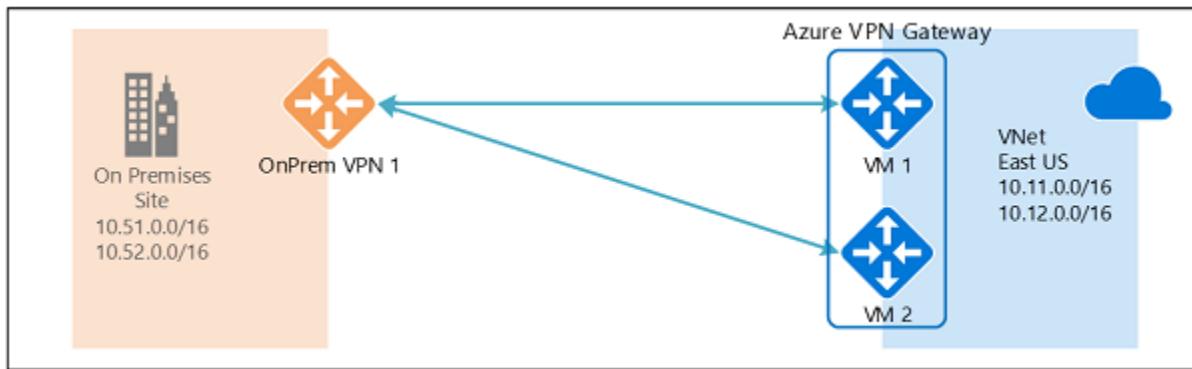
This configuration provides multiple active tunnels from the same Azure VPN gateway to your on-premises devices in the same location. There are some requirements and constraints:

1. You need to create multiple S2S VPN connections from your VPN devices to Azure. When you connect multiple VPN devices from the same on-premises network to Azure, you need to create one local network gateway for each VPN device, and one connection from your Azure VPN gateway to each local network gateway.
2. The local network gateways corresponding to your VPN devices must have unique public IP addresses in the `GatewayIpAddress` property.
3. BGP is required for this configuration. Each local network gateway representing a VPN device must have a unique BGP peer IP address specified in the `BgpPeerIpAddress` property.
4. You should use BGP to advertise the same prefixes of the same on-premises network prefixes to your Azure VPN gateway, and the traffic will be forwarded through these tunnels simultaneously.
5. You must use Equal-cost multi-path routing (ECMP).
6. Each connection is counted against the maximum number of tunnels for your Azure VPN gateway, 10 for Basic and Standard SKUs, and 30 for HighPerformance SKU.

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

Active-active VPN gateways

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



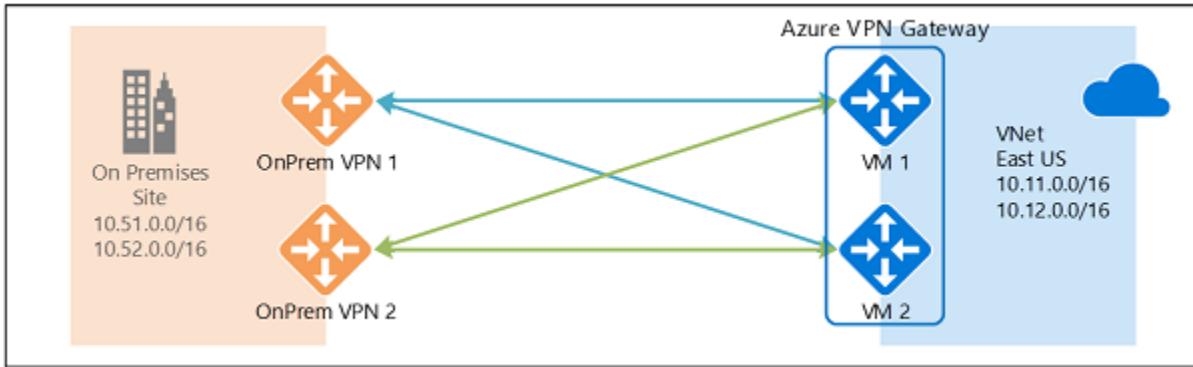
In this configuration, each Azure gateway instance will have a unique public IP address, and each will establish an IPsec/IKE S2S VPN tunnel to your on-premises VPN device specified in your local network gateway and connection. Note that both VPN tunnels are part of the same connection. You will still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

Because the Azure gateway instances are in active-active configuration, the traffic from your Azure virtual network to your on-premises network will be routed through both tunnels simultaneously, even if your on-premises VPN device may favor one tunnel over the other. For a single TCP or UDP flow, Azure attempts to use the same tunnel when sending packets to your on-premises network. However, your on-premises network could use a different tunnel to send packets to Azure.

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

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

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



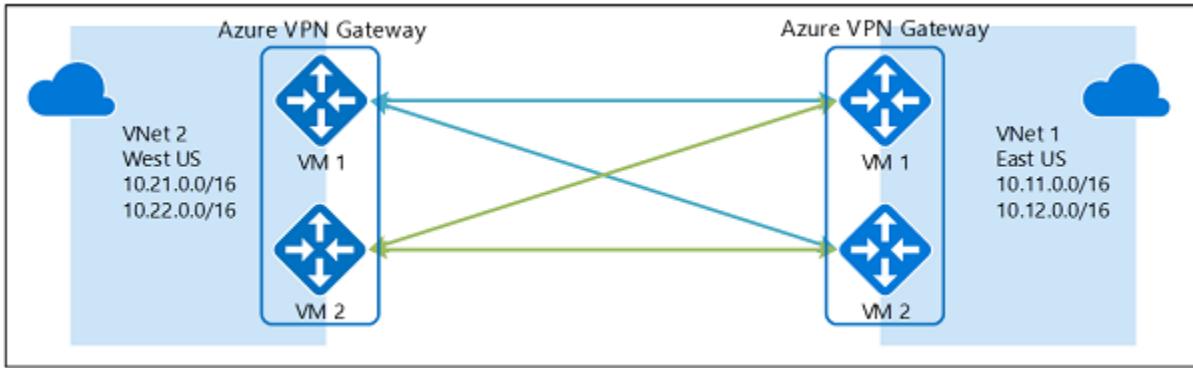
Here you create and set up the Azure VPN gateway in an active-active configuration and create two local network gateways and two connections for your two on-premises VPN devices as described above. The result is a full mesh connectivity of 4 IPsec tunnels between your Azure virtual network and your on-premises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

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

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

Highly Available VNet-to-VNet

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



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

Troubleshoot Azure VPN Gateway

VPN Gateway connections can fail for a variety of reasons. Although a network engineer will be able to troubleshoot many connectivity issues from experience, the following Microsoft documentation provides help and guidance for resolving many common problems.

Validate VPN throughput to a VNet

A VPN gateway connection enables you to establish secure, cross-premises connectivity between your Virtual Network within Azure and your on-premises IT infrastructure. This article shows how to validate network throughput from the on-premises resources to an Azure virtual machine (VM). It also provides troubleshooting guidance. See [Validate VPN throughput to a virtual network - Azure VPN Gateway](#).

Point-to-Site connections

This article lists common point-to-site connection problems that you might experience. It also discusses possible causes and solutions for these problems. See [Troubleshoot Azure point-to-site connection problems - Azure VPN Gateway](#).

Site-to-Site connections

After you configure a site-to-site VPN connection between an on-premises network and an Azure virtual network, the VPN connection suddenly stops working and cannot be

reconnected. This article provides troubleshooting steps to help you resolve this problem. See [Troubleshoot an Azure site-to-site VPN connection that cannot connect - Azure VPN Gateway](#).

VPN and Firewall device settings

This article provides several suggested solutions for third-party VPN or firewall devices that are used with VPN Gateway. Technical support for third-party VPN or firewall devices is provided by the device vendor. See [Community-suggested third-party VPN or firewall device settings for Azure VPN Gateway](#).

Troubleshoot Azure VPN Gateway using diagnostic logs

Using diagnostic logs, you can troubleshoot multiple VPN gateway related events including configuration activity, VPN Tunnel connectivity, IPsec logging, BGP route exchanges, Point to Site advanced logging.

There are several diagnostic logs you can use to help troubleshoot a problem with your VPN Gateway.

- **GatewayDiagnosticLog** - Contains diagnostic logs for gateway configuration events, primary changes, and maintenance events.
- **TunnelDiagnosticLog** - Contains tunnel state change events. Tunnel connect/disconnect events have a summarized reason for the state change if applicable.
- **RouteDiagnosticLog** - Logs changes to static routes and BGP events that occur on the gateway.
- **IKEDiagnosticLog** - Logs IKE control messages and events on the gateway.
- **P2SDiagnosticLog** - Logs point-to-site control messages and events on the gateway.

Use Azure Monitor to analyze the data collected in the diagnostic logs.

Exercise: Create and configure a virtual network gateway

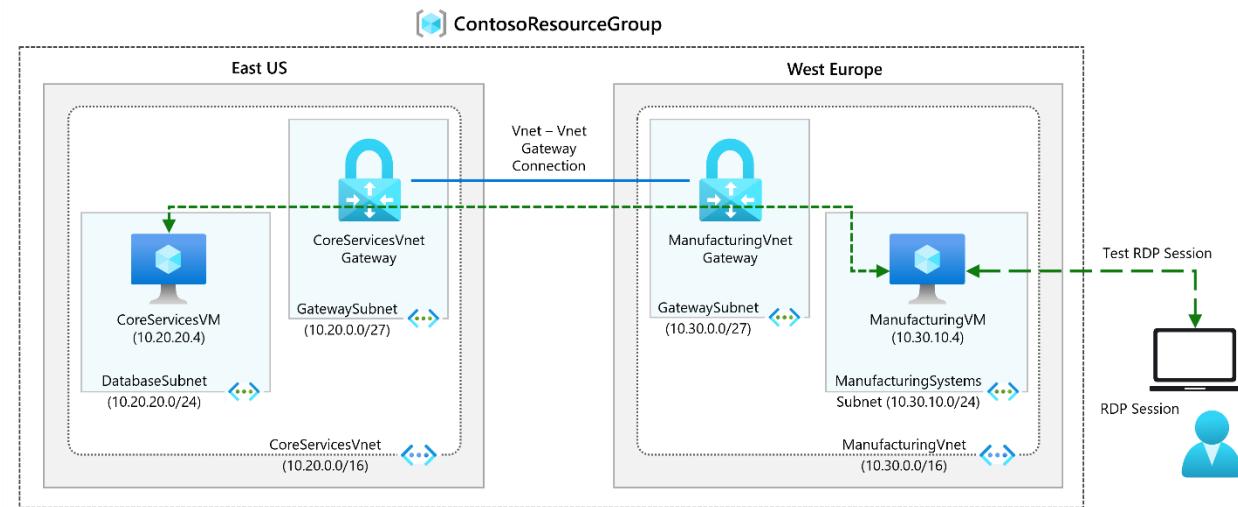
Completed 100 XP

- 10 minutes

Lab scenario

In this exercise you will configure a virtual network gateway to connect the Contoso Core Services VNet and Manufacturing VNet.

Architecture diagram



Objectives

- **Task 1:** Create CoreServicesVnet and ManufacturingVnet
 - Use a template to create the virtual networks. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 2:** Create CoreServicesTestVM
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 3:** Create ManufacturingTestVM
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 4:** Connect to the VMs using RDP
- **Task 5:** Test the connection between the VMs
- **Task 6:** Create CoreServicesVnet Gateway
- **Task 7:** Create ManufacturingVnet Gateway
- **Task 8:** Connect CoreServicesVnet to ManufacturingVnet
- **Task 9:** Connect ManufacturingVnet to CoreServicesVnet

- **Task 10:** Verify that the connections connect
- **Task 11:** Test the connection between the VMs

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



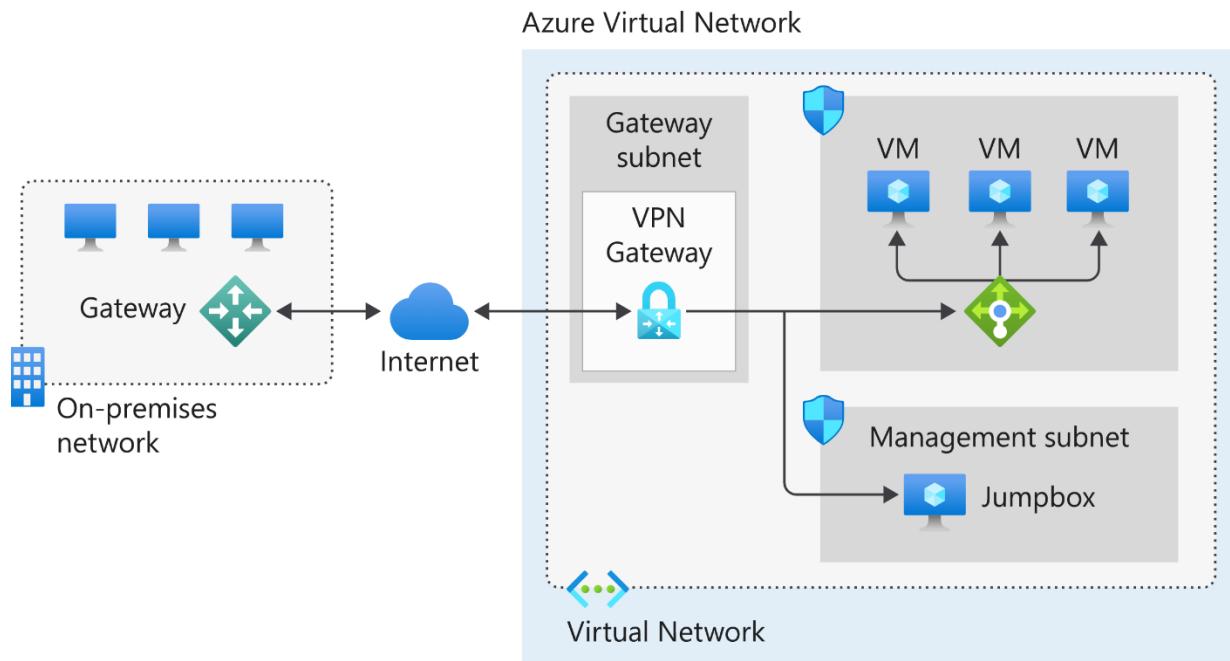
<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20and%20configure%20a%20virtual%20network%20gateway>

Connect networks with Site-to-site VPN connections

Completed 100 XP

- 3 minutes

A site-to-site (S2S) VPN gateway connection lets you create a secure connection to your virtual network from another virtual network or a physical network. The following diagram illustrates how you would connect an on-premises network to the Azure platform. The internet connection uses an IPsec VPN tunnel.



In the diagram:

- The on-premises network represents your on-premises Active Directory and any data or resources.
- The gateway is responsible for sending encrypted traffic to a virtual IP address when it uses a public connection.
- The Azure virtual network holds all your cloud applications and any Azure VPN gateway components.
- An Azure VPN gateway provides the encrypted link between the Azure virtual network and your on-premises network. An Azure VPN gateway is made up of these elements:
 - Virtual network gateway
 - Local network gateway
 - Connection
 - Gateway subnet
- Cloud applications are the ones you've made available through Azure.
- An internal load balancer, located in the front end, routes cloud traffic to the correct cloud-based application or resource.

Using this architecture offers several benefits, including:

- Configuration and maintenance are simplified.
- Having a VPN gateway helps ensure that all data and traffic are encrypted between the on-premises gateway and the Azure gateway.
- The architecture can be scaled and extended to meet your organization's networking needs.

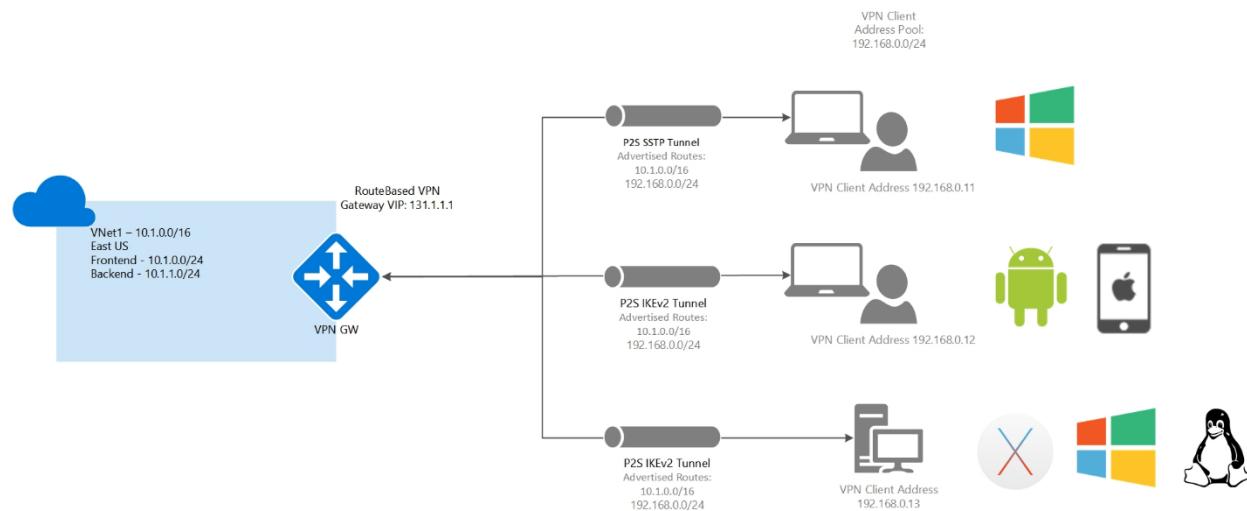
This architecture isn't applicable in all situations because it uses an existing internet connection as the link between the two gateway points. Bandwidth constraints can cause latency issues that result from reuse of the existing infrastructure.

Connect devices to networks with Point-to-site VPN connections

200 XP

- 6 minutes

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



Point-to-site protocols

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

- OpenVPN® Protocol, an SSL/TLS based VPN protocol. A TLS VPN solution can penetrate firewalls, since most firewalls open TCP port 443 outbound, which TLS uses. OpenVPN can be used to connect from Android, iOS (versions 11.0 and above), Windows, Linux, and Mac devices (macOS versions 10.13 and above).
- Secure Socket Tunneling Protocol (SSTP), a proprietary TLS-based VPN protocol. A TLS VPN solution can penetrate firewalls, since most firewalls open TCP port 443 outbound, which TLS uses. SSTP is only supported on Windows devices. Azure supports all versions of Windows that have SSTP (Windows 7 and later).
- IKEv2 VPN, a standards-based IPsec VPN solution. IKEv2 VPN can be used to connect from Mac devices (macOS versions 10.11 and above).

Point-to-site authentication methods

The user must be authenticated before Azure accepts a P2S VPN connection. There are two mechanisms that Azure offers to authenticate a connecting user.

Authenticate using native Azure certificate authentication

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

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

Authenticate using native Microsoft Entra authentication

Microsoft Entra authentication allows users to connect to Azure using their Microsoft Entra credentials. Native Microsoft Entra authentication is only supported for OpenVPN protocol and Windows 10 and requires the use of the Azure VPN Client.

With native Microsoft Entra authentication, you can leverage Microsoft Entra Conditional Access as well as multifactor authentication (MFA) features for VPN.

At a high level, you need to perform the following steps to configure Microsoft Entra authentication:

- Configure a Microsoft Entra tenant
- Enable Microsoft Entra authentication on the gateway
- Download and configure Azure VPN Client

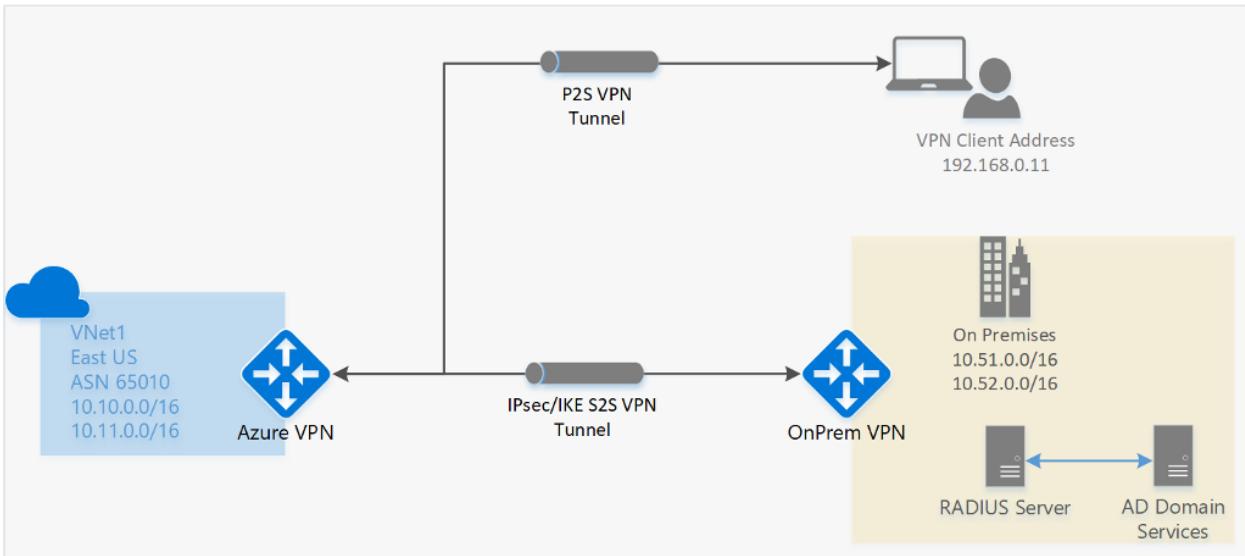
Authenticate using Active Directory (AD) Domain Server

AD Domain authentication is a popular option because it allows users to connect to Azure using their organization domain credentials. It requires a RADIUS server that integrates with the AD server. Organizations can also leverage their existing RADIUS deployment.

The RADIUS server is deployed either on-premises or in your Azure VNet. During authentication, the Azure VPN Gateway passes authentication messages back and forth between the RADIUS server and the connecting device. Thus, the Gateway must be able to communicate with the RADIUS server. If the RADIUS server is present on-premises, then a VPN S2S connection from Azure to the on-premises site is required for reachability.

The RADIUS server can also integrate with AD certificate services. This lets you use the RADIUS server and your enterprise certificate deployment for P2S certificate authentication as an alternative to the Azure certificate authentication. Integrating the RADIUS server with AD certificate services means that you can do all your certificate management in AD, you don't need to upload root certificates and revoked certificates to Azure.

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



Configure point-to-site clients

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

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



The screenshot shows the Microsoft Azure portal interface. At the top, there's a dark header bar with the Microsoft Azure logo on the left and the text "Microsoft Azure" in white. Below this, the main content area has a title "VNet1GW | Point-to-site configuration" with a blue briefcase icon to its left. Underneath the title, it says "Virtual network gateway". On the left side, there's a vertical navigation menu with icons and text for "Overview", "Activity log", "Access control (IAM)", "Tags", and "Diagnose and solve problems". In the center, there's a message "Point-to-site is not configured" with a red rectangular box around the blue "Configure now" button. At the top right, there are buttons for "Save", "Discard", and "Download VPN client". Below the central message, there's a section titled "Settings" with a horizontal line. Under "Settings", there are five items: "Configuration", "Connections" (which has a red rectangular box around it), "Point-to-site configuration" (which also has a red rectangular box around it), "Properties", and "Locks". At the bottom left, there's a section titled "Monitoring" with a horizontal line. The entire page has a clean, modern design with a white background and light gray borders for sections.

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

Note

That for Windows clients, you must have administrator rights on the client device to initiate the VPN connection from the client device to Azure.

Connect remote resources by using Azure Virtual WANs

200 XP

- 3 minutes

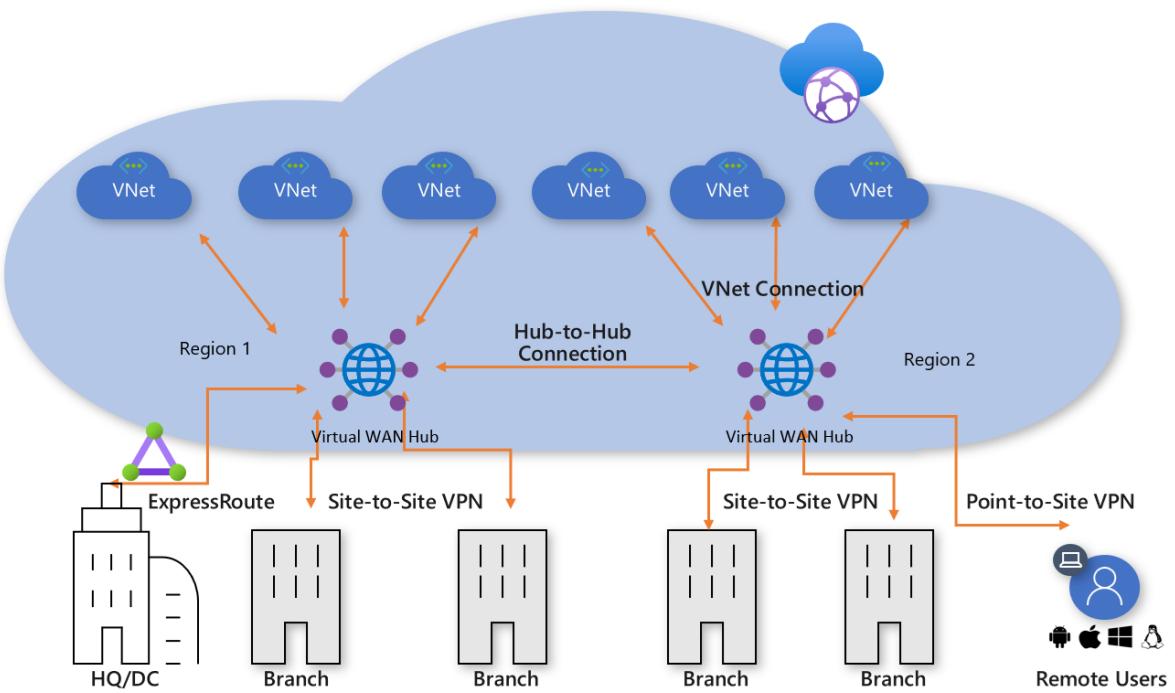
Today's workforce is more distributed than ever before. Organizations are exploring options that enable their employees, partners, and customers to connect to the resources they need from wherever they are. It's not unusual for organizations to operate across national/regional boundaries, and across time zones.

What is Azure Virtual WAN?

Azure Virtual WAN is a networking service that brings many networking, security, and routing functionalities together to provide a single operational interface. Some of the main features include:

- Branch connectivity (via connectivity automation from Virtual WAN Partner devices such as SD-WAN or VPN CPE).
- Site-to-site VPN connectivity.
- Remote user VPN connectivity (point-to-site).
- Private connectivity (ExpressRoute).
- Intra-cloud connectivity (transitive connectivity for virtual networks).
- VPN ExpressRoute inter-connectivity.
- Routing, Azure Firewall, and encryption for private connectivity.

The following diagram shows an organization with two Virtual WAN hubs connecting the spokes. VNets, Site-to-site and point-to-site VPNs, SD WANs, and ExpressRoute connectivity are all supported.



To configure an end-to-end virtual WAN, you create the following resources:

- **Virtual WAN**
- **Hub**
- **Hub virtual network connection**
- **Hub-to-hub connection**
- **Hub route table**

Choose a Virtual WAN SKU

The virtualWAN resource represents a virtual overlay of your Azure network and is a collection of multiple resources. It contains links to all your virtual hubs that you would like to have within the virtual WAN. Virtual WANs are isolated from each other and can't contain a common hub. Virtual hubs in different virtual WANs don't communicate with each other.

There are two types of Virtual WANs: Basic and Standard. The following table shows the available configurations for each type.

Virtual WAN type	Hub type	Available configurations
Basic	Basic	Site-to-site VPN only
Standard	Standard	ExpressRoute User VPN (P2S) VPN (site-to-site) Inter-hub and VNet-to-VNet transiting through the virtual hub Azure Firewall NVA in a virtual WAN

Hub private address space

A virtual hub is a Microsoft-managed virtual network. The hub contains various service endpoints to enable connectivity. From your on-premises network (`vpngatewaysite`), you can connect to a VPN gateway inside the virtual hub, connect ExpressRoute circuits to a virtual hub, or even connect mobile users to a point-to-site gateway in the virtual hub. The hub is the core of your network in a region. Multiple virtual hubs can be created in the same region.

The minimum address space is /24 to create a hub. If you use anything in the range from /25 to /32, it will produce an error during creation. You don't need to explicitly plan the subnet address space for the services in the virtual hub. Because Azure Virtual WAN is a managed service, it creates the appropriate subnets in the virtual hub for the different gateways/services (for example, VPN gateways, ExpressRoute gateways, User VPN point-to-site gateways, Firewall, routing, etc.).

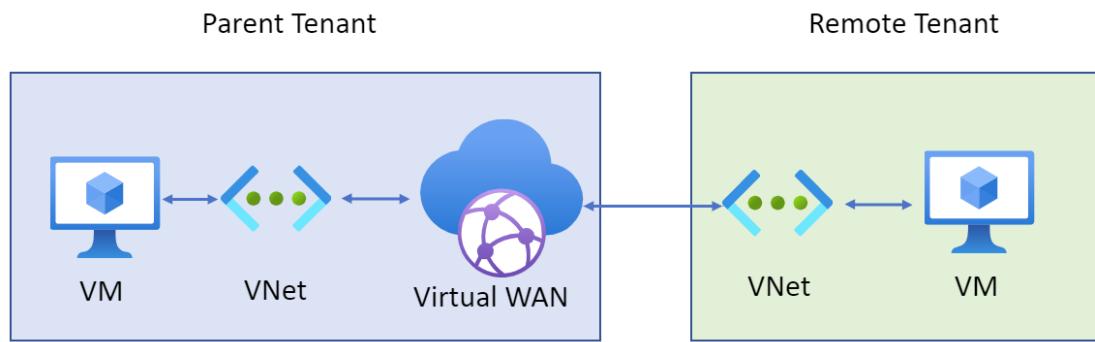
Gateway scale

A hub gateway isn't the same as a virtual network gateway that you use for ExpressRoute and VPN Gateway. For example, when using Virtual WAN, you don't create a site-to-site connection from your on-premises site directly to your VNet. Instead, you create a site-to-site connection to the hub. The traffic always goes through the hub gateway. This means that your VNets don't need their own virtual network gateway. Virtual WAN lets your VNets take advantage of scaling easily through the virtual hub and the virtual hub gateway.

Gateway scale units allow you pick the aggregate throughput of the gateway in the virtual hub. Each type of gateway scale unit (site-to-site, user-vpn, and ExpressRoute) is configured separately.

Connect cross-tenant VNets to a Virtual WAN hub

You can use Virtual WAN to connect a VNet to a virtual hub in a different tenant. This architecture is useful if you have client workloads that must be connected to be the same network but are on different tenants. For example, as shown in the following diagram, you can connect a non-Contoso VNet (the Remote Tenant) to a Contoso virtual hub (the Parent Tenant).



Before you can connect a cross-tenant VNet to a Virtual WAN hub, you must have the following configuration already set up:

- A Virtual WAN and virtual hub in the parent subscription.
- A virtual network configured in a subscription in the remote tenant.
- Non-overlapping address spaces in the remote tenant and address spaces within any other VNets already connected to the parent virtual hub.

Virtual Hub routing

The routing capabilities in a virtual hub are provided by a router that manages all routing between gateways using Border Gateway Protocol (BGP). A virtual hub can contain multiple gateways such as a Site-to-site VPN gateway, ExpressRoute gateway, Point-to-site gateway, Azure Firewall. This router also provides transit connectivity between virtual networks that connect to a virtual hub and can support up to an aggregate throughput of 50 Gbps. These routing capabilities apply to Standard Virtual WAN customers.

To learn more about how to configure routing, see [How to configure virtual hub routing](#).

Hub route table

You can create a virtual hub route and apply the route to the virtual hub route table. You can apply multiple routes to the virtual hub route table.

Exercise: create a Virtual WAN by using the Azure portal

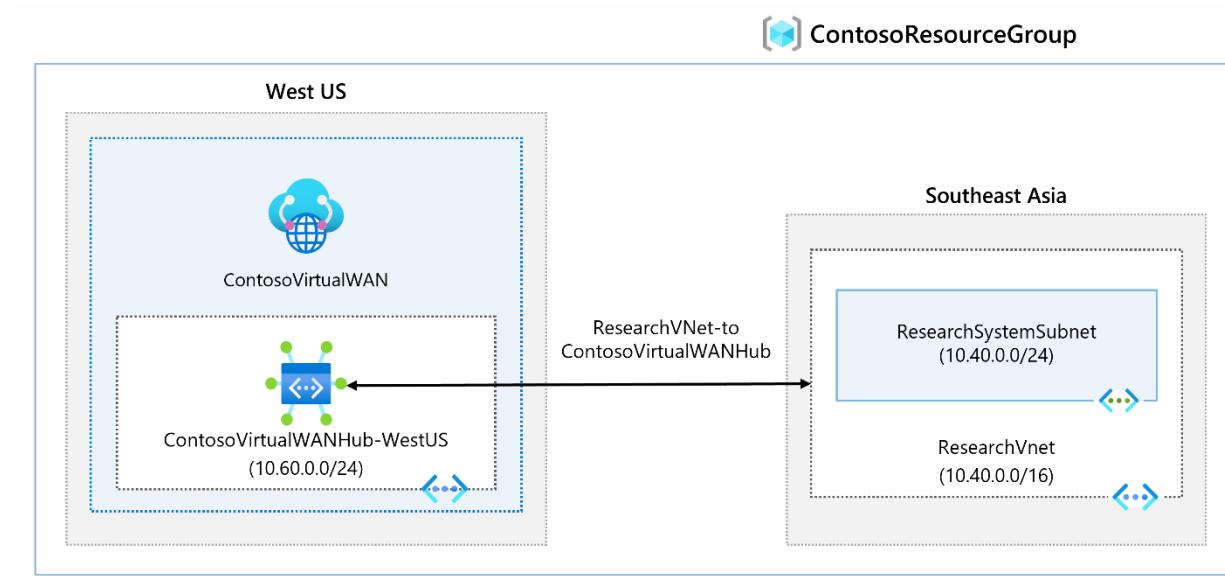
Completed 100 XP

- 8 minutes

Lab scenario

In this lab, you'll create a Virtual WAN for Contoso.

Architecture diagram



Objectives

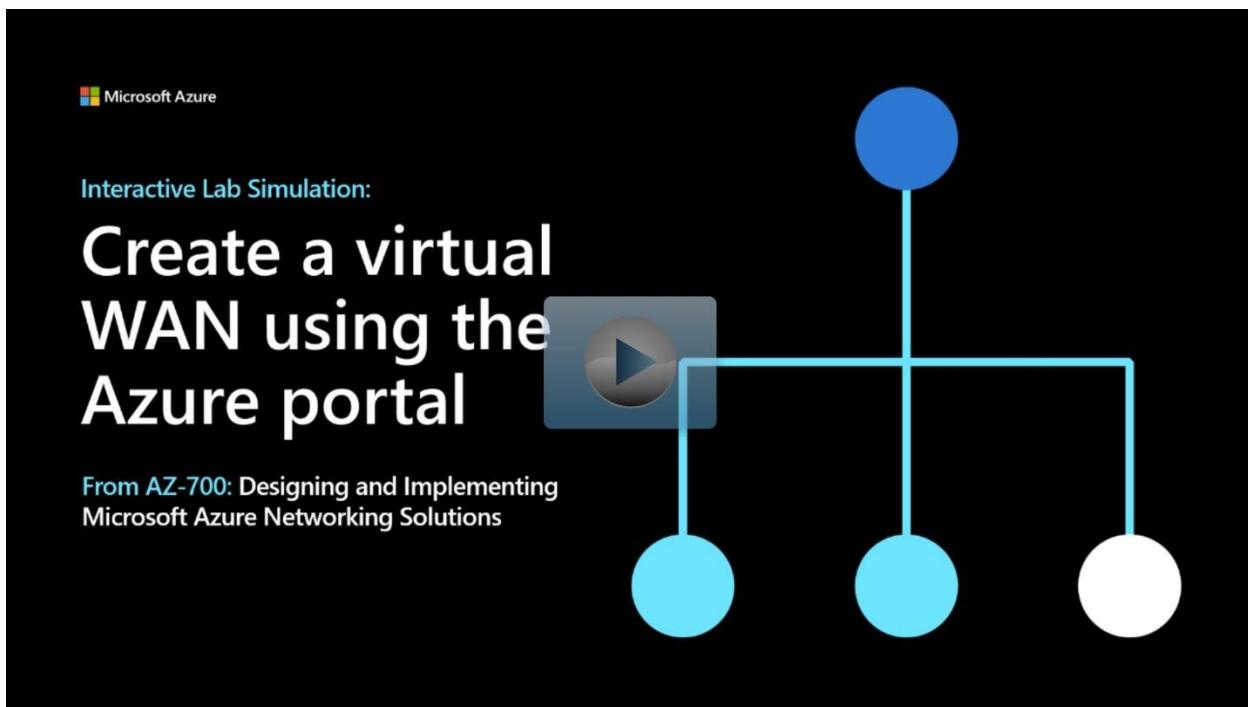
- **Task 1:** Create a Virtual WAN
- **Task 2:** Create a hub by using Azure portal
- **Task 3:** Connect a VNet to the Virtual Hub

Note

Select the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20a%20virtual%20WAN%20using%20the%20Azure%20portal>

Create a network virtual appliance (NVA) in a virtual hub

200 XP

- 7 minutes

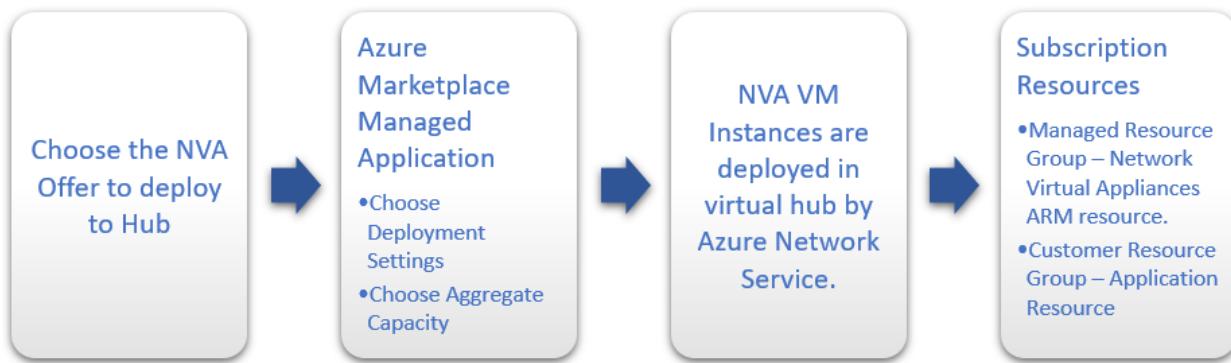
One of the benefits of Azure Virtual WAN is the ability to support reliable connections from many different technologies, whether Microsoft based, such as ExpressRoute or a

VPN Gateway, or from a networking partner, such as Barracuda CloudGen WAN, Cisco Cloud OnRamp for Multi-Cloud, and VMware SD-WAN. These types of devices are known as network virtual appliances (NVAs); they are deployed directly into a Virtual WAN hub and have an externally facing public IP address. This capability enables customers who want to connect their branch Customer Premises Equipment (CPE) to the same brand NVA in the virtual hub to take advantage of proprietary end-to-end SD-WAN capabilities. Once VNets are connected to the virtual hub, NVAs enable transitive connectivity throughout the organization's Virtual WAN.

Manage an NVA in a Virtual Hub

The NVAs available in the Azure Marketplace can be deployed directly into a virtual hub and nowhere else. Each is deployed as a Managed Application, which allows Azure Virtual WAN to manage the configuration of the NVA. They cannot be deployed within an arbitrary VNet.

The following diagram shows the NVA deployment process:



Although each NVA offers support for different CPEs and has a slightly different user experience, they all offer a Managed Application experience through Azure Marketplace, NVA Infrastructure Unit-based capacity and billing, and Health Metrics surfaced through Azure Monitor.

Deploy an NVA in your Virtual Hub

To deploy an NVA in your virtual hub, you can access the Azure Marketplace through the Azure portal and select the Managed Application for the NVA partner that you need to enable connectivity for your devices. When you create an NVA in the Virtual WAN hub, like all Managed Applications, there will be two Resource Groups created in your subscription.

- **Customer Resource Group** - This will contain an application placeholder for the Managed Application. Partners can use this resource group to expose whatever customer properties they choose here.
- **Managed Resource Group** - Customers cannot configure or change resources in this resource group directly, as this is controlled by the publisher of the Managed Application. This Resource Group will contain the NetworkVirtualAppliances resource.

The NVA is configured automatically as part of the deployment process. Once the NVA has been provisioned into the virtual hub, any additional configuration must be performed via the NVA partners portal or management application. You cannot access the NVA directly.

Unlike Azure VPN Gateway configurations, you do not need to create Site resources, Site-to-Site connection resources, or point-to-site connection resources to connect your branch sites to your NVA in the Virtual WAN hub. This is all managed via the NVA partner.

You still need to create Hub-to-VNet connections to connect your Virtual WAN hub to your Azure VNets.

Create the Network Virtual Appliance in the hub

In this step, you will create a Network Virtual Appliance in the hub. The procedure for each NVA will be different for each NVA partner's product. For this example, we are creating a Barracuda CloudGen WAN Gateway.

1. Locate the Virtual WAN hub you created in the previous step and open it.

The screenshot shows the Azure portal interface for a 'NVAHub' Virtual WAN hub. The 'Overview' tab is selected. Key details shown include:

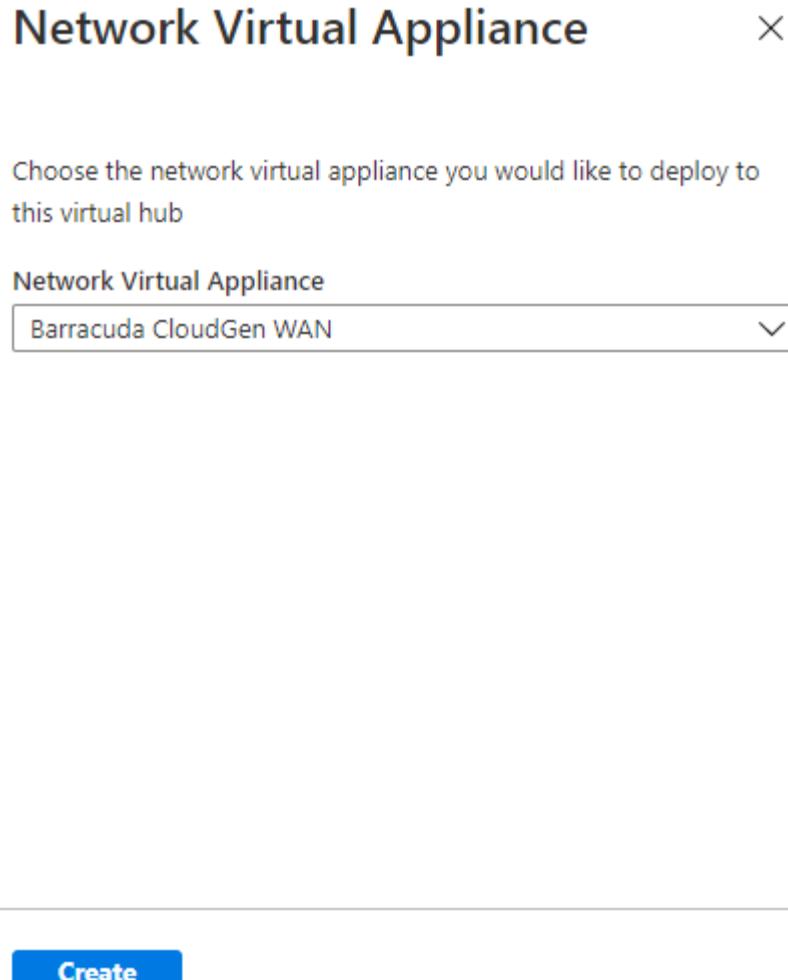
- Name:** NVAHub
- Status:** Hub VNET creation successful (Configure your hub)
- Private address space:** 10.0.0.0/24
- Location:** North Europe

The 'Virtual network connections' section indicates 0 vNet connections and 0 routes. Below this, there are five 'Create' links for different connection types:

- VPN (Site to site) - No gateway (Create)
- User VPN (Point to site) - No gateway (Create)
- ExpressRoute - No gateway (Create)
- Azure Firewall - No firewall (Create)
- Network Virtual Appliance - No gateway (Create)

2. Find the Network Virtual Appliances tile and select the Create link.

3. On the Network Virtual Appliance blade, select Barracuda CloudGen WAN, then select the Create



button.

4. This will take you to the Azure Marketplace offer for the Barracuda CloudGen WAN gateway. Read the terms, then select the **Create** button when you're

ready.

Basics CloudGen WAN gateway Review + create

Project details

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

Subscription * ⓘ ExpressRoute-Lab

Resource group * ⓘ SEA-Cust13
Create new

Instance details

Region * ⓘ North Europe

Managed Application Details

Provide a name for your managed application, and its managed resource group. Your application's managed resource group holds all the resources that are required by the managed application which the consumer has limited access to.

Application Name * barracuda

Managed Resource Group * ⓘ mrg-barracuda_clougenwan_gateway-p-20200702115052

Review + create < Previous Next : CloudGen WAN gateway >

5. On the Basics page you will need to provide the following information:
 - **Subscription** - Choose the subscription you used to deploy the Virtual WAN and hub.
 - **Resource Group** - Choose the same Resource Group you used to deploy the Virtual WAN and hub.
 - **Region** - Choose the same Region in which your Virtual hub resource is located.
 - **Application Name** - The Barracuda NextGen WAN is a Managed Application. Choose a name that makes it easy to identify this resource, as this is what it will be called when it appears in your subscription.
 - **Managed Resource Group** - This is the name of the Managed Resource Group in which Barracuda will deploy resources that are managed by them. The name should be pre-populated.

6. Select the Next: CloudGen WAN gateway button.

Basics **CloudGen WAN gateway** Review + create

i Before configuring your first gateway, you need to subscribe to the service in the CloudGen WAN Service in the Azure Marketplace. [Click here to go to the Barracuda CloudGen WAN service and subscribe.](#)

Virtual WAN hub [\(i\)](#) [\(v\)](#)

NVA infrastructure scale unit * [\(i\)](#) [\(v\)](#)

i To automatically configure your CloudGen WAN gateway, you must supply a valid token below. [Click here to go to the Barracuda CloudGen WAN service to generate your gateway token.](#)

Token * [\(i\)](#)

[Review + create](#) [< Previous](#) [Next : Review + create >](#)

7. Provide the following information here:

- **Virtual WAN Hub** - The Virtual WAN hub you want to deploy this NVA into.
- **NVA Infrastructure Units** - Indicate the number of NVA Infrastructure Units you want to deploy this NVA with. Choose the amount of aggregate bandwidth capacity you want to provide across all of the branch sites that will be connecting to this hub through this NVA.
- **Token** - Barracuda requires that you provide an authentication token here in order to identify yourself as a registered user of this product.

NVA Infrastructure Units

When you create an NVA in the Virtual WAN hub, you must choose the number of NVA Infrastructure Units you want to deploy it with. An NVA Infrastructure Unit is a unit of aggregate bandwidth capacity for an NVA in the Virtual WAN hub. An NVA Infrastructure Unit is similar to a VPN Scale Unit in terms of the way you think about capacity and sizing.

- One NVA Infrastructure Unit represents 500 Mbps of aggregate bandwidth for all branch site connections coming into this NVA.
- Azure supports from 1-80 NVA Infrastructure Units for a given NVA virtual hub deployment.
- Each partner may offer different NVA Infrastructure Unit bundles that are a subset of all supported NVA Infrastructure Unit configurations.

To learn more about deploying an NVA, see [How to create a Network Virtual Appliance in an Azure Virtual WAN hub \(Preview\)](#).

Summary

Completed 100 XP

- 1 minute

As your organization moves to Azure, you must design and implement a hybrid connectivity solution that will address the short term and long-term goals of the organization's global enterprise IT footprint.

In this module you learned about three ways to connect your on premises data center and remote users to an Azure virtual network.

You now have the fundamental knowledge required to design and implement hybrid networking in Azure.

Design and implement Azure ExpressRoute

Explore Azure ExpressRoute

200 XP

- 22 minutes

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider. With ExpressRoute, you can establish connections to various Microsoft cloud services, such as Microsoft Azure and Microsoft 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 colocation facility. Since ExpressRoute connections do not go over the public Internet, this approach allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security.

ExpressRoute capabilities

Some key benefits of ExpressRoute are:

- 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
- Built-in redundancy in every peering location for higher reliability

Azure ExpressRoute is used to create private connections between Azure datacenters and infrastructure on your premises or in a colocation environment. ExpressRoute connections do not go over the public Internet, and they offer more reliability, faster speeds, and lower latencies than typical Internet connections.

Understand use cases for Azure ExpressRoute

Faster and Reliable connection to Azure services - Organizations leveraging Azure services look for reliable connections to Azure services and data centers. Public internet is dependent upon many factors and may not be suitable for a business. Azure ExpressRoute is used to create private connections between Azure data centers and infrastructure on your premises or in a colocation environment. Using ExpressRoute

connections to transfer data between on-premises systems and Azure can also give significant cost benefits.

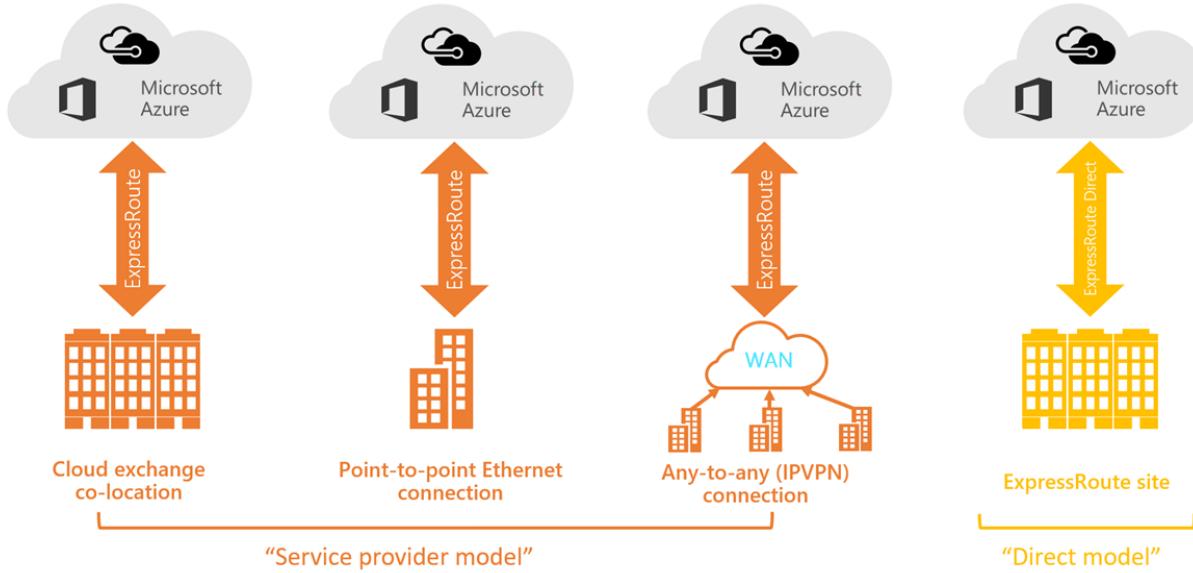
Storage, backup, and Recovery - Backup and Recovery are important for an organization for business continuity and recovering from outages. ExpressRoute gives you a fast and reliable connection to Azure with bandwidths up to 100 Gbps, which makes it excellent for scenarios such as periodic data migration, replication for business continuity, disaster recovery and other high-availability strategies.

Extends Data center capabilities - ExpressRoute can be used to connect and add compute and storage capacity to your existing data centers. With high throughput and fast latencies, Azure will feel like a natural extension to or between your data centers, so you enjoy the scale and economics of the public cloud without having to compromise on network performance.

Predictable, reliable, and high-throughput connections - With predictable, reliable, and high-throughput connections offered by ExpressRoute, enterprises can build applications that span on-premises infrastructure and Azure without compromising privacy or performance. For example, run a corporate intranet application in Azure that authenticates your customers with an on-premises Active Directory service, and serve all your corporate customers without traffic ever routing through the public Internet.

ExpressRoute connectivity models

You can create a connection between your on-premises network and the Microsoft cloud in four different ways, CloudExchange Co-location, Point-to-point Ethernet Connection, Any-to-any (IPVPN) Connection, and ExpressRoute Direct. Connectivity providers may offer one or more connectivity models.



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.

Direct from ExpressRoute sites

You can connect directly into the Microsoft's global network at a peering location strategically distributed across the world. ExpressRoute Direct provides dual 100 Gbps or 10-Gbps connectivity, which supports Active/Active connectivity at scale.

Design considerations for ExpressRoute deployments

When planning an ExpressRoute deployment, there are many decisions to make. This section discusses a few key areas that you must consider as you design your deployment.

Choose between provider and direct model (ExpressRoute Direct)

ExpressRoute Direct

ExpressRoute Direct gives you the ability to connect directly into Microsoft's global network at peering locations strategically distributed around the world. ExpressRoute Direct provides dual 100 Gbps or 10-Gbps connectivity, which supports Active/Active connectivity at scale. You can work with any service provider for ExpressRoute Direct.

Key features that ExpressRoute Direct provides includes:

- 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

Using ExpressRoute direct vs using a Service Provider

ExpressRoute using a Service Provider	ExpressRoute Direct
Uses 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	Customer may select a combination of the following circuit SKUs on 100-Gbps ExpressRoute Direct: 5 Gbps 10 Gbps 40 Gbps 100 Gbps Customer may select a combination of the following circuit SKUs on 10-Gbps ExpressRoute Direct: 1 Gbps 2 Gbps 5 Gbps 10 Gbps
Optimized for single tenant	Optimized for single tenant with multiple business units and multiple work environments

Route advertisement

When Microsoft peering gets configured on your ExpressRoute circuit, the Microsoft Edge routers establish a pair of Border Gateway Protocol (BGP) sessions with your edge routers through your connectivity provider. No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter.

In order to associate a route filter:

- You must have an active ExpressRoute circuit that has Microsoft peering provisioned.
- Create an ExpressRoute circuit and have the circuit enabled by your connectivity provider before you continue. 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.

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.

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.

Bidirectional Forwarding Detection

ExpressRoute supports Bidirectional Forwarding Detection (BFD) both over private and Microsoft peering. When you enable BFD over ExpressRoute, you can speed up the link failure detection between Microsoft Enterprise edge (MSEE) devices and the routers that your ExpressRoute circuit gets configured (CE/PE). You can configure ExpressRoute over your edge routing devices or your Partner Edge routing devices (if you went with managed Layer 3 connection service). This section walks you through the need for BFD, and how to enable BFD over ExpressRoute.

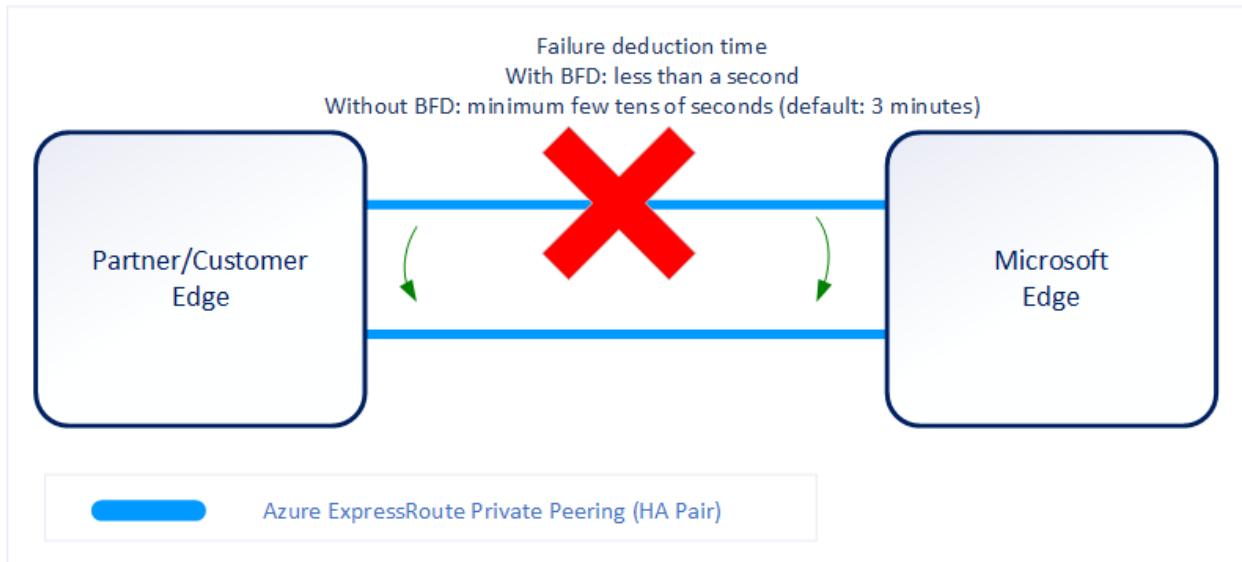
You can enable ExpressRoute circuit either by Layer 2 connections or managed Layer 3 connections. In both cases, if there are more than one Layer-2 devices in the ExpressRoute connection path, the responsibility of detecting any link failures in the path lies with the overlying BGP session.

On the MSEE devices, BGP keep-alive and hold-time are typically configured as 60 and 180 seconds, respectively. For that reason, when a link failure happens it can take up to three minutes to detect any link failure and switch traffic to alternate connection.

You can control the BGP timers by configuring a lower BGP keep-alive and hold-time on your edge peering device. If the BGP timers are not the same between the two peering devices, the BGP session will establish using the lower time value. The BGP keep-alive can be set as low as three seconds, and the hold-time as low as 10 seconds. However, setting a very aggressive BGP timer isn't recommended because the protocol is process intensive.

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

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



Enabling BFD

BFD is configured by default under all the newly created ExpressRoute private peering interfaces on the MSEEs. As such, to enable BFD, you only need to configure BFD on both your primary and secondary devices. Configuring BFD is two-step process. You configure the BFD on the interface and then link it to the BGP session.

When you disable a peering, the Border Gateway Protocol (BGP) session for both the primary and the secondary connection of your ExpressRoute circuit is shut down. When you enable a peering, the BGP session on both the primary and the secondary connection of your ExpressRoute circuit is restored.

Note

The first time you configure the peering on your ExpressRoute circuit, the Peerings are enabled by default.

Resetting your ExpressRoute Peerings might be helpful in the following scenarios:

You are testing your disaster recovery design and implementation. For example, assume that you have two ExpressRoute circuits. You can disable the Peerings of one circuit and force your network traffic to use the other circuit.

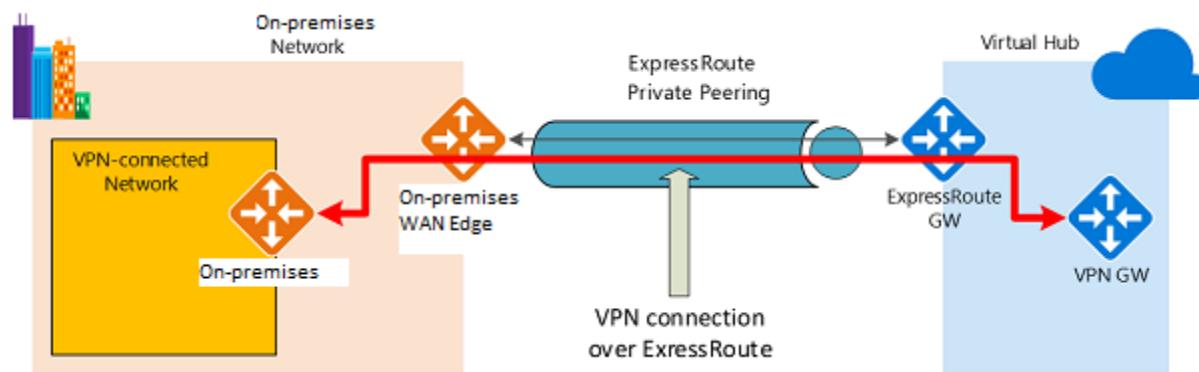
You want to enable Bidirectional Forwarding Detection (BFD) on Azure private peering or Microsoft peering. If your ExpressRoute circuit was created before August 1, 2018, on Azure private peering or before January 10, 2020, on Microsoft peering, BFD was not enabled by default. Reset the peering to enable BFD.

Configure encryption over ExpressRoute

This section shows you how to use Azure Virtual WAN to establish an IPsec/IKE VPN connection from your on-premises network to Azure over the private peering of an Azure ExpressRoute circuit. This technique can provide an encrypted transit between the on-premises networks and Azure virtual networks over ExpressRoute, without going over the public internet or using public IP addresses.

Topology and routing

The following diagram shows an example of VPN connectivity over ExpressRoute private peering:



The diagram shows a network within the on-premises network connected to the Azure hub VPN gateway over ExpressRoute private peering. The connectivity establishment is straightforward:

- Establish ExpressRoute connectivity with an ExpressRoute circuit and private peering.
- Establish the VPN connectivity.

An important aspect of this configuration is routing between the on-premises networks and Azure over both the ExpressRoute and VPN paths.

Traffic from on-premises networks to Azure

For traffic from on-premises networks to Azure, the Azure prefixes (including the virtual hub and all the spoke virtual networks connected to the hub) are advertised via both the ExpressRoute private peering BGP and the VPN BGP. This results in two network routes (paths) toward Azure from the on-premises networks:

- One over the IPsec-protected path

- One directly over ExpressRoute without IPsec protection

To apply encryption to the communication, you must make sure that for the VPN-connected network in the diagram, the Azure routes via on-premises VPN gateway are preferred over the direct ExpressRoute path.

Traffic from Azure to on-premises networks

The same requirement applies to the traffic from Azure to on-premises networks. To ensure that the IPsec path is preferred over the direct ExpressRoute path (without IPsec), you have two options:

- Advertise more specific prefixes on the VPN BGP session for the VPN-connected network. You can advertise a larger range that encompasses the VPN-connected network over ExpressRoute private peering, then more specific ranges in the VPN BGP session. For example, advertise 10.0.0.0/16 over ExpressRoute, and 10.0.1.0/24 over VPN.
- Advertise disjoint prefixes for VPN and ExpressRoute. If the VPN-connected network ranges are disjoint from other ExpressRoute connected networks, you can advertise the prefixes in the VPN and ExpressRoute BGP sessions, respectively. For example, advertise 10.0.0.0/24 over ExpressRoute, and 10.0.1.0/24 over VPN.

In both examples, Azure will send traffic to 10.0.1.0/24 over the VPN connection rather than directly over ExpressRoute without VPN protection.

[!WARNING]

If you advertise the same prefixes over both ExpressRoute and VPN connections, Azure will use the ExpressRoute path directly without VPN protection.

Design redundancy for an ExpressRoute deployment

There are 2 ways in which redundancy can be planned for an ExpressRoute deployment.

- Configure ExpressRoute and site to site coexisting connections
- Create a zone redundant VNET gateway in Azure Availability zones

Configure ExpressRoute and site to site coexisting connections

This section 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.

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.

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

Network Limits and limitations

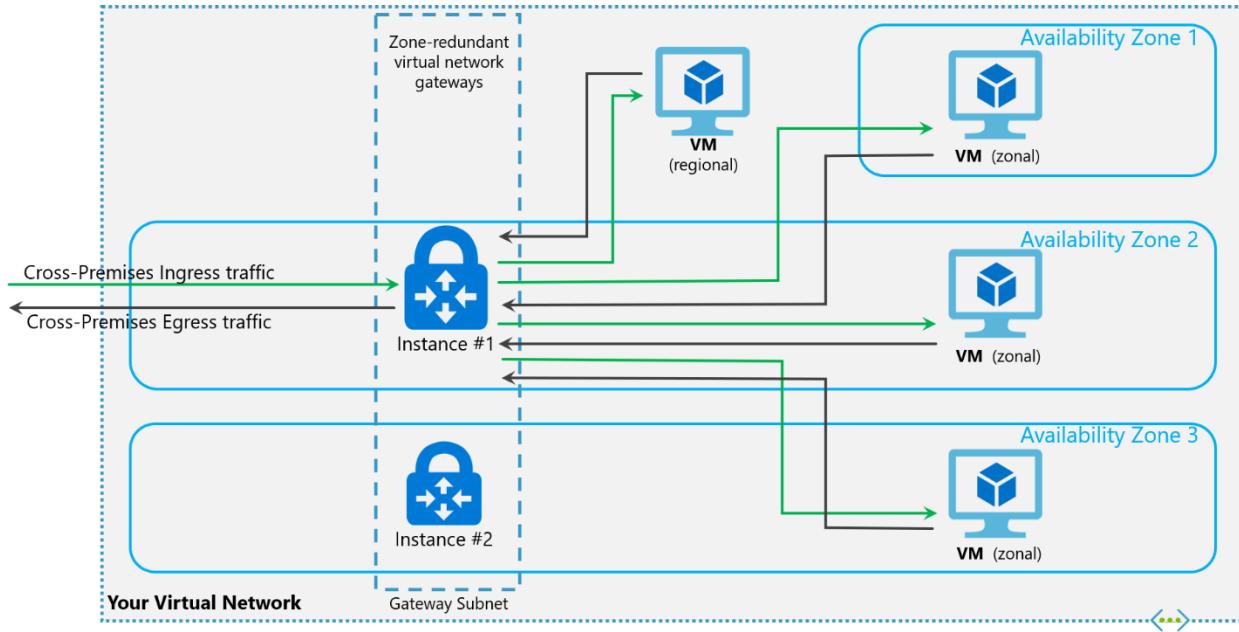
- **Only route-based VPN gateway is supported.** You must use a route-based VPN gateway. You also can use a route-based VPN gateway with a VPN connection configured for 'policy-based traffic selectors'.
- **The ASN of Azure VPN Gateway must be set to 65515.** Azure VPN Gateway supports the BGP routing protocol. For ExpressRoute and Azure VPN to work together, you must keep the Autonomous System Number of your Azure VPN gateway at its default value, 65515. If you previously selected an ASN other than 65515 and you change the setting to 65515, you must reset the VPN 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.
- **Coexistence in a dual stack VNet is not supported.** If you are using ExpressRoute IPv6 support and a dual-stack ExpressRoute gateway, coexistence with VPN Gateway will not be possible.

Create a zone redundant VNet gateway in Azure availability zones

You can deploy VPN and ExpressRoute gateways in [Azure Availability Zones](#). This brings resiliency, scalability, and higher availability to virtual network gateways. Deploying gateways in Azure Availability Zones physically and logically separates gateways within a region, while protecting your on-premises network connectivity to Azure from zone-level failures.

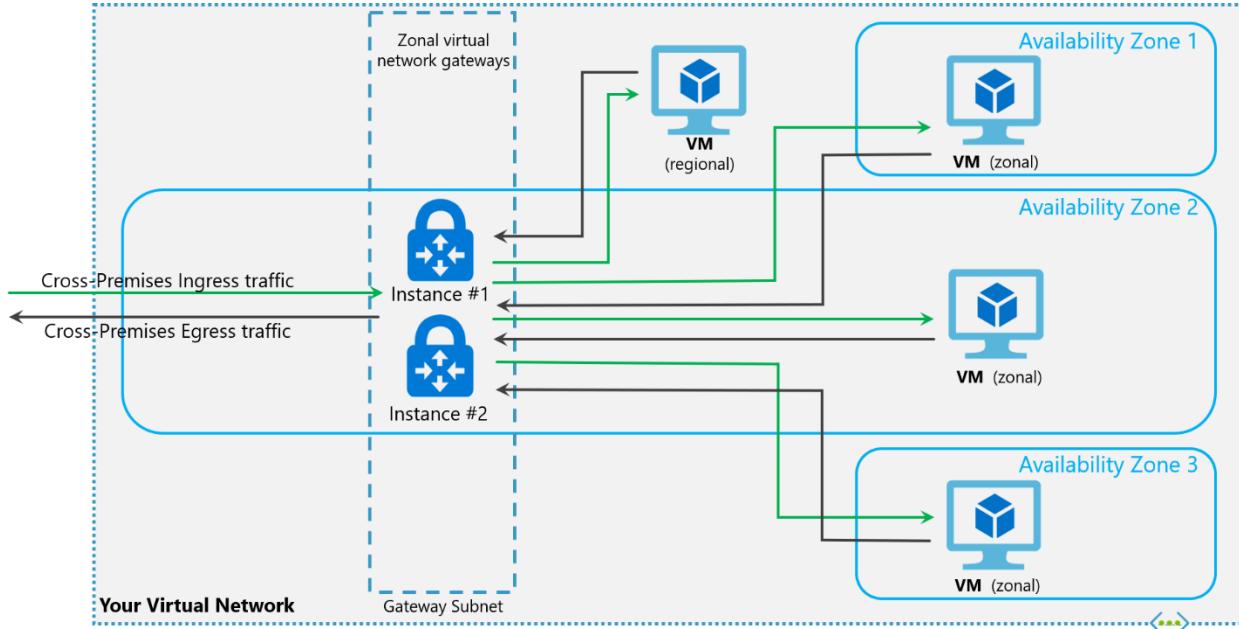
Zone-redundant gateways

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



Zonal gateways

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



Gateway SKUs

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

Public IP SKUs

Zone-redundant gateways and zonal gateways both rely on the Azure public IP resource Standard SKU. The configuration of the Azure public IP resource determines whether the gateway that you deploy is zone-redundant, or zonal. If you create a public IP resource with a Basic SKU, the gateway will not have any zone redundancy, and the gateway resources will be regional.

- Zone-redundant gateways
 - When you create a public IP address using the **Standard** public IP SKU without specifying a zone, the behavior differs depending on whether the gateway is a VPN gateway, or an ExpressRoute gateway.
 - For a VPN gateway, the two gateway instances will be deployed in any 2 out of these three zones to provide zone-redundancy.
 - For an ExpressRoute gateway, since there can be more than two instances, the gateway can span across all the three zones.
- Zonal gateways
 - When you create a public IP address using the **Standard** public IP SKU and specify the Zone (1, 2, or 3), all the gateway instances will be deployed in the same zone.
- Regional gateways
 - When you create a public IP address using the **Basic** public IP SKU, the gateway is deployed as a regional gateway and does not have any zone-redundancy built into the gateway.

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

If you have ExpressRoute Microsoft Peering enabled, you can receive the public IP address of your Azure VPN gateway on the ExpressRoute connection. To set up your site-to-site VPN connection as a backup, you must configure your on-premises network so that the VPN connection is routed to the Internet.

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.

Design an ExpressRoute deployment

200 XP

- 7 minutes

ExpressRoute enables us to connect on Premises to Azure services seamlessly. Let's review some design decisions you will make before deploying an ExpressRoute circuit.

ExpressRoute circuit SKUs

Azure ExpressRoute has three different circuit SKUs: [Local](#), Standard, and [Premium](#). The way you are charged for your ExpressRoute usage varies between these three SKU types.

- **Local SKU** - With Local SKU, you are automatically charged with an Unlimited data plan.
- **Standard and Premium SKU** - You can select between a Metered or an Unlimited data plan. All ingress data are free of charge except when using the Global Reach add-on.

Important

Based on requirements of workloads and data plan, selection of SKU types can help optimize cost and budget.

Explore pricing based on ExpressRoute SKU

SKU models have been discussed previously as Local, Standard and Premium. It is a good practice to estimate costs before using Azure ExpressRoute as the price might affect your design decisions.

Use the [Azure pricing calculator](#) to estimate costs before you create an Azure ExpressRoute circuit.

1. On the left, select **Networking**, then select **Azure ExpressRoute** to begin.
2. Select the appropriate Zone depending on your peering location.
3. Then select the SKU, Circuit Speed, and the Data Plan you would like an estimate for.
4. In the Additional outbound data transfer, enter an estimate in GB of how much outbound data you might use over the course of a month.
5. Lastly, you can add the Global Reach Add-on to the estimate.

Choose a peering location

Peering location is of importance when working with ExpressRoute.

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 (Peering 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.

Azure regions to ExpressRoute locations within a geopolitical region.

The following link provides a list of [Azure regions to ExpressRoute locations](#) within a geopolitical region. This page is kept up to date with the latest ExpressRoute locations and providers.

ExpressRoute connectivity providers

The following link lists locations by service provider. This page is kept up to date with the latest available providers by location, see [Service providers by location](#).

Connectivity through Exchange providers

If your connectivity provider is not listed in previous sections, you can still create a connection. Several connectivity providers are already connected to Ethernet exchanges.

Connectivity through satellite operators

If you are remote and do not have fiber connectivity or want to explore other connectivity options, you can check the following satellite operators.

Additional Connectivity options:

- Through additional service providers
- Datacenter providers
- National Research and Education networks (NERN)
- System integrators

Choose the right ExpressRoute circuit and billing model

Microsoft offers various Express Route options depending on the desired bandwidth of this private connection between the customer on premises network and the selected Azure region. Typically, enterprises need to evaluate their current usage and determine how much data they use monthly to start with.

The next step is to figure out which of the available ExpressRoute is the best choice depending upon the requirements of the Enterprise keeping in mind the budget and SLA requirements.

When you deploy ExpressRoute, you must choose between the Local, Standard and Premium SKUs. The Standard and Premium SKU are available in a metered version, where you pay per used GB and an unlimited option.

The other option is the ExpressRoute Direct, connecting your network to the closest Microsoft Edge node which then connects to the Microsoft Global Network, to connect to other customers offices or factories and any Azure Region. The usage of the Microsoft Global Network is charged on top of the ExpressRoute Direct.

Please refer to the [Express Route pricing](#) for details on metered and unlimited data plan based on the bandwidth.

You can purchase ExpressRoute circuits for a wide range of bandwidths. The supported bandwidths are listed as follows. 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

Choose a billing model

You can pick a billing model that works best for you. Choose between the billing models listed as followed.

- **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 every 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.

Exercise: configure an ExpressRoute gateway

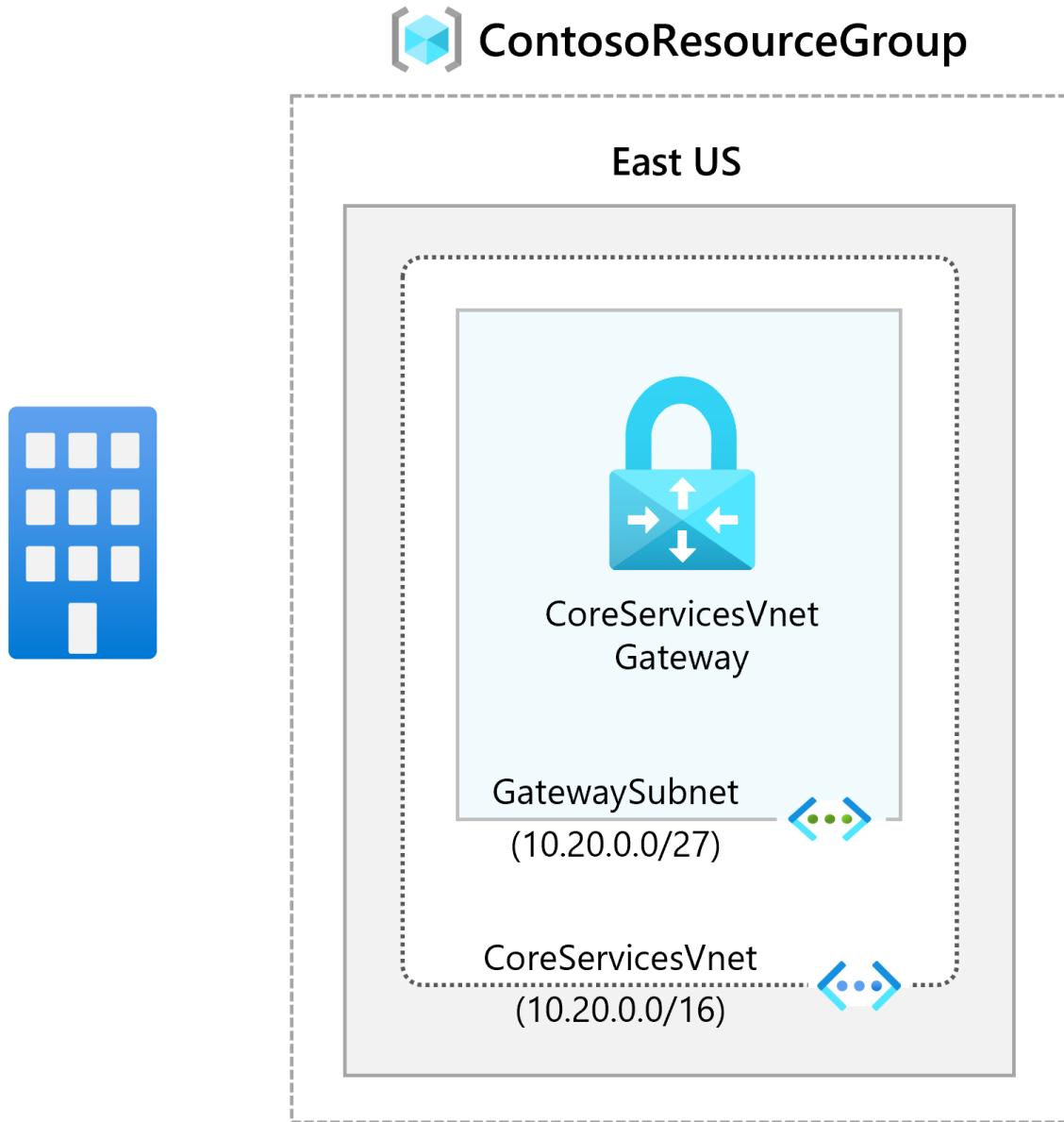
Completed 100 XP

- 5 minutes

Lab scenario

In this lab, you'll create a virtual network gateway for ExpressRoute.

Architecture diagram



Objectives

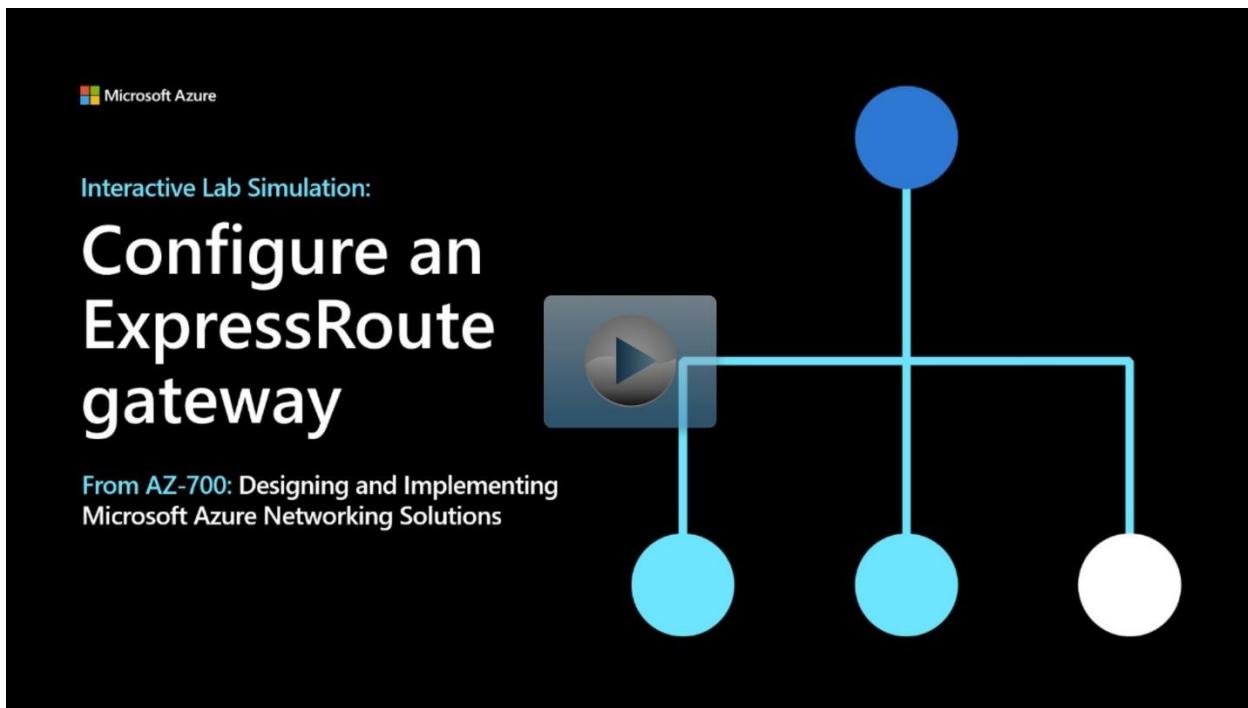
- Task 1: Create the VNet and gateway subnet
- Task 2: Create the virtual network gateway

Note

Select the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Configure%20an%20ExpressRoute%20gateway>

Exercise: provision an ExpressRoute circuit

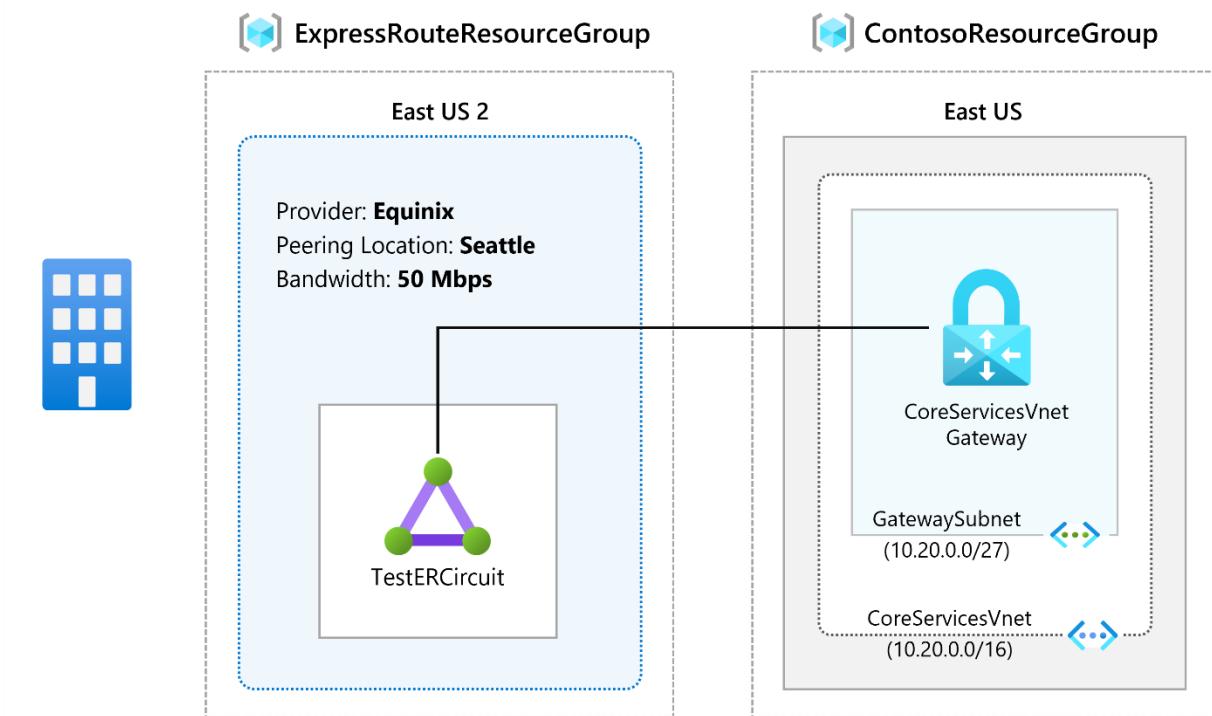
Completed 100 XP

- 5 minutes

Lab scenario

In this exercise, you will create an ExpressRoute circuit using the Azure portal and the Azure Resource Manager deployment model.

Architecture diagram



Objectives

- **Task 1:** Create and provision an ExpressRoute circuit
- **Task 2:** Retrieve your Service key
- **Task 3:** Deprovisioning an ExpressRoute circuit

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

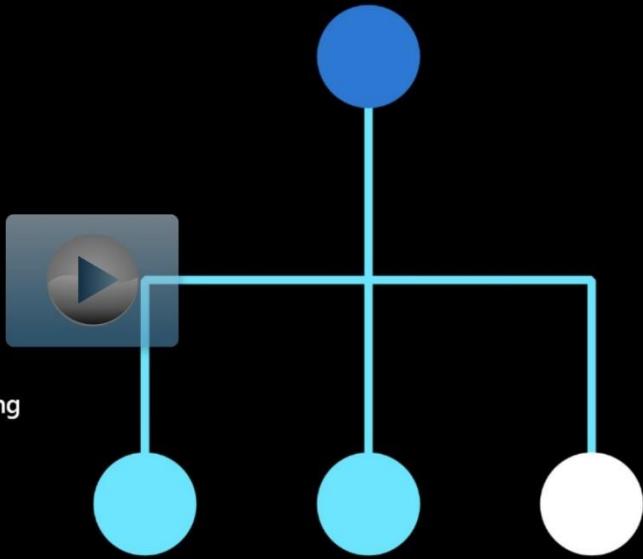
You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



Interactive Lab Simulation:

Provision an ExpressRoute circuit

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions



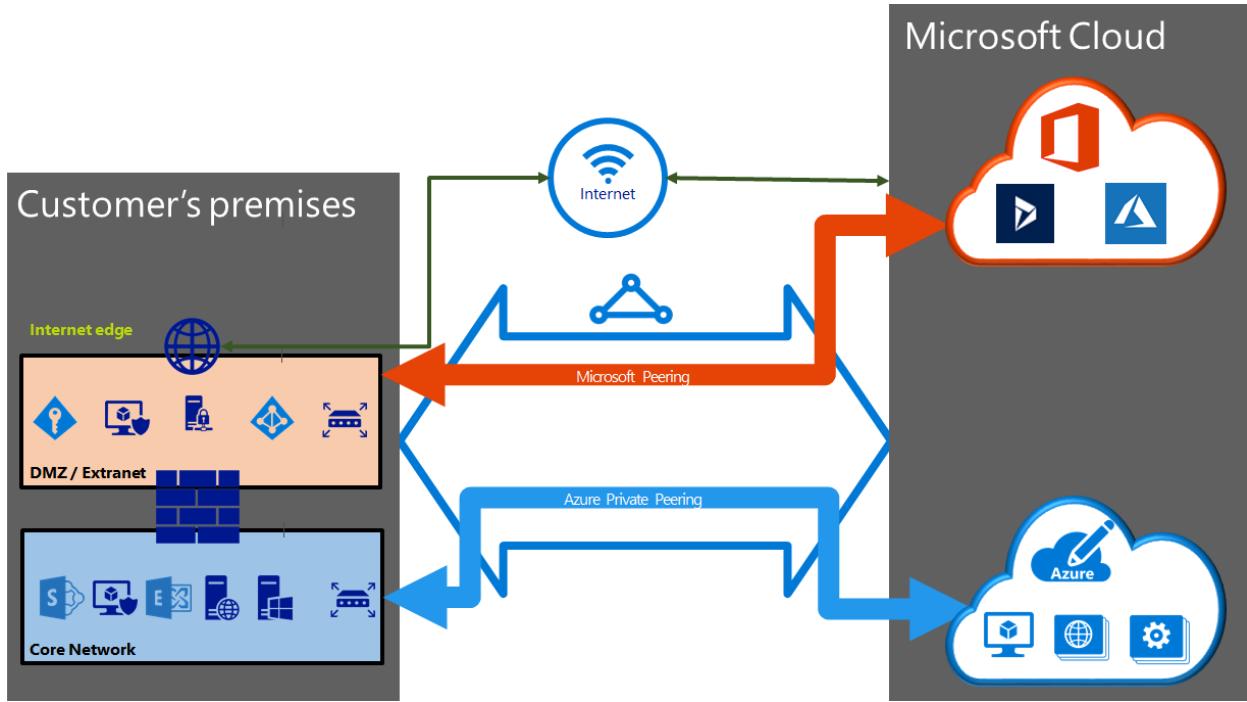
<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Provision%20an%20ExpressRoute%20circuit>

Configure peering for an ExpressRoute deployment

200 XP

- 10 minutes

An ExpressRoute circuit has two peering options associated with it: 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.



Create Peering configuration

- You can configure private peering and Microsoft peering for an ExpressRoute circuit. Peering's 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.
- You must have an active ExpressRoute circuit. Have the circuit enabled by your connectivity provider before you continue. 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 the key on both sides of the tunnel. The limit is a maximum of 25 alphanumeric characters. Special characters are not supported.
- This only applies 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 the routing for you.

Choose between private peering only, Microsoft peering only, or both

The following table compares the two peering. Public peering is deprecated for new peering.

Features	Private Peering	Microsoft Peering
Max. # prefixes supported per peering	4000 by default, 10,000 with ExpressRoute Premium	200
IP address ranges supported	Any valid IP address within your WAN.	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.
IP protocols supported	IPv4, IPv6 (preview)	IPv4, IPv6
Routing Interface IP addresses	RFC1918 and public IP addresses	Public IP addresses registered to you in routing registries.
MD5 Hash support	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. 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.

Important

IPv6 support for private peering is currently in Public Preview. If you would like to connect your virtual network to an ExpressRoute circuit with IPv6-based private peering configured, please make sure that your virtual network is dual stack and follows the guidelines for [IPv6 for Azure VNet](#).

Configure 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 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. You can visit the [Azure Subscription and Service Limits, Quotas, and Constraints](#) page for up-to-date information on limits.

Configure Microsoft peering

Microsoft 365 was created to be accessed securely and reliably via the Internet. Because of this, it is recommended to use ExpressRoute for specific scenarios.

Connectivity to Microsoft online services (Microsoft 365 and Azure PaaS services) occurs through Microsoft peering. You can enable bidirectional 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.

Configure route filters for Microsoft Peering

Route filters are a way to consume a subset of supported services through Microsoft peering.

Microsoft 365 services such as Exchange Online, SharePoint Online, and Skype for Business, are accessible through the Microsoft peering. When Microsoft peering gets configured in an ExpressRoute circuit, all prefixes related to these services gets 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.

Connectivity to all Azure and Microsoft 365 services causes many prefixes to get advertised through BGP. The large number of prefixes 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. Route filtering 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 gets configured on your ExpressRoute circuit, the Microsoft Edge routers establish a pair of BGP sessions with your edge routers through your connectivity provider. 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 allowed list of all the BGP community values. Once a route filter resource gets defined and attached to an ExpressRoute circuit, all prefixes that map to the BGP community values gets advertised to your network.

To attach route filters with Microsoft 365 services, you must have authorization to consume Microsoft 365 services through ExpressRoute. If you are not authorized to consume Microsoft 365 services through ExpressRoute, the operation to attach route filters fails.

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.

- Select **Create a resource** then search for Route filter as shown in the following image:

Route filter

Microsoft



Route filter

Save for later

Microsoft

Create

[Overview](#) [Plans](#) [Usage Information + Support](#)

Route filters are a way to consume a subset of supported services through a Microsoft peering.

More offers from Microsoft



Workspace

Microsoft

Windows Virtual Desktop resource



Wire Data 2.0

Microsoft

Provides the ability to explore wire data and helps identify network related issues.



Microsoft HPC Pack 2012 R2

Microsoft

Enterprise-class HPC solution. Easy to deploy, cost-effective and supports Windows/Linux workloads.



Windows Server 2019 Datacenter (zh-cn)

Microsoft

Azure Hybrid Benefit for Windows Server

- Place the route filter in a resource group. Ensure the location is the same as the ExpressRoute circuit. Select **Review + create** and then **Create**.

Create route filter

Basics Tags Review + create

Route filters help you filter the traffic going through your ExpressRoute. Essentially a white list of all the BGP community values, a route filter lets you identify services you want to consume through your ExpressRoute circuit's Microsoft peering. 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. [Learn more about route filters](#)

Project details

Subscription * 

Azure Subscription 

Resource group * 

ExpressRouteResourceGroup 

[Create new](#)

Instance details

Name *

MyRouteFilter 

Region *

(US) West US 2 

 Route filter must be created in the same location as the ExpressRoutes it will be associated with.

[Review + create](#)

< Previous

Next : Tags >

[Download a template for automation](#)

Create a filter rule

To add and update rules, select the manage rule tab for your route filter.

The screenshot shows the Azure portal interface for a 'Route filter' named 'MyRouteFilter'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings (Circuits, Properties, Locks), Automation (Tasks, Export template), Support + troubleshooting (New support request), and Help & feedback. The main content area is titled 'Essentials' and includes sections for Resource group (change) to 'ExpressRouteResourceGroup', Status (Succeeded), Location (West US 2), Subscription (change) to 'Azure Subscription', and Subscription ID. Below this is a 'Tags (change)' section with a link to 'Click here to add tags'. Under 'Automation', there is a 'Allowed service communities' section with a search bar and a table for adding new communities. The 'Circuits' section also has a search bar and a table for managing circuits. The top navigation bar includes 'Search (Ctrl+)', 'Move', 'Delete', 'Refresh', 'Manage rule' (which is highlighted with a red box), and 'Add circuit'. A 'JSON View' link is located in the top right corner.

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

Manage rule

X

MyRouteFilter

 Save  Discard

Rule name *

Rule1 

Allowed service communities *

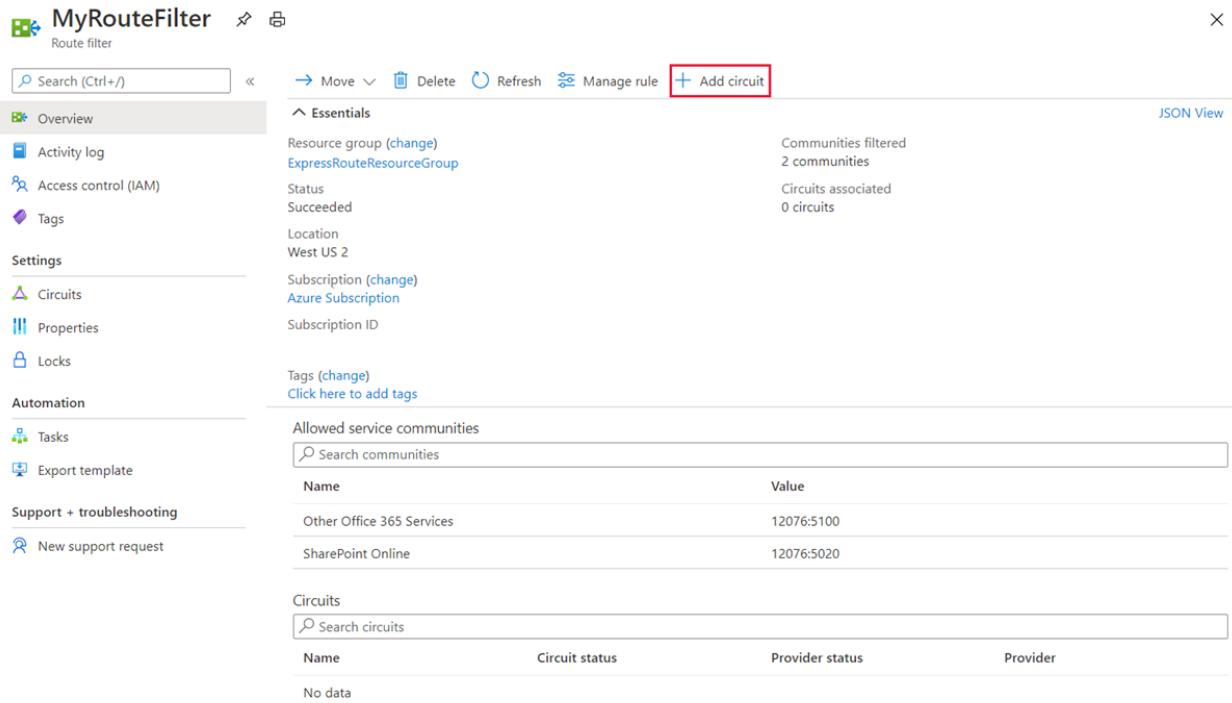
2 selected 

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 Active Directory (12076:5060)
- Azure Australia Central (12076:5103)
- Azure Australia Central 2 (12076:510)
- Azure Australia East (12076:51015)
- Azure Australia Southeast (12076:51014)
- Azure Brazil South (12076:51014)
- Azure Canada Central (12076:51020)
- Azure Canada East (12076:51021)
- Azure Central India (12076:51017)
- Azure Central US (12076:51009)
- Azure Central US EUAP (12076:51009)
- Azure East Asia (12076:51010)
- Azure East US (12076:51004)
- Azure East US 2 (12076:51005)

Attach the route filter to an ExpressRoute circuit

- 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 contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings (with Circuits, Properties, and Locks), Automation (Tasks and Export template), Support + troubleshooting (New support request), and Help. The main content area is titled 'Essentials' and displays resource group information (ExpressRouteResourceGroup), status (Succeeded), location (West US 2), and subscription details (Azure Subscription, Subscription ID). It also includes sections for Tags, Allowed service communities (with a search bar and table for Other Office 365 Services and SharePoint Online), and Circuits (with a search bar and a 'No data' message). At the top right, there are buttons for Move, Delete, Refresh, Manage rule, and the '+ Add circuit' button, which is highlighted with a red box. A 'JSON View' link is also present at the top right.

- If the connectivity provider configures peering for your ExpressRoute circuit, refresh the circuit from the ExpressRoute circuit page before you select the **+ Add Circuit** button.

The screenshot shows the Azure portal interface for an ExpressRoute circuit named "TestERCircuit". The left sidebar lists navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration, Connections, Authorizations, Peerings, Properties, Locks), Monitoring, and Metrics. The main content area displays the "Essentials" section with the following details:

Setting	Value
Resource group (change)	ExpressRouteResourceGroup
Circuit status	Enabled
Location	West US 2
Subscription (change)	Azure Subscription
Subscription ID	[redacted]
Tags (change)	Click here to add tags

Below this is the "Peering" section, which contains a table:

Type	Status	Primary subnet	Secondary subnet	Last modified by
Azure public	Not provisioned	-	-	-
Azure private	Provisioned	> One subnet configured	One subnet configured	Customer
Microsoft	Provisioned	> One subnet configured	One subnet configured	Customer

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 interface for a 'Route filter' named 'MyRouteFilter'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Settings (with Circuits, Properties, Locks), Automation (Tasks, Export template), Support + troubleshooting (New support request), and Help & feedback. The main content area is titled 'Overview' under 'Essentials'. It displays resource group (ExpressRouteResourceGroup), status (Succeeded), location (West US 2), subscription (Azure Subscription), and a list of communities filtered (2 communities). Below this is a 'Tags' section with a link to add tags. The 'Allowed service communities' section lists 'Other Office 365 Services' and 'SharePoint Online' with their respective values. The 'Circuits' section shows one circuit named 'TestERCircuit' with an enabled status and provisioned provider (Equinix). Top navigation includes Move, Delete, Refresh, Manage rule, and Add circuit buttons, along with a JSON View link.

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 Manage rule Add circuit

Activity log Access control (IAM) Tags

Settings Circuits Properties Locks

Automation Tasks Export template

Support + troubleshooting New support request

Essentials

Resource group (change)
ExpressRouteResourceGroup

Status Succeeded

Location West US 2

Subscription (change)
Azure Subscription

Subscription ID

Tags (change)
Click here to add tags

Allowed service communities

Search communities

Name	Value
Other Office 365 Services	12076:5100
SharePoint Online	12076:5020

Circuits

Search circuits

Name	Circuit status	Provider status	Provider
TestERCircuit	Enabled	Provisioned	Equinix

- Select the service communities you want and then select **Save**.

Manage rule

X

MyRouteFilter

 Save  Discard

Rule name *

Rule1 

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 Active Directory (12076:5060)
- Azure Australia Central (12076:5103)
- Azure Australia Central 2 (12076:510)
- Azure Australia East (12076:51015)
- Azure Australia Southeast (12076:51014)
- Azure Brazil South (12076:51014)
- Azure Canada Central (12076:51020)
- Azure Canada East (12076:51021)
- Azure Central India (12076:51017)
- Azure Central US (12076:51009)
- Azure Central US EUAP (12076:51009)
- Azure East Asia (12076:51010)
- Azure East US (12076:51004)
- Azure East US 2 (12076:51005)

To detach a route filter from an ExpressRoute circuit

- To detach a circuit from the route filter, right-click on the circuit and select **Disassociate**.

The screenshot shows the Azure portal interface for managing a route filter named 'MyRouteFilter'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings (with Circuits, Properties, and Locks), Automation (Tasks and Export template), and Support + troubleshooting (New support request). The main content area is titled 'Essentials' and displays resource group information: ExpressRouteResourceGroup, Status Succeeded, Location West US 2, Subscription Azure Subscription, and Subscription ID. Below this is a 'Tags' section with a link to add tags. The 'Allowed service communities' section lists 'Other Office 365 Services' and 'SharePoint Online' with their respective values. The 'Circuits' section shows a table with one row for 'TestERCircuit', which is Enabled, Provisioned, and associated with Equinix. The 'Dissociate' button next to the circuit name is highlighted with a red box.

Clean up resources

- You can delete a route filter by selecting the **Delete** button. Ensure the Route filter is not associate to any circuits before doing so.

The screenshot shows the Azure portal interface for a 'Route filter' named 'MyRouteFilter'. The top navigation bar includes 'Search (Ctrl+)', 'Move', 'Delete' (which is highlighted with a red box), 'Refresh', 'Manage rule', and 'Add circuit'. Below the navigation is a breadcrumb trail: 'Route filter > MyRouteFilter'. A left sidebar lists sections: Overview (selected), Activity log, Access control (IAM), Tags, Settings (Circuits, Properties, Locks), Automation (Tasks, Export template), and Support + troubleshooting (New support request). The main content area is titled 'Essentials' and shows details: Resource group (change) to 'ExpressRouteResourceGroup', Status 'Succeeded', Location 'West US 2', Subscription (change) to 'Azure Subscription', and Subscription ID. It also shows 'Communities filtered' (2 communities) and 'Circuits associated' (0 circuits, which is highlighted with a red box). The 'Settings' section contains a table for 'Allowed service communities' with two entries: 'Other Office 365 Services' (Value: 12076:5100) and 'SharePoint Online' (Value: 12076:5020). The 'Circuits' section is shown below, also highlighted with a red box, and contains a table with one entry: 'No data'.

Reset peering

Sign into the Azure portal

From a browser, go to the [Azure portal](#), and then sign in with your Azure account.

Reset a peering

You can reset the Microsoft peering and the Azure private peering on an ExpressRoute circuit independently.

- Choose the circuit that you want to change.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information. The left sidebar contains links for creating a resource, Home, Dashboard, All services, Favorites, Resource groups, All resources, Recent, App Services, and Virtual machines (classic). The main content area is titled 'ExpressRoute circuits' and shows a single record: 'TestERCircuit'. The circuit is listed as Enabled, Provisioned, Equinix, Seattle, in the ExpressRouteResourceGroup, and Azure Subscription. There are filters at the top for Subscription, Resource group, Location, and grouping.

- Choose the peering configuration that you want to reset.

The screenshot shows the 'TestERCircuit' settings page. The left sidebar lists Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration, Connections, Authorizations, Peerings, Properties, Locks), Monitoring, and Metrics. The main content area has tabs for Overview, Essentials, and Peering. The Essentials tab displays basic information like Resource group, Circuit status, Location, Subscription, and Service key. The Peering tab lists three entries: Azure public, Microsoft, and Azure private. The Azure private entry is highlighted with a red border and shows it is Provisioned with one subnet configured for both primary and secondary.

- Clear the **Enable Peering** check box, and then select **Save** to disable the peering configuration.

 Private peering ✖️ 🖨️

TestERCircuit

Enable Peering ⓘ

Peer ASN ⓘ
65020

Subnets
 Both
 IPv4
 IPv6

IPv4 Primary subnet ⓘ
192.168.11.16/30

IPv4 Secondary subnet ⓘ
192.168.11.20/30

VLAN ID ⓘ
110

Shared key

Enable Global Reach ⓘ

Save Cancel

- Select the **Enable Peering** check box, and then select **Save** to re-enable the peering configuration.

Connect an ExpressRoute circuit to a virtual network

Completed 100 XP

- 5 minutes

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).

In the previous exercises you created an ExpressRoute Gateway and an ExpressRoute circuit. You then learned how to configure peering for an express route circuit. You will now learn how to create a connection between your ExpressRoute circuit and Azure virtual network.

Connect a virtual network to an ExpressRoute circuit

- You must have an active ExpressRoute circuit.
- Ensure that you have Azure private peering configured for your circuit.
- Ensure that Azure private peering gets configured and establishes BGP peering between your network and Microsoft for end-to-end connectivity.
- Ensure that you have a virtual network and a virtual network gateway created and fully provisioned. 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 16 ExpressRoute circuits. Use the following process 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 virtual networks outside of the geopolitical region of the ExpressRoute circuit. The premium add-on will also allow you to connect more than 10 virtual networks to your ExpressRoute circuit depending on the bandwidth chosen.
- To create the connection from the ExpressRoute circuit to the target ExpressRoute virtual network gateway, the number of address spaces advertised from the local or peered virtual networks needs to be equal to or less than **200**. Once the connection has been successfully created, you can add additional address spaces, up to 1,000, to the local or peered virtual networks.

Add a VPN to an ExpressRoute deployment

This section 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 high availability and redundancy, you can configure multiple tunnels over the two MSEE-PE pairs of an 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 this section, 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.

Steps

- Configure Microsoft peering for your ExpressRoute circuit.
- Advertise selected Azure regional public prefixes to your on-premises network via Microsoft peering.
- Configure a VPN gateway and establish IPsec tunnels
- Configure the on-premises VPN device.
- Create the site-to-site IPsec/IKE connection.
- (Optional) Configure firewalls/filtering on the on-premises VPN device.

- Test and validate the IPsec communication over the ExpressRoute circuit.

Connect geographically dispersed networks with ExpressRoute global reach

200 XP

- 6 minutes

Use cross-region connectivity to link multiple ExpressRoute locations

There are various ways of designing and implementing ExpressRoute based on specific organizational requirements.

ExpressRoute connections enable access to the following services:

- Microsoft Azure services
- Microsoft 365 services

Connectivity to all regions within a geopolitical region

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

For example, if you connect to Microsoft in Amsterdam through ExpressRoute, you will have access to all Microsoft cloud services hosted in Northern and Western Europe.

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. You can also access services deployed in South America or Australia the same way you access North and West Europe regions. National clouds are excluded.

Local connectivity with ExpressRoute Local

You can transfer data cost-effectively by enabling the Local SKU. With Local SKU, you can bring your data to an ExpressRoute location near the Azure region you want. 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 an ExpressRoute circuit in Silicon Valley and another private data center in Texas connected to an ExpressRoute circuit in Dallas. With ExpressRoute Global Reach, you can connect your private data centers together through these two ExpressRoute circuits. Your cross-data-center traffic will traverse through Microsoft's network.

Rich connectivity partner ecosystem

ExpressRoute has a constantly growing ecosystem of connectivity providers and systems integrator partners. You can refer to [ExpressRoute partners and peering locations](#).

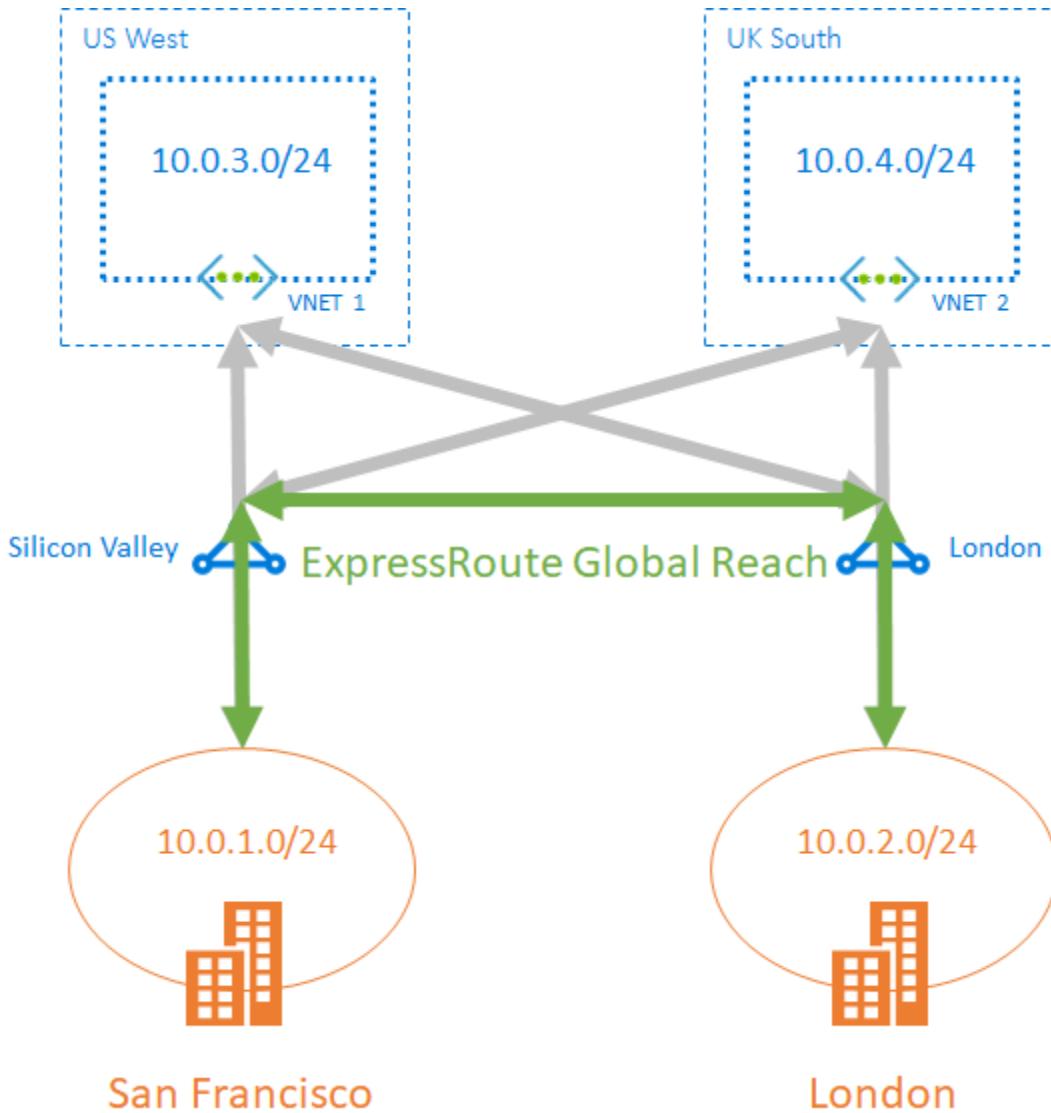
Connectivity to national clouds

Microsoft operates isolated cloud environments for special geopolitical regions and customer segments.

ExpressRoute Direct

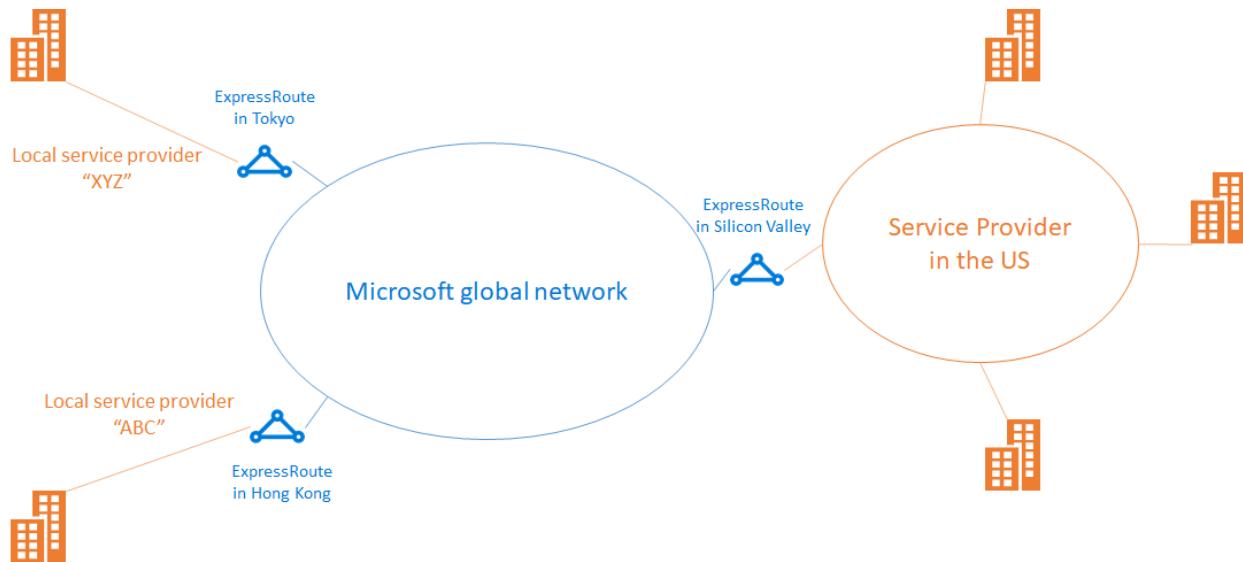
ExpressRoute Direct provides customers the opportunity to connect directly into Microsoft's global network at peering locations strategically distributed across the world. ExpressRoute Direct provides dual 100-Gbps connectivity, which supports Active/Active connectivity at scale.

ExpressRoute is a private and resilient way to connect your on-premises networks to the Microsoft Cloud. You can access many Microsoft cloud services such as Azure and Microsoft 365 from your private data center or your corporate network. For example, you might 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 have high-speed connectivity to Azure resources in US West and UK South. However, the branch offices cannot connect and send data directly with one another. In other words, 10.0.1.0/24 can send data to 10.0.3.0/24 and 10.0.4.0/24 network, but NOT to 10.0.2.0/24 network.



Choose when to use ExpressRoute global reach

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 your branches in the U.S., but the service provider does not operate in Japan and Hong Kong SAR, 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 the Microsoft global network.



Configure ExpressRoute global reach

These steps show you how to configure ExpressRoute Global Reach using Azure portal.

Before you begin

Before you start configuration, confirm the following criteria:

- 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.

Identify circuits

Identify the ExpressRoute circuits that you want use. You can enable ExpressRoute Global Reach between the private peering of any two ExpressRoute circuits, if they are in the supported countries/regions. The circuits are required to be 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.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with options like 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES' (which has 'Resource groups' selected), 'All resources', 'Recent', 'App Services', and 'Virtual machines (classic)'. The main area is titled 'ExpressRoute circuits' under 'Microsoft'. It includes a search bar, filter buttons for 'Subscription == Azure Subscription', 'Resource group == all', and 'Location == all', and buttons for 'Add', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', and 'Feedback'. Below these are buttons for 'Filter by name...', 'No grouping', and 'List view'. A table lists one record: 'TestERCircuit' (Enabled, Provisioned, Equinix, Seattle, ExpressRouteResourceGroup, Azure Subscription). There are also 'Add filter' and '...' buttons.

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. Select the **Azure private** peering configuration.

The screenshot shows the 'TestERCircuit' details page. The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration, Connections, Authorizations, Peerings, Properties, Locks, Monitoring, and Metrics. The 'Peerings' section is expanded, showing a table with columns: Type, Status, Primary subnet, Secondary subnet, and Last modified by. It lists 'Azure public' (Not provisioned, -, -, -) and 'Microsoft' (Not provisioned, -, -, -). The 'Azure private' row is highlighted with a red border. The right side shows essential details: Resource group (change) ExpressRouteResourceGroup, Provider Equinix, Circuit status Enabled, Location West US 2, Subscription (change) Azure Subscription, Bandwidth 50 Mbps, Service key 01234567-abcd-ef01-2345-6789abcdef01, and Tags (change) Click here to add tags.

2. Select **Add Global Reach** to open the Add Global Reach configuration page.

 Private peering 

TestERCircuit

Enable Peering [\(i\)](#)

Peer ASN * [\(i\)](#)
65020 

IPv4 Primary subnet * [\(i\)](#)
192.168.11.16/30

IPv4 Secondary subnet * [\(i\)](#)
192.168.11.20/30

VLAN ID * [\(i\)](#)
110 

Shared key

Add Global Reach 

Global Reach name
ExpressRoute Circuit name [\(i\)](#)
Global Reach subnet [\(i\)](#)

Save **Cancel**

3. On the Add Global Reach configuration page, give a name to this configuration. Select the ExpressRoute circuit you want to connect this circuit to and enter in a **/29 IPv4** for the Global Reach subnet. Azure uses IP addresses in this subnet to establish connectivity between the two ExpressRoute circuits. Do not use the addresses in this subnet in your Azure virtual networks, or in your on-premises network. Select **Add** to add the circuit to the private peering configuration.

Add Global Reach

X

TestERCircuit

Global Reach name *

TestERCircuit1-TestERCircuit2



Redeem authorization ⓘ

ExpressRoute circuit * ⓘ

TestERCircuit2

resourceGroup: ExpressRouteResourceGroup2, location: eastus



Global Reach subnet * ⓘ

192.168.11.24/29



Add

Cancel

4. Select **Save** to complete the Global Reach configuration. When the operation completes, you will have connectivity between your two on-premises networks through both ExpressRoute circuits.

Private peering

TestERCircuit

Enable Peering 

Peer ASN * 

65020



IPv4 Primary subnet * 

192.168.11.16/30

IPv4 Secondary subnet * 

192.168.11.20/30

VLAN ID * 

110



Shared key

Add Global Reach

Global Reach name

ExpressRoute Circuit name 

Global Reach subnet 

TestERCircuit1-TestERCirc...

TestERCircuit2

192.168.11.24/29



Save

Cancel

Verify the configuration

Verify the Global Reach configuration by selecting Private peering under the ExpressRoute circuit configuration. When configured correctly your configuration should look as follows:

 **Private peering** X

TestERCircuit

Peer ASN * ⓘ
65020 ✓

IPv4 Primary subnet * ⓘ
192.168.20.16/30 ✓

IPv4 Secondary subnet * ⓘ
192.168.20.20/30 ✓

Enable IPv4 Peering ⓘ

VLAN ID * ⓘ
200 ✓

Shared key

Add Global Reach

Global Reach name	ExpressRoute Circuit name ⓘ	IPv4 Subnet ⓘ	IPv6 Subnet ⓘ
TestERCircuit1-TestE...	TestERCircuit2 (ER...	192.168.11.24/29	

Save Cancel

Disable connectivity

To disable connectivity between an individual circuit, select the delete button next to the Global Reach name to remove connectivity between them. Then select **Save** to complete the operation.

 **Private peering** X

TestERCircuit

Peer ASN * ⓘ
65020 ✓

Subnets
 Both
 IPv4
 IPv6

IPv4 Primary subnet * ⓘ
192.168.20.16/30 ✓

IPv4 Secondary subnet * ⓘ
192.168.20.20/30 ✓

Enable IPv4 Peering ⓘ

VLAN ID * ⓘ
200 ✓

Shared key

Add Global Reach

Global Reach name	ExpressRoute Circuit name ⓘ	IPv4 Subnet ⓘ	IPv6 Subnet ⓘ
TestERCircuit1-TestE...	ASH-Cust20-ER (AS...)	192.168.11.24/29	

Save Cancel

Improve data path performance between networks with ExpressRoute FastPath

200 XP

- 10 minutes

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.

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.

Gateway requirements for ExpressRoute FastPath

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

- Ultra-Performance
- ErGw3AZ

Important

If you plan to use FastPath with IPv6-based private peering over ExpressRoute, make sure to select ErGw3AZ for SKU. Note that this is only available for circuits using ExpressRoute Direct.

Limitations

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

- UDR on the gateway subnet: This UDR has no impact on the network traffic that FastPath sends directly from your on-premises network to the virtual machines in Azure virtual network.
- 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.

Configure ExpressRoute FastPath

To enable FastPath, connect a virtual network to an ExpressRoute circuit using the Azure portal.

This section shows you how to 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 be part of another subscription.

Prerequisites

- Review the 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.
- Ensure that Azure private peering gets configured and establishes BGP peering between your network and Microsoft for end-to-end connectivity.
- Ensure that you have a virtual network and a virtual network gateway created and fully provisioned. 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 16 ExpressRoute circuits. Use the following process 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 virtual networks outside of the geopolitical region of the ExpressRoute circuit. The premium add-on will also allow you to connect more than 10 virtual networks to your ExpressRoute circuit depending on the bandwidth chosen.
- To create the connection from the ExpressRoute circuit to the target ExpressRoute virtual network gateway, the number of address spaces advertised from the local or peered virtual networks needs to be equal to or less than **200**. Once the

connection has been successfully created, you can add additional address spaces, up to 1,000, to the local or peered virtual networks.

Connect a VNet to a circuit - same subscription

Note

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

1. To create a connection Ensure that your ExpressRoute circuit and Azure private peering have been configured successfully. Your ExpressRoute circuit should look like the following image:

The screenshot shows the Azure portal interface for managing an ExpressRoute circuit named "TestERCircuit". The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Connections, Authorizations, Peerings, Properties, Locks, Monitoring, and Metrics. The "Overview" link is currently selected. The main content area displays the "Essentials" section with the following details:

Resource group (change)	Provider
ExpressRouteResourceGroup	Equinix
Circuit status	Provider status
Enabled	Provisioned
Location	Peering location
West US 2	Seattle
Subscription (change)	Bandwidth
Azure Subscription	50 Mbps
Subscription ID	Service key
	01234567-abcd-ef01-2345-6789abcdef01

Below the essentials section is a "Tags" section with a link to "Click here to add tags". The "Peerings" section contains a table:

Type	Status	Primary subnet	Secondary subnet	Last modified by
Azure public	Not provisioned	-	-	-
Microsoft	Not provisioned	-	-	-
Azure private	Provisioned	> One subnet configured	One subnet configured	Customer

2. You can now start provisioning a connection to link your virtual network gateway to your ExpressRoute circuit. Select **Connection** > **Add** to open the **Add connection** page.

TestERCircuit | Connections

ExpressRoute circuit

Search (Ctrl+/) < **+ Add** Delete Refresh

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

Settings Configuration **Connections** Authorizations

Search connections

Name ↑↓	Status ↑↓	Connection type ↑↓	Peer ↑↓
No results.			

3. Enter a name for the connection and then select **Next: Settings >**.

Create connection

Basics Settings Tags Review + create

Create a secure connection to your virtual network by using VPN Gateway or ExpressRoute. [Learn more about VPN Gateway](#) [Learn more about ExpressRoute](#)

Project details

Subscription: Azure Subscription

Resource group: ExpressRouteResourceGroup [Create new](#)

Instance details

Connection type: ExpressRoute

Name: ER-VNet-Connection

Region: (US) West US 2

Review + create < Previous **Next : Settings >** Download a template for automation

4. Select the gateway that belongs to the virtual network that you want to link to the circuit and select **Review + create**. Then select **Create** after validation completes.

Create connection ⊕

Basics **Settings** Tags Review + create

Virtual network gateway

To use a virtual network with a connection, it must be associated to a virtual network gateway.

Virtual network gateway * ⓘ

ERGW

ExpressRoute circuit ⓘ

TestERCircuit

Redeem authorization ⓘ



Routing weight *

0

Review + create

< Previous

Next : Tags >

[Download a template for automation](#)

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

The screenshot shows the Azure portal interface for managing ExpressRoute circuit connections. The main title is 'TestERCircuit | Connections'. On the left, there's a sidebar with various navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration, Connections (which is highlighted), and Authorizations. The main content area has a search bar 'Search connections' and a table with columns: Name ↑↓, Status ↑↓, Connection type ↑↓, and Peer ↑↓. A single row is listed: 'ER-VNet-Connection' with status 'Succeeded', connection type 'ExpressRoute', and peer 'ERGW'. The entire row for 'ER-VNet-Connection' is highlighted with a red border.

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

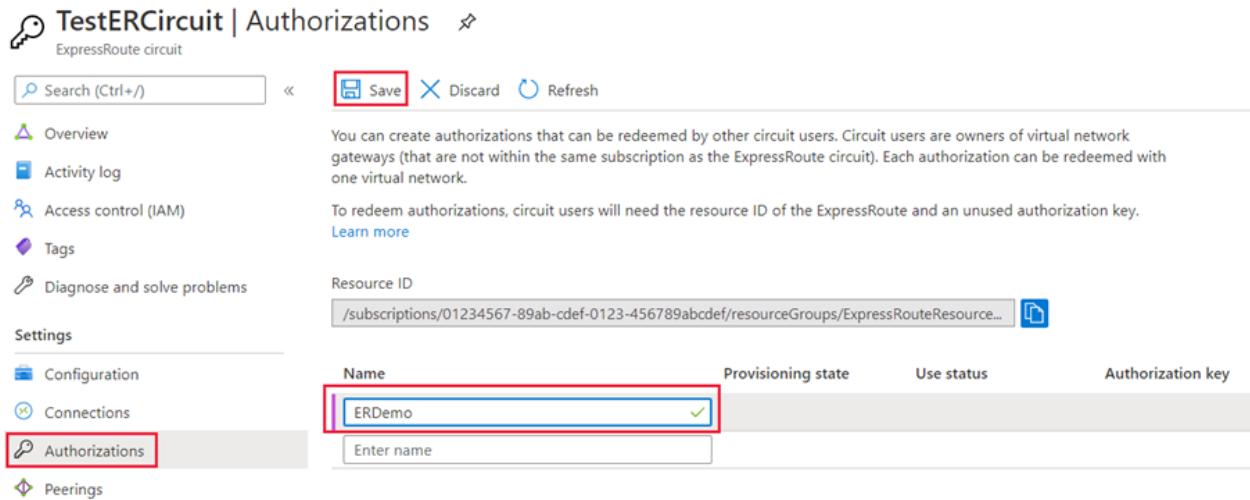
To create a connection authorization

The circuit owner creates an authorization, which creates an authorization key to 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, select **Authorizations** and then type a **name** for the authorization and select **Save**.

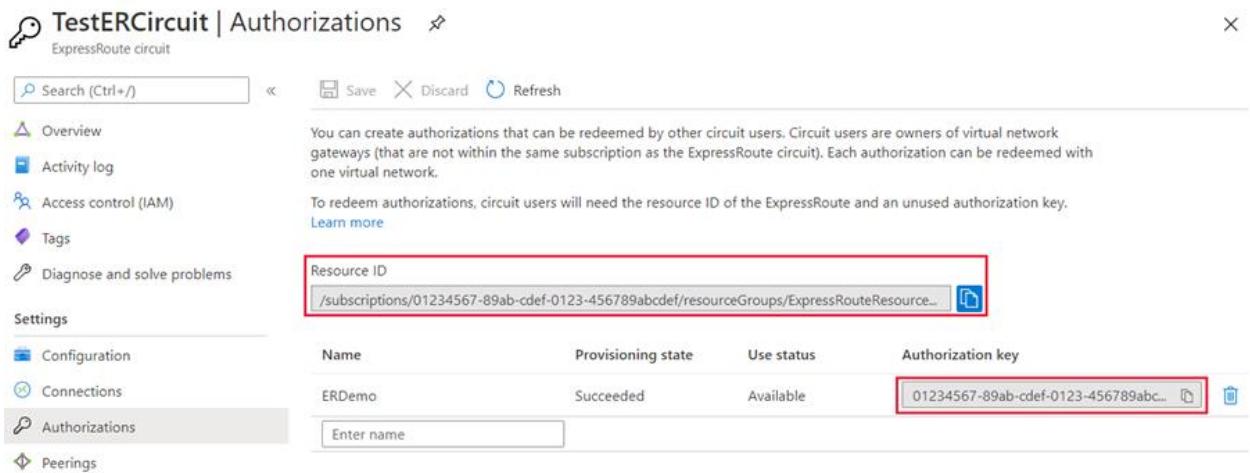


You can create authorizations that can be redeemed by other circuit users. Circuit users are owners of virtual network gateways (that are not within the same subscription as the ExpressRoute circuit). Each authorization can be redeemed with one virtual network.

To redeem authorizations, circuit users will need the resource ID of the ExpressRoute and an unused authorization key. [Learn more](#)

Name	Provisioning state	Use status	Authorization key
ERDemo	Succeeded	Available	01234567-89ab-cdef-0123-456789abcdef
<input type="text" value="Enter name"/>			

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



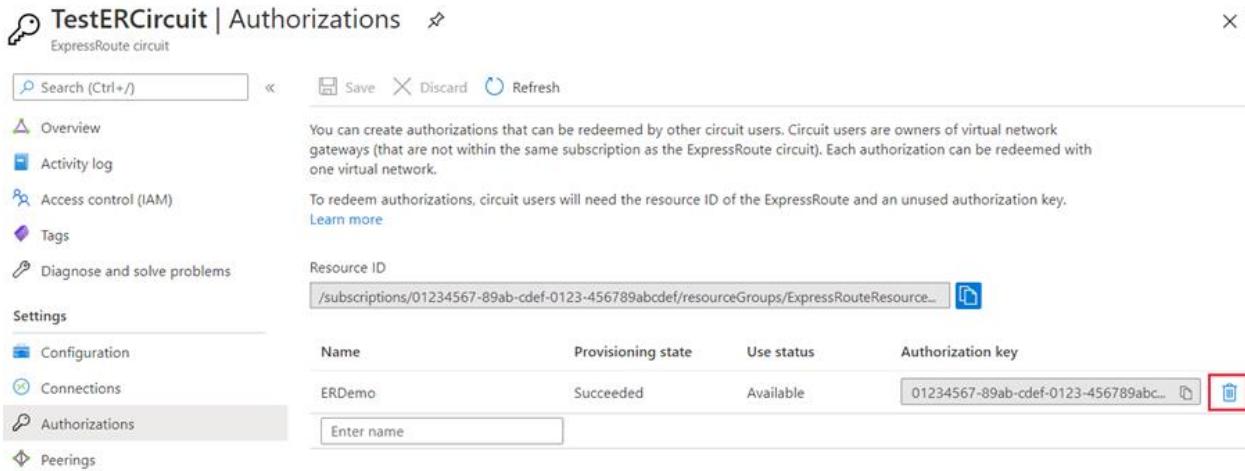
You can create authorizations that can be redeemed by other circuit users. Circuit users are owners of virtual network gateways (that are not within the same subscription as the ExpressRoute circuit). Each authorization can be redeemed with one virtual network.

To redeem authorizations, circuit users will need the resource ID of the ExpressRoute and an unused authorization key. [Learn more](#)

Name	Provisioning state	Use status	Authorization key
ERDemo	Succeeded	Available	01234567-89ab-cdef-0123-456789abcdef
<input type="text" value="Enter name"/>			

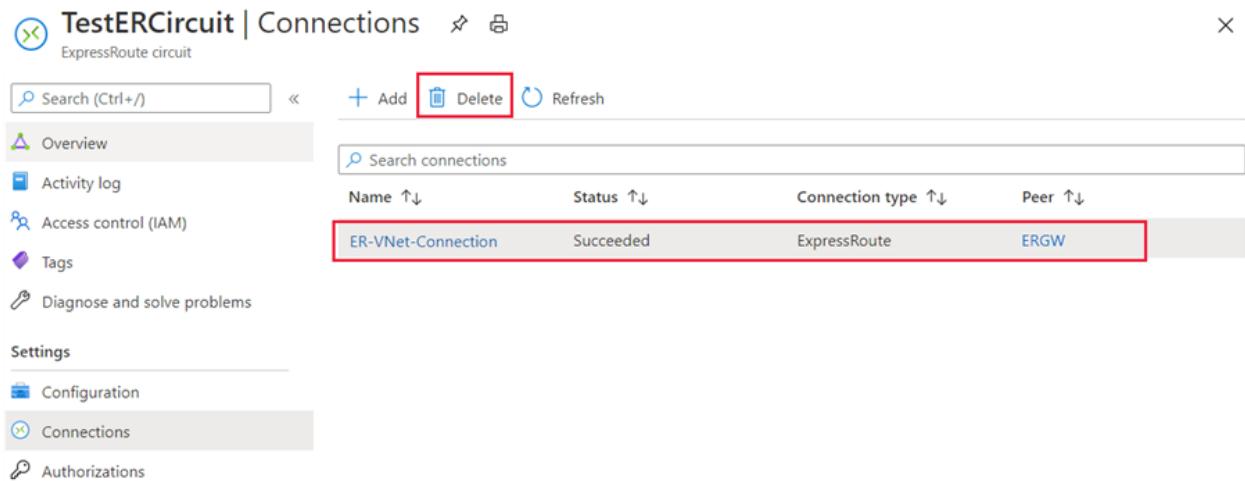
- To delete a connection authorization

You can delete a connection by selecting the Delete icon for the authorization key for your connection.

Screenshot of the 'Authorizations' blade in the Azure portal for the 'TestERCircuit' ExpressRoute circuit. The blade shows a summary of authorizations and a table where a new authorization named 'ERDemo' is listed. The 'Authorization key' column contains the value '01234567-89ab-cdef-0123-456789abcdef...' with a red box highlighting the copy icon.

Name	Provisioning state	Use status	Authorization key
ERDemo	Succeeded	Available	01234567-89ab-cdef-0123-456789abcdef...

If you want to delete the connection but retain the authorization key, you can delete the connection from the connection page of the circuit.

Screenshot of the 'Connections' blade in the Azure portal for the 'TestERCircuit' ExpressRoute circuit. The blade lists a single connection named 'ER-VNet-Connection' with a status of 'Succeeded'. The 'Delete' button for this connection is highlighted with a red box.

Circuit user operations

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

To redeem a connection authorization

1. Select the + Create a resource button. Search for Connection and select Create.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons and a 'Create a resource' button highlighted with a red box. The main content area is titled 'Connection' with a Microsoft logo. It features a large blue 'Create' button at the top right, also highlighted with a red box. Below the button, there's a section titled 'Azure benefit eligible'. The 'Overview' tab is selected, showing a brief description of a VPN connection. Other tabs include 'Plans', 'Usage Information + Support'. At the bottom, there's a section titled 'More offers from Microsoft' with four items: 'Workspace', 'Wire Data 2.0', 'Microsoft HPC Pack 2012 R2', and 'Windows Server 2019 Datacenter (zh-cn)'. Each item has a small icon, a title, a company name, and a brief description.

2. Make sure the Connection type is set to **ExpressRoute**. Select the Resource group and Location, then select **OK** in the Basics page.

Note

The location must match the virtual network gateway location you are creating the connection for.

Basics

Connection type * ⓘ

ExpressRoute

Subscription *

Azure Subscription

Resource group * ⓘ

ExpressRouteResourceGroup

[Create new](#)

Location *

West US 2

OK

3. In the **Settings** page, Select the Virtual network gateway and check the **Redeem authorization** check box. Enter the Authorization key and the Peer circuit URI and give the connection a name. Select **OK**.

Note

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).

Settings

*Virtual network gateway ⓘ



ERGW

Redeem authorization ⓘ

Authorization key *

01234567-89ab-cdef-0123-456789abcdef ✓

Peer circuit URI *

/subscriptions/01234567-89ab-cdef-012... ✓

Connection name *

ER-VNet-Connection ✓

Routing weight *

0

OK

4. Review the information in the **Summary** page and select **OK**.

Summary

Basics

Connection type	ExpressRoute
Subscription	Azure Subscription
Resource Group	ExpressRouteResourceGroup
Location	East US

Settings

Virtual network gateway	ERGW
Redeem ExpressRoute authorization	Yes
Authorization key	01234567-89ab-cdef-0123-456789abcdef
Peer circuit URI	/subscriptions/01234567-89ab-cdef-0123-45678...
Connection name	ER-VNet-Connection

OK

Clean up resources

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

The screenshot shows the Azure portal interface for managing an ExpressRoute connection named 'ER-VNet-Connection'. The top navigation bar includes 'Subscriptions', 'Groups', 'Storage', 'Compute', 'Networking', 'Databases', 'Machine Learning', and 'Marketplace'. On the left, a sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings', 'Configuration', 'Properties', 'Locks', and 'Monitoring'. The main content area displays 'Essentials' information: Resource group (change) to 'ExpressRouteResourceGroup', Virtual network to 'TestVNet'; Status to 'Succeeded', Virtual network gateway to 'ERGW'; Location to 'West US 2', Circuit to 'TestERCircuit'; Subscription (change) to 'Azure Subscription', Routing weight to '0'; and Subscription ID. At the bottom, there's a 'Tags (change)' section with a link 'Click here to add tags'. In the top right corner, there are 'Move' and 'Delete' buttons, with the 'Delete' button being highlighted by a red box.

Troubleshoot ExpressRoute connection issues

200 XP

- 6 minutes

As an Azure network engineer supporting an ExpressRoute deployment, you will have to diagnose and resolve any ExpressRoute connection issues that arise.

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.

Verify circuit provisioning and state through the Azure portal

Provisioning an ExpressRoute circuit establishes a redundant Layer 2 connections between CEs/PE-MSEEs (2)/(4) and MSEEs (5).

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.

In the Azure portal, open the ExpressRoute circuit blade. In the section of the blade, the ExpressRoute essentials are listed as shown in the following screenshot:

The screenshot shows the Azure portal interface for managing an ExpressRoute circuit. At the top, the circuit name is 'ER-Demo-Ckt-SV' and it is identified as an 'ExpressRoute circuit'. Below the title, there are two buttons: 'Settings' and 'Delete'. The main content area is divided into sections: 'Essentials' and 'Peerings'. The 'Essentials' section contains various configuration details, including:

Resource group USWest-ER-Demo-RG	Provider Equinix
Circuit status Enabled	Provider status Provisioned
Location West US	Peering location Silicon Valley
Subscription name ExpressRoute-Demo	Bandwidth 200 Mbps
Subscription ID	Service key

An 'All settings' link is located at the bottom right of the 'Essentials' section. The 'Peerings' section lists three entries:

TYPE	STATUS	PRIMARY SUBNET	SECONDARY SUBNET	...
Azure private	Disabled	-	-	...
Azure public	Disabled	-	-	...
Microsoft	Disabled	-	-	...

An 'Add a section' button is located at the bottom of the 'Peerings' 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.

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).

Note

In IPVPN connectivity model, service providers handle the responsibility of configuring the peering (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 peering would be listed as shown in the following screenshot:

Type	Status	Primary subnet	Secondary subnet	Last modified by	...
Azure private	Provisioned	192.168.20.16/30	192.168.20.20/30	Customer	...
Azure public	Not provisioned	-	-	-	...
Microsoft	Not provisioned	-	-	-	...

In the preceding example, as noted Azure private peering is provisioned, whereas Azure public and Microsoft peering 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 peering 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 map 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 Address Resolution Protocol (ARP)

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. ARP tables can help validate layer 2 configuration and troubleshooting basic layer 2 connectivity issues.

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.

ARP table when Microsoft side has problems

- You won't 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're still experiencing issues.

ExpressRoute monitoring tools

ExpressRoute uses Network insights to provide a detailed topology mapping of all ExpressRoute components (peerings, connections, gateways) in relation with one another. Network insights for ExpressRoute also have preloaded metrics dashboard for availability, throughput, packet drops, and gateway metrics.

You can analyze metrics for Azure ExpressRoute with metrics from other Azure services using metrics explorer by opening Metrics from the Azure Monitor menu.

- 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.

Summary and resources

Completed 100 XP

- 1 minute

Now that you have reviewed this module, you should be able to:

- Learn about Express Route, its use cases and implementation
- Understand Express Route configuration and when to choose the appropriate SKU
- Learn about ExpressRoute Global Reach and connecting branch offices
- Explore Express Route FastPath to improve performance
- Understand Express Route peering, Private and Microsoft peering

Load balance non-HTTP(S) traffic in Azure

Explore load balancing

200 XP

- 6 minutes

The term load balancing refers to the even distribution of workloads (that is, incoming network traffic), across a group of backend computing resources or servers. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overloading any single resource. It can also improve availability by sharing a workload across redundant computing resources.

Load Balancing options for Azure

Azure provides various load balancing services that you can use to distribute your workloads across multiple computing resources, but the following are the main services:

- **Azure Load Balancer** - high-performance, ultra-low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle millions of requests per second while ensuring your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.
- **Traffic Manager** - DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness. Because Traffic Manager is a DNS-based load-balancing service, it load-balances only at the domain level. For that reason, it can't fail over as quickly as Front Door, because of common challenges around DNS caching and systems not honoring DNS time-to-live values (TTLs).
- **Azure Application Gateway** - provides application delivery controller (ADC) as a service, offering various Layer 7 load-balancing capabilities. Use it to optimize web farm productivity by offloading CPU-intensive SSL termination to the gateway.
- **Azure Front Door** - application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like SSL offload, path-based routing, fast failover, caching, etc. to improve performance and high-availability of your applications.

Categorizing load balancing services

The above load balancing services can be categorized in two ways: global versus regional, and HTTP(S) versus non-HTTP(S).

Global versus regional

Global load-balancing services distribute traffic across regional backends, clouds, or hybrid on-premises services. These services route end-user traffic to the closest available backend. They also react to changes in service reliability or performance, in order to maximize availability and performance. You can think of them as systems that load balance between application stamps, endpoints, or scale-units hosted across different regions/geographies.

In contrast, **Regional** load-balancing services distribute traffic within virtual networks across virtual machines (VMs) or zonal and zone-redundant service endpoints within a region. You can think of them as systems that load balance between VMs, containers, or clusters within a region in a virtual network.

HTTP(S) versus non-HTTP(S)

HTTP(S) load-balancing services are Layer 7 load balancers that only accept HTTP(S) traffic. They're intended for web applications or other HTTP(S) endpoints. They include features such as SSL offload, web application firewall, path-based load balancing, and session affinity.

In contrast, **non-HTTP(S)** load-balancing services can handle non-HTTP(S) traffic and are recommended for non-web workloads.

The table below summarizes these categorizations for each Azure load balancing service.

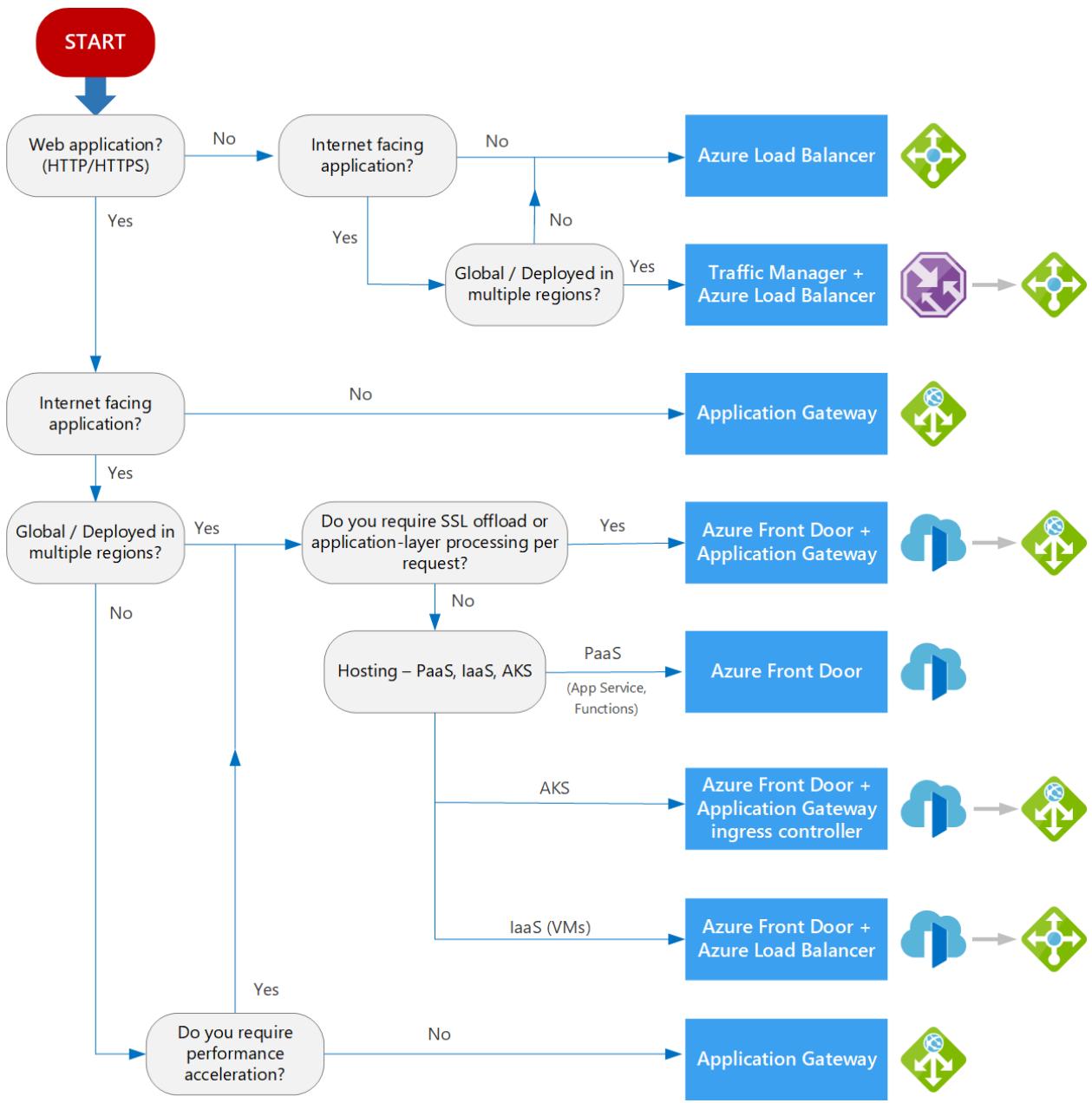
Service	Global/regional	Recommended traffic
Azure Front Door	Global	HTTP(S)
Traffic Manager	Global	non-HTTP(S)
Application Gateway	Regional	HTTP(S)
Azure Load Balancer	Regional or Global	non-HTTP(S)

Choosing a load balancing option for Azure

When choosing an appropriate load balancing option, there are some key factors to consider:

- **Type of traffic** - is it for a web application? Is it a public-facing or private application?
- **Scope** - do you need to load balance virtual machines and containers within a virtual network, or load balance across regions, or both? (see 'Global versus regional' above)
- **Availability** - what is the Service Level Agreement (SLA) for the service?
- **Cost** - In addition to the cost of the actual service itself, consider the operational cost to manage and maintain a solution built on that service. See [Load balancing pricing](#).
- **Features and limitations** - what features and benefits does each service provide, and what are its limitations? See [Load balancer limits](#).

The flowchart below will help you to select the most appropriate load-balancing solution for your application, by guiding you through a set of key decision criteria in order to reach a recommendation.



As every application will have its own unique requirements, you should only use this flowchart and the suggested recommendation as a starting point, and then perform a more detailed evaluation yourself in order to select the best option for your environment.

If your application consists of multiple workloads, evaluate each workload separately. A complete solution may incorporate two or more load-balancing solutions.

Selecting a load balancing solution by using the Azure portal

You can use the **Azure Load Balancing** page in the Azure portal to help you guide to the right load-balancing solution for your business need. Azure Load Balancing includes the decision-making queries described in the workflow diagram above.

- Sign into the Azure portal at <https://portal.azure.com>.
- In the search box at the top of the page, type load balancing. When **Load balancing - help me choose (Preview)** appears in the search results, select it.

The screenshot shows the Azure portal search interface. The search bar at the top contains the text "load balancing". Below the search bar, the "Services" section lists several Azure services, with "Load balancing - help me choose (Preview)" highlighted by a red box. The "Marketplace" section lists external load balancing solutions like Citrix ADC and Pulse Virtual Traffic Manager. The "Documentation" section provides links to Azure load-balancing documentation. The "Resource Groups" section indicates no results were found. At the bottom, there are suggestions for further search: "Try searching in Activity Log" and "Try searching in Azure Active Directory".

- Answer the Yes or No questions on this page to get a recommended solution. Note that the final recommended solution may be a combination of multiple load balancing services.

Microsoft Azure

Home > Load balancing - help me choose (Preview)

Search (Ctrl+ /) Help me choose Service comparison Tutorial

Load Balancing Services

- Application Gateway
- Front Door
- Load Balancer
- Traffic Manager

Does your application use HTTP/HTTPS?

Yes No

Application Gateway

- Internal and public configurations
- Regional layer 7 load balancer
- SSL/TLS offloading

Create Show more

Front Door

- Global layer 7 load balancer
- Site acceleration
- SSL/TLS offloading

Create Show more

Load Balancer

- Layer 4 load balancing
- Internal and public configurations
- High availability across zones

Create Show more

Traffic Manager

- DNS-based traffic load balancer
- Global across Azure regions
- High availability

Create Show more

- Depending on what answers you give, the list of potential load balancing services change.

Microsoft Azure

Home > Load balancing - help me choose (Preview)

Search (Ctrl+ /) Help me choose Service comparison Tutorial

Load Balancing Services

- Application Gateway
- Front Door
- Load Balancer
- Traffic Manager

Is your application public (internet facing)?

Yes No

← Previous Is this helpful? Start over

Load Balancer

- Layer 4 load balancing
- Internal and public configurations
- High availability across zones

Create Show more

Traffic Manager

- DNS-based traffic load balancer
- Global across Azure regions
- High availability

Create Show more

- Optionally, you can also select the **Service comparison** or **Tutorial** tabs for more information and training on the different load balancing services.

Now let's look at each of the main Azure load balancing services in more detail.

Design and implement Azure load balancer using the Azure portal

200 XP

- 17 minutes

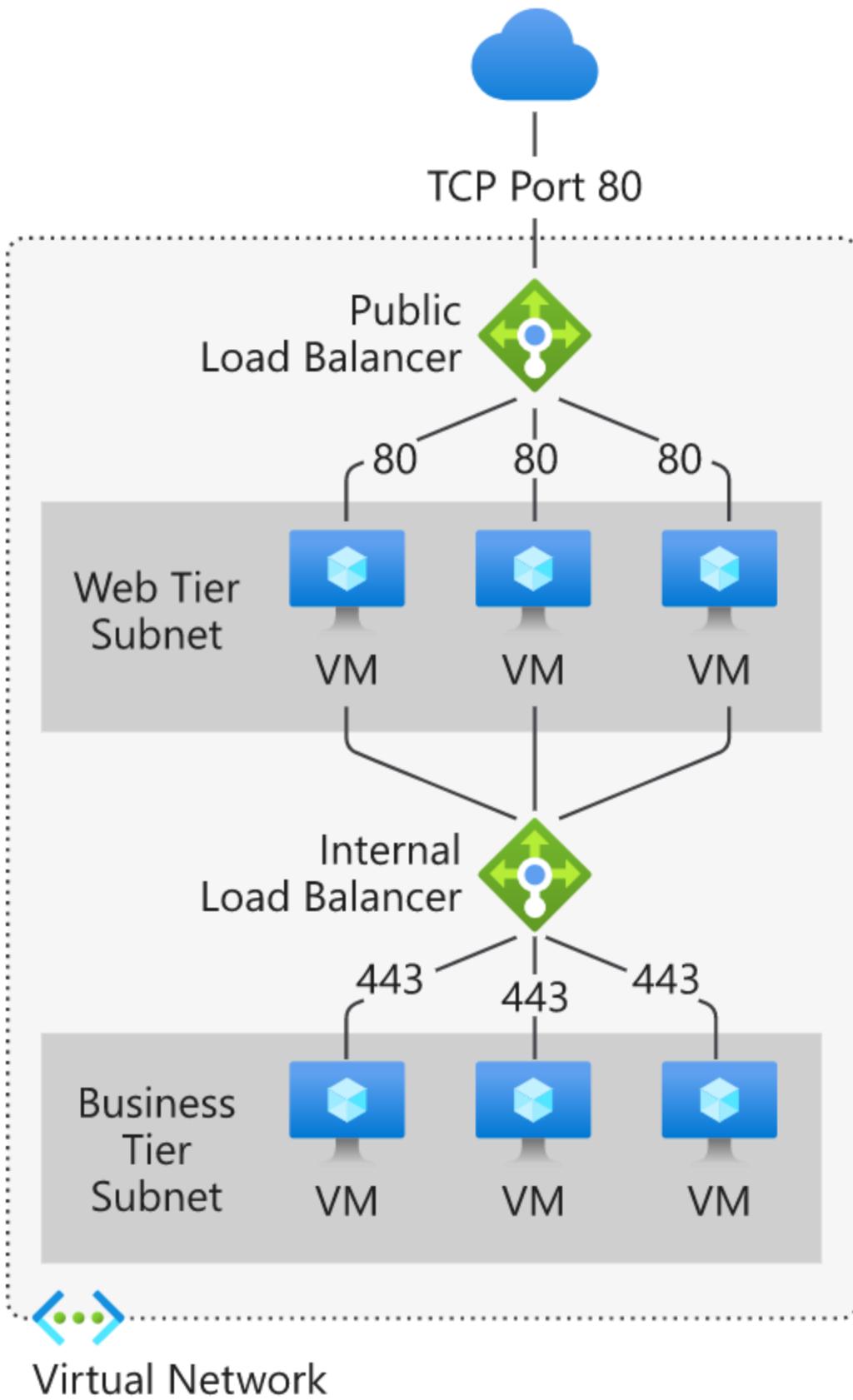
Azure Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model. It's the single point of contact for clients. Azure Load Balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. These flows are according to configured load-balancing rules and health probes. The backend pool instances can be Azure Virtual Machines or instances in a virtual machine scale set.

Choosing a load balancer type

Load balancers can be public (also known as external) or internal (also known as private).

A **public load balancer** can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. External load balancers are used to distribute client traffic from the internet across your VMs. That internet traffic might come from web browsers, module apps, or other sources.

An **internal load balancer** is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic from internal Azure resources to other Azure resources inside a virtual network. A load balancer frontend can also be accessed from an on-premises network in a hybrid scenario.



Azure load balancer and availability zones

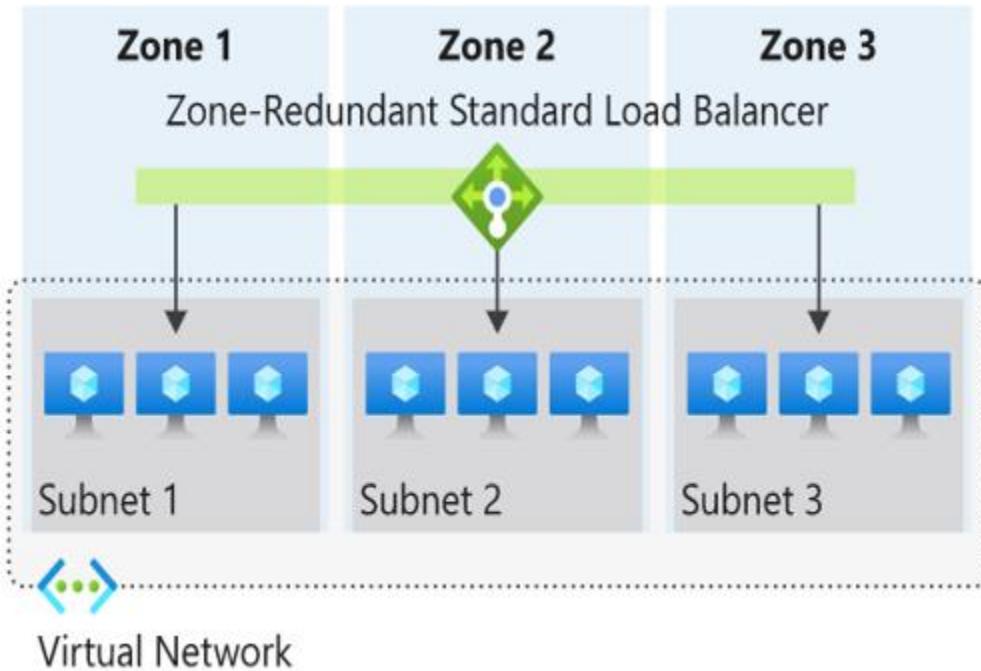
Azure services that support availability zones fall into three categories:

- Zonal services: Resources can be pinned to a specific zone. For example, virtual machines, managed disks, or standard IP addresses can be pinned to a specific zone, which allows for increased resilience by having one or more instances of resources spread across zones.
- Zone-redundant services: Resources are replicated or distributed across zones automatically. Azure replicates the data across three zones so that a zone failure doesn't impact its availability.
- Non-regional services: Services are always available from Azure geographies and are resilient to zone-wide outages and region-wide outages.

Azure Load Balancer supports availability zones scenarios. You can use Standard Load Balancer to increase availability throughout your scenario by aligning resources with, and distribution across zones. Review this document to understand these concepts and fundamental scenario design guidance.

A Load Balancer can either be zone redundant, zonal, or non-zonal. To configure the zone related properties (mentioned above) for your load balancer, select the appropriate type of frontend needed.

Zone redundant



In a region with Availability Zones, a Standard Load Balancer can be zone-redundant. This traffic is served by a single IP address.

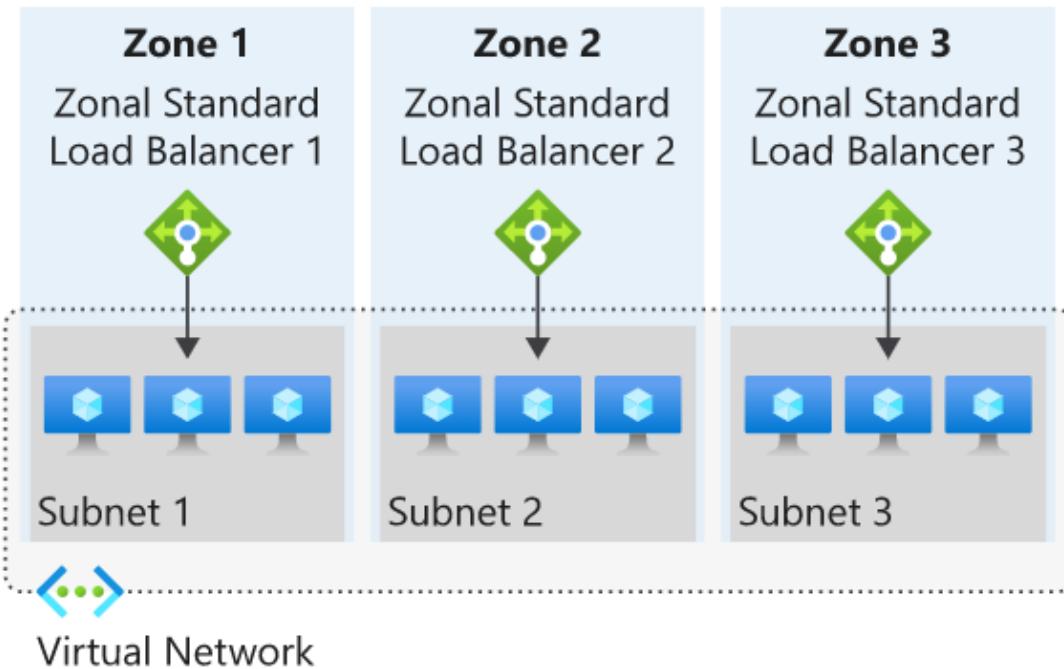
A single frontend IP address survives zone failure. The frontend IP may be used to reach all (non-impacted) backend pool members no matter the zone. One or more availability zones can fail and the data path survives as long as one zone in the region remains healthy.

The frontend's IP address is served simultaneously by multiple independent infrastructure deployments in multiple availability zones. Any retries or reestablishment succeed in other zones not affected by the zone failure.

Zonal

You can choose to have a frontend guaranteed to a single zone, which is known as a zonal. This scenario means any inbound or outbound flow is served by a single zone in a region. Your frontend shares fate with the health of the zone. The data path is unaffected by failures in zones other than where it was guaranteed. You can use zonal frontends to expose an IP address per Availability Zone.

Additionally, the use of zonal frontends directly for load balanced endpoints within each zone is supported. You can use this configuration to expose per zone load-balanced endpoints to individually monitor each zone. For public endpoints, you can integrate them with a DNS load-balancing product like Traffic Manager and use a single DNS name.



For a public load balancer frontend, you add a `zones` parameter to the public IP. This public IP is referenced by the frontend IP configuration used by the respective rule.

For an internal load balancer frontend, add a `zones` parameter to the internal load balancer frontend IP configuration. A zonal frontend guarantees an IP address in a subnet to a specific zone.

Selecting an Azure load balancer SKU

Two SKUs are available when you create a load balancer in Azure: Basic load balancers and Standard load balancers. These SKUs differ in terms of their scenario scope and scale, features, and cost. Any scenario that is possible with the Basic load balancer can also be created with the Standard load balancer.

To compare and understand the differences, review the table below.

Features	Standard Load Balancer	Basic Load Balancer
Backend pool size	Supports up to 1000 instances.	Supports up to 300 instances.
Backend pool endpoints	Any virtual machines or virtual machine scale sets in a single virtual network.	Virtual machines in a single availability set or virtual machine scale set.
Health probes	TCP, HTTP, HTTPS	TCP, HTTP
Health probe down behavior	TCP connections stay alive on an instance probe down and on all probes down.	TCP connections stay alive on an instance probe down. All TCP connections end when all probes are down.
Availability Zones	Zone-redundant and zonal frontends for inbound and outbound traffic.	Not available
Diagnostics	Azure Monitor multi-dimensional metrics	Azure Monitor logs
HA Ports	Available for Internal Load Balancer	Not available
Secure by default	Closed to inbound flows unless allowed by a network security group. Internal traffic from the virtual network to the internal load balancer is allowed.	Open by default. Network security group optional.
Outbound Rules	Declarative outbound NAT configuration	Not available
TCP Reset on Idle	Available on any rule	Not available
Multiple front ends	Inbound and outbound	Inbound only
Management Operations	Most operations < 30 seconds	60-90+ seconds typical
SLA	99.99%	Not available

Microsoft recommends Standard load balancer. Standalone VMs, availability sets, and virtual machine scale sets can be connected to only one SKU, never both. Load balancer and the public IP address SKU must match when you use them with public IP addresses.

SKUs aren't mutable; therefore, you cannot change the SKU of an existing resource.

Creating and configuring an Azure load balancer

There are several tasks you need to perform to successfully create and configure an Azure Load Balancer.

Create the load balancer

In this example, we're looking at the tasks required to create and configure a **Public** (external) **load balancer** in a **Standard SKU**. The first task is to create the load balancer itself. During the creation process, a frontend IP configuration is created and a public IP address is assigned. You can also add a backend pool, create a health probe, and add load balancing rules during the creation process, but we add these components later in the process.

From the Azure portal home page, navigate to the Global Search bar and search **Load Balancer** then select **Load balancers**.

Choose **+ Create** or **Create load balancer** to start the process.

On the **Create load balancer** page, you must supply the following required information on the **Basics** tab:

Setting	Value
Subscription	Select the Azure subscription that you want to create your new load balancer resource in.
Resource group	Here you can select an existing resource group or create a new one.
Name	Provide a unique name for the instance.
Region	Select the region where the virtual machines were created.
Type	This is where you select whether your load balancer is going to be Internal (private) or Public (external). If you choose Internal , you need to specify a virtual network and IP address assignment, but if you choose Public , you need to specify several Public IP address details.
SKU	Here you can select either the Standard SKU or the Basic SKU (for production workloads you should choose Standard , but for testing and evaluation and training purposes, you could choose Basic , but you won't get all the possible load balancer features). Depending on which SKU you select here, the remaining configuration options differ slightly.
Tier	This is where you select whether your load balancer is balancing within a region (Regional) or across regions (Global) - If you select the Basic SKU above, this setting is greyed out.

Create load balancer

Basics Frontend IP configuration Backend pools Inbound rules Outbound rules Tags Review + create

Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses. [Learn more](#).

Project details

Subscription * Contoso Subscription

Resource group * Contoso-ResourceGroup [Create new](#)

Instance details

Name * Public-Std-LB

Region * West US 3

SKU * Standard
 Gateway
 Basic

Type * Public
 Internal

Tier * Regional
 Global

[Review + create](#) < Previous [Next : Frontend IP configuration >](#) [Download a template for automation](#) [Give feedback](#)

After you select **Next: Frontend IP configuration**, select **+ Add frontend IP address** to add a public IP address for your public-facing front-end. You add a name for the frontend IP configuration, choose IP version and IP type, then add a Public IP Address. You can create a new public IP address for your public-facing front-end, or use an existing one. When creating a new public IP address, you specify the name and you also specify a name for your public IP address, and whether to use a dynamic or statically assigned IP address. You can optionally also assign an IPv6 address to your load balancer in addition to the default IPv4 one.

The screenshot shows the Azure portal interface for creating a load balancer. On the left, the 'Create load balancer' wizard is open, with the 'Frontend IP configuration' step selected. A red box highlights the 'Add a frontend IP configuration' button. Below it, there's a table with columns for 'Name' and 'IP address'. A red box highlights the 'Name' column for the first row, which contains 'Contoso-Frontend-IP-Config'. On the right, a modal window titled 'Add frontend IP configuration' is displayed. It has fields for 'Name *' (containing 'Contoso-Frontend-IP-Config'), 'IP version' (set to IPv4), 'IP type' (set to IP address), 'Public IP address *' (set to '(New) Contoso-Public-IP'), and 'Gateway Load balancer' (set to 'None'). A red box highlights the 'Add' button at the bottom right of the modal.

Once completed with the frontend IP configuration, select **Review + Create**, where the configuration settings for the new load balancer resource will be validated, and then you can select **Create** to deploy the resource.

Create load balancer

...

Validation passed

Basics Frontend IP configuration Backend pools Inbound rules Outbound rules Tags Review + create

Basics

Subscription	Contoso Subscription
Resource group	Contoso-ResourceGroup
Name	Public-Std-LB
Region	West US 3
SKU	Standard
Tier	Regional
Type	Public

Frontend IP configuration

Frontend IP configuration name	Contoso-Frontend-IP-Config
Frontend IP configuration IP address	To be created

Backend pools

None

Inbound rules

None

Outbound rules

None

[Create](#)

< Previous

Next >

[Download a template for automation](#)

[Give feedback](#)

The resource starts to be deployed.

 Microsoft.LoadBalancer-20231206111324 | Overview ⚡ ...

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview Inputs Outputs Template

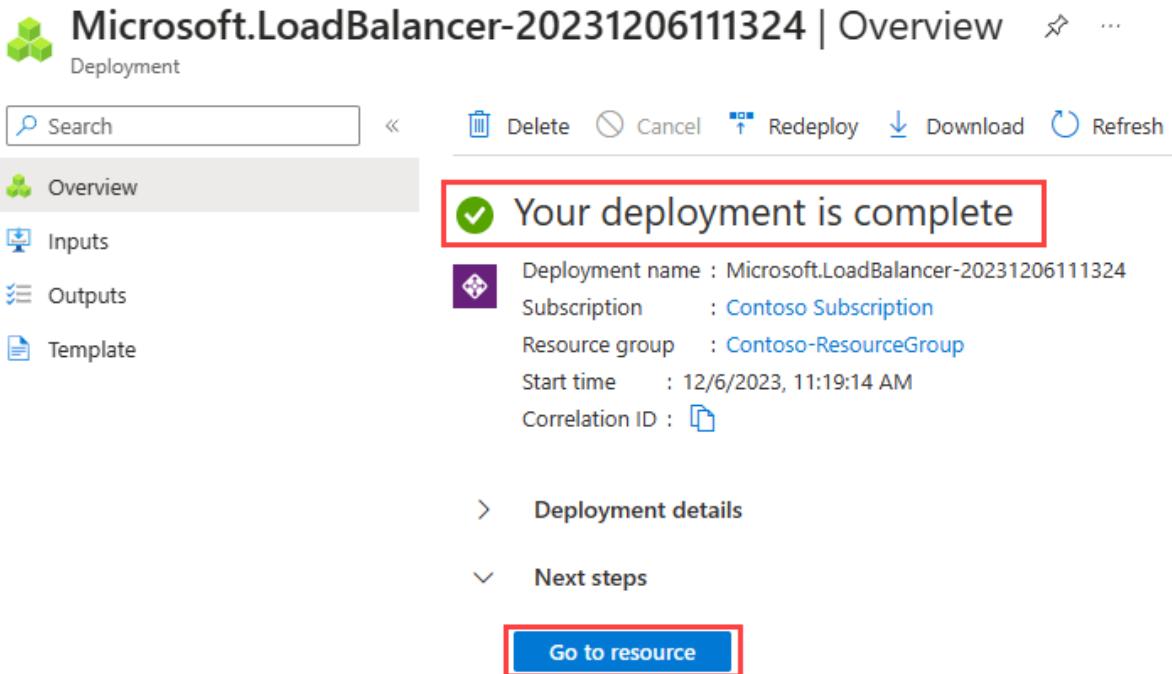
 Your deployment is complete

Deployment name : Microsoft.LoadBalancer-20231206111324
Subscription : Contoso Subscription
Resource group : Contoso-ResourceGroup
Start time : 12/6/2023, 11:19:14 AM
Correlation ID : 

> Deployment details

▽ Next steps

Go to resource



When it completes, you can select **Go to resource** to view the new load balancer resource in the portal.

Public-Std-LB Load balancer X

Search Move Delete Refresh Give feedback JSON View

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Frontend IP configuration

Backend pools

Health probes

Load balancing rules

Inbound NAT rules

Outbound rules

Properties

Locks

Monitoring

Insights

Diagnostic settings

Logs

Alerts

Metrics

Automation

CLI / PS

Essentials

Resource group ([move](#)) [Contoso-ResourceGroup](#)

Location West US 3

Subscription ([move](#)) [Contoso Subscription](#)

Subscription ID 0000000-0000-0000-0000-000000000000

SKU Standard

Tags ([edit](#)) [Add tags](#)

See more

Configure high availability and scalability for your applications

Create highly-available and scalable applications in minutes by using built-in load balancing for cloud services and virtual machines. Azure Load Balancer supports TCP/UDP-based protocols and protocols used for real-time voice and video messaging applications. [Learn more ↗](#)

 **Balance IPv4 and IPv6 addresses**
Native dual-stack endpoints help meet regulatory requirements and address the fast-growing number of devices in mobile and IoT. [Learn more ↗](#)

 **Build highly reliable applications**
Load Balancer improves application uptime by routing traffic to healthy nodes. [Learn more ↗](#)

View frontend IP configuration **View health probes** **View load balancing rules**

Add a backend pool

The next task is to create a backend pool in the load balancer and then adds your virtual machines to it.

From the Load balancer Overview page for your load balancer, select **Backend pools** under Settings and select **+ Add**.

The screenshot shows the 'Backend pools' section of the Azure portal for a specific load balancer. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Frontend IP configuration, Backend pools (which is selected and highlighted with a red box), Health probes, and Load balancing rules. The main area displays a table with the following columns: Backend pool, Resource N..., IP address, Network int..., Availability ..., Rules count, and Resource St... A red box highlights the 'Add' button at the top of the table. A note below the table states: 'The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more](#)'.

You need to enter the following information on the **Add backend pool** page:

Expand table

Setting	Value
Name	Enter a unique name for the backend pool.
Virtual network	Select the name of the virtual network where the resources are located that you're adding to.
Backend Pool Configuration	Select whether you want to associate the backend pool using the NIC or IP address of a resource.

You could add existing virtual machines to the backend pool at this point, or you can create and add them later. You then select **Save** to add the backend pool.

Add backend pool

Public-Std-LB

Name *

Virtual network

Backend Pool Configuration NIC IP address

IP configurations

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

+ Add | X Remove

Resource Name	Resource group	Type	IP config...	IP Addr...	Availabi...

Save | Cancel | Give feedback

Add virtual machines to the backend pool

The next task is to add the virtual machines to the existing back-end pool.

On the **Backend pools** page, select the backend pool from the list.

Home > Contoso-ResourceGroup > Public-Std-LB

Public-Std-LB | Backend pools

Load balancer

Add Refresh

The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more](#).

Backend pool

Resource Name	IP address	Network int...	Availability ...	Ri
Contoso-bepool (1)				

You need to enter the following information to add the virtual machines to the backend pool:

Expand table

Setting	Value
Virtual network	Specify the name of the virtual network where the resources are located that you're adding
Backend Pool Configuration	Select whether you want to associate the backend pool using the NIC or IP address of a resource.

Then under the **IP configurations** section, select **+ Add**.

Home > Contoso-ResourceGroup > Public-Std-LB | Backend pools >

Contoso-bepool ...

Public-Std-LB

Name *

Virtual network

Backend Pool Configuration NIC IP address

IP configurations

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

Add

Resource Name	Resource group	Type	IP config...	IP Addr...	Availabi...
---------------	----------------	------	--------------	------------	-------------

Used by

The list of load balancing rules, inbound NAT rules, and outbound rules using this backend pool.

Name	Type
------	------

[Give feedback](#)

Select the virtual machines you want to add to the backend pool and select **Add**.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons like Home, Dashboard, All services, Favorites (with All resources selected), Resource groups, App Services, Function App, SQL databases, Azure Cosmos DB, Virtual machines, Load balancers, Storage accounts, Virtual networks, Microsoft Entra ID, Monitor, Advisor, Microsoft Defender for Cloud, Cost Management + Billing, and Help + support. The main area is titled "Add IP configurations to backend pool". It displays a table with columns: Resource N..., Resource ..., Type, IP config..., IP Address, Availability..., and Tags. Under the "Virtual machine (2)" heading, two rows are shown: TestVM1 and TestVM2. Both rows have checkboxes next to them, which are highlighted with a red box. The IP address for TestVM1 is 10.0.0.4 and for TestVM2 is 10.0.0.5. The "Add" button at the bottom left is also highlighted with a red box.

Then select **Save** to add them to the backend pool.

Contoso-bepool

X

Public-Std-LB

Name *	Contoso-bepool
Virtual network ⓘ	CoreServicesVNet (Contoso-ResourceGroup)
Backend Pool Configuration	<input checked="" type="radio"/> NIC <input type="radio"/> IP address

IP configurations

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

+ Add

| Remove

Resource Name	Resource group	Type	IP config...	IP Addr...	Availabi...	
TestVM1	Contoso-ResourceGroup	Virtual mach	ipconfig1	10.0.0.4	CONTOSO-I	
TestVM2	Contoso-ResourceGroup	Virtual mach	ipconfig1	10.0.0.5	CONTOSO-I	

Used by

The list of load balancing rules, inbound NAT rules, and outbound rules using this backend pool.

Name

Type

Save

Cancel

Give feedback

Home > Contoso-ResourceGroup > Public-Std-LB

Public-Std-LB | Backend pools

Load balancer

» + Add ⏪ Refresh

The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more](#).

🔍 Add filter

Backend pool	Resource Name	IP address	Network int...	Availability ...	Rules count	Resource St...
Contoso-bepool (2)						
Contoso-bepool	TestVM2	10.0.0.5	testvm2183	-	0	Running
Contoso-bepool	TestVM1	10.0.0.4	testvm1588	-	0	Running

Add health probes

The next task is to create a health probe to monitor the virtual machines in the back-end pool.

On the **Backend pools** page of the load balancer, select **Health probes** under **Settings**, and then select **+**

Add.

Home > Public-Std-LB

Public-Std-LB | Health probes

Load balancer

🔍 Search + Add ⏪ Refresh ⏪ Give feedback

Type to start filtering ...

No results.

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Frontend IP configuration

Backend pools

Health probes

Load balancing rules

Inbound NAT rules

You need to enter the following information on the **Add health probe** page:

Expand table

Setting	Value
Name	Enter a unique name for the health probe.
Protocol	Select either TCP , HTTP , or HTTPS .
Port	Specify the destination port number for the health signal. The default is port 80 .
Interval (seconds)	Specify the interval time in seconds between probe attempts. The default is 5 seconds.

You then select **Save** to add the health probe.

Home > Public-Std-LB | Health probes >

Add health probe

Public-Std-LB X

ⓘ Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Name *	Contoso-HealthProbe
Protocol *	TCP
Port * ⓘ	80
Interval (seconds) * ⓘ	5
Used by * ⓘ	Not used

Save Cancel Give feedback

The screenshot shows the Azure portal interface for managing a load balancer. The top navigation bar includes 'Home', 'Load balancing | Load Balancer', and 'Public-Std-LB'. The main title is 'Public-Std-LB | Health probes'. On the left, a sidebar menu lists various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Frontend IP configuration, Backend pools, and Health probes (which is selected and highlighted in grey). The main content area displays a table of health probes. The table has columns for Name, Protocol, Port, Path, and Used By. One row is shown, with the entire row highlighted by a red box: 'Contoso-HealthProbe' (Protocol: Tcp, Port: 80). A delete icon is visible at the end of this row.

Add a load balancer rule

The last task is to create a load balancing rule for the load balancer. A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.

On the **Health probes** page of the load balancer, select **Load balancing rules** under **Settings**, and then select **+ Add**.

Home > Load balancing | Load Balancer > Public-Std-LB

Public-Std-LB | Load balancing rules

Load balancer

Search Add Refresh Delete

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

Frontend IP configuration Backend pools Health probes **Load balancing rules** Inbound NAT rules Outbound rules

A load balancer rule is used to define how incoming traffic is distributed to the all the instances within the backend pool. A load-balancing rule maps a given frontend IP configuration and port to multiple backend IP addresses and ports. An example would be a rule created on port 80 to load balance web traffic. [Learn more.](#)

Filter by name... Name ↑ Protocol ↑ Backend pool ↑ Health probe ↑

No results.

The screenshot shows the Azure portal interface for managing a load balancer named 'Public-Std-LB'. On the left, there's a navigation sidebar with links like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Frontend IP configuration', 'Backend pools', 'Health probes', and 'Load balancing rules'. The 'Load balancing rules' link is highlighted with a red box. On the right, the main content area has a heading 'Load balancing rules' and a sub-section titled 'A load balancer rule is used to define how incoming traffic is distributed to the all the instances within the backend pool...'. Below this is a search bar and a table header with columns: 'Name ↑', 'Protocol ↑', 'Backend pool ↑', and 'Health probe ↑'. A message 'No results.' is displayed below the table. At the top of the main content area, there are buttons for 'Add', 'Refresh', and 'Delete'.

You need to enter the following information on the **Add load balancing rule** page:

Setting	Value
Name	Enter a unique name for the load balancing rule.
IP Version	Select either IPv4 or IPv6.
Frontend IP address	Select the existing public-facing IP address of the load balancer.
Backend pool	Select an existing backend pool. The virtual machines in this backend pool are the target for the load balanced traffic of this rule.
Protocol	Select either the TCP or UDP protocol.
Port	Specify the port number for the load balancing rule. The default is port 80.
Backend port	You can choose to route traffic to the virtual machine in the backend pool using a different port than the one that clients use by default to communicate with the load balancer (port 80).
Health probe	Select an existing health probe or create a new one. The load balancing rule uses the health probe to determine which virtual machines in the backend pool are healthy and therefore can receive load balanced traffic.
Session persistence	You can choose None , or Client IP , or Client IP and protocol . Session persistence specifies that traffic from a client should be handled by the same virtual machine in the backend pool for the duration of a session. None specifies that successive requests from the same client may be handled by any virtual machine. Client IP specifies that successive requests from the same client IP address will be handled by the same virtual machine. Client IP and protocol specifies that successive requests from the same client IP address and protocol combination will be handled by the same virtual machine.
Idle timeout (minutes)	Specify the time to keep a TCP or HTTP connection open without relying on clients to send <i>keep-alive</i> messages. The default idle timeout is 4 minutes, which is also the minimum setting. The maximum setting is 30 minutes.
Enable TCP Reset	Choose between Disabled or Enabled . With TCP Reset on Idle set to Disabled , Azure doesn't send a TCP reset packet when the idle timeout period is reached. With TCP Reset on Idle set to Enabled , Azure sends a TCP reset packet when the idle timeout period is reached.
Enable Floating IP	Choose between Disabled or Enabled . With Floating IP set to Disabled , Azure exposes a traditional load balancing IP address mapping scheme for ease of use (the VM instances' IP). With Floating IP set to Enabled , it changes the IP address mapping to the Frontend IP of the load balancer to allow for more flexibility.
Outbound source network address translation (SNAT)	Choose between Disabled or Enabled . With Outbound SNAT set to Disabled , Azure doesn't translate the source IP address of outbound flows to public IP addresses. With Outbound SNAT set to Enabled , Azure translates the source IP address of outbound flows to public IP addresses.

Add load balancing rule

Public-Std-LB

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.

Name *	<input type="text" value="Contoso-lbrule"/> *
IP Version *	<input checked="" type="radio"/> IPv4 <input type="radio"/> IPv6
Frontend IP address * ⓘ	<input type="text" value="Contoso-Frontend-IP-Config"/>
Backend pool * ⓘ	<input type="text" value="Contoso-bepool"/>
Protocol	<input checked="" type="radio"/> TCP <input type="radio"/> UDP
Port *	<input type="text" value="80"/>
Backend port * ⓘ	<input type="text" value="80"/>
Health probe * ⓘ	<input type="text" value="Contoso-HealthProbe (TCP:80)"/> Create new
Session persistence ⓘ	<input type="text" value="None"/>
Idle timeout (minutes) * ⓘ	<input type="text" value="4"/>
Enable TCP Reset	<input checked="" type="checkbox"/>
Enable Floating IP ⓘ	<input type="checkbox"/>
Outbound source network address translation (SNAT) ⓘ	<input checked="" type="radio"/> (Recommended) Use outbound rules to provide backend pool members access to the internet. Learn more.  <input type="radio"/> Use default port allocation to provide backend pool members with a minimal set of SNAT ports. This is not recommended because it can cause SNAT port exhaustion. Learn more. 

Save

Cancel

The screenshot shows the Azure portal interface for managing load balancing rules. On the left, a navigation sidebar lists various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Frontend IP configuration, Backend pools, Health probes, Load balancing rules (which is selected and highlighted in grey), and Inbound NAT rules. The main content area is titled "Public-Std-LB | Load balancing rules". It includes a search bar, an "Add" button, a "Refresh" button, and a "Delete" button. A descriptive text block explains what a load balancer rule does. Below this is a table with columns: Name, Protocol, Backend pool, and Health probe. The first row in the table is highlighted with a red box around the "Name" column value "Contoso-lbrule".

Test the load balancer

Having completed the various tasks to create and configure your public load balancer and its components, you should then test your configuration to ensure it works successfully. The simplest way to do this is to copy the **Public IP Address** from the public load balancer resource you created and paste it into a web browser. You should receive a response from one of the VMs in your load balancer. You could then stop whichever VM randomly responds, and once that VM has stopped, refresh the browser page to verify that you receive a response from the other VM in the load balancer instead.

Exercise: Create and configure an Azure load balancer

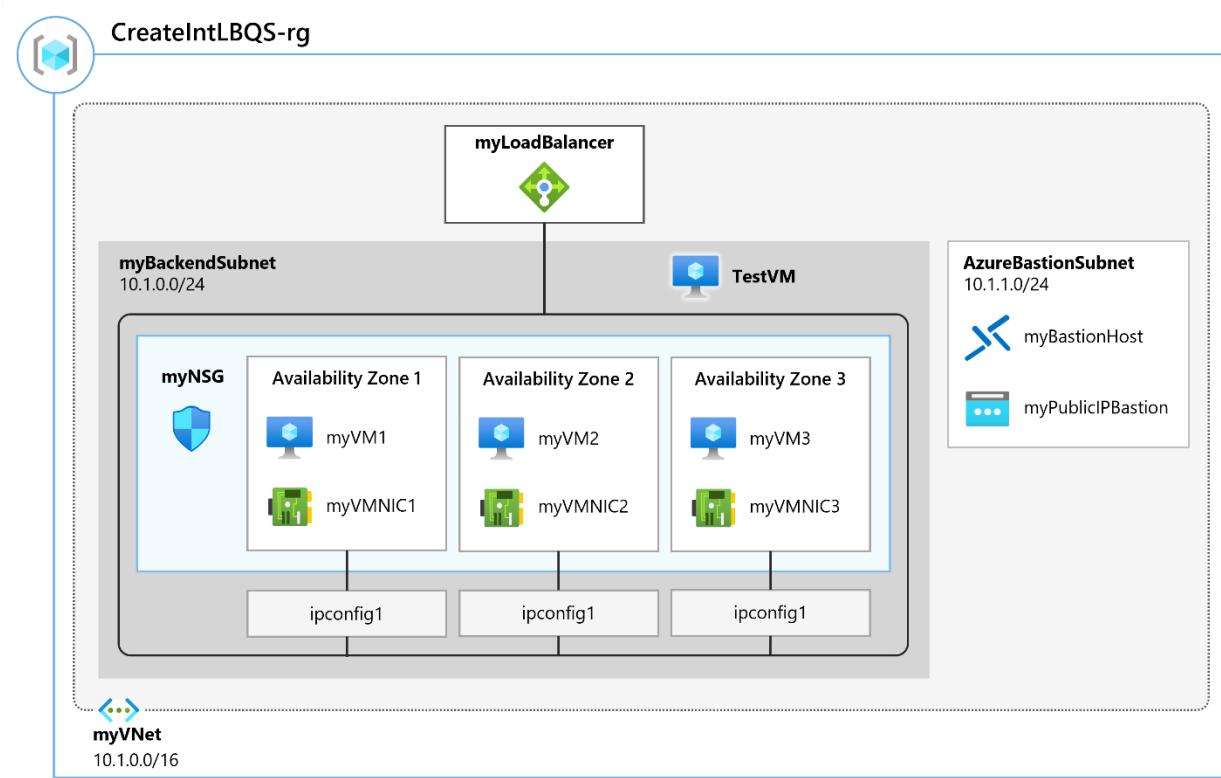
Completed 100 XP

- 15 minutes

Lab scenario

In this lab, you will create an internal load balancer for the fictional Contoso Ltd organization.

Architecture diagram



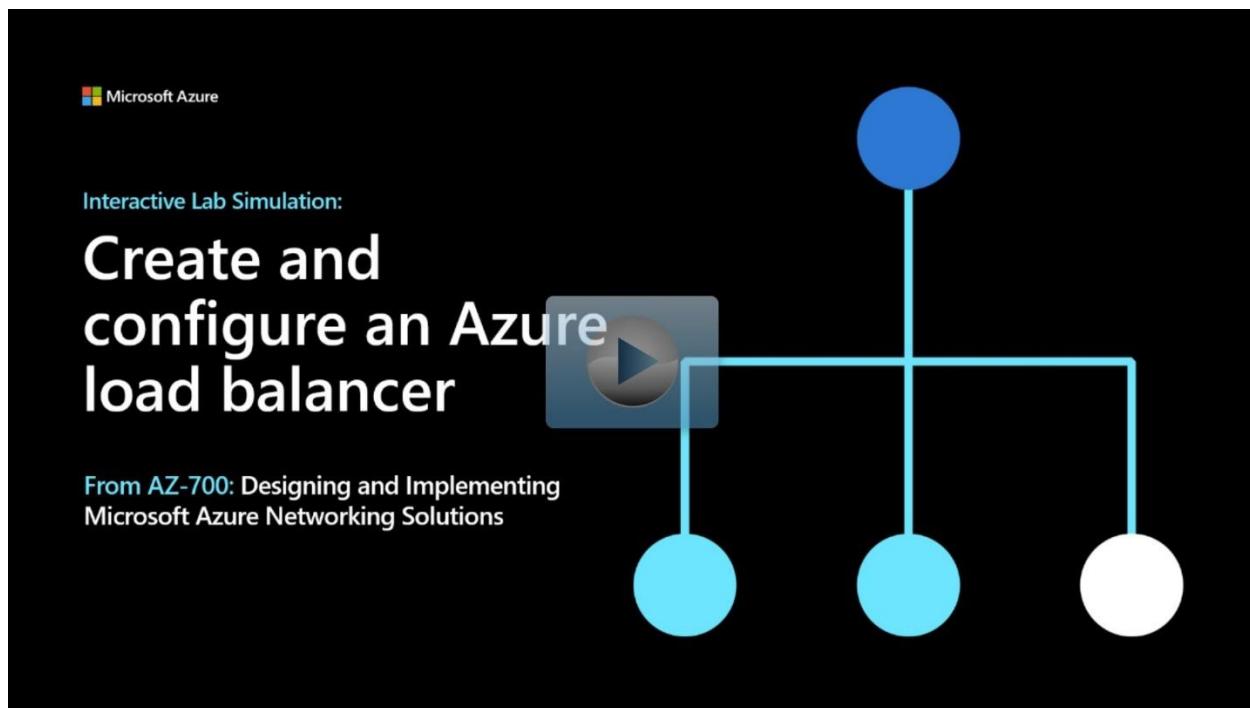
Objectives

- **Task 1:** Create the virtual network
- **Task 2:** Create backend servers
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 3:** Create the load balancer
- **Task 4:** Create load balancer resources
- **Task 5:** Test the load balancer

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20and%20configure%20an%20Azure%20load%20balancer>

Explore Azure Traffic Manager

200 XP

- 14 minutes

Azure Traffic Manager is a DNS-based traffic load balancer. This service allows you to distribute traffic to your public facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness.

Traffic Manager uses DNS to direct the client requests to the appropriate service endpoint based on a traffic-routing method. Traffic manager also provides health monitoring for every endpoint. The endpoint can be any Internet-facing service hosted inside or outside of Azure. Traffic Manager provides a range of traffic-routing methods and endpoint monitoring options to suit different application needs and automatic failover models. Traffic Manager is resilient to failure, including the failure of an entire Azure region.

Key features of Traffic Manager

Traffic Manager offers the several key features.

Feature	Description
Increase application availability	Traffic Manager delivers high availability for your critical applications by monitoring your endpoints and providing automatic failover when an endpoint goes down.
Improve application performance	Azure allows you to run cloud services and websites in datacenters located around the world. Traffic Manager can improve the responsiveness of your website by directing traffic to the endpoint with the lowest latency.
Service maintenance without downtime	You can have planned maintenance done on your applications without downtime. Traffic Manager can direct traffic to alternative endpoints while the maintenance is in progress.
Combine hybrid applications	Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including the burst-to-cloud, migrate-to-cloud, and failover-to-cloud scenarios.
Distribute traffic for complex deployments	Using nested Traffic Manager profiles, multiple traffic-routing methods can be combined to create sophisticated and flexible rules to scale to the needs of larger, more complex deployments.

How Traffic Manager works

Azure Traffic Manager enables you to control the distribution of traffic across your application endpoints. An endpoint is any Internet-facing service hosted inside or outside of Azure.

Traffic Manager provides two key benefits:

- Distribution of traffic according to one of several **traffic-routing methods**
- Continuous **monitoring of endpoint** health and automatic failover when endpoints fail

When a client attempts to connect to a service, it must first resolve the DNS name of the service to an IP address. The client then connects to that IP address to access the service.

Traffic Manager uses DNS to direct clients to specific service endpoints based on the rules of the traffic-routing method. Clients connect to the selected endpoint directly. Traffic Manager isn't a proxy or a gateway. Traffic Manager doesn't see the traffic passing between the client and the service.

Traffic Manager works at the DNS level which is at the Application layer (Layer-7).

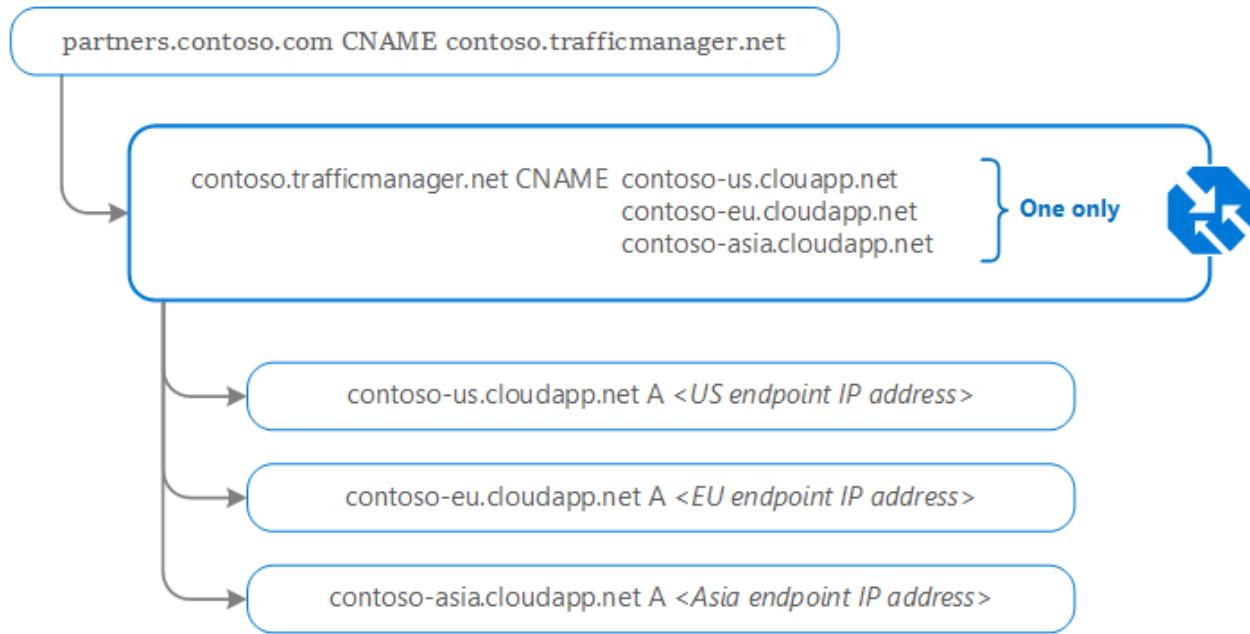
Traffic Manager example deployment

Contoso Corp have developed a new partner portal. The URL for this portal is <https://partners.contoso.com/login.aspx>.

The application is hosted in three regions of Azure. To improve availability and maximize global performance, they use Traffic Manager to distribute client traffic to the closest available endpoint.

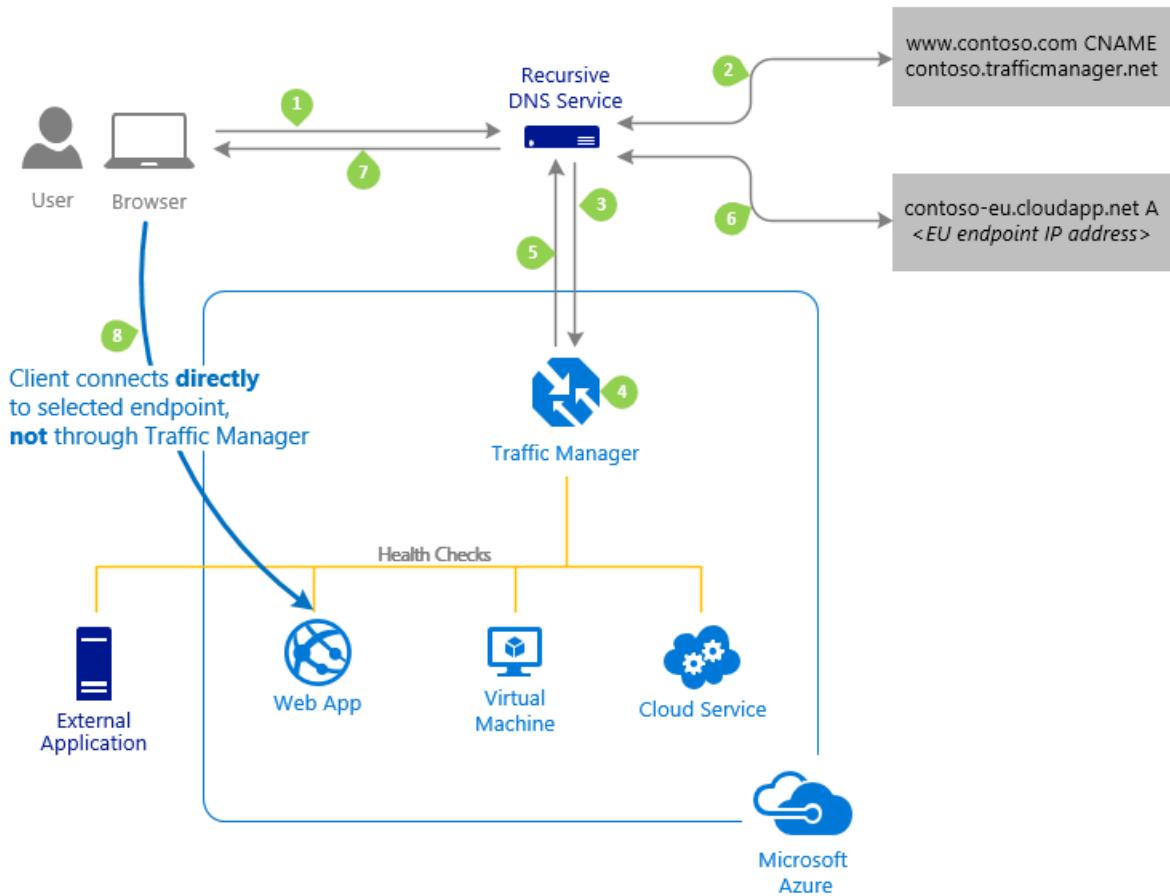
To achieve this configuration, they complete the following steps:

1. Deploy three instances of their service. The DNS names of these deployments are contoso-us.cloudapp.net, contoso-eu.cloudapp.net, and contoso-asia.cloudapp.net.
2. Create a Traffic Manager profile, named contoso.trafficmanager.net, and configure it to use the 'Performance' traffic-routing method across the three endpoints.
3. Configure their vanity domain name, partners.contoso.com, to point to contoso.trafficmanager.net, using a DNS CNAME record.



Traffic Manager example client usage

Following on from the deployment example above; when a client requests the page <https://partners.contoso.com/login.aspx>, the client performs the following steps to resolve the DNS name and establish a connection:



1. The client sends a DNS query to its configured recursive DNS service to resolve the name 'partners.contoso.com'. A recursive DNS service, sometimes called a 'local DNS' service, doesn't host DNS domains directly. Rather, the client off-loads the work of contacting the various authoritative DNS services across the Internet needed to resolve a DNS name.
2. To resolve the DNS name, the recursive DNS service finds the name servers for the 'contoso.com' domain. It then contacts those name servers to request the 'partners.contoso.com' DNS record. The contoso.com DNS servers return the CNAME record that points to contoso.trafficmanager.net.
3. Next, the recursive DNS service finds the name servers for the 'trafficmanager.net' domain, which are provided by the Azure Traffic Manager service. It then sends a request for the 'contoso.trafficmanager.net' DNS record to those DNS servers.
4. The Traffic Manager name servers receive the request. They choose an endpoint based on:
 - The configured state of each endpoint (disabled endpoints aren't returned)
 - The current health of each endpoint, as determined by the Traffic Manager health checks.

- The chosen traffic-routing method.
5. The chosen endpoint is returned as another DNS CNAME record. In this case, let us suppose contoso-eu.cloudapp.net is returned.
 6. Next, the recursive DNS service finds the name servers for the 'cloudapp.net' domain. It contacts those name servers to request the 'contoso-eu.cloudapp.net' DNS record. A DNS 'A' record containing the IP address of the EU-based service endpoint is returned.
 7. The recursive DNS service consolidates the results and returns a single DNS response to the client.
 8. The client receives the DNS results and connects to the given IP address. The client connects to the application service endpoint directly, not through Traffic Manager. Since it's an HTTPS endpoint, the client performs the necessary SSL/TLS handshake, and then makes an HTTP GET request for the '/login.aspx' page.

The recursive DNS service caches the DNS responses it receives. The DNS resolver on the client device also caches the result. Caching enables subsequent DNS queries to be answered more quickly by using data from the cache rather than querying other name servers. The duration of the cache is determined by the 'time-to-live' (TTL) property of each DNS record. Shorter values result in faster cache expiry and thus more round-trips to the Traffic Manager name servers. Longer values mean that it can take longer to direct traffic away from a failed endpoint. Traffic Manager allows you to configure the TTL used in Traffic Manager DNS responses to be as low as 0 seconds and as high as 2,147,483,647 seconds (the maximum range compliant with RFC-1035), enabling you to choose the value that best balances the needs of your application.

Traffic routing methods

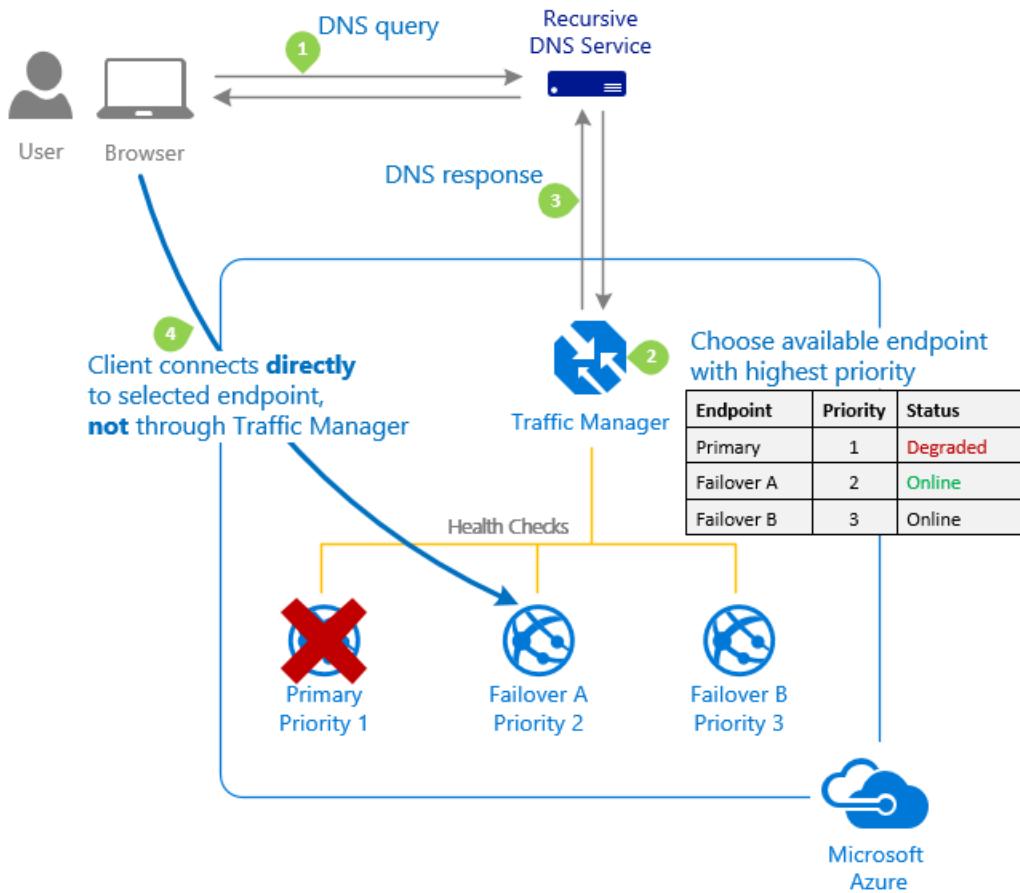
Azure Traffic Manager supports six traffic-routing methods to determine how to route network traffic to the various service endpoints. For any profile, Traffic Manager applies the traffic-routing method associated to it to each DNS query it receives. The traffic-routing method determines which endpoint is returned in the DNS response.

The following traffic routing methods are available in Traffic Manager:

Routing method	When to use
Priority	Select this routing method when you want to have a primary service endpoint for all traffic. You can provide multiple backup endpoints in case the primary or one of the backup endpoints is unavailable.
Weighted	Select this routing method when you want to distribute traffic across a set of endpoints based on their weight. Set the weight the same to distribute evenly across all endpoints.
Performance	Select the routing method when you have endpoints in different geographic locations, and you want end users to use the "closest" endpoint for the lowest network latency.
Geographic	Select this routing method to direct users to specific endpoints (Azure, External, or Nested) based on where their DNS queries originate from geographically. With this routing method, it enables you to be compliant with scenarios such as data sovereignty mandates, localization of content & user experience and measuring traffic from different regions.
MultiValue	Select this routing method for Traffic Manager profiles that can only have IPv4/IPv6 addresses as endpoints. When a query is received for this profile, all healthy endpoints are returned.
Subnet	Select this routing method to map sets of end-user IP address ranges to a specific endpoint. When a request is received, the endpoint returned will be the one mapped for that request's source IP address.

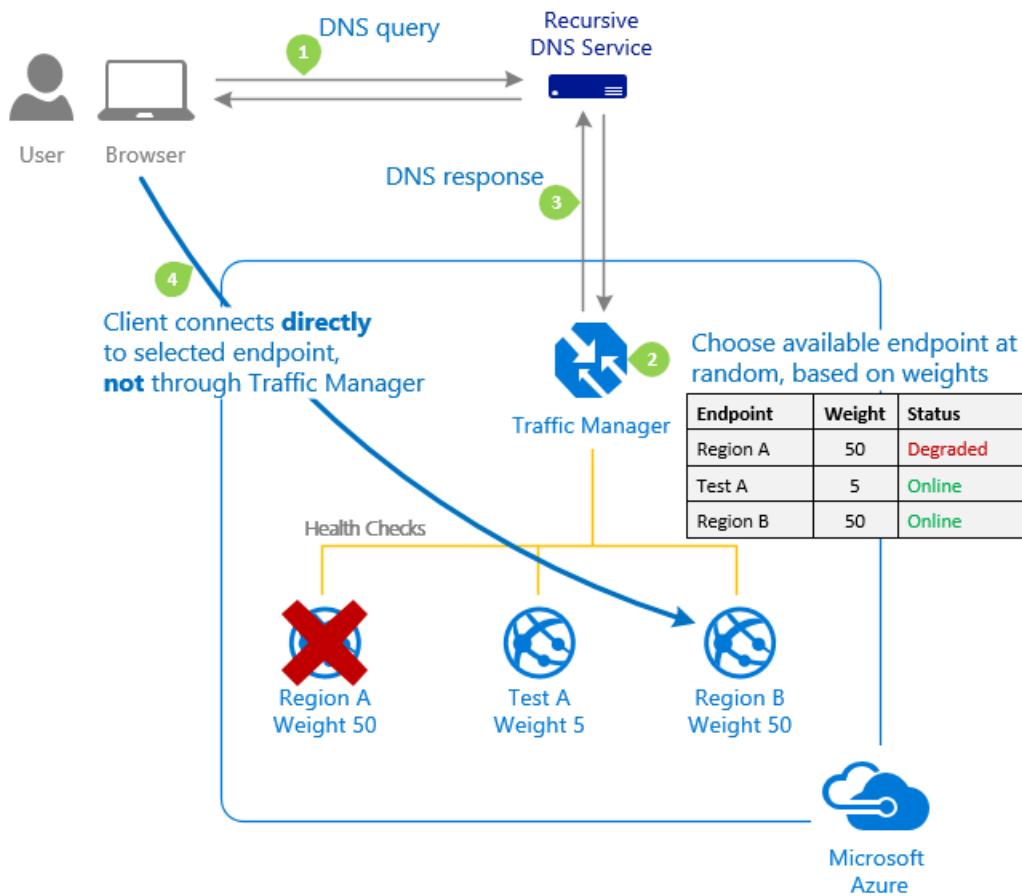
Routing method examples

This is an example of the **Priority** routing method.



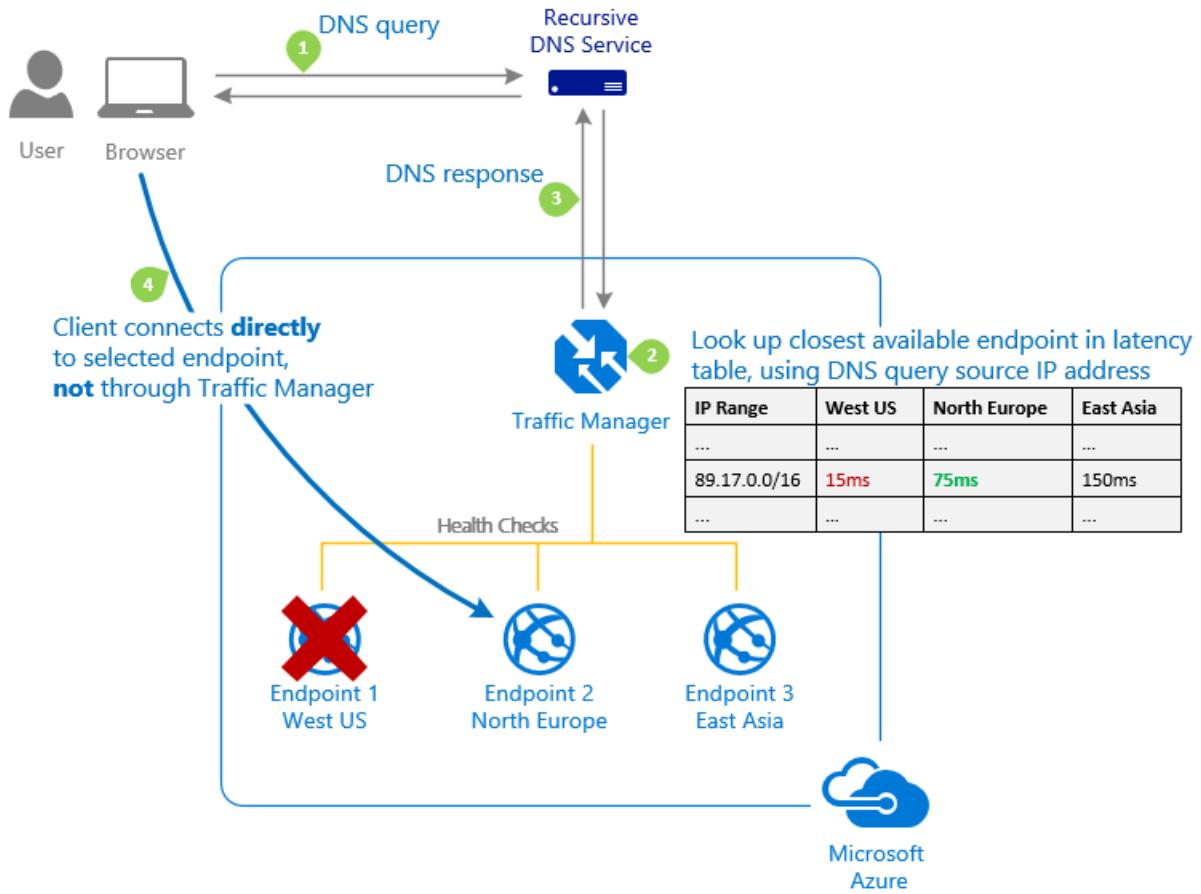
For more information, see [Priority traffic-routing method](#).

This is an example of the **Weighted** routing method.



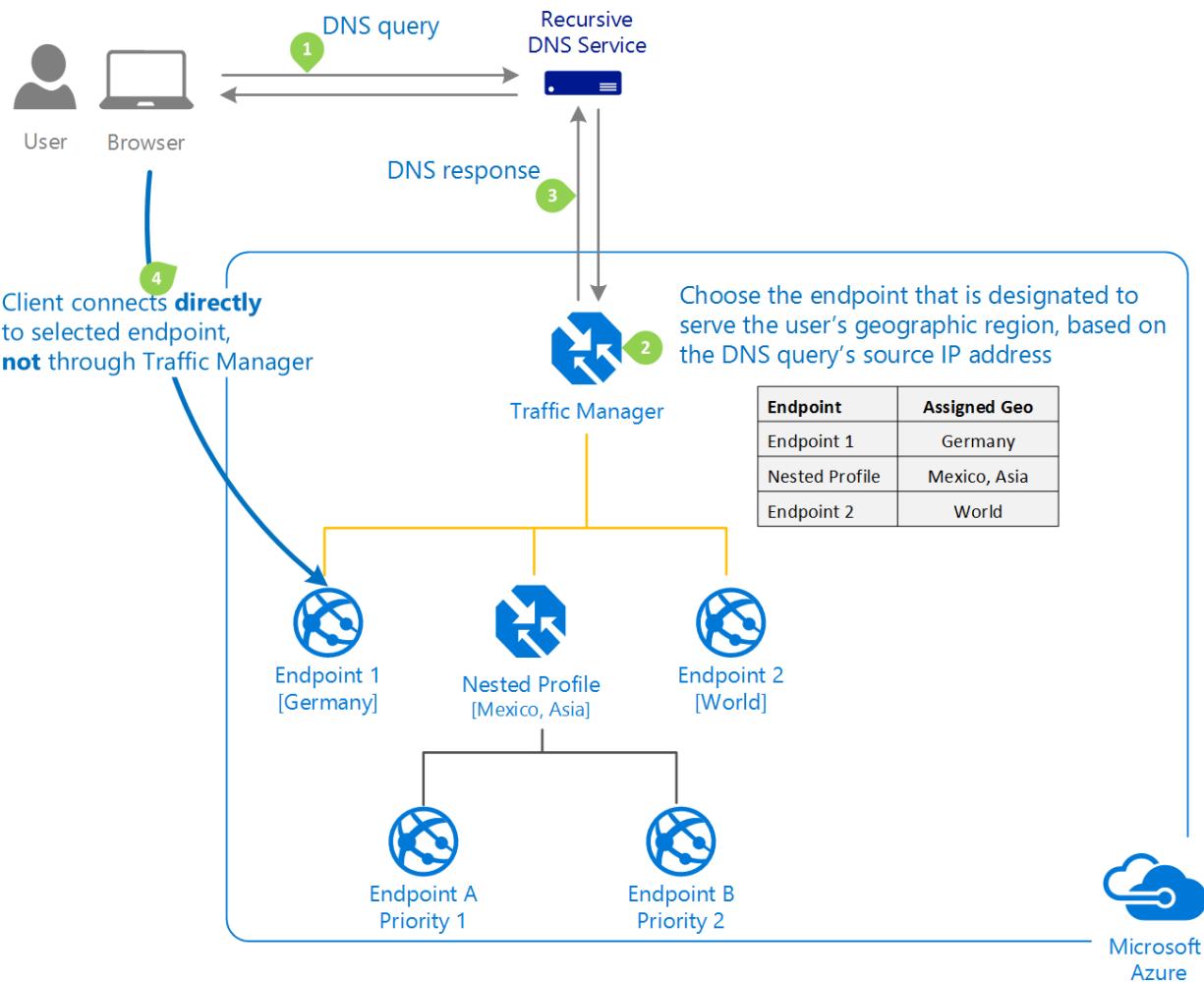
For more information, see [Weighted traffic-routing method](#).

This is an example of the **Performance** routing method.



For more information, see [Performance traffic-routing method](#).

This is an example of the **Geographic** routing method.



For more information, see [Geographic traffic-routing method](#).

Traffic Manager profiles

Within a Traffic Manager profile, you can only configure one traffic routing method at a time. You can select a different traffic routing method for your profile at any time. Your changes are applied within a minute without any downtime.

All Traffic Manager profiles have health monitoring and automatic failover of endpoints.

Nested Traffic Manager profiles

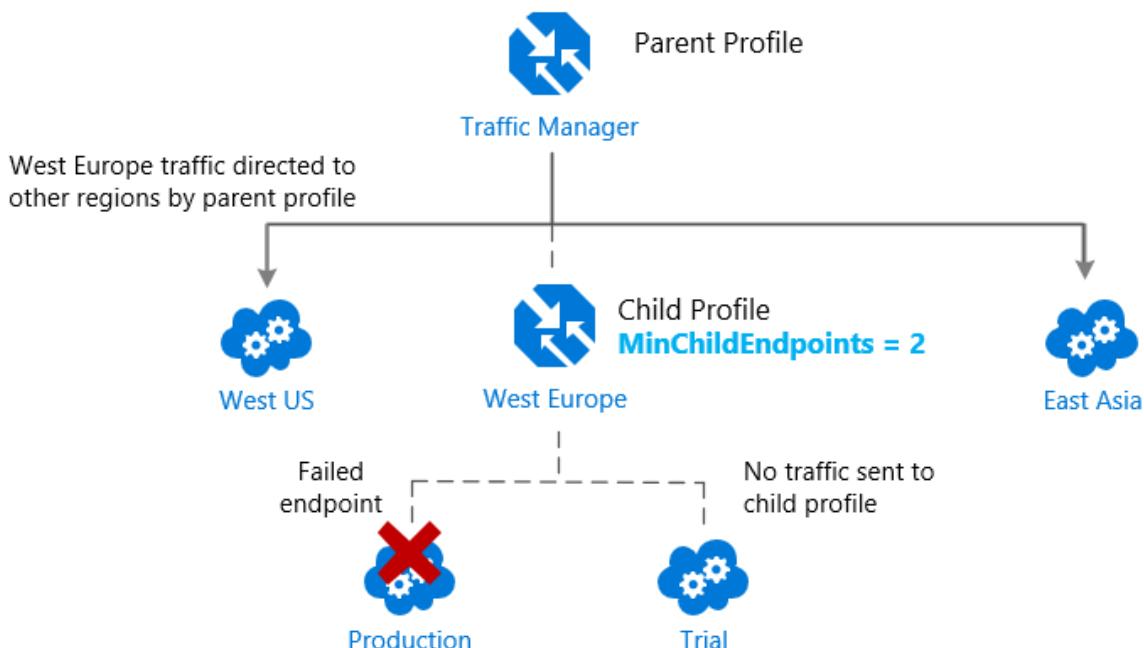
As mentioned earlier, each Traffic Manager profile can only specify one traffic-routing method. However, you may have scenarios that require more complicated traffic routing

than the routing that can be provided by a single Traffic Manager profile. In these situations, you can combine traffic routing methods by using nested Traffic Manager profiles to gain the benefits of multiple traffic-routing methods. Nested profiles enable you to override the default Traffic Manager behavior to support larger and more complex traffic-routing configurations for your application deployments.

The example and diagrams below illustrate the combining of the **Performance** and **Weighted** traffic-routing methods in nested profiles.

Example: Combining 'performance' and 'weighted' traffic routing methods using nested profiles

Suppose that you deployed an application in the following Azure regions: West US, West Europe, and East Asia. You use the **Performance** traffic-routing method to distribute traffic to the region closest to the user.

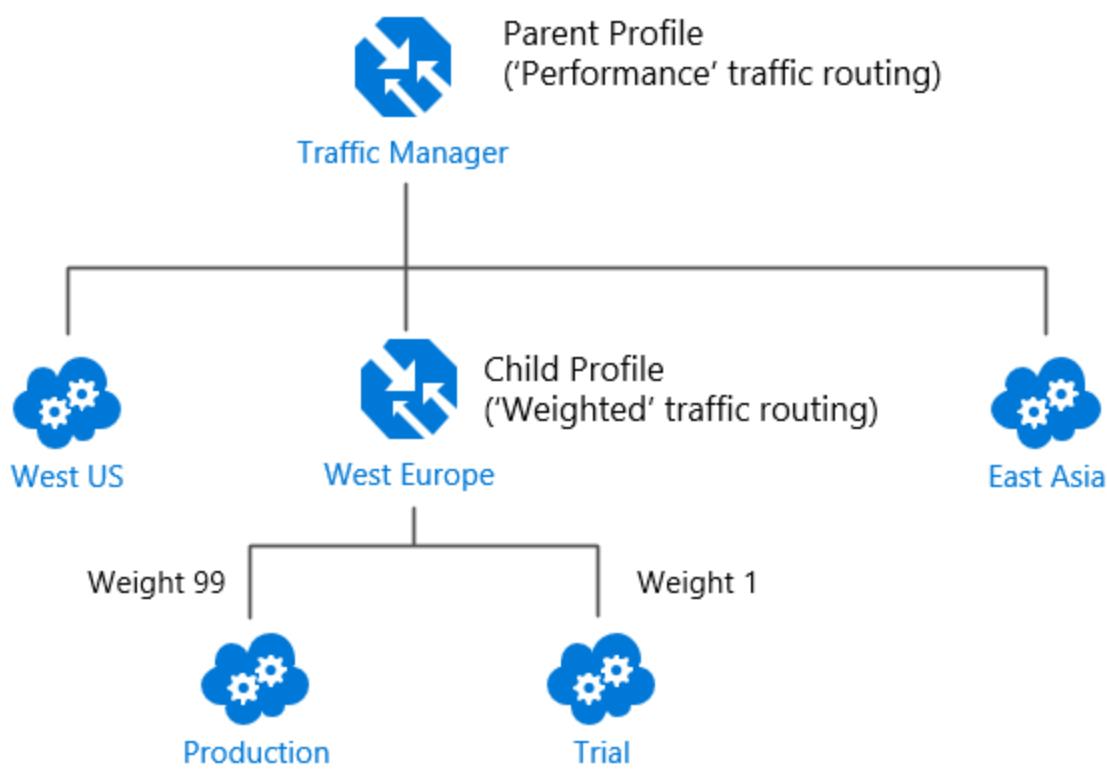


But what if you wanted to test an update to your service before rolling it out more widely, and you wanted to use the **Weighted** traffic-routing method to direct a small percentage of traffic to your test deployment?

You would set up the test deployment alongside the existing production deployment in West Europe.

As you just learned, you can't combine both the **Weighted** and **Performance** traffic-routing methods in a single profile. Therefore, to support this scenario, you would create a Traffic Manager profile using the two West Europe endpoints and the **Weighted** traffic-routing method. Then you would add this child profile as an endpoint to the parent profile. The parent profile would still use the **Performance** traffic-routing method and would contain the other global deployments as endpoints.

The diagram below illustrates this example scenario:



With the above configuration, traffic directed via the parent profile (using the **Performance** routing method) distributes traffic across regions normally. While, within West Europe, the nested child profile (using the **Weighted** routing method) distributes traffic to the production and test endpoints according to the weights assigned.

When the parent profile uses the **Performance** traffic-routing method, each endpoint must be assigned a location, which is done when you configure the endpoint. Choose the Azure region closest to your deployment.

For more information, and for more example scenarios, see [Nested Traffic Manager profiles](#).

Traffic Manager endpoints

Azure Traffic Manager enables you to control how network traffic is distributed to application deployments running in your different datacenters. You configure each application deployment as an endpoint in Traffic Manager. When Traffic Manager receives a DNS request, it chooses an available endpoint to return in the DNS response. Traffic manager bases the choice on the current endpoint status and the traffic-routing method.

Traffic Manager supports three types of endpoints:

- **Azure endpoints** - Use this type of endpoint to load-balance traffic to a cloud service, web app, or public IP address in the same subscription within Azure.
- **External endpoints** - Use this type of endpoint to load balance traffic for IPv4/IPv6 addresses, FQDNs, or for services hosted outside Azure. These services can either be on-premises or with a different hosting provider.
- **Nested endpoints** - Use this type of endpoint to combine Traffic Manager profiles to create more flexible traffic-routing schemes to support the needs of larger, more complex deployments. With Nested endpoints, a child profile is added as an endpoint to a parent profile. Both the child and parent profiles can contain other endpoints of any type, including other nested profiles.

There are no restrictions on how different endpoints types can be combined in a single Traffic Manager profile; each profile can contain any mix of endpoint types.

You add endpoints to existing Traffic Manager profiles from the **Endpoints** page of a Traffic Manager profile in the Azure portal.

For more information, visit [Traffic Manager endpoints](#).

Configuring Traffic Manager profiles

This example shows how to create and configure a new Traffic Manager profile to direct client traffic based on endpoint priority.

From the Azure portal home page, From the Azure portal home page, navigate to the Global Search bar and search **Traffic Manager profile**. Then select **Traffic Manager profiles**.

Load balancing | Traffic Manager

[Search](#) [Create](#) [Manage view](#) [Refresh](#) [Export to CSV](#) [Open query](#) [Assign tags](#)

[Overview](#) [Filter for any field...](#) [Add filter](#) [More \(3\)](#)

Load Balancing Services

- [Application Gateway](#)
- [Front Door and CDN profiles](#)
- [Load Balancer](#)
- [Traffic Manager](#) (selected)

Showing 0 to 0 of 0 records.

Name ↑↓	Status ↑↓	Routing method ↑↓	Resource group ↑↓	Subscription ↑↓
---------	-----------	-------------------	-------------------	-----------------



No traffic manager profiles to display

Azure Traffic Manager is a DNS-based traffic load balancer. This service allows you to distribute traffic to your public-facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness.

[Create traffic manager profile](#)

[Learn more](#)

Select **+ Create** or the **Create traffic manager profile** button.

You need to enter the following information on the **Create Traffic Manager profile** page:

Expand table

Field	Information
Name	Enter a unique name for the Traffic Manager profile.
Routing method	Select the routing method to use in this profile.
Subscription	Select the subscription from the list that you want this profile to be applied to.
Resource group	Select the appropriate resource group from the list or create a new one.

Select **Create** to create the profile.

Home > Create a resource > Traffic Manager profile >

Create Traffic Manager profile

Name *

Contoso-TMprofile ✓
.trafficmanager.net

Routing method

Priority

Subscription *

Free Trial

Resource group *

Contoso-ResourceGroup

[Create new](#)

Resource group location ⓘ

West US

Create

[Automation options](#)

The next step is to add endpoints to the Traffic Manager profile.

From the Azure portal home page, select **All resources**, then select the Traffic Manager profile from the list.

Home >

All resources 🔗 ...

Flexible Road LLC (skillup.tech)

+ Create Manage view Refresh Export to CSV Open query | Assign tags Delete | Feedback

Filter for any field... Subscription == all Resource group == all Type == all Location == all + Add filter

Showing 1 to 52 of 52 records. Show hidden types ⓘ

No grouping ▼ List view

<input type="checkbox"/> Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
<input type="checkbox"/> Contoso-LB-AS	Availability set	Contoso-ResourceGroup	West US	Free Trial
<input type="checkbox"/> Contoso-Public-IP	Public IP address	Contoso-ResourceGroup	West US	Free Trial
<input type="checkbox"/> Contoso-TMprofile	Traffic Manager profile	Contoso-ResourceGroup	Global	Free Trial
<input type="checkbox"/> contoso.com	Private DNS zone	Contoso-ResourceGroup	Global	Free Trial
<input type="checkbox"/> CoreServicesVnet	Virtual network	Contoso-ResourceGroup	West US	Free Trial
<input type="checkbox"/> ManufacturingVM	Virtual machine	Contoso-ResourceGroup	North Europe	Free Trial
<input type="checkbox"/> ManufacturingVM-ip	Public IP address	Contoso-ResourceGroup	North Europe	Free Trial
<input type="checkbox"/> ManufacturingVM-nsg	Network security group	Contoso-ResourceGroup	North Europe	Free Trial
<input type="checkbox"/> manufacturingvm489	Network interface	Contoso-ResourceGroup	North Europe	Free Trial

On the Traffic manager profile page, under **Settings**, select **Endpoints**, then select **Add**.

Home > All resources > Contoso-TMprofile

All resources 🔗 ...

Flexible Road LLC (skillup.tech)

+ Create Manage view ...

Filter for any field...

Name ↑↓

- Contoso-LB-AS ...
- Contoso-Public-IP ...
- Contoso-TMprofile ...
- contoso.com ...
- CoreServicesVnet ...
- ManufacturingVM ...
- ManufacturingVM-ip ...
- ManufacturingVM-nsg ...
- manufacturingvm489 ...
- ManufacturingVM_OsDisk_1_b3c65a... ...

Page 1 of 1

+ Add Refresh

🔍 Search (Ctrl+ /)

Contoso-TMprofile | Endpoints 🔗 ...

Traffic Manager profile

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

Settings

- Configuration
- Real user measurements
- Traffic view
- Endpoints
- Properties
- Locks

Monitoring

- Alerts

🔍 Search endpoints

Name ↑↓	Status ↑↓	Monitor status ↑↓
No results.		

You then enter the required information on the **Add endpoint** page:

Field	Information
Type	Select the type of endpoint to add. You can select from the following endpoint types: Azure endpoints External endpoints Nested endpoints Depending on which endpoint type you select here, the remaining options differ.
Name	Enter a unique name for the endpoint.
Target resource type (for Azure endpoints only)	If you select the Azure endpoint type, you can select from the following resource types: Cloud service App Service App Service slot Public IP address
Target resource (for Azure and Nested endpoints only)	Select the appropriate target service, IP address, or profile from the list. The available options differ depending on which endpoint type and target resource type are selected above.
fully qualified domain name (FQDN) or IP (for External endpoints only)	Specify the FQDN or IP address for the external endpoint.
Priority	Specify the priority for this endpoint. If you enter 1, then all traffic goes to this endpoint when it's healthy.
Minimum child endpoints (for Nested endpoints only)	Specify the minimum number of endpoints that must be available in the <i>child</i> Traffic Manager profile for it to receive traffic. If the available-endpoints number in the child profile falls below this threshold, this endpoint is considered as degraded.
Custom Header setting (optional setting)	You can configure custom headers for your endpoint, using the following paired formatting: <i>host:contoso.com,customheader:contoso</i> The maximum number of supported pairs is 8, and they're applicable for both the HTTP and HTTPS protocols. These endpoint Custom Header settings override the settings configured in a profile.
Add as disabled (optional setting)	Disabling an endpoint in Traffic Manager can be useful to temporarily remove traffic from an endpoint that is in maintenance mode or being redeployed. Once the endpoint is running again, it can be re-enabled.

Select **Add** to add the endpoint to the Traffic Manager profile.

 **Add endpoint**

Contoso-TMprofile-SR

Type * ⓘ

Azure endpoint

Name *

myPrimaryEndpoint ✓

Target resource type

App Service

Target resource *

myWebAppEastUS-SR (East US)

Priority *

1

Custom Header settings ⓘ

Add as disabled

Add

If you're adding a failover endpoint for another Azure region, then you would add another endpoint for that region. This would point to the application target resource in the other region and would have a priority setting of **2**.



Add endpoint

Contoso-TMprofile-SR

X

Type * ⓘ

Azure endpoint

Name *

myFailoverEndpoint



Target resource type

App Service

Target resource *

myWebAppWestEurope-SR (West Europe)

Priority *

2

Custom Header settings ⓘ

Add as disabled

Add

When you add endpoints to a Traffic Manager profile, their status is checked.

Contoso-TMprofile-SR | Endpoints

Traffic Manager profile

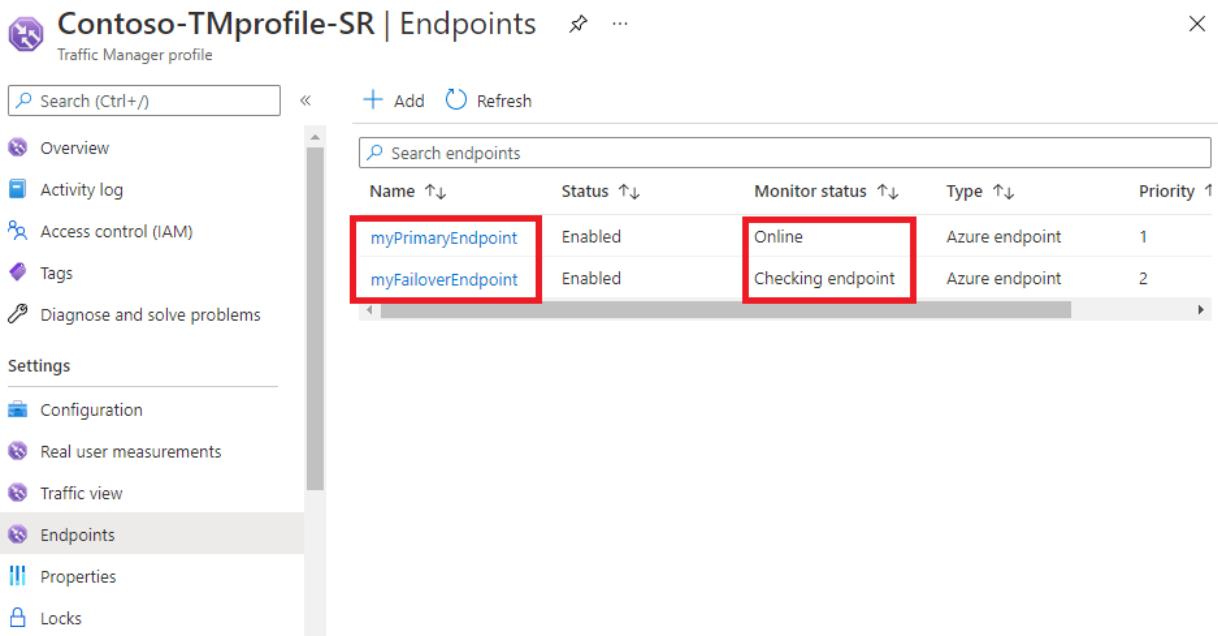
Search (Ctrl+ /) Add Refresh

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

Settings Configuration Real user measurements Traffic view Endpoints Properties Locks

Name ↑↓	Status ↑↓	Monitor status ↑↓	Type ↑↓	Priority ↑↓
myPrimaryEndpoint	Enabled	Online	Azure endpoint	1
myFailoverEndpoint	Enabled	Checking endpoint	Azure endpoint	2

Once they have been checked their **Monitor status** changes to Online.



Contoso-TMprofile-SR | Endpoints

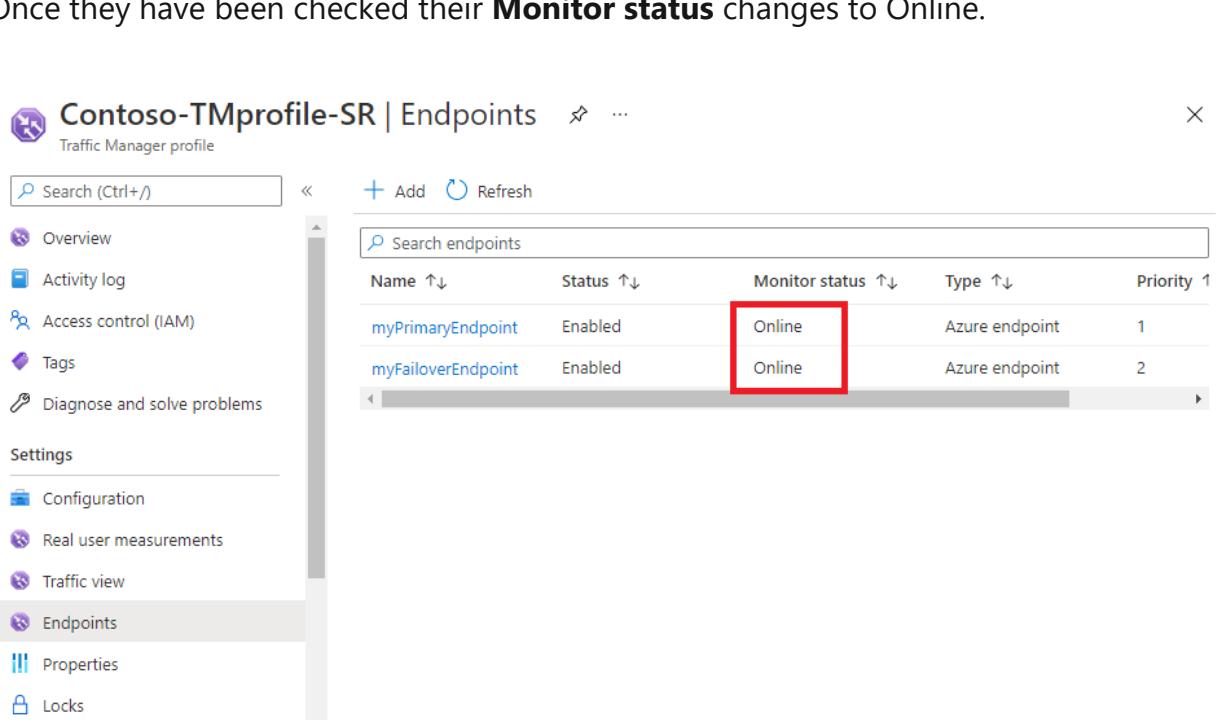
Traffic Manager profile

Search (Ctrl+ /) Add Refresh

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

Settings Configuration Real user measurements Traffic view Endpoints Properties Locks

Name ↑↓	Status ↑↓	Monitor status ↑↓	Type ↑↓	Priority ↑↓
myPrimaryEndpoint	Enabled	Online	Azure endpoint	1
myFailoverEndpoint	Enabled	Online	Azure endpoint	2



Configuring endpoint monitoring

Azure Traffic Manager includes built-in endpoint monitoring and automatic endpoint failover. This feature helps you deliver high-availability applications that are resilient to endpoint failure, including Azure region failures.

To configure endpoint monitoring, you open the **Configuration** page for the Traffic Manager profile.

Then, under the **Endpoint monitor settings** section, you specify the following settings for the Traffic Manager profile:

Setting	Description
Protocol	Choose HTTP, HTTPS, or TCP as the protocol that Traffic Manager uses when probing your endpoint to check its health. HTTPS monitoring doesn't verify whether your TLS/SSL certificate is valid; it only checks that the certificate is present.
Port	Choose the port used for the request.
Path	This configuration setting is valid only for the HTTP and HTTPS protocols, for which specifying the path setting is required. Providing this setting for the TCP monitoring protocol results in an error. For HTTP and HTTPS protocol, give the relative path and the name of the webpage or the file that the monitoring accesses. A forward slash (/) is a valid entry for the relative path. This value implies that the file is in the root directory (default).
Custom Header settings	This configuration setting helps you add specific HTTP headers to the health checks that Traffic Manager sends to endpoints under a profile. The custom headers can be specified at a profile level to be applicable for all endpoints in that profile and / or at an endpoint level applicable only to that endpoint. You can use custom headers for health checks of endpoints in a multitenant environment. That way it can be routed correctly to their destination by specifying a host header. You can also use this setting by adding unique headers that can be used to identify Traffic Manager originated HTTP(S) requests and processes them differently. You can specify up to eight <i>header:value</i> pairs separated by a comma. Example - header1:value1, header2:value2
Expected Status Code Ranges	This setting allows you to specify multiple success code ranges in the format 200-299, 301-301. If these status codes are received as response from an endpoint when a health check is done, Traffic Manager marks those endpoints as healthy. You can specify a maximum of eight status code ranges. This setting is applicable only to HTTP and HTTPS protocol and to all endpoints. This setting is at the Traffic Manager profile level and by default the value 200 is defined as the success status code.

Probing interval	This value specifies how often an endpoint is checked for its health from a Traffic Manager probing agent. You can specify two values here: 30 seconds (normal probing) and 10 seconds (fast probing). If no values are provided, the profile sets to a default value of 30 seconds. Visit the Traffic Manager Pricing page to learn more about fast probing pricing.
Tolerated number of failures	This value specifies how many failures a Traffic Manager probing agent tolerates before marking that endpoint as unhealthy. Its value can range between 0 and 9. A value of 0 means a single monitoring failure can cause that endpoint to be marked as unhealthy. If no value is specified, it uses the default value of 3.
Probe timeout	This property specifies the amount of time the Traffic Manager probing agent should wait before considering a health probe check to an endpoint a failure. If the Probing Interval is set to 30 seconds, then you can set the Timeout value between 5 and 10 seconds. If no value is specified, it uses a default value of 10 seconds. If the Probing Interval is set to 10 seconds, then you can set the Timeout value between 5 and 9 seconds. If no Timeout value is specified, it uses a default value of 9 seconds.

Select **Save** when you have finished endpoint monitor configuration.

The screenshot shows the Azure Traffic Manager configuration interface for a profile named "Contoso-TMprofile-SR". The left sidebar lists various tabs: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration (which is selected and highlighted with a red box), Real user measurements, Traffic view, Endpoints, Properties, Locks, Monitoring (Alerts, Metrics, Diagnostic settings, Logs), Automation (Tasks (preview), Export template), and a Search bar.

The main configuration area includes:

- Routing method:** Priority
- DNS time to live (TTL):** 60 seconds
- Endpoint monitor settings:**
 - Protocol:** HTTP
 - Port:** 80
 - Path:** /
- Custom Header settings:** (empty)
- Expected Status Code Ranges (default: 200):** 200-299
- Fast endpoint failover settings:**
 - Probing interval:** 30
 - Tolerated number of failures:** 3
 - Probe timeout:** 10 seconds

How endpoint monitoring works

When the monitoring protocol is set as HTTP or HTTPS, the Traffic Manager probing agent makes a GET request to the endpoint using the protocol, port, and relative path given. An endpoint is considered healthy if probing agent receives a 200-OK response, or any of the responses configured in the Expected status code *ranges. If the response is a different value or no response get received within the timeout period, the Traffic Manager probing agent reattempts according to the Tolerated Number of Failures setting. No reattempts are done if this setting is 0. The endpoint is marked unhealthy if the number of consecutive failures is higher than the Tolerated Number of Failures setting.

When the monitoring protocol is TCP, the Traffic Manager probing agent creates a TCP connection request using the port specified. If the endpoint responds to the request with a response to establish the connection, that health check is marked as a success. The Traffic Manager probing agent resets the TCP connection. In cases where the

response is a different value or no response get received within the timeout period, the Traffic Manager probing agent reattempts according to the Tolerated Number of Failures setting. No reattempts are made if this setting is 0. If the number of consecutive failures is higher than the Tolerated Number of Failures setting, then that endpoint is marked unhealthy.

In all cases, Traffic Manager probes from multiple locations. The consecutive failure determines what happen within each region. That's why endpoints are receiving health probes from Traffic Manager with a higher frequency than the setting used for Probing Interval.

For HTTP or HTTPS monitoring protocol, a common practice on the endpoint side is to implement a custom page within your application - for example, /health.aspx. Using this path for monitoring, you can perform application-specific checks, such as checking performance counters or verifying database availability. Based on these custom checks, the page returns an appropriate HTTP status code.

All endpoints in a Traffic Manager profile share monitoring settings. If you need to use different monitoring settings for different endpoints, you can create [nested Traffic Manager profiles](#).

Exercise: Create a Traffic Manager profile using the Azure portal

Completed 100 XP

- 7 minutes

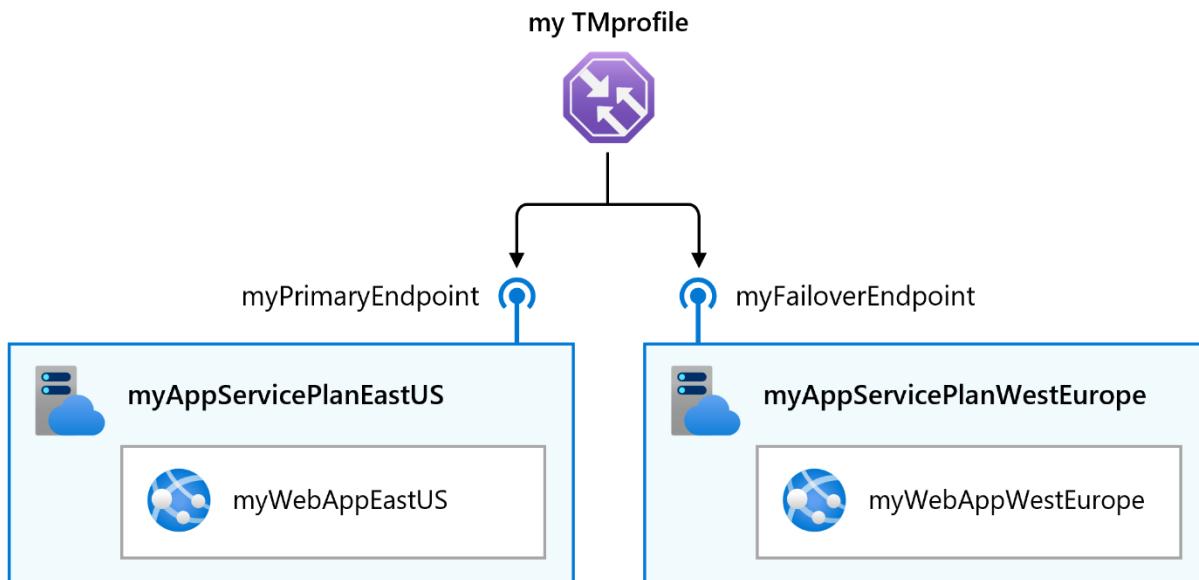
Lab scenario

In this lab, you create a Traffic Manager profile to deliver high availability for the fictional Contoso Ltd organization's web application.

You create two instances of a web application deployed in two different regions (East US and West Europe). The East US region is the primary endpoint for Traffic Manager, and the West Europe region is the failover endpoint.

Then you create a Traffic Manager profile based on endpoint priority. This profile directs user traffic to the primary site running the web application. Traffic Manager continuously monitors the web application, and if the primary site in East US is unavailable, it provides automatic failover to the backup site in West Europe.

Architecture diagram



Objectives

- **Task 1:** Create the web apps
- **Task 2:** Create a Traffic Manager profile
- **Task 3:** Add Traffic Manager endpoints
- **Task 4:** Test the Traffic Manager profile

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.

Microsoft Azure

Interactive Lab Simulation:

Create a Traffic Manager profile using the Azure portal

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions

```
graph TD; Top(( )) --- Vertical(( )); Vertical --- Horizontal(( )); Horizontal --- Blue1(( )); Horizontal --- White1(( )); Horizontal --- White2(( ));
```

<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20a%20Traffic%20Manager%20profile%20using%20the%20Azure%20portal>

Summary

Completed 100 XP

- 1 minute

In this module, you had a high-level overview of the different load-balancing options available to you in Azure. You learned in detail about two of those Azure load-balancing technologies, namely Azure Load Balancer and Azure Traffic Manager.

You now have the knowledge required to help you to load balance network traffic in your Azure networks.

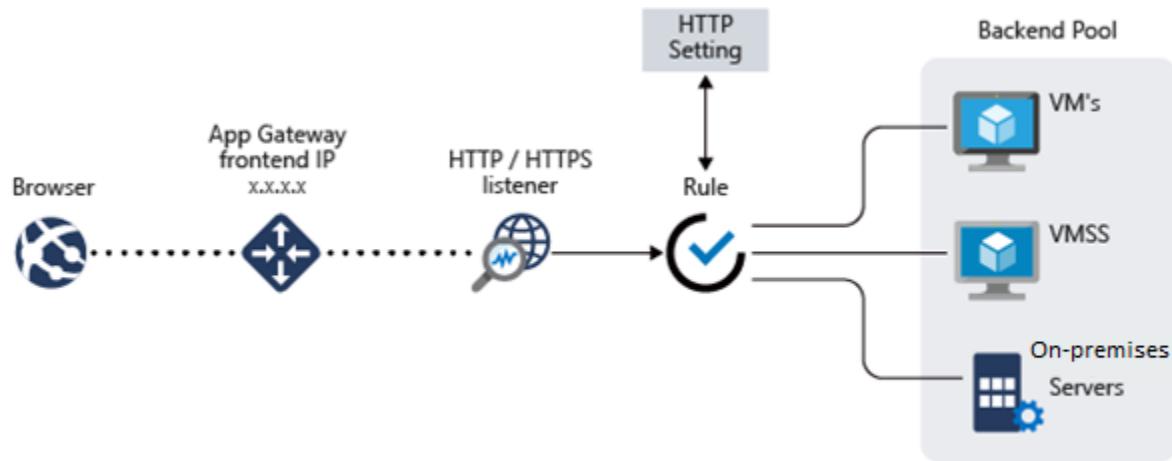
Load balance HTTP(S) traffic in Azure

Design Azure Application Gateway

Completed 100 XP

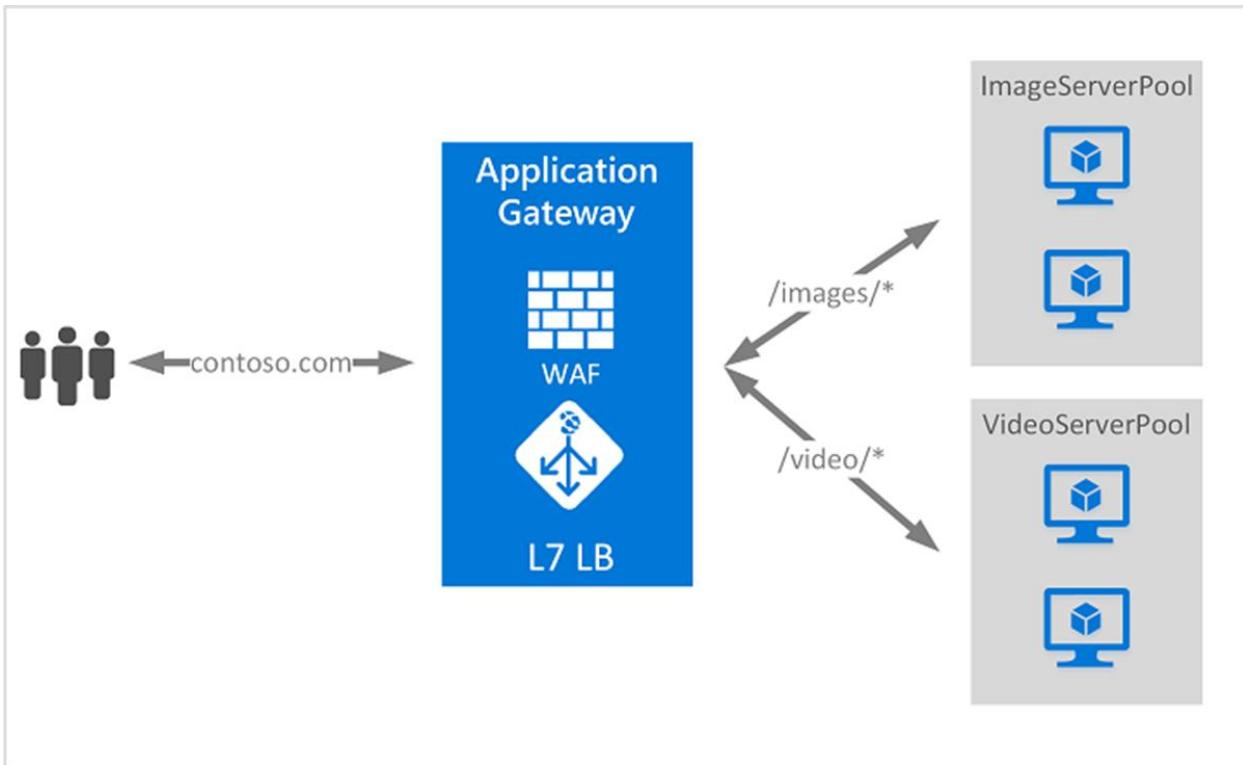
- 10 minutes

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications. Traditional load balancers operate at the transport layer (OSI layer 4 - TCP and UDP) and route traffic based on source IP address and port, to a destination IP address and port.



Application Gateway can make routing decisions based on additional attributes of an HTTP request, for example URI path or host headers. For example, you can route traffic based on the incoming URL. So, if /images is in the incoming URL, you can route traffic to a specific set of servers (known as a pool) configured for images. If /video is in the URL, that traffic is routed to another pool that's optimized for videos.

This type of routing is known as application layer (OSI layer 7) load balancing. Azure Application Gateway can do URL-based routing and more.



Application Gateway features

- Support for the HTTP, HTTPS, HTTP/2 and WebSocket protocols.
- A web application firewall to protect against web application vulnerabilities.
- End-to-end request encryption.
- Autoscaling, to dynamically adjust capacity as your web traffic load change.
- **Redirection:** Redirection can be used to another site, or from HTTP to HTTPS.
- **Rewrite HTTP headers:** HTTP headers allow the client and server to pass parameter information with the request or the response.
- **Custom error pages:** Application Gateway allows you to create custom error pages instead of displaying default error pages. You can use your own branding and layout using a custom error page.

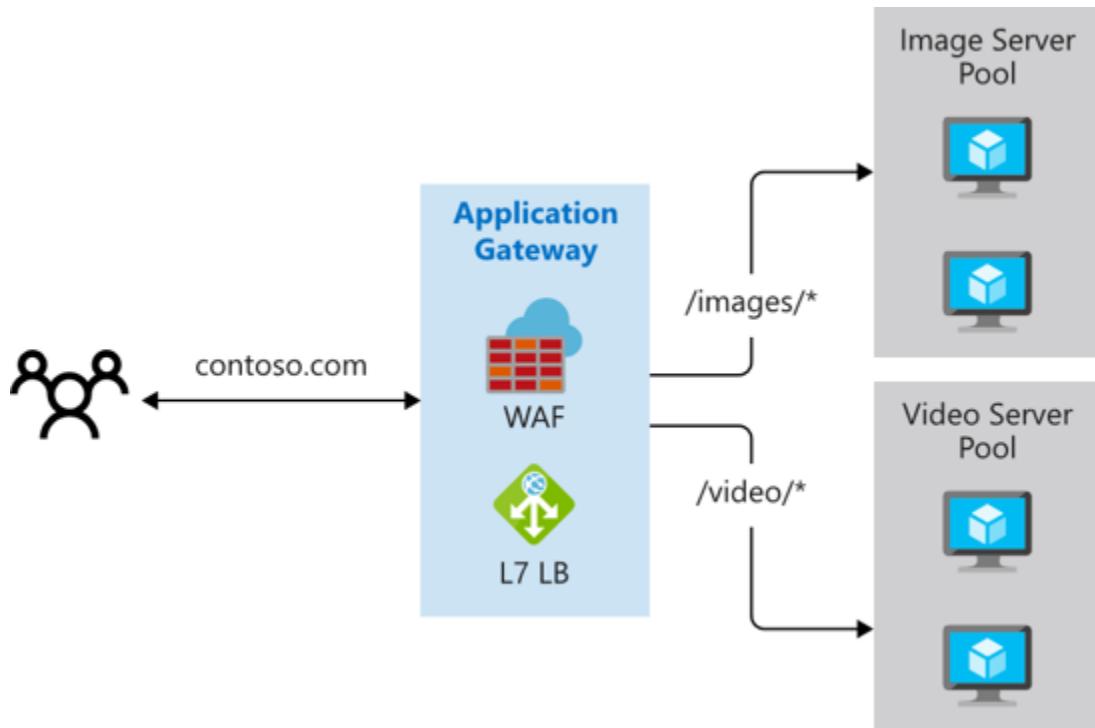
Determine Application Gateway routing

Clients send requests to your web apps to the IP address or DNS name of the gateway. The gateway routes requests to a selected web server in the back-end pool, using a set of rules configured for the gateway to determine where the request should go.

There are two primary methods of routing traffic, path-based routing, and multiple site routing.

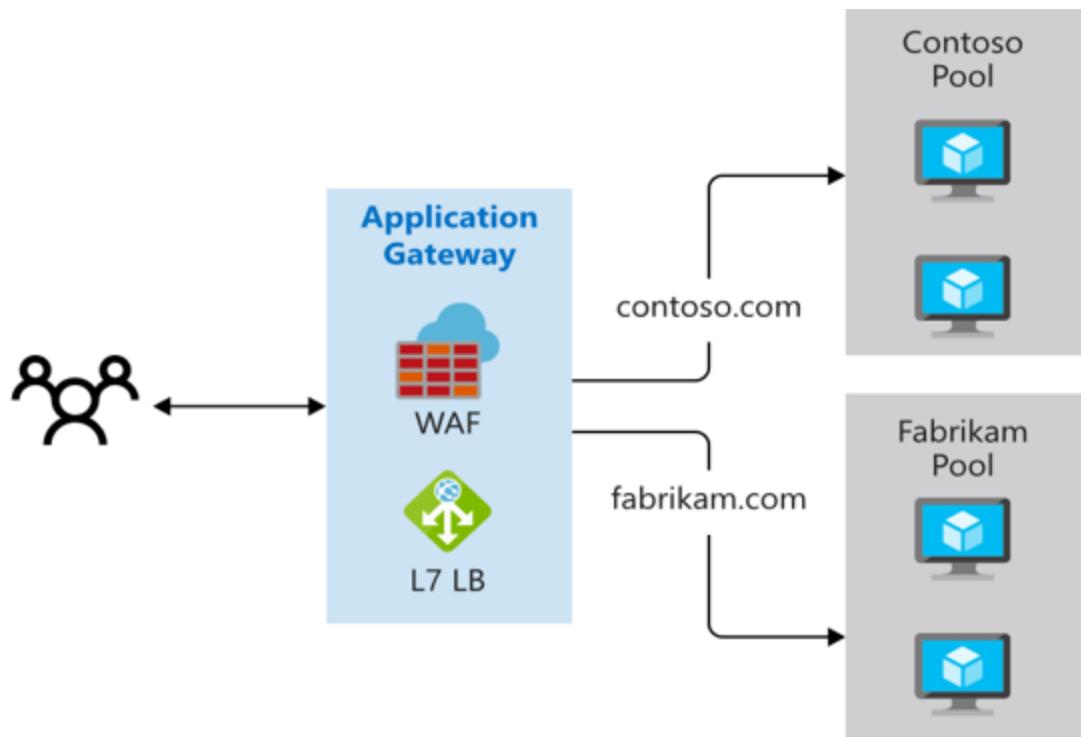
Path-based routing

Path-based routing sends requests with different URL paths different pools of back-end servers. For example, you could direct requests with the path `/video/*` to a back-end pool containing servers that are optimized to handle video streaming, and direct `/images/*` requests to a pool of servers that handle image retrieval.



Multiple site routing

Multiple site routing configures more than one web application on the same application gateway instance. In a multi-site configuration, you register multiple DNS names (CNAMEs) for the IP address of the Application Gateway, specifying the name of each site. Application Gateway uses separate listeners to wait for requests for each site. Each listener passes the request to a different rule, which can route the requests to servers in a different back-end pool. For example, you could direct all requests for <https://contoso.com> to servers in one back-end pool, and requests for <https://fabrikam.com> to another back-end pool. The following diagram shows this configuration.



Multi-site configurations are useful for supporting multi-tenant applications, where each tenant has its own set of virtual machines or other resources hosting a web application.

Choosing an Azure Application Gateway SKU

Application Gateway is available under a Standard_v2 SKU. Web Application Firewall (WAF) is available under a WAF_v2 SKU. The v2 SKU offers performance enhancements and adds support for critical new features like autoscaling, zone redundancy, and support for static VIPs. Existing features under the Standard and WAF SKU continue to be supported in the new v2 SKU.

Review the [feature comparison table](#) between v1 and v2 SKU to determine which SKU meets your deployment needs.

Choosing between Azure Application Gateway v2 and Web Application Firewall V2 SKUs

When choosing whether to deploy an Application Gateway or a Web Application Firewall, there are several factors you must consider, including the scaling strategy you want to follow.

Scaling Application Gateway and WAF v2

Application Gateway and WAF can be configured to scale in two modes:

Autoscaling: With autoscaling enabled, the Application Gateway and WAF v2 SKUs scale up or down based on application traffic requirements. This mode offers better elasticity to your application and eliminates the need to guess the application gateway size or instance count. This mode also allows you to save cost by not requiring the gateway to run at peak provisioned capacity for anticipated maximum traffic load. You must specify a minimum and optionally maximum instance count. Minimum capacity ensures that Application Gateway and WAF v2 don't fall below the minimum instance count specified, even in the absence of traffic. Each instance is roughly equivalent to 10 additional reserved Capacity Units. Zero signifies no reserved capacity and is purely autoscaling in nature. You can also optionally specify a maximum instance count, which ensures that the Application Gateway doesn't scale beyond the specified number of instances. You will only be billed for traffic served by the Gateway. The instance counts can range from 0 to 125. The default value for maximum instance count is 20 if not specified.

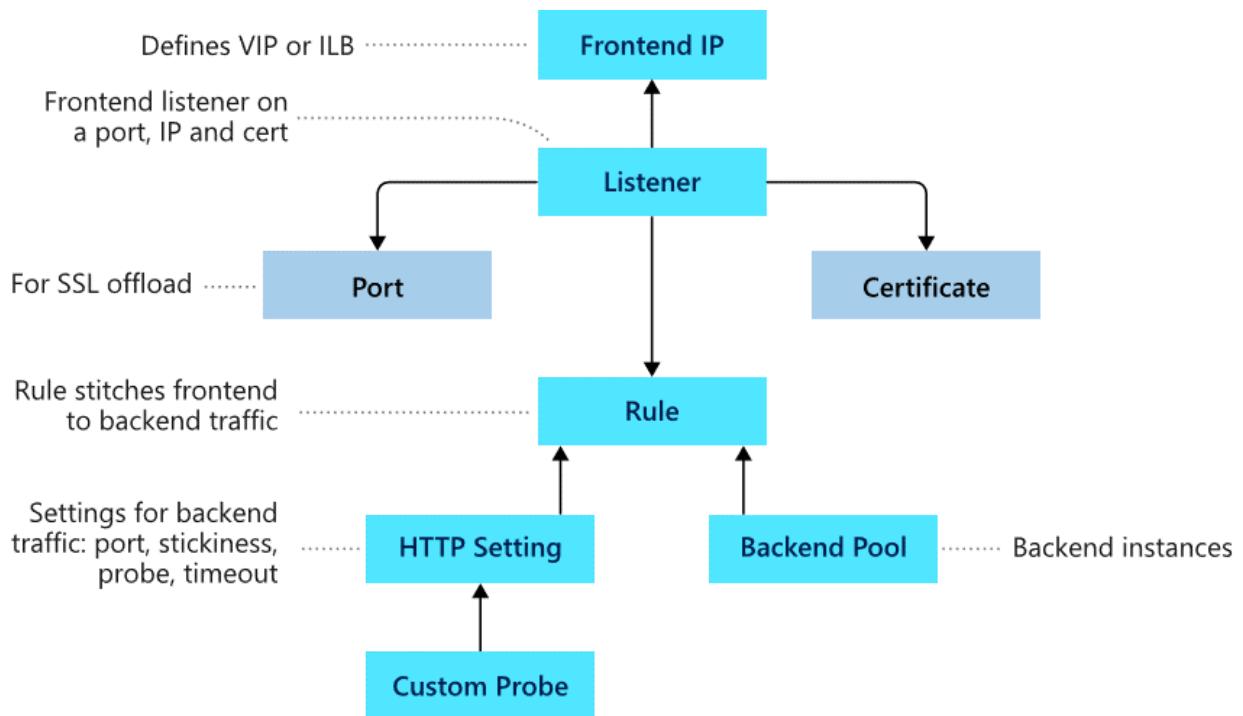
Manual: You can alternatively choose Manual mode where the gateway doesn't autoscale. In this mode, if there is more traffic than the Application Gateway or WAF can handle, it could result in traffic loss. With manual mode, specifying instance count is mandatory. Instance count can vary from 1 to 125 instances.

Configure Azure Application Gateway

200 XP

- 25 minutes

Application Gateway has a series of components that combine to route requests to a pool of web servers and to check the health of these web servers.



Frontend configuration

You can configure the application gateway to have a public IP address, a private IP address, or both. A public IP address is required when you host a back end that clients must access over the Internet via an Internet-facing virtual IP.

Backend configuration

The backend pool is used to route requests to the backend servers that serve the request. Backend pools can be composed of NICs, virtual machine scale sets, public IP addresses, internal IP addresses, fully qualified domain names (FQDN), and multi-tenant back-ends like Azure App Service. You can create an empty backend pool with your application gateway and then add backend targets to the backend pool.

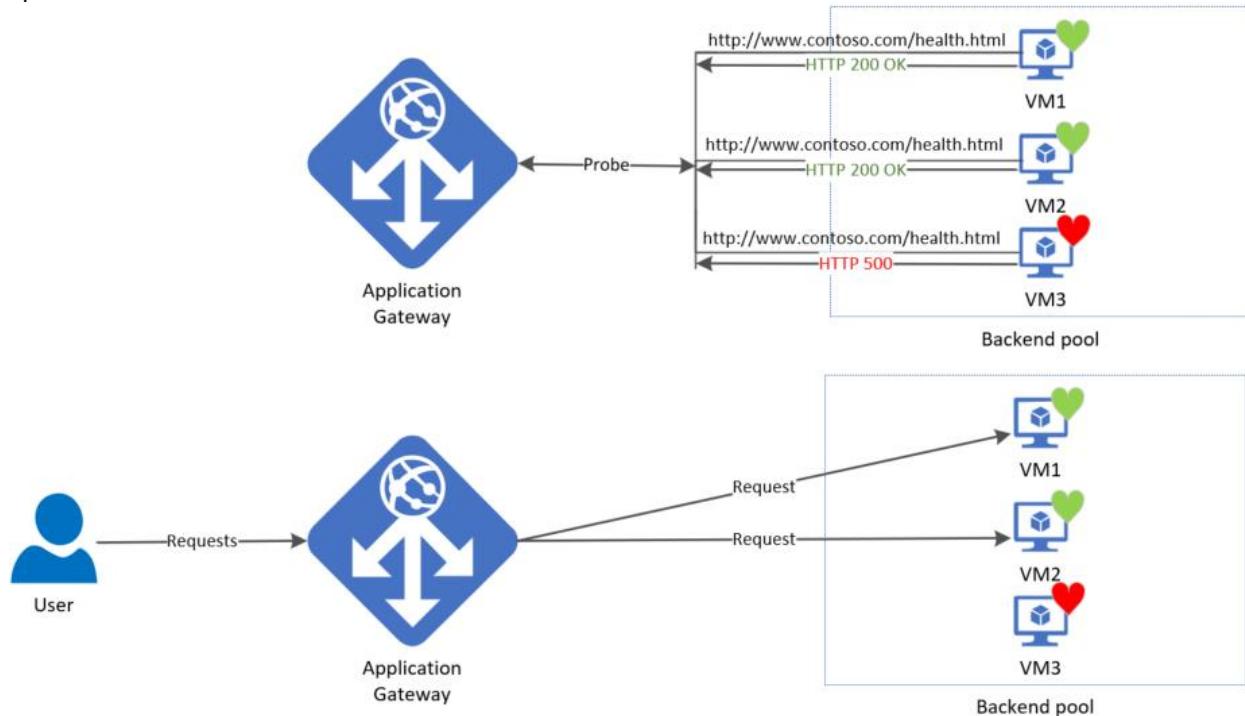
Configure health probes

Azure Application Gateway by default monitors the health of all resources in its backend pool and automatically removes any resource considered unhealthy from the pool. Application Gateway continues to monitor the unhealthy instances and adds them back to the healthy back-end pool once they become available and respond to health probes. By default, Application gateway sends the health probes with the same port that is

defined in the back-end HTTP settings. A custom probe port can be configured using a custom health probe.

The source IP address that the Application Gateway uses for health probes depends on the backend pool:

- If the server address in the backend pool is a public endpoint, then the source address is the application gateway's frontend public IP address.
- If the server address in the backend pool is a private endpoint, then the source IP address is from the application gateway subnet's private IP address space.



Default health probe

An application gateway automatically configures a default health probe when you don't set up any custom probe configurations. The monitoring behavior works by making an HTTP GET request to the IP addresses or FQDN configured in the back-end pool. For default probes if the backend http settings are configured for HTTPS, the probe uses HTTPS to test health of the backend servers.

For example: You configure your application gateway to use back-end servers A, B, and C to receive HTTP network traffic on port 80. The default health monitoring tests the three servers every 30 seconds for a healthy HTTP response with a 30 second timeout

for each request. A healthy HTTP response has a status code between 200 and 399. In this case, the HTTP GET request for the health probe looks like <http://127.0.0.1/>.

If the default probe check fails for server A, the application gateway stops forwarding requests to this server. The default probe continues to check for server A every 30 seconds. When server A responds successfully to one request from a default health probe, application gateway starts forwarding the requests to the server again.

Default health probe settings

The following table lists the default health probe settings:

Probe property	Value	Description
Probe URL	<protocol>://127.0.0.1: <port>/	The protocol and port are inherited from the backend HTTP settings to which the probe is associated
Interval	30	The amount of time in seconds to wait before the next health probe is sent.
Time-out	30	The amount of time in seconds the application gateway waits for a probe response before marking the probe as unhealthy. If a probe returns as healthy, the corresponding backend is immediately marked as healthy.
Unhealthy threshold	3	Governs how many probes to send in case there's a failure of the regular health probe. In v1 SKU, these additional health probes are sent in quick succession to determine the health of the backend quickly and don't wait for the probe interval. In the case of v2 SKU, the health probes wait the interval. The back-end server is marked down after the consecutive probe failure count reaches the unhealthy threshold.

Probe intervals

All instances of Application Gateway probe the backend independent of each other. The same probe configuration applies to each Application Gateway instance. For example, if the probe configuration is to send health probes every 30 seconds and the application gateway has two instances, then both instances send the health probe every 30 seconds.

If there are multiple listeners, then each listener probes the backend independent of each other.

Custom health probe

Custom probes give you more granular control over the health monitoring. When using custom probes, you can configure a custom hostname, URL path, probe interval, and how many failed responses to accept before marking the back-end pool instance as unhealthy, etc.

Custom health probe settings

The following table provides definitions for the properties of a custom health probe.

Probe property	Description
Name	Name of the probe. This name is used to identify and refer to the probe in back-end HTTP settings.
Protocol	Protocol used to send the probe. This property must match with the protocol defined in the back-end HTTP settings it is associated to
Host	Host name to send the probe with. In v1 SKU, this value is used only for the host header of the probe request. In v2 SKU, it is used both as host header and SNI
Path	Relative path of the probe. A valid path starts with '/'
Port	If defined, this property is used as the destination port. Otherwise, it uses the same port as the HTTP settings that it is associated to. This property is only available in the v2 SKU
Interval	Probe interval in seconds. This value is the time interval between two consecutive probes
Time-out	Probe time-out in seconds. If a valid response isn't received within this time-out period, the probe is marked as failed
Unhealthy threshold	Probe retry count. The back-end server is marked down after the consecutive probe failure count reaches the unhealthy threshold

Probe matching

By default, an HTTP(S) response with status code between 200 and 399 is considered healthy. Custom health probes additionally support two matching criteria. Matching criteria can be used to optionally modify the default interpretation of what makes a healthy response.

The following are matching criteria:

- HTTP response status code match - Probe matching criterion for accepting user specified http response code or response code ranges. Individual comma-separated response status codes or a range of status code is supported.
- HTTP response body match - Probe matching criterion that looks at HTTP response body and matches with a user specified string. The match only looks for presence of user specified string in response body and isn't a full regular expression match.

Match criteria can be specified using the New-AzApplicationGatewayProbeHealthResponseMatch cmdlet.

Configure listeners

A listener is a logical entity that checks for incoming connection requests by using the port, protocol, host, and IP address. When you configure a listener, you must enter values that match the corresponding values in the incoming request on the gateway.

When you create an application gateway by using the Azure portal, you also create a default listener by choosing the protocol and port for the listener. You can choose whether to enable HTTP2 support on the listener. After you create the application gateway, you can edit the settings of that default listener (appGatewayHttpListener) or create new listeners.

Home > Application gateways > TestAppGwv2 - Listeners > appGatewayHttpListener

appGatewayHttpListener

TestAppGwv2

Save Discard Delete

Name
appGatewayHttpListener

* Frontend IP configuration
appGatewayFrontendIP

* Frontend port
appGatewayFrontendPort (443)

Protocol
HTTPS

* Certificate
scrap

Renew or edit selected certificate

Associated rule
rule1

CUSTOM ERROR PAGES ⓘ

ERROR CODE	URL
Forbidden - 403	https://mycustomerrorpages.blob.core.v...
Bad Gateway - 502	https://mycustomerrorpages.blob.core.wind...

Listener type

When you create a new listener, you must choose between basic and multi-site.

- **Basic:** All requests for any domain will be accepted and forwarded to backend pools.
- **Multi-site:** Forward requests to different backend pools based on the host header or host names. You must specify a host name that matches with the incoming

request. This is because Application Gateway relies on HTTP 1.1 host headers to host more than one website on the same public IP address and port.

Order of processing listeners

For the v1 SKU, requests are matched according to the order of the rules and the type of listener. If a rule with basic listener comes first in the order, it's processed first and accepts any request for that port and IP combination. To avoid this, configure the rules with multi-site listeners first and push the rule with the basic listener to the last in the list.

For the v2 SKU, multi-site listeners are processed before basic listeners.

Front-end IP address

Choose the front-end IP address that you plan to associate with this listener. The listener will listen to incoming requests on this IP.

Front-end port

Choose the front-end port. Select an existing port or create a new one. Choose any value from the allowed range of ports. You can use not only well-known ports, such as 80 and 443, but any allowed custom port that's suitable. A port can be used for public-facing listeners or private-facing listeners.

Protocol

Choose HTTP or HTTPS:

- **HTTP:** traffic between the client and the application gateway is unencrypted.
- **HTTPS:** enables TLS termination or end-to-end TLS encryption. The TLS connection terminates at the application gateway. Traffic between the client and the application gateway is encrypted. If you want end-to-end TLS encryption, you must choose HTTPS and configure the back-end HTTP setting. This ensures that traffic is re-encrypted when it travels from the application gateway to the back end.

To configure TLS termination and end-to-end TLS encryption, you must add a certificate to the listener to enable the application gateway to derive a symmetric key. The symmetric key is used to encrypt and decrypt the traffic that's sent to the gateway. The

gateway certificate must be in Personal Information Exchange (PFX) format. This format lets you export the private key that the gateway uses to encrypt and decrypt traffic.

Redirection overview

You can use application gateway to redirect traffic. It has a generic redirection mechanism which allows for redirecting traffic received at one listener to another listener or to an external site. This simplifies application configuration, optimizes the resource usage, and supports new redirection scenarios including global and path-based redirection.

A common redirection scenario for many web applications is to support automatic HTTP to HTTPS redirection to ensure all communication between application and its users occurs over an encrypted path. In the past, customers have used techniques such as creating a dedicated backend pool whose sole purpose is to redirect requests it receives on HTTP to HTTPS. With redirection support in Application Gateway, you can accomplish this simply by adding a new redirect configuration to a routing rule and specifying another listener with HTTPS protocol as the target listener.

The following types of redirection are supported:

- 301 Permanent Redirect
- 302 Found
- 303 See Other
- 307 Temporary Redirect

Application Gateway redirection support offers the following capabilities:

- **Global redirection:** Redirects from one listener to another listener on the gateway. This enables HTTP to HTTPS redirection on a site.
- **Path-based redirection:** Enables HTTP to HTTPS redirection only on a specific site area, for example a shopping cart area denoted by /cart/*.
- **Redirect to external site:** Requires a new redirect configuration object, which specifies the target listener or external site to which redirection is desired. The configuration element also supports options to enable appending the URI path and query string to the redirected URL. The redirect configuration is attached to the source listener via a new rule.

For more information on configuring redirection in Application Gateway, see [URL path-based redirection using PowerShell - Azure Application Gateway | Microsoft Docs](#).

Application Gateway request routing rules

When you create an application gateway using the Azure portal, you create a default rule (rule1). This rule binds the default listener (appGatewayHttpListener) with the default back-end pool (appGatewayBackendPool) and the default back-end HTTP settings (appGatewayBackendHttpSettings). After you create the gateway, you can edit the settings of the default rule or create new rules.

Rule types:

- Basic forwards all requests on the associated listener (for example, blog.contoso.com/*) to a single back-end pool.
- Path-based routes requests from specific URL paths to specific back-end pools.

Order of processing rules

For the v1 and v2 SKU, pattern matching of incoming requests is processed in the order that the paths are listed in the URL path map of the path-based rule. If a request matches the pattern in two or more paths in the path map, the path that's listed first is matched. And the request is forwarded to the back end that's associated with that path.

Associated listener

Associate a listener to the rule so that the request-routing rule that's associated with the listener is evaluated to determine the back-end pool to route the request to.

Associated back-end pool

Associate to the rule the back-end pool that contains the back-end targets that serve requests that the listener receives.

For a basic rule, only one back-end pool is allowed. All requests on the associated listener are forwarded to that back-end pool.

For a path-based rule, add multiple back-end pools that correspond to each URL path. The requests that match the URL path that's entered are forwarded to the corresponding back-end pool. Also, add a default back-end pool. Requests that don't match any URL path in the rule are forwarded to that pool.

Associated back-end HTTP setting

Add a back-end HTTP setting for each rule. Requests are routed from the application gateway to the back-end targets by using the port number, protocol, and other information that's specified in this setting.

For a basic rule, only one back-end HTTP setting is allowed. All requests on the associated listener are forwarded to the corresponding back-end targets by using this HTTP setting.

For a path-based rule, add multiple back-end HTTP settings that correspond to each URL path. Requests that match the URL path in this setting are forwarded to the corresponding back-end targets by using the HTTP settings that correspond to each URL path. Also, add a default HTTP setting. Requests that don't match any URL path in this rule are forwarded to the default back-end pool by using the default HTTP setting.

Redirection setting

If redirection is configured for a basic rule, all requests on the associated listener are redirected to the target. This is global redirection. If redirection is configured for a path-based rule, only requests in a specific site area are redirected. An example is a shopping cart area that's denoted by /cart/*. This is path-based redirection.

Redirection type

Choose the type of redirection required: Permanent(301), Temporary(307), Found(302), or See other(303).

Redirection target

Choose another listener or an external site as the redirection target.

Listener

Choose listener as the redirection target to redirect traffic from one listener to another on the gateway.

External site

Choose external site when you want to redirect the traffic on the listener that's associated with this rule to an external site. You can choose to include the query string

from the original request in the request that's forwarded to the redirection target. You can't forward the path to the external site that was in the original request.

Rewrite HTTP headers and URL

By using rewrite rules, you can add, remove, or update HTTP(S) request and response headers as well as URL path and query string parameters as the request and response packets move between the client and backend pools via the application gateway.

The headers and URL parameters can be set to static values or to other headers and server variables. This helps with important use cases, such as extracting client IP addresses, removing sensitive information about the backend, adding more security, and so on.

Configure URL-based routing

URL Path Based Routing allows you to route traffic to back-end server pools based on URL Paths of the request. One use case is to route requests for different content types to different backend server pools.

For the v1 SKU, rules are processed in the order they are listed in the portal. If a basic listener is listed first and matches an incoming request, it gets processed by that listener. For the v2 SKU, exact matches have higher precedence. However, it is highly recommended to configure multi-site listeners first prior to configuring a basic listener. This ensures that traffic gets routed to the right back end.

UrlPathMap configuration element

The urlPathMap element is used to specify Path patterns to back-end server pool mappings. The following code example is the snippet of urlPathMap element from template file.

JSONCopy

```
"urlPathMaps": [{

  "name": "{urlpathMapName}",

  "id": "/subscriptions/{subscriptionId}.../microsoft.network/applicationGateways/{gatewayName}/urlPathMaps/{urlpathMapName}",
```

```
"properties": {  
    "defaultBackendAddressPool": {  
        "id":  
        "/subscriptions/{subscriptionId}.../microsoft.network/applicationGateways/{gatewayName}/backendAddressPools/{poolName1}"  
    },  
    "defaultBackendHttpSettings": {  
        "id":  
        "/subscriptions/{subscriptionId}.../microsoft.network/applicationGateways/{gatewayName}/backendHttpSettingsList/{settingname1}"  
    },  
    "pathRules": [ {  
        "name": "{pathRuleName}",  
        "properties": {  
            "paths": [  
                "{pathPattern}"  
            ],  
            "backendAddressPool": {  
                "id":  
                "/subscriptions/{subscriptionId}.../microsoft.network/applicationGateways/{gatewayName}/backendAddressPools/{poolName2}"  
            },  
            "backendHttpsettings": {  
                "id":  
                "/subscriptions/{subscriptionId}.../microsoft.network/applicationGateways/{gatewayName}/backendHttpsettingsList/{settingName2}"  
            }  
        }  
    }]  
}
```

PathPattern

PathPattern is a list of path patterns to match. Each must start with / and the only place a "*" is allowed is at the end following a "/." The string fed to the path matcher does not include any text after the first? or #, and those chars are not allowed here. Otherwise, any characters allowed in a URL are allowed in PathPattern. The supported patterns depend on whether you deploy Application Gateway v1 or v2.

PathBasedRouting rule

RequestRoutingRule of type PathBasedRouting is used to bind a listener to a urlPathMap. All requests that are received for this listener are routed based on policy specified in urlPathMap.

Configure rewrite policies in Application Gateway

Application Gateway allows you to rewrite selected content of requests and responses. With this feature, you can translate URLs, query string parameters as well as modify request and response headers. It also allows you to add conditions to ensure that the URL or the specified headers are rewritten only when certain conditions are met. These conditions are based on the request and response information.

HTTP header and URL rewrite features are only available for the Application Gateway v2 SKU.

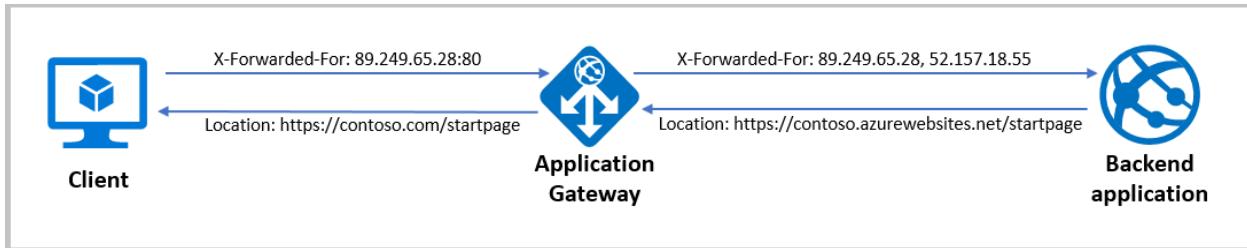
Supported rewrite types

Application Gateway supports multiple rewrite types.

Request and response headers

HTTP headers allow a client and server to pass additional information with a request or response. By rewriting these headers, you can accomplish important tasks, such as adding security-related header fields like HSTS/ X-XSS-Protection, removing response header fields that might reveal sensitive information, and removing port information from X-Forwarded-For headers.

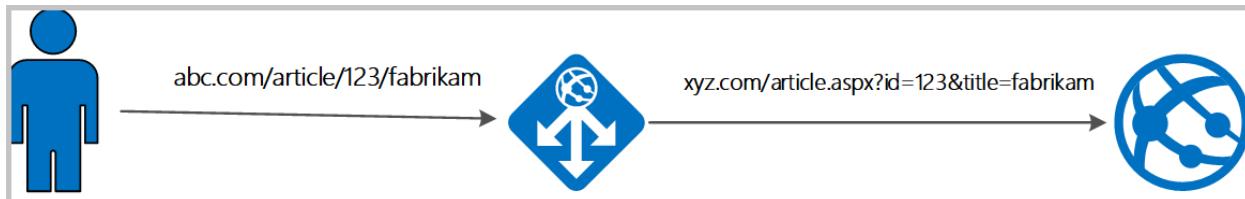
Application Gateway allows you to add, remove, or update HTTP request and response headers while the request and response packets move between the client and back-end pools.



URL path and query string

With URL rewrite capability in Application Gateway, you can:

- Rewrite the host name, path, and query string of the request URL
- Choose to rewrite the URL of all requests on a listener or only those requests which match one or more of the conditions you set. These conditions are based on the request and response properties (request, header, response header and server variables).
- Choose to route the request (select the backend pool) based on either the original URL or the rewritten URL.



Rewrite actions

You use rewrite actions to specify the URL, request headers or response headers that you want to rewrite and the new value to which you intend to rewrite them to. The value of a URL or a new or existing header can be set to these types of values:

- Text
- Request header. To specify a request header, you need to use the syntax {http_req_headerName}
- Response header. To specify a response header, you need to use the syntax {http_resp_headerName}
- Server variable. To specify a server variable, you need to use the syntax {var_serverVariable}. See the list of supported server variables

A combination of text, a request header, a response header, and a server variable.

Rewrite conditions

You can use rewrite conditions, an optional configuration, to evaluate the content of HTTP(S) requests and responses and perform a rewrite only when one or more conditions are met. The application gateway uses these types of variables to evaluate the content of requests and responses:

- HTTP headers in the request
- HTTP headers in the response
- Application Gateway server variables

You can use a condition to evaluate whether a specified variable is present, whether a specified variable matches a specific value, or whether a specified variable matches a specific pattern.

Rewrite configuration

To configure a rewrite rule, you need to create a rewrite rule set and add the rewrite rule configuration in it.

A rewrite rule set contains:

- **Request routing rule association:** The rewrite configuration is associated to the source listener via the routing rule. When you use a basic routing rule, the rewrite configuration is associated with a source listener and is a global header rewrite. When you use a path-based routing rule, the rewrite configuration is defined on the URL path map. In that case, it applies only to the specific path area of a site. You can create multiple rewrite sets and apply each rewrite set to multiple listeners. But you can apply only one rewrite set to a specific listener.
- **Rewrite Condition:** It is an optional configuration. Rewrite conditions evaluate the content of the HTTP(S) requests and responses. The rewrite action will occur if the HTTP(S) request or response matches the rewrite condition. If you associate more than one condition with an action, the action occurs only when all the conditions are met. In other words, the operation is a logical AND operation.
- **Rewrite type:** There are three types of rewrites available:
 - Rewriting request headers
 - Rewriting response headers
 - Rewriting URL components
 - **URL path:** The value to which the path is to be rewritten to.

- **URL Query String:** The value to which the query string is to be rewritten to.
- **Re-evaluate path map:** Used to determine whether the URL path map is to be re-evaluated or not. If kept unchecked, the original URL path will be used to match the path-pattern in the URL path map. If set to true, the URL path map will be re-evaluated to check the match with the rewritten path.
Enabling this switch helps in routing the request to a different backend pool post rewrite.

For more information on Configuring rewrites in application Gateway, see [Rewrite HTTP headers and URL with Azure Application Gateway | Microsoft Docs](#).

Exercise: deploy Azure Application Gateway

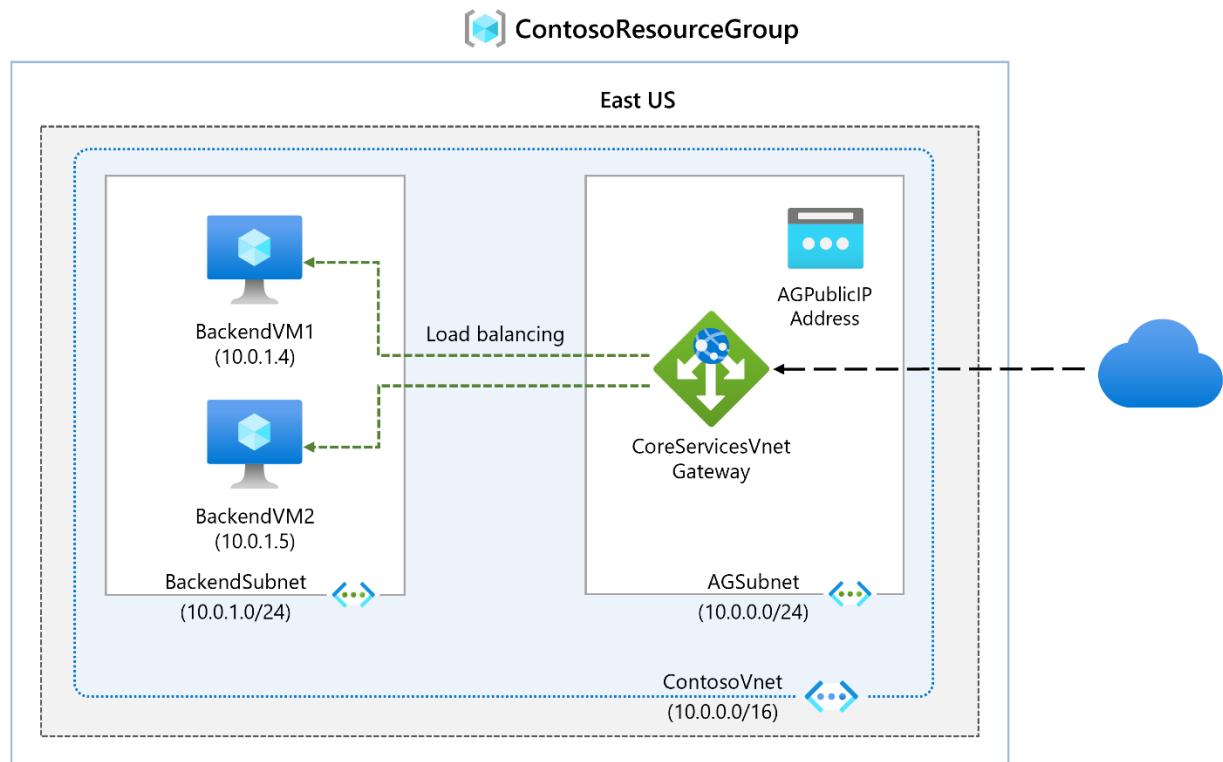
Completed 100 XP

- 8 minutes

Lab scenario

In this lab, you use the Azure portal to create an application gateway. Then you test it to make sure it works correctly.

Architecture diagram



Objectives

- **Task 1:** Create an application gateway
- **Task 2:** Add backend targets
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 3:** Add backend servers to backend pool
- **Task 4:** Test the application gateway

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

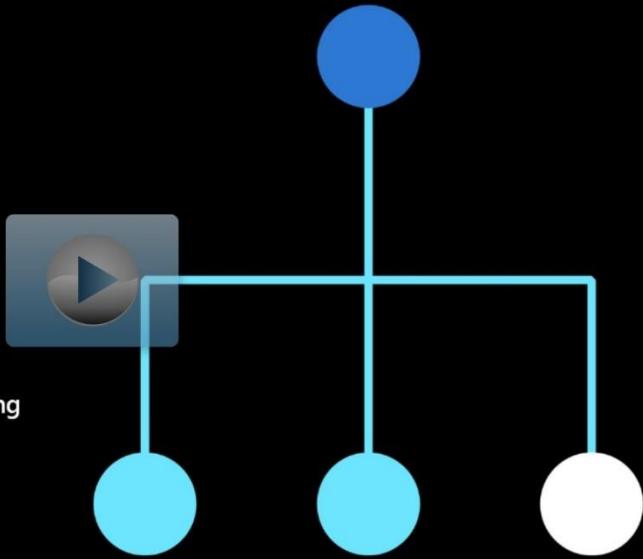
You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



Interactive Lab Simulation:

Deploy Azure Application Gateway

From AZ-700: Designing and Implementing Microsoft Azure Networking Solutions



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Deploy%20Azure%20Application%20Gateway>

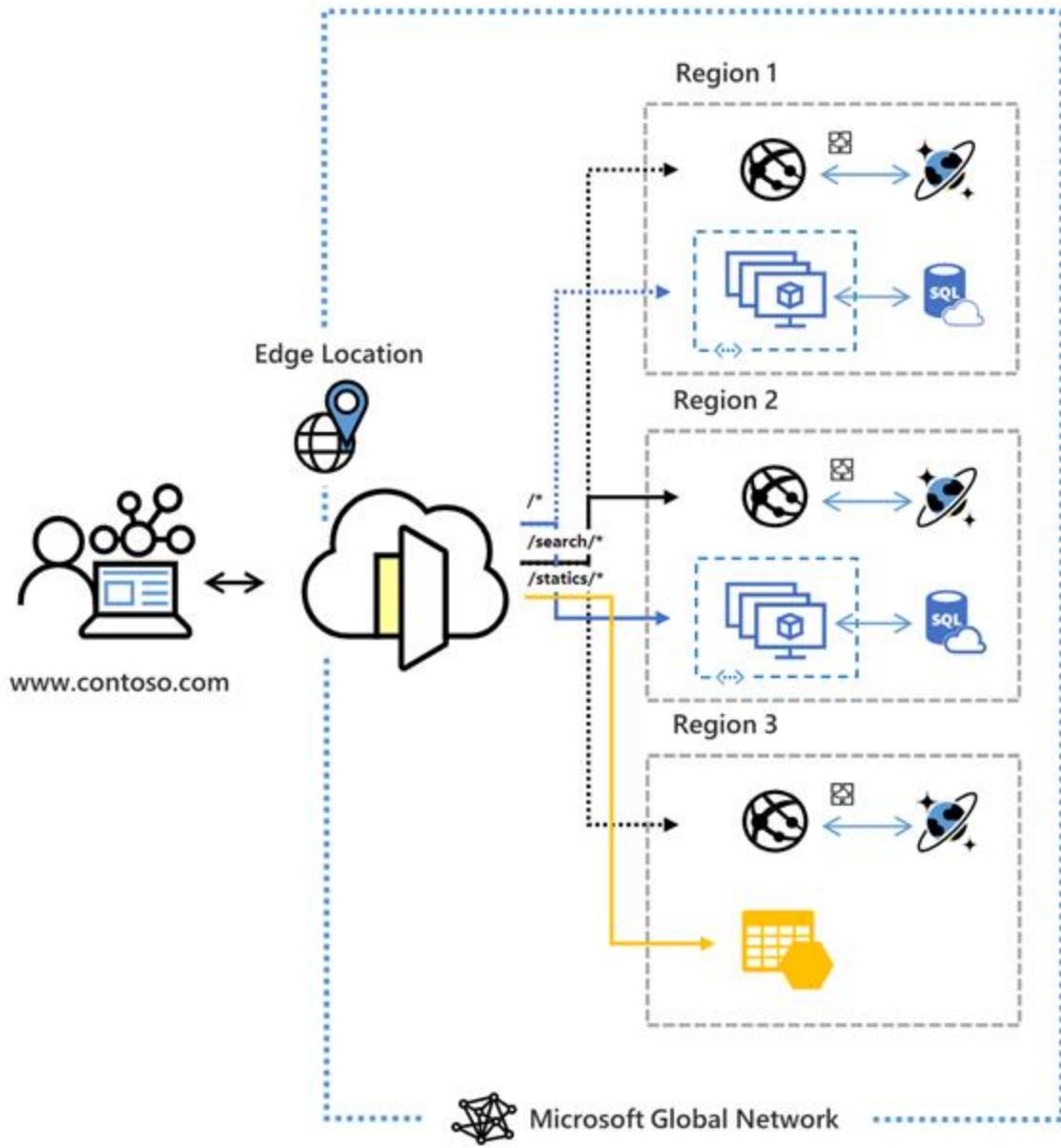
Design and configure Azure Front Door

200 XP

- 14 minutes

Azure Front Door is Microsoft's modern cloud Content Delivery Network (CDN) that provides fast, reliable, and secure access between your users and your applications' static and dynamic web content across the globe. Azure Front Door delivers your content using the Microsoft's global edge network with hundreds of global and local POPs distributed around the world close to both your enterprise and consumer end users.

Many organizations have applications they want to make available to their customers, their suppliers, and almost certainly their users. The tricky part is making sure those applications are highly available. In addition, they need to be able to quickly respond while being appropriately secured. Azure Front Door provides different SKUs (pricing tiers) that meet these requirements. Let's briefly review the features and benefits of these SKUs so you can determine which option best suits your requirements.



A secure, modern cloud CDN provides a distributed platform of servers. This helps minimize latency when users are accessing webpages. Historically, IT staff might have used a CDN and a web application firewall to control HTTP and HTTPS traffic flowing to and from target applications.

If an organization uses Azure, they might achieve these goals by implementing the products described in the following table

Product	Description
Azure Front Door	Enables an entry point to your apps positioned in the Microsoft global edge network. Provides faster, more secure, and scalable access to your web applications.
Azure Content Delivery Network	Delivers high-bandwidth content to your users by caching their content at strategically placed physical nodes around the world.
Azure Web Application Firewall	Helps provide centralized, greater protection for web applications from common exploits and vulnerabilities.

Azure Front Door tier comparison

Azure Front Door is offered in 2 different tiers, Azure Front Door Standard and Azure Front Door Premium. Azure Front Door Standard and Premium tier combines capabilities of Azure Front Door (classic), Azure CDN Standard from Microsoft (classic), and Azure WAF into a single secure cloud CDN platform with intelligent threat protection. Azure Front Door resides in the edge locations and manages user requests to your hosted applications. Users connect to your application through the Microsoft global network. Azure Front Door then routes user requests to the fastest and most available application backend.

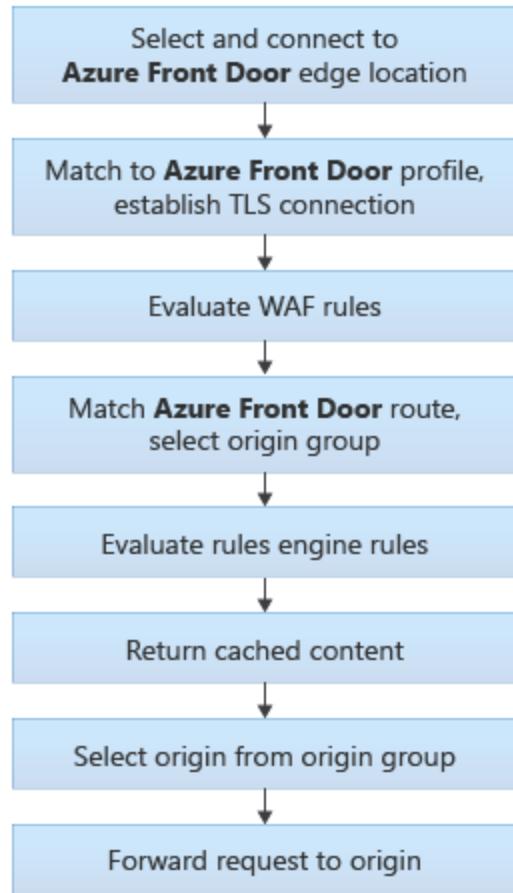
For a comparison of supported features in Azure Front Door, [Review the feature comparison table](#).

Create a Front Door in the Azure portal

Review the following [QuickStart](#) to learn how to create an Azure Front Door profile using the Azure portal. You can create an Azure Front Door profile through Quick Create with basic configurations or through the Custom create which allows a more advanced configuration.

Routing architecture overview

Front Door traffic routing takes place over multiple stages. First, traffic is routed from the client to Front Door. Then, Front Door uses your configuration to determine the origin to send the traffic to. The Front Door web application firewall, routing rules, rules engine, and caching configuration all affect the routing process. The following diagram illustrates the routing architecture:



Configure redirection rules in Front Door

After establishing a connection and completing a TLS handshake, when a request lands on a Front Door environment one of the first things that Front Door does is determine which routing rule to match the request to and then take the defined action in the configuration.

Front Door route rules configuration structure

A Front Door routing rule configuration is composed of two major parts: a "left-hand side" and a "right-hand side". Front Door matches the incoming request to the left-hand side of the route. The right-hand side defines how Front Door processes the request.

Incoming match

The following properties determine whether the incoming request matches the routing rule (or left-hand side):

- HTTP Protocols (HTTP/HTTPS)
- Hosts (for example, [www.foo.com](#), *.bar.com)
- Paths (for example, /, /users/, /file.gif)

These properties are expanded out internally so that every combination of Protocol/Host/Path is a potential match set.

Route data

Front Door speeds up the processing of requests by using caching. If caching is enabled for a specific route, it uses the cached response. If there is no cached response for the request, Front Door forwards the request to the appropriate backend in the configured backend pool.

Route matching

Front Door attempts to match to the **most-specific match first** looking only at the left-hand side of the route. It first matches based on HTTP protocol, then Frontend host, then the Path.

- **Frontend host matching:**
 - Look for any routing with an exact match on the host.
 - If no exact frontend hosts match, reject the request and send a 400 Bad Request error.
- **Path matching:**
 - Look for any routing rule with an exact match on the Path.
 - If no exact match Paths, look for routing rules with a wildcard Path that matches.
 - If no routing rules are found with a matching Path, then reject the request and return a 400: Bad Request error HTTP response.

If there are no routing rules for an exact-match frontend host with a catch-all route Path (/*), then there will not be a match to any routing rule.

Azure Front Door redirects traffic at each of the following levels: protocol, hostname, path, query string. These functionalities can be configured for individual microservices since the redirection is path-based. This can simplify application configuration by optimizing resource usage and supports new redirection scenarios including global and path-based redirection.

ROUTE DETAILS

Once a route for a Front Door is matched, the configuration below defines the behavior of the route - forward and serve from the cache, or redirect. [Learn more](#)

Route type 

Forward Redirect

Redirect type 

Moved (301)



Redirect protocol 

- HTTPS only
 HTTP only
 Match request

Destination host 

Preserve Replace

Destination path 

Preserve Replace

Query string 

Preserve Replace

Destination fragment 

Example: section-header-2

Redirection types

A redirect type sets the response status code for the clients to understand the purpose of the redirect. The following types of redirection are supported:

Redirection type	Action	Description
301	Moved permanently	Indicates that the target resource has been assigned a new permanent URI. Any future references to this resource will use one of the enclosed URIs. Use 301 status code for HTTP to HTTPS redirection.
302	Found	Indicates that the target resource is temporarily under a different URI. Since the redirection can change on occasion, the client should continue to use the effective request URI for future requests.
307	Temporary redirect	Indicates that the target resource is temporarily under a different URI. The user agent MUST NOT change the request method if it does an automatic redirection to that URI. Since the redirection can change over time, the client ought to continue using the original effective request URI for future requests.
308	Permanent redirect	Indicates that the target resource has been assigned a new permanent URI. Any future references to this resource should use one of the enclosed URIs.

Redirection protocol

You can set the protocol that will be used for redirection. The most common use case of the redirect feature is to set HTTP to HTTPS redirection.

- **HTTPS only:** Set the protocol to HTTPS only, if you're looking to redirect the traffic from HTTP to HTTPS. Azure Front Door recommends that you should always set the redirection to HTTPS only.
- **HTTP only:** Redirects the incoming request to HTTP. Use this value only if you want to keep your traffic HTTP that is, non-encrypted.
- **Match request:** This option keeps the protocol used by the incoming request. So, an HTTP request remains HTTP and an HTTPS request remains HTTPS post redirection.

Destination host

As part of configuring a redirect routing, you can also change the hostname or domain for the redirect request. You can set this field to change the hostname in the URL for the redirection or otherwise preserve the hostname from the incoming request. So, using this field you can redirect all requests sent on https://www.contoso.com/* to https://www.fabrikam.com/*.

Destination path

For cases where you want to replace the path segment of a URL as part of redirection, you can set this field with the new path value. Otherwise, you can choose to preserve the path value as part of redirect. So, using this field, you can redirect all requests sent to https://www.contoso.com/* to <https://www.contoso.com/redirected-site>.

Destination fragment

The destination fragment is the portion of URL after '#', which is used by the browser to land on a specific section of a web page. You can set this field to add a fragment to the redirect URL.

Query string parameters

You can also replace the query string parameters in the redirected URL. To replace any existing query string from the incoming request URL, set this field to 'Replace' and then set the appropriate value. Otherwise, keep the original set of query strings by setting the field to 'Preserve'. As an example, using this field, you can redirect all traffic sent to <https://www.contoso.com/foo/bar> to https://www.contoso.com/foo/bar?utm_referrer=https%3A%2F%2Fwww.bing.com%2F.

Configure rewrite policies

Azure Front Door supports URL rewrite by configuring an optional Custom Forwarding Path to use when constructing the request to forward to the backend. By default, if a custom forwarding path isn't provided, the Front Door will copy the incoming URL path to the URL used in the forwarded request. The Host header used in the forwarded request is as configured for the selected backend. Read Backend Host Header to learn what it does and how you can configure it.

The powerful part of URL rewrite is that the custom forwarding path will copy any part of the incoming path that matches to a wildcard path to the forwarded path.

Configure health probes, including customization of HTTP response codes

To determine the health and proximity of each backend for a given Front Door environment, each Front Door environment periodically sends a synthetic HTTP/HTTPS

request to each of your configured backends. Front Door then uses these responses from the probe to determine the "best" backend resources to route your client requests.

Since Front Door has many edge environments globally, health probe volume for your backends can be quite high - ranging from 25 requests every minute to as high as 1200 requests per minute, depending on the health probe frequency configured. With the default probe frequency of 30 seconds, the probe volume on your backend should be about 200 requests per minute.

Supported HTTP methods for health probes

Front Door supports sending probes over either HTTP or HTTPS protocols. These probes are sent over the same TCP ports configured for routing client requests and cannot be overridden.

Front Door supports the following HTTP methods for sending the health probes:

GET: The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI.

HEAD: The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response. Because it has lower load and cost on your backends, for new Front Door profiles, by default, the probe method is set as HEAD.

Health probe responses

The following table describes responses to the health probe:

Response	Description
Determining Health	A 200 OK status code indicates the backend is healthy. Everything else is considered a failure. If for any reason (including network failure) a valid HTTP response isn't received for a probe, the probe is counted as a failure.
Measuring Latency	Latency is the wall-clock time measured from the moment immediately before the probe request is sent to the moment the last byte of the response is received. A new TCP connection is used for each request, so this measurement isn't biased towards backends with existing warm connections.

Azure Front Door uses the same three-step process below across all algorithms to determine health.

1. Exclude disabled backends.
2. Exclude backends that have health probe errors:
 - This selection is done by looking at the last n health probe responses. If at least x are healthy, the backend is considered healthy.
 - n is configured by changing the SampleSize property in load-balancing settings.
 - x is configured by changing the SuccessfulSamplesRequired property in load-balancing settings.
3. For the sets of healthy backends in the backend pool, Front Door additionally measures and maintains the latency (round-trip time) for each backend.

If you have a single backend in your backend pool, you can choose to disable the health probes reducing the load on your application backend. Even if you have multiple backends in the backend pool but only one of them is in enabled state, you can disable health probes.

Secure Front Door with SSL

Use the HTTPS protocol on your custom domain (for example, <https://www.contoso.com>), you ensure that your sensitive data is delivered securely via TLS/SSL encryption when it's sent across the internet. When your web browser is connected to a web site via HTTPS, it validates the web site's security certificate and verifies that it was issued by a legitimate certificate authority. This process provides security and protects your web applications from attacks.

Some of the key attributes of the custom HTTPS feature are:

- **No extra cost:** There are no costs for certificate acquisition or renewal and no extra cost for HTTPS traffic.
- **Simple enablement:** One-click provisioning is available from the Azure portal. You can also use REST API or other developer tools to enable the feature.
- **Complete certificate management:** All certificate procurement and management is handled for you. Certificates are automatically provisioned and renewed before expiration, which removes the risks of service interruption because of a certificate expiring.

You can enable the HTTPS protocol for a custom domain that's associated with your Front Door under the frontend hosts section.

For more information on how to configure HTTPS on Front door, see [Tutorial - Configure HTTPS on a custom domain for Azure Front Door | Microsoft Docs](#).

Exercise: create a Front Door for a highly available web application

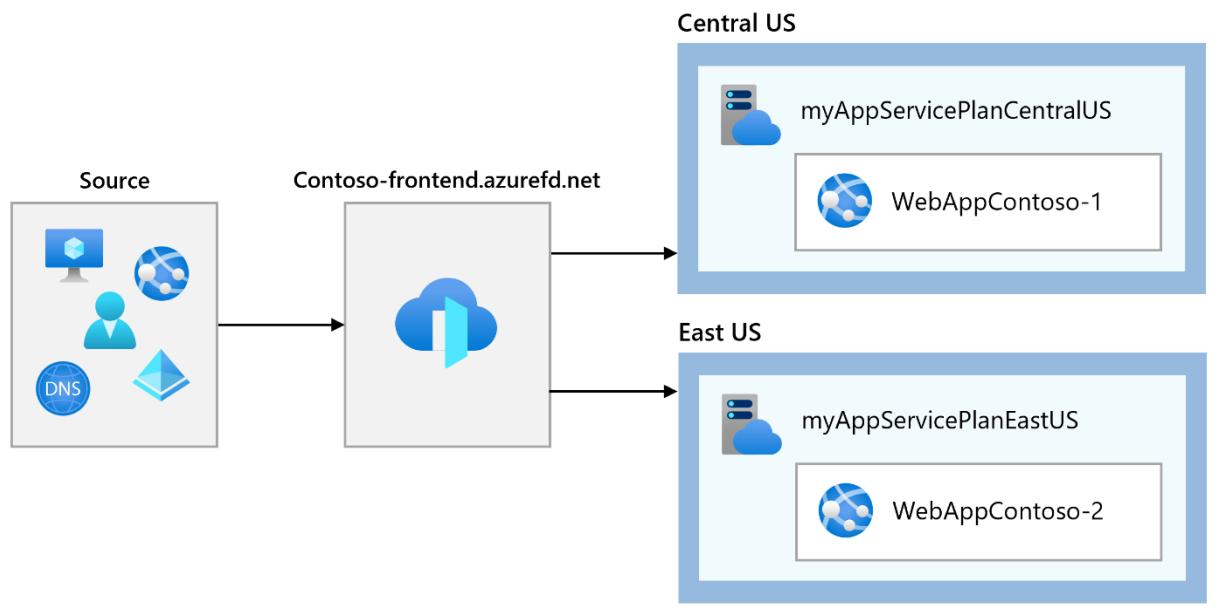
Completed 100 XP

- 7 minutes

Lab scenario

In this lab, you set up an Azure Front Door configuration that pools two instances of a web application that runs in different Azure regions. This configuration directs traffic to the nearest site that runs the application. Azure Front Door continuously monitors the web application. You demonstrate automatic failover to the next available site when the nearest site is unavailable.

Architecture diagram



Objectives

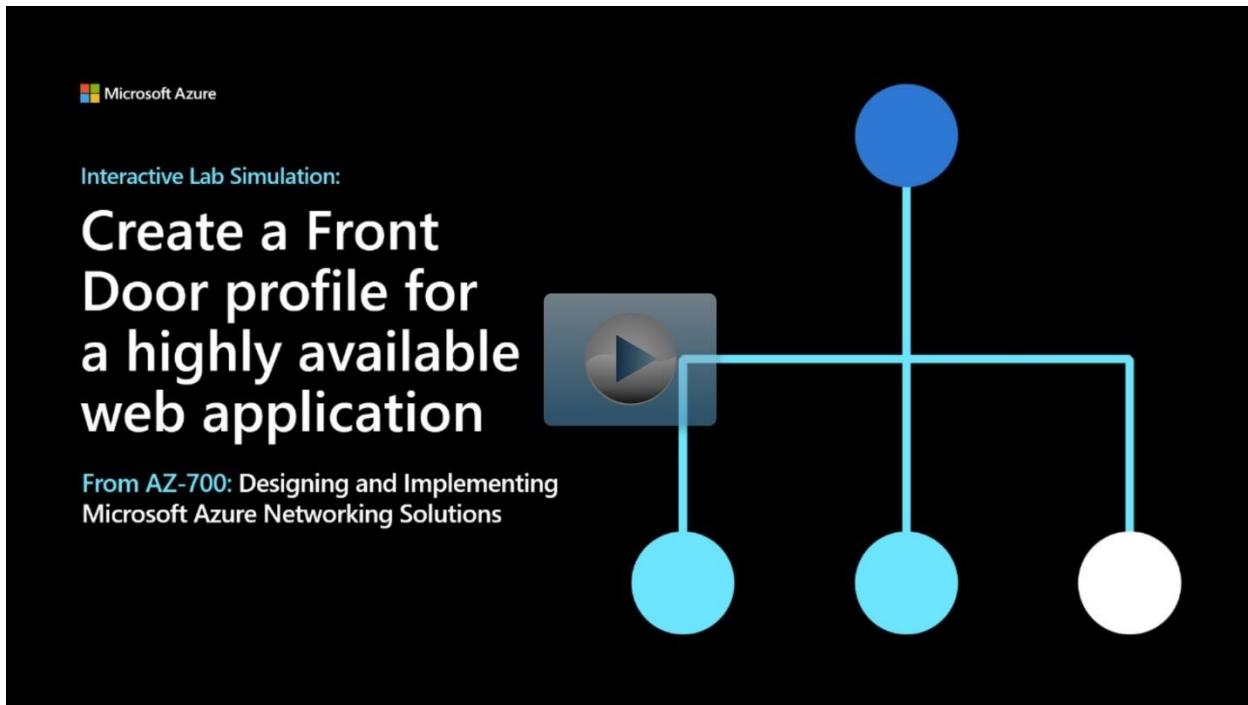
- **Task 1:** Create two instances of a web app
- **Task 2:** Create a Front Door for your application
- **Task 3:** View Azure Front Door in action

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20a%20Front%20Door%20profile%20for%20a%20highly%20available%20web%20application>

Summary

Completed 100 XP

- 1 minute

In this module, you had an in-depth look at Azure Front Door and Azure Application Gateway. You learned how to load balance network traffic effectively to ensure high availability of services.

Design and implement network security

Get network security recommendations with Microsoft Defender for Cloud

200 XP

- 12 minutes

Network security covers a multitude of technologies, devices, and processes. It provides a set of rules and configurations designed to protect the integrity, confidentiality and accessibility of computer networks and data. Every organization, regardless of size, industry, or infrastructure, requires a degree of network security solutions in place to protect it from the ever-growing risks of attacks.

For Microsoft Azure, securing or providing the ability to secure resources like microservices, VMs, data, and others is paramount. Microsoft Azure ensures it through a distributed virtual firewall.

A virtual network in Microsoft Azure is isolated from other networks, while communicating through private IP addresses.

Network Security

Network Security covers controls to secure and protect Azure networks, including securing virtual networks, establishing private connections, preventing and mitigating external attacks, and securing DNS. Full description of the controls can be found at [Security Control V3: Network Security on Microsoft Docs](#).

NS-1: Establish network segmentation boundaries

Security Principle: Ensure that your virtual network deployment aligns to your enterprise segmentation strategy defined in the GS-2 security control. Any workload

that could incur higher risk for the organization should be in isolated virtual networks. Examples of high-risk workload include:

- An application storing or processing highly sensitive data.
- An external network-facing application accessible by the public or users outside of your organization.
- An application using insecure architecture or containing vulnerabilities that cannot be easily remediated.

To enhance your enterprise segmentation strategy, restrict or monitor traffic between internal resources using network controls. For specific, well-defined applications (such as a 3-tier app), this can be a highly secure "deny by default, permit by exception" approach by restricting the ports, protocols, source, and destination IPs of the network traffic. If you have many applications and endpoints interacting with each other, blocking traffic may not scale well, and you may only be able to monitor traffic.

Azure Guidance: Create a virtual network (VNet) as a fundamental segmentation approach in your Azure network, so resources such as VMs can be deployed into the VNet within a network boundary. To further segment the network, you can create subnets inside VNet for smaller sub-networks.

Use network security groups (NSG) as a network layer control to restrict or monitor traffic by port, protocol, source IP address, or destination IP address.

You can also use application security groups (ASGs) to simplify complex configuration. Instead of defining policy based on explicit IP addresses in network security groups, ASGs enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups.

NS-2: Secure cloud services with network controls

Security Principle: Secure cloud services by establishing a private access point for the resources. You should also disable or restrict access from public network when possible.

Azure Guidance: Deploy private endpoints for all Azure resources that support the Private Link feature, to establish a private access point for the resources. You should also disable or restrict public network access to services where feasible.

For certain services, you also have the option to deploy VNet integration for the service where you can restrict the VNET to establish a private access point for the service.

NS-3: Deploy firewall at the edge of enterprise network

Security Principle: Deploy a firewall to perform advanced filtering on network traffic to and from external networks. You can also use firewalls between internal segments to support a segmentation strategy. If required, use custom routes for your subnet to override the system route when you need to force the network traffic to go through a network appliance for security control purpose.

At a minimum, block known bad IP addresses and high-risk protocols, such as remote management (for example, RDP and SSH) and intranet protocols (for example, SMB and Kerberos).

Azure Guidance: Use Azure Firewall to provide fully stateful application layer traffic restriction (such as URL filtering) and/or central management over a large number of enterprise segments or spokes (in a hub/spoke topology).

If you have a complex network topology, such as a hub/spoke setup, you may need to create user-defined routes (UDR) to ensure the traffic goes through the desired route. For example, you have option to use an UDR to redirect egress internet traffic through a specific Azure Firewall or a network virtual appliance.

NS-4: Deploy intrusion detection/intrusion prevention systems (IDS/IPS)

Security Principle: Use network intrusion detection and intrusion prevention systems (IDS/IPS) to inspect the network and payload traffic to or from your workload. Ensure that IDS/IPS is always tuned to provide high-quality alerts to your SIEM solution.

For more in-depth host level detection and prevention capability, use host-based IDS/IPS or a host-based endpoint detection and response (EDR) solution in conjunction with the network IDS/IPS.

Azure Guidance: Use Azure Firewall's IDPS capability on your network to alert on and/or block traffic to and from known malicious IP addresses and domains.

For more in-depth host level detection and prevention capability, deploy host-based IDS/IPS or a host-based endpoint detection and response (EDR) solution, such as Microsoft Defender for Endpoint, at the VM level in conjunction with the network IDS/IPS.

NS-5: Deploy DDOS protection

Security Principle: Deploy distributed denial of service (DDoS) protection to protect your network and applications from attacks.

Azure Guidance: Enable DDoS Network Protection plan on your VNet to protect resources that are exposed to the public networks.

NS-6: Deploy web application firewall

Security Principle: Deploy a web application firewall (WAF) and configure the appropriate rules to protect your web applications and APIs from application-specific attacks.

Azure Guidance: Use web application firewall (WAF) capabilities in Azure Application Gateway, Azure Front Door, and Azure Content Delivery Network (CDN) to protect your applications, services and APIs against application layer attacks at the edge of your network. Set your WAF in "detection" or "prevention mode", depending on your needs and threat landscape. Choose a built-in ruleset, such as OWASP Top 10 vulnerabilities, and tune it to your application.

NS-7: Simplify network security configuration

Security Principle: When managing a complex network environment, use tools to simplify, centralize and enhance the network security management.

Azure Guidance: Use the following features to simplify the implementation and management of the NSG and Azure Firewall rules:

- Use Microsoft Defender for Cloud Adaptive Network Hardening to recommend NSG hardening rules that further limit ports, protocols and source IPs based on threat intelligence and traffic analysis result.
- Use Azure Firewall Manager to centralize the firewall policy and route management of the virtual network. To simplify the firewall rules and network security groups implementation, you can also use the Azure Firewall Manager ARM (Azure Resource Manager) template.

NS-8: Detect and disable insecure services and protocols

Security Principle: Detect and disable insecure services and protocols at the OS, application, or software package layer. Deploy compensating controls if disabling insecure services and protocols are not possible.

Azure Guidance: Use Azure Sentinel's built-in Insecure Protocol Workbook to discover the use of insecure services and protocols such as SSL/TLSv1, SSHv1, SMBv1, LM/NTLMv1, wDigest, Unsigned LDAP Binds, and weak ciphers in Kerberos. Disable insecure services and protocols that do not meet the appropriate security standard.

Note

If disabling insecure services or protocols is not possible, use compensating controls such as blocking access to the resources through network security group, Azure Firewall, or Azure Web Application Firewall to reduce the attack surface.

NS-9: Connect on-premises or cloud network privately

Security Principle: Use private connections for secure communication between different networks, such as cloud service provider datacenters and on-premises infrastructure in a colocation environment.

Azure Guidance: Use private connections for secure communication between different networks, such as cloud service provider datacenters and on-premises infrastructure in a colocation environment.

For lightweight connectivity between site-to-site or point-to-site, use Azure virtual private network (VPN) to create a secure connection between your on-premises site or end-user device to the Azure virtual network.

For enterprise-level high performance connection, use Azure ExpressRoute (or Virtual WAN) to connect Azure datacenters and on-premises infrastructure in a co-location environment.

When connecting two or more Azure virtual networks together, use virtual network peering. Network traffic between peered virtual networks is private and is kept on the Azure backbone network.

NS-10: Ensure Domain Name System (DNS) security

Security Principle: Ensure that Domain Name System (DNS) security configuration protects against known risks:

- Use trusted authoritative and recursive DNS services across your cloud environment to ensure the client (such as operating systems and applications) receive the correct resolution result.
- Separate the public and private DNS resolution so the DNS resolution process for the private network can be isolated from the public network.
- Ensure your DNS security strategy also includes mitigations against common attacks, such as dangling DNS, DNS amplifications attacks, DNS poisoning and spoofing, and so on.

Azure Guidance: Use Azure recursive DNS or a trusted external DNS server in your workload recursive DNS setup, such as in VM's operating system or in the application.

Use Azure Private DNS for private DNS zone setup where the DNS resolution process does not leave the virtual network. Use a custom DNS to restrict the DNS resolution which only allows the trusted resolution to your client.

Use Azure Defender for DNS for the advanced protection against the following security threats to your workload or your DNS service:

- Data exfiltration from your Azure resources using DNS tunneling
- Malware communicating with command-and-control server
- Communication with malicious domains as phishing and crypto mining
- DNS attacks in communication with malicious DNS resolvers

You can also use Azure Defender for App Service to detect dangling DNS records if you decommission an App Service website without removing its custom domain from your DNS registrar.

Microsoft cloud security benchmark

Microsoft has found that using security benchmarks can help you quickly secure cloud deployments. A comprehensive security best practice framework from cloud service providers can give you a starting point for selecting specific security configuration settings in your cloud environment, across multiple service providers and allow you to monitor these configurations using a single pane of glass.

The Microsoft cloud security benchmark (MCSB) includes a collection of high-impact security recommendations you can use to help secure your cloud services in a single or multicloud environment. MCSB recommendations include two key aspects:

- **Security controls:** These recommendations are generally applicable across your cloud workloads. Each recommendation identifies a list of stakeholders that are typically involved in planning, approval, or implementation of the benchmark.
- **Service baselines:** These apply the controls to individual cloud services to provide recommendations on that specific service's security configuration. We currently have service baselines available only for Azure.

Implement Microsoft cloud security benchmark

- **Plan** your MCSB implementation by reviewing the documentation for the enterprise controls and service-specific baselines to plan your control framework and how it maps to guidance like Center for Internet Security (CIS) Controls, National Institute of Standards and Technology (NIST), and the Payment Card Industry Data Security Standard (PCI-DSS) framework.
- **Monitor** your compliance with MCSB status (and other control sets) using the Microsoft Defender for Cloud – Regulatory Compliance Dashboard for your multicloud environment.
- **Establish** guardrails to automate secure configurations and enforce compliance with MCSB (and other requirements in your organization) using features such as Azure Blueprints, Azure Policy, or the equivalent technologies from other cloud platforms.

Terminology

The terms **control**, and **baseline** are used often in the Microsoft cloud security benchmark documentation, and it is important to understand how Azure uses those terms.

Term	Description	Example
Control	A control is a high-level description of a feature or activity that needs to be addressed and is not specific to a technology or implementation.	Data Protection is one of the security control families. Data Protection contains specific actions that must be addressed to help ensure data is protected.
Baseline	A baseline is the implementation of the control on the individual Azure services. Each organization dictates a benchmark recommendation and corresponding configurations are needed in Azure. Note: Today we have service baselines available only for Azure.	The Contoso company looks to enable Azure SQL security features by following the configuration recommended in the Azure SQL security baseline.

Using Microsoft Defender for Cloud for regulatory compliance

Microsoft Defender for Cloud helps streamline the process for meeting regulatory compliance requirements, using the regulatory compliance dashboard.

The regulatory compliance dashboard shows the status of all the assessments within your environment for your chosen standards and regulations. As you act on the recommendations and reduce risk factors in your environment, your compliance posture improves.

Regulatory compliance dashboard

The screenshot shows the Microsoft Defender for Cloud Regulatory compliance dashboard. At the top left, it says "Dashboard > Microsoft Defender for Cloud". Below that is the title "Microsoft Defender for Cloud | Regulatory compliance" with a shield icon. It shows "Showing subscription 'VEH Doc Test'". A search bar is followed by navigation icons: a magnifying glass, back/forward arrows, and a refresh symbol. To the right are links for "Download report", "Manage compliance policies", "Open query", "Compliance over time workbook", "Audit reports", and "Compliance offerings". A message box says: "You can now fully customize the standards you track in the dashboard. Update your dashboard by selecting 'Manage compliance policies' above. →". On the left, a sidebar menu includes "General", "Cloud Security", "Security posture", "Regulatory compliance" (which is selected), "Workload protections", "Data security", "Firewall Manager", "DevOps security", and "Management". The main content area has three sections: "Microsoft cloud security benchmark" (55 of 63 passed controls), "Lowest compliance regulatory standards" (No additional standards are currently monitored), and a callout for "Cloud compliance data now integrated in Microsoft Purview Compliance Manager". The "Microsoft cloud security benchmark" section contains a table of 10 compliance categories, each with a status icon (green checkmark or red X) and a link to "Manage compliance policies". The categories are: NS. Network Security, IM. Identity Management, PA. Privileged Access, DP. Data Protection, AM. Asset Management, LT. Logging and Threat Detection, IR. Incident Response, PV. Posture and Vulnerability Management, ES. Endpoint Security, BR. Backup and Recovery, DS. DevOps Security, and GS. Governance and Strategy.

The dashboard shows an overview of your compliance status with the set of supported compliance regulations. You will see your overall compliance score, and the number of passing vs. failing assessments associated with each standard.

Compliance controls

The screenshot shows the Microsoft cloud security benchmark interface. At the top, it says "Microsoft cloud security benchmark". Below that, a note states: "Recommendations from Microsoft Defender for Cloud - Regulatory Compliance should not be interpreted as a guarantee of compliance. It is up to you to evaluate and validate the effectiveness of customer controls per your regulatory environment. These services are subject to the terms and conditions in the [licensing terms](#)." A button "Microsoft cloud security benchmark is applied to the subscription VEH Doc Test" is shown with a circled number 1 above it. There is also a checkbox "Expand all compliance controls".

Section 2: NS. Network Security contains two items:

- NS-1. Establish network segmentation boundaries: Status MS C
- NS-2. Secure cloud services with network controls: Status MS C

Section 3: Automated assessments - Azure lists five items:

Assessment Detail	Resource type	Failed resources	Resource compliance status
Storage account should use a private link connection	Storage accounts	1 of 1	Red bar
Storage accounts should restrict network access using virtual network rules	Storage accounts	1 of 1	Red bar
Storage account public access should be disabled	Storage accounts	1 of 1	Red bar
Access to storage accounts with firewall and virtual network configurations should be n	Storage accounts	1 of 1	Red bar
Azure AI Services resources should restrict network access	Azure resources	0 of 0	Green bar

Section 4: Resource compliance status shows a summary of failed resources across different resource types.

1. Subscriptions the standard is applied on.
2. List of all controls for that standard.
3. View the details of passing and failing assessments associated with that control.
4. Number of affected resources.

Some controls are grayed out. These controls do not have any Microsoft Defender for Cloud assessments associated with them. Check their requirements and assess them in your environment. Some of these might be process-related and not technical.

Exploring the details of compliance with a specific standard

To generate a PDF report with a summary of your current compliance status for a particular standard, select Download report.

The screenshot shows the Microsoft Defender for Cloud Regulatory compliance dashboard. On the left, there's a sidebar with various navigation options like General, Overview, Getting started, Recommendations, Attack path analysis, Security alerts, Inventory, Cloud Security Explorer, Workbooks, Community, Diagnose and solve problems, Cloud Security, Security posture, and Regulatory compliance. The Regulatory compliance section is currently selected. At the top center, there's a 'Download report' button highlighted with a red box. To the right, a 'Download report' modal is open, also with a red box around its title. Inside the modal, there's a 'Report standard' dropdown set to 'Microsoft cloud security benchmark', another red box. Below it is a 'Format' dropdown with options: PDF (selected), PDF, CSV, and Scope. A scope dropdown shows 'VEH Doc Test (f054d65b-172e-41e0-b0cb-cdd68243...'. At the bottom right of the modal is a large blue 'Download' button.

The report provides a high-level summary of your compliance status for the selected standard based on Microsoft Defender for Cloud assessments data. The report is organized according to the controls of that standard. The report can be shared with relevant stakeholders and might provide evidence to internal and external auditors.

Alerts in Microsoft Defender for Cloud

Microsoft Defender for Cloud automatically collects, analyzes, and integrates log data from your Azure resources, the network, and connected partner solutions - like firewall and endpoint protection solutions - to detect real threats and reduce false positives. A list of prioritized security alerts is shown in Microsoft Defender for Cloud along with the information you need to quickly investigate the problem and steps to take to remediate an attack.

Manage your security alerts

The Microsoft Defender for Cloud overview page shows the Security alerts tile at the top of the page, and as a link from the sidebar.

The screenshot shows the Microsoft Defender for Cloud Security alerts dashboard. On the left, there's a navigation sidebar with sections like General, Security alerts (which is selected), and Cloud Security. The main area displays four summary counts: 0 Open alerts, 0 Active alerts, 0 In progress alerts, and 0 Affected resources. Below these are filter options for Subscription (All), Status (Active, In Progress), Severity (Low, Medium, High), and other metrics like Resource Group, Activity start time, MITRE ATT&CK tactics, and Status. A large central circle contains a shield icon with a plus sign, and the text "No alerts found".

The security alerts page shows the active alerts. You can sort the list by Severity, Alert title, Affected resource, Activity start time, MITRE ATTACK tactics, and status.

This screenshot shows the detailed list of security alerts. At the top, it displays 644 Active alerts and 34 Affected resources. A color-coded bar indicates the distribution of alerts by severity: High (166), Medium (414), and Low (64). The main table lists individual alerts with columns for Severity, Alert title, Affected resource, Activity start time, MITRE ATT&CK tactics, and Status. Each alert row includes a checkbox for selection. At the bottom, there are navigation controls for page 1 of 17.

Severity	Alert title	Affected resource	Activity start time (UTC+2)	MITRE ATT&CK® tactics	Status
High	Suspicious process executed [seen ...]	CH-VictimVM00-Dev	11/22/20, 3:00 AM	Credential Access	Active
High	Suspicious process executed [seen ...]	CH-VictimVM00	11/22/20, 1:00 AM	Credential Access	Active
High	Suspicious process executed [seen ...]	dockervm-redhat	11/21/20, 3:00 AM	Credential Access	Active
High	Suspicious process executed [seen ...]	dockeroniasdemolo	11/21/20, 1:00 AM	Credential Access	Active
High	Suspicious process executed [seen ...]	samplecrmwebblobstor...	11/20/20, 7:00 AM	Credential Access	Active
High	Suspicious process executed	dockervm-redhat	11/20/20, 6:00 AM	Credential Access	Active
High	Suspicious process executed	dockervm-redhat	11/20/20, 5:00 AM	Credential Access	Active
High	Azure Security Center test alert for A...	ASC-AKS-CLOUD-TALK	11/20/20, 3:00 AM	Persistence	Active
High	Exposed Kubernetes dashboard det...	ASC-WORKLOAD-PRO...	11/20/20, 12:00 AM	Initial Access	Active
High	Suspicious process executed [seen ...]	CH-VictimVM00-Dev	11/19/20, 7:00 PM	Credential Access	Active

To filter the alerts list, select any of the relevant filters. You can add further filters with the Add filter option.

Security alerts

The screenshot shows the 'Security alerts' page in Microsoft Defender XDR. At the top, there are summary counts: 644 Active alerts and 34 Affected resources. A progress bar indicates 'Active alerts by severity' with segments for High (166), Medium (414), and Low (64). Below this, a search bar and filter buttons for Status == Active, Severity == High, and Time == Last month are visible. A large table lists security alerts, including columns for Severity, Alert title, Affected resource, and Activity start time (UTC+2). A red box highlights the 'Add filter' modal, which lists various filtering options like Alert name, Affected resource, Resource type, MITRE ATT&CK® tactics, Tags, Creator, Owner, and environment.

Severity ↑↓	Alert title ↑↓	Affected resource ↑↓	Activity start time (UTC+2) ↑↓
High	Suspicious process executed [seen ...]	CH-VictimVM00	10/30/20, 2:00 AM
High	Suspicious process executed	CH-VictimVM00	10/30/20, 1:00 AM
High	Suspicious process executed [seen ...]	CH-VictimVM00-Dev	10/30/20, 12:00 AM
High	Suspicious process executed	CH-VictimVM00	10/30/20, 12:00 AM
High	Suspicious process executed	CH-VictimVM00-Dev	10/29/20, 11:00 PM
High	Suspicious process executed	CH-VictimVM00	10/29/20, 10:00 PM
High	Suspicious process executed	CH-VictimVM00-Dev	10/29/20, 9:00 PM

The list updates according to the filtering options you have selected. Filtering can be very helpful. For example, you might want to address security alerts that occurred in the last 24 hours because you are investigating a potential breach in the system.

Respond to security alerts

From the Security alerts list, select an alert. A side pane opens and shows a description of the alert and all the affected resources.

Security alerts

The screenshot shows the Azure Security Center Security alerts dashboard. At the top, there are navigation links: Refresh, Change status, Open query, Suppression rules, Security alerts map (Preview), Create sample alerts, and Active alerts (3). Below this, there are two summary cards: Active alerts (3) and Affected resources (1).

The main area displays a list of security alerts. The filters applied are Status == Active and Severity == High. The results are grouped by activity time (Last month). The columns in the list are: Severity, Alert title, Affected resource, and Activity start time.

Severity	Alert title	Affected resource	Activity start time
High	Exposed Kubernetes dashboard detected	ASC-AKS-CLOUD-TALK	11/05/20, 1:58 PM
High	Azure Security Center test alert for AKS...	ASC-AKS-CLOUD-TALK	11/04/20, 11:50 AM
High	Exposed Kubernetes dashboard detect...	ASC-AKS-CLOUD-TALK	10/26/20, 10:44 PM

The first alert in the list is highlighted with a red box and a cursor is hovering over it. To the right, a detailed view of this specific alert is shown:

Exposed Kubernetes dashboard detected

High Severity | **Active Status** | **11/05/20, 13:58 Activity time**

Alert description
Kubernetes audit log analysis detected exposure of the Kubernetes Dashboard by a LoadBalancer service. Exposed dashboard allows an unauthenticated access to the cluster management and poses a security threat.

Affected resource

- ASC-AKS-CLOUD-TALK (Kubernetes service)
- ASC DEMO (Subscription)

MITRE ATT&CK® tactics

- Initial Access

View full details | **Take action**

View full details displays further information, as shown in the following image:

The screenshot shows the Azure Security Center interface for a security alert titled "Potential SQL Injection". The left pane provides high-level information: the alert is at "High Severity" and is currently "Active". It occurred on "06/11/20, 1..." and was detected on the "Demo" database on the "R-DEV\SQLEXPRESS" server. The affected resource is an "Azure Arc machine" named "R-DEV" under the "Env: Development" tag. Another resource listed is a "Subscription" named "DS-ThreatDetection_Demo". The "Intent" section shows a timeline with a single step labeled "Pre-attack" highlighted. Below this, there's a "Was this useful?" poll with "Yes" and "No" options. The right pane is the "Take action" tab, which contains several sections: "Mitigate the threat" (with a link to more information), "Prevent future attacks" (listing recommendations like enabling Windows Defender Exploit Guard and remediating SQL server vulnerabilities), "Trigger automated response", and "Suppress similar alerts (preview)". A "Next: Take Action >>" button is located at the bottom of the right pane.

The left pane of the security alert page shows high-level information regarding the security alert: title, severity, status, activity time, description of the suspicious activity, and the affected resource. Alongside the affected resource are the Azure tags relevant to the resource. Use these to infer the organizational context of the resource when investigating the alert.

The right pane includes the Alert details tab containing further details of the alert to help you investigate the issue: IP addresses, files, processes, and more.

Also in the right pane is the Take action tab. Use this tab to take further actions regarding the security alert. Actions such as:

- **Mitigate the threat:** Provides manual remediation steps for this security alert.
- **Prevent future attacks:** Provides security recommendations to help reduce the attack surface, increase security posture, and thus prevent future attacks.
- **Trigger automated response:** Provides the option to trigger a logic app as a response to this security alert.
- **Suppress similar alerts:** Provides the option to suppress future alerts with similar characteristics if the alert isn't relevant for your organization.

Deploy Azure DDoS Protection by using the Azure portal

200 XP

- 10 minutes

Distributed Denial of Service (DDoS)

A denial of service attack (DoS) is an attack that has the goal of preventing access to services or systems. If the attack originates from one location, it's called a DoS. If the attack originates from multiple networks and systems, it's called distributed denial of service (DDoS).

Distributed Denial of Service (DDoS) attacks are some of the largest availability and security concerns facing customers that are moving their applications to the cloud. A DDoS attack tries to drain an API's or application's resources, making that application unavailable to legitimate users. DDoS attacks can be targeted at any endpoint that is publicly reachable through the internet.

DDoS implementation

Azure DDoS Protection, combined with application design best practices, provide defense against DDoS attacks. Azure DDoS Protection provides the following service tiers:

- **Network Protection**

Provides additional mitigation capabilities over DDoS infrastructure Protection that are tuned specifically to Azure Virtual Network resources. Azure DDoS Protection is simple to enable, and requires no application changes. Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses associated to resources deployed in virtual networks, such as Azure Load Balancer, Azure Application Gateway, and Azure Service Fabric instances, but this protection doesn't apply to App Service Environments. Real-time telemetry is available through Azure Monitor views during an attack, and for history. Rich attack mitigation analytics are available via diagnostic settings. Application layer protection can be added through the Azure Application

Gateway Web Application Firewall or by installing a third party firewall from Azure Marketplace. Protection is provided for IPv4 and IPv6 Azure public IP addresses.

- **IP Protection**

DDoS IP Protection is a pay-per-protected IP model. DDoS IP Protection contains the same core engineering features as DDoS Network Protection, but will differ in value-added services like DDoS rapid response support, cost protection, and discounts on WAF.

DDoS Protection protects resources in a virtual network including public IP addresses associated with virtual machines, load balancers, and application gateways. When coupled with the Application Gateway web application firewall, or a third-party web application firewall deployed in a virtual network with a public IP, DDoS Protection can provide full layer 3 to layer 7 mitigation capability.

Every property in Azure is protected by Azure's DDoS infrastructure (Basic) Protection at no additional cost. Azure DDoS Protection is a paid service, designed for services that are deployed in a virtual network.

Types of DDoS attacks

DDoS Protection can mitigate the following types of attacks:

- **Volumetric attacks**

These attacks flood the network layer with a substantial amount of seemingly legitimate traffic. They include UDP floods, amplification floods, and other spoofed-packet floods. DDoS Protection mitigates these potential multi-gigabyte attacks by absorbing and scrubbing them, with Azure's global network scale, automatically.

- **Protocol attacks**

These attacks render a target inaccessible, by exploiting a weakness in the layer 3 and layer 4 protocol stack. They include SYN flood attacks, reflection attacks, and other protocol attacks. DDoS Protection mitigates these attacks, differentiating between malicious and legitimate traffic, by interacting with the client, and blocking malicious traffic.

- **Resource (application) layer attacks**

These attacks target web application packets, to disrupt the transmission of data between hosts. They include HTTP protocol violations, SQL injection, cross-site scripting, and other layer 7 attacks. Use a Web Application Firewall, such as the Azure Application Gateway web application firewall, and DDoS Protection to provide defense against these attacks. There are also third-party web application firewall offerings available in the Azure Marketplace.

Azure DDoS protection features

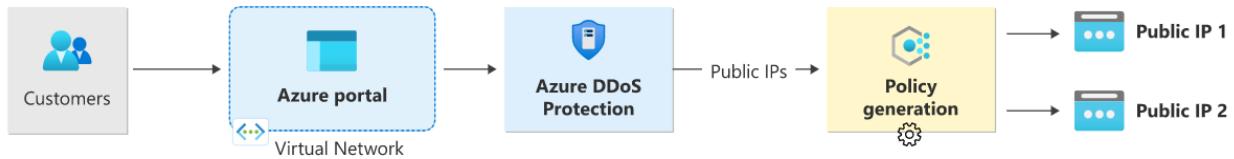
Some of Azure DDoS protection features include:

- **Native platform integration:** Natively integrated into Azure and configured through portal.
- **Turnkey protection:** Simplified configuration protecting all resources immediately.
- **Always-on traffic monitoring:** Your application traffic patterns are monitored 24 hours a day, 7 days a week, looking for indicators of DDoS attacks.
- **Adaptive tuning:** Profiling and adjusting to your service's traffic.
- **Attack analytics:** Get detailed reports in five-minute increments during an attack, and a complete summary after the attack ends.
- **Attack metrics and alerts:** Summarized metrics from each attack are accessible through Azure Monitor. Alerts can be configured at the start and stop of an attack, and over the attack's duration, using built-in attack metrics.
- **Multi-layered protection:** When deployed with a web application firewall (WAF), DDoS Protection protects both at the network layer (Layer 3 and 4, offered by Azure DDoS Protection) and at the application layer (Layer 7, offered by a WAF).

Let us have a look in a bit more detail at some of those key features.

Always-on traffic monitoring

DDoS Protection monitors actual traffic utilization and constantly compares it against the thresholds defined in the DDoS Policy. When the traffic threshold is exceeded, DDoS mitigation is initiated automatically. When traffic returns below the thresholds, the mitigation is stopped.



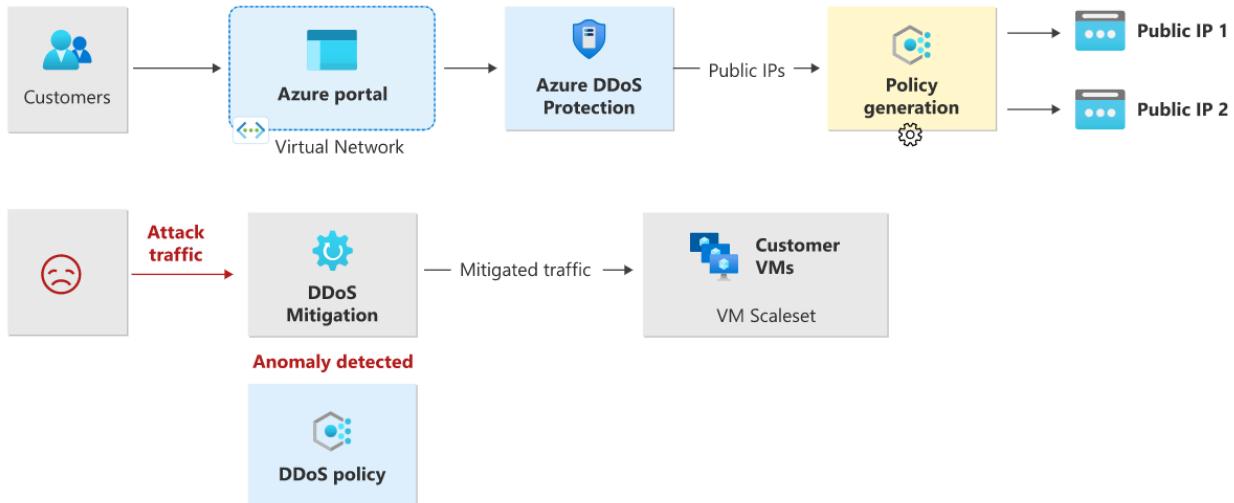
During mitigation, traffic sent to the protected resource is redirected by the DDoS protection service and several checks are performed, such as:

- Ensure packets conform to internet specifications and aren't malformed.
- Interact with the client to determine if the traffic is potentially a spoofed packet (e.g: SYN Auth or SYN Cookie or by dropping a packet for the source to retransmit it).
- Rate-limit packets if no other enforcement method can be performed.

DDoS protection drops attack traffic and forwards the remaining traffic to its intended destination. Within a few minutes of attack detection, you're notified using Azure Monitor metrics. By configuring logging on DDoS Protection telemetry, you can write the logs to available options for future analysis. Metric data in Azure Monitor for DDoS Protection is retained for 30 days.

Adaptive real-time tuning

The Azure DDoS Protection service helps protect customers and prevent impacts to other customers. For example, if a service is provisioned for a typical volume of legitimate incoming traffic that is smaller than the trigger rate of the infrastructure-wide DDoS Protection policy, a DDoS attack on that customer's resources might go unnoticed. More generally, the complexity of recent attacks (for example, multi-vector DDoS) and the application-specific behaviors of tenants call for per-customer, tailored protection policies.



The service accomplishes this by using two insights:

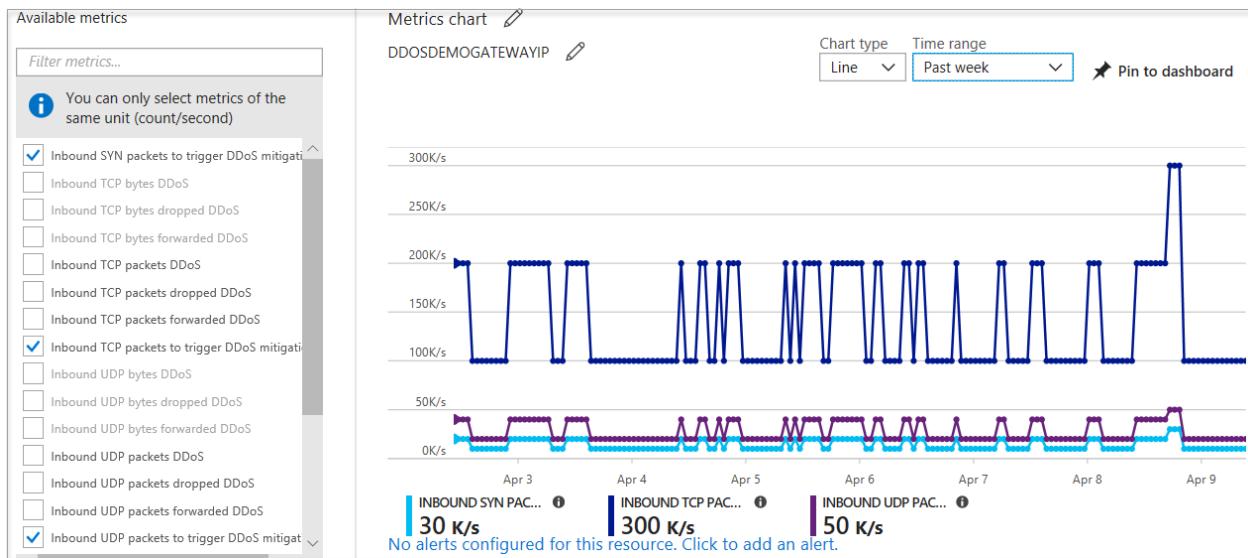
- Automatic learning of per-customer (per- Public IP) traffic patterns for Layer 3 and 4.
- Minimizing false positives, considering that the scale of Azure allows it to absorb a significant amount of traffic.

Attack metrics, alerts, and logs

DDoS Protection exposes rich telemetry via the Azure Monitor tool. You can configure alerts for any of the Azure Monitor metrics that DDoS Protection uses. You can integrate logging with Splunk (Azure Event Hubs), Azure Monitor logs, and Azure Storage for advanced analysis via the Azure Monitor Diagnostics interface.

In the Azure portal, select **Monitor > Metrics**. In the **Metrics** pane, select the resource group, select a resource type of **Public IP Address**, and select your Azure public IP address. DDoS metrics are visible in the **Available metrics** pane.

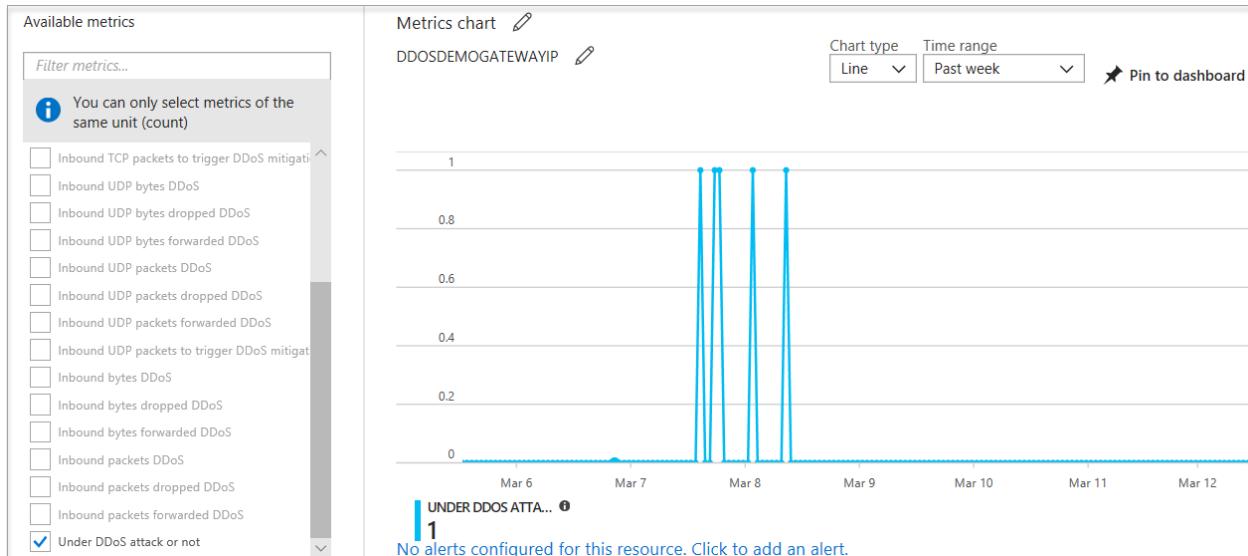
DDoS Protection applies three autotuned mitigation policies (SYN, TCP, and UDP) for each public IP of the protected resource, in the virtual network that has DDoS enabled. You can view the policy thresholds by selecting the **Inbound [SYN/TCP/UDP] packets to trigger DDoS mitigation** metrics as shown in the example screenshot below.



The policy thresholds are autoconfigured via machine learning-based network traffic profiling. DDoS mitigation occurs for an IP address under attack only when the policy threshold is exceeded.

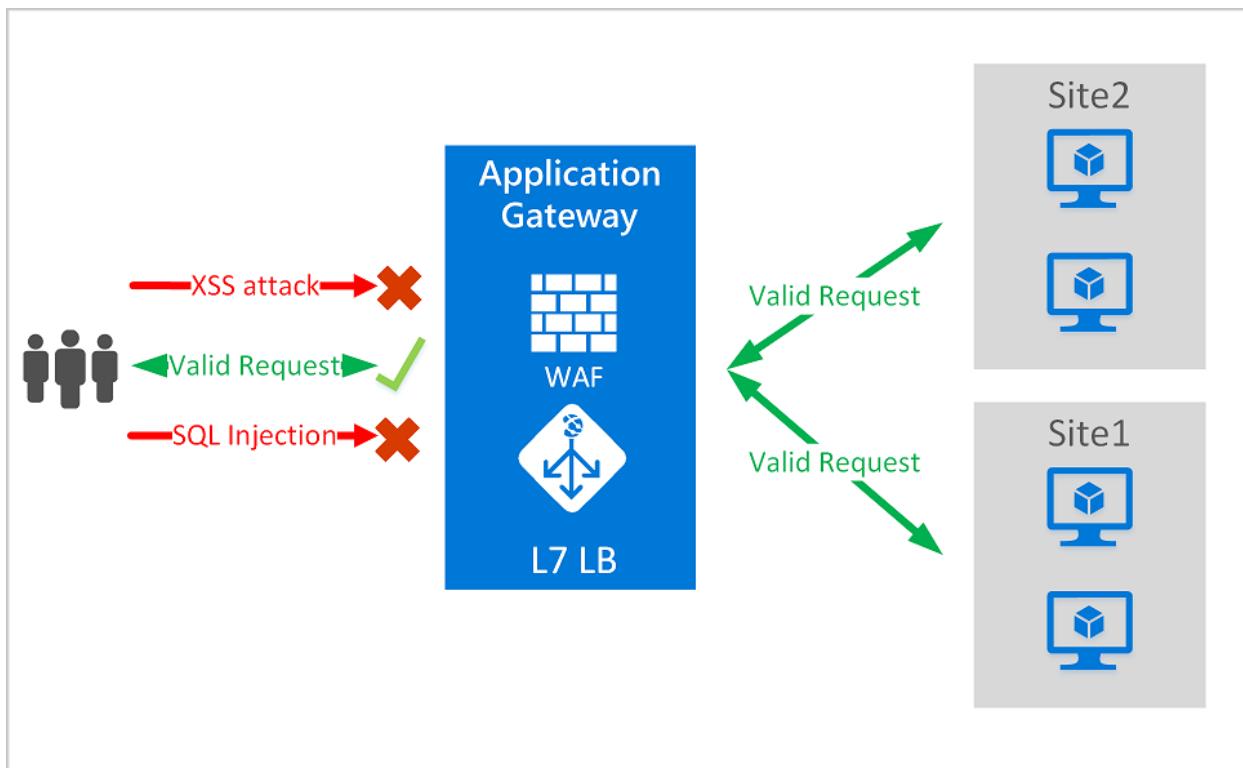
If the public IP address is under attack, the value for the **Under DDoS attack or not** metric changes to **1** as DDoS Protection performs mitigation on the attack traffic.

It's recommended to configure an alert on this metric as you'll then get notified if there's an active DDoS mitigation performed on your public IP address.



Multi-layered protection

Specific to resource attacks at the application layer, you should configure a web application firewall (WAF) to help secure web applications. A WAF inspects inbound web traffic to block SQL injections, cross-site scripting, DDoS, and other Layer 7 attacks. Azure provides WAF as a feature of **Application Gateway** for centralized protection of your web applications from common exploits and vulnerabilities. There are other WAF offerings available from Azure partners that might be more suitable for your needs via the Azure Marketplace.



Even web application firewalls are susceptible to volumetric and state exhaustion attacks. Therefore, it's firmly recommended to enable DDoS Protection on the WAF virtual network to help protect from volumetric and protocol attacks.

Deploying a DDoS protection plan

The key stages of deploying a DDoS Protection plan are as follows:

- Create a resource group
- Create a DDoS Protection Plan
- Enable DDoS protection on a new or existing virtual network or IP address

- Configure DDoS telemetry
- Configure DDoS diagnostic logs
- Configure DDoS alerts
- Run a test DDoS attack and monitor the results.

Exercise: Configure DDoS Protection on a virtual network using the Azure portal

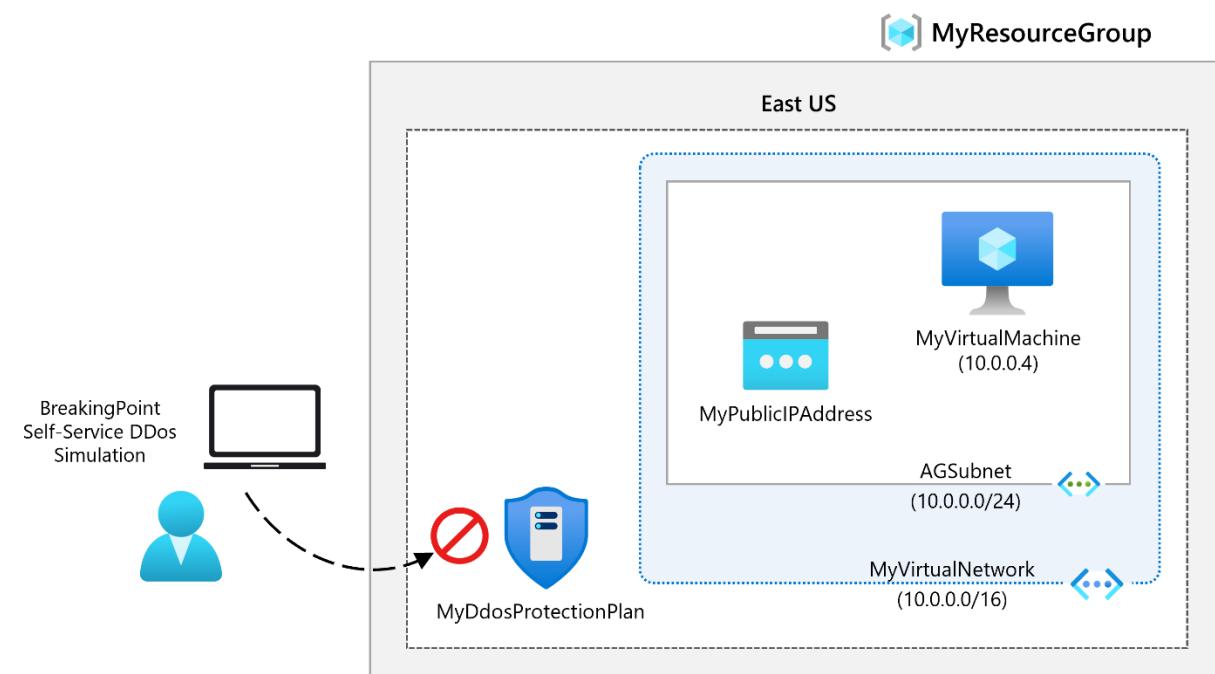
Completed 100 XP

- 3 minutes

Lab scenario

In this lab, you're going to run a mock DDoS attack on the virtual network. The following steps walk you through creating a virtual network, configuring DDoS Protection, and creating an attack which you can observe and monitor with the help of telemetry and metrics.

Architecture diagram



Objectives

- **Task 1:** Create a DDoS Protection plan
- **Task 2:** Enable DDoS Protection on a new virtual network
- **Task 3:** Configure DDoS telemetry
- **Task 4:** Configure DDoS diagnostic logs
- **Task 5:** Configure DDoS alerts
- **Task 6:** Monitor a DDos test attack

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Configure%20Azure%20DDoS%20Protection%20on%20a%20virtual%20network>

Deploy Network Security Groups by using the Azure portal

200 XP

- 14 minutes

A Network Security Group (NSG) in Azure allows you to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

NSG security rules

A network security group contains zero, or as many rules as desired, within Azure subscription limits. Each rule specifies the following properties:

- **Name** - Must be a unique name within the network security group.
- **Priority** - Can be any number between 100 and 4096. Rules are processed in priority order, with lower numbers processed before higher numbers, because lower numbers have higher priority. Once traffic matches a rule, processing stops. As a result, any rules that exist with lower priorities (higher numbers) that have the same attributes as rules with higher priorities aren't processed.
- **Source or destination** - Can be set to Any, or an individual IP address, or classless inter-domain routing (CIDR) block (10.0.0.0/24, for example), service tag, or application security group.
- **Protocol** - Can be TCP, UDP, ICMP, ESP, AH, or Any.
- **Direction** - Can be configured to apply to inbound, or outbound traffic.
- **Port range** - Can be specified either as an individual port or range of ports. For example, you could specify 80 or 10000-10005. Specifying ranges enables you to create fewer security rules.
- **Action** - Can be set to Allow or deny.

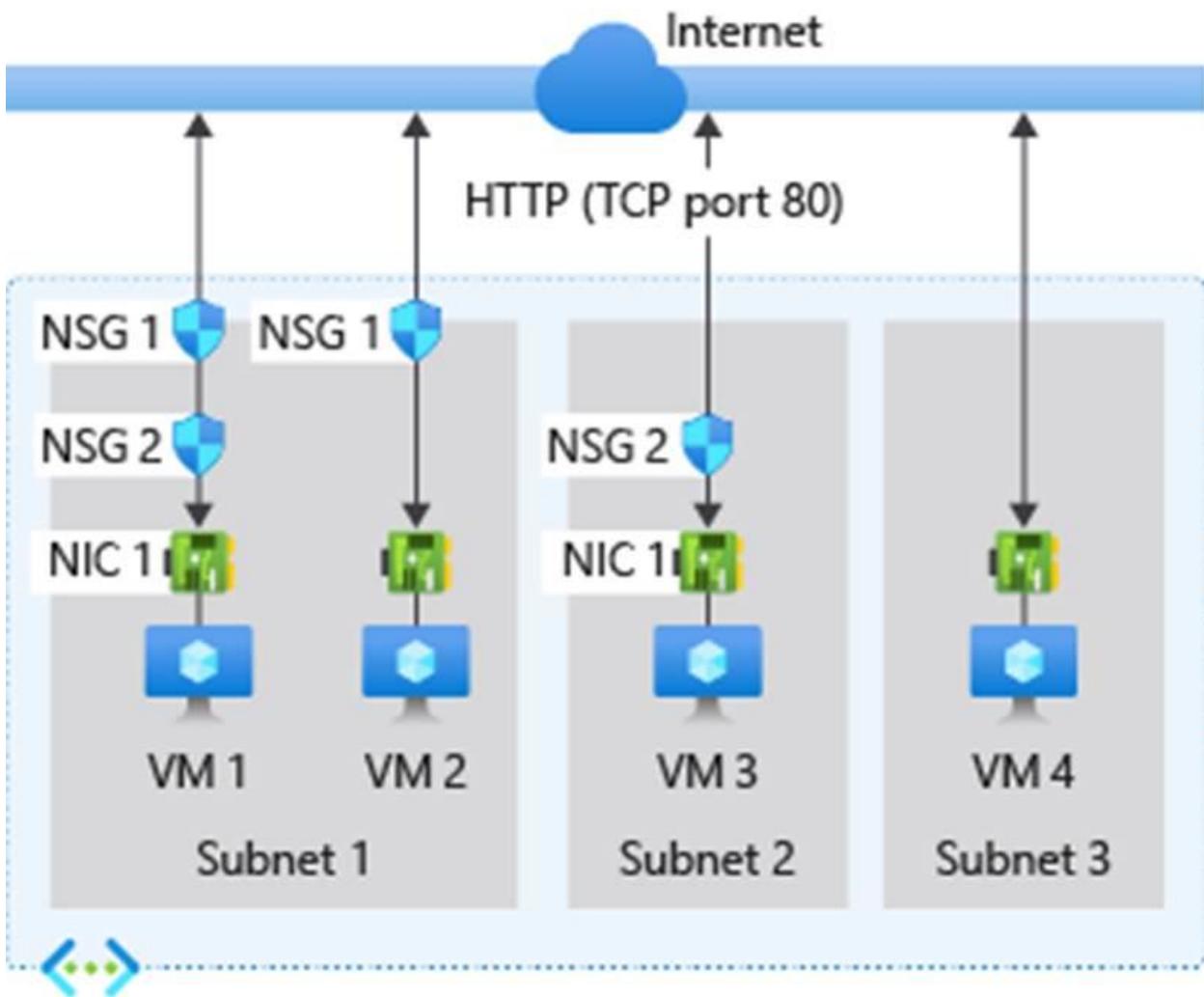
The firewall evaluates network security group security rules by priority, using the 5-tuple information (source, source port, destination, destination port, and protocol) to either allow or deny the traffic.

Default security rules

Azure creates the following default rules in each network security group that you create:

Direction	Name	Priority	Source	Source Ports	Destination	Destination Ports	Protocol	Access
Inbound	AllowVNetInBound	65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow
Inbound	AllowAzureLoadBalancerInBound	65001	AzureLoadBalancer	0-65535	0.0.0.0/0	0-65535	Any	Allow
Inbound	DenyAllInbound	65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny
Outbound	AllowVNetOutBound	65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow
Outbound	AllowInternetOutBound	65001	0.0.0.0/0	0-65535	Internet	0-65535	Any	Allow
Outbound	DenyAllOutBound	65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny

The following diagram and bullet points illustrate different scenarios for how network security groups might be deployed to allow network traffic to and from the internet over TCP port 80:



For **inbound traffic** Azure processes the rules in a network security group associated to a subnet first, if there's one, and then the rules in a network security group associated to the network interface, if there's one.

- **VM1:** Subnet1 is associated with NSG1, so the security rules are processed, and VM1 is in Subnet1. Unless you created a rule that allows port 80 inbound, the DenyAllInbound default security rule denies the traffic, and never evaluated by NSG2, since NSG2 is associated to the network interface. If NSG1 has a security rule that allows port 80, NSG2 then processes the traffic. To allow port 80 to the virtual machine, both NSG1 and NSG2 must have a rule that allows port 80 from the internet.
- **VM2:** The rules in NSG1 are processed because VM2 is also in Subnet1. Since VM2 doesn't have a network security group associated to its network interface, it receives all traffic allowed through NSG1 or is denied all traffic denied by NSG1.

Traffic is either allowed or denied to all resources in the same subnet when a network security group is associated to a subnet.

- **VM3:** Since there's no network security group associated to Subnet2, traffic is allowed into the subnet and processed by NSG2, because NSG2 is associated to the network interface attached to VM3.
- **VM4:** Traffic is allowed to VM4, because a network security group isn't associated to Subnet3, or the network interface in the virtual machine. All network traffic is allowed through a subnet and network interface if they don't have a network security group associated to them.

For **outbound traffic**, Azure processes the rules in a network security group associated to a network interface first, if there's one, and then the rules in a network security group associated to the subnet, if there's one.

- **VM1:** The security rules in NSG2 are processed. Unless you create a security rule that denies port 80 outbound to the internet, the AllowInternetOutbound default security rule allows the traffic in both NSG1 and NSG2. If NSG2 has a security rule that denies port 80, the traffic is denied, and NSG1 never evaluates it. To deny port 80 from the virtual machine, either, or both of the network security groups must have a rule that denies port 80 to the internet.
- **VM2:** All traffic is sent through the network interface to the subnet, since the network interface attached to VM2 doesn't have a network security group associated to it. The rules in NSG1 are processed.
- **VM3:** If NSG2 has a security rule that denies port 80, the traffic is denied. If NSG2 has a security rule that allows port 80, then port 80 is allowed outbound to the internet, since a network security group isn't associated to Subnet2.
- **VM4:** All network traffic is allowed from VM4, because a network security group isn't associated to the network interface attached to the virtual machine, or to Subnet3.

Application Security Groups

An Application Security Group (ASG) enables you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. You can reuse your security policy at scale without manual maintenance of explicit IP addresses. The platform handles the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic.

To minimize the number of security rules you need, and the need to change the rules, plan out the application security groups you need and create rules using service tags or

application security groups, rather than individual IP addresses, or ranges of IP addresses, whenever possible.

Filter network traffic with an NSG using the Azure portal

You can use a network security group to filter network traffic inbound and outbound from a virtual network subnet. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet.

The key stages to filter network traffic with an NSG using the Azure portal are:

1. **Create a resource group** - this can either be done beforehand or as you create the virtual network in the next stage. All other resources that you create must be in the same region specified here.
2. **Create a virtual network** - this must be deployed in the same resource group you created previously.
3. **Create application security groups** - the application security groups you create here enables you to group together servers with similar functions, such as web servers or management servers. You would create two application security groups here; one for web servers and one for management servers (for example, MyAsgWebServers and MyAsgMgmtServers)
4. **Create a network security group** - the network security group secures network traffic in your virtual network. This NSG will be associated with a subnet in the next stage.
5. **Associate a network security group with a subnet** - this is where you associate the network security group you create previously, with the subnet of the virtual network you created in stage 2 above.
6. **Create security rules** - this is where you create your inbound security rules. Here you would create a security rule to allow ports 80 and 443 to the application security group for your web servers (for example, MyAsgWebServers). Then you would create another security rule to allow RDP traffic on port 3389 to the application security group for your management servers (for example, MyAsgMgmtServers). These rules control from where you can access your VM remotely and your IIS Webserver.
7. **Create virtual machines** - this is where you create the web server (for example, MyVMWeb) and management server (for example, MyVMMgmt) virtual machines that will be associated with their respective application security group in the next stage.
8. **Associate NICs to an ASG** - this is where you associate the network interface card (NIC) attached to each virtual machine with the relevant application security group that you created in stage 3 above.

9. **Test traffic filters** - the final stage is where you test that your traffic filtering is working as expected.
- To test this, you would attempt to connect to the management server virtual machine (for example, MyVMMgmt) by using an RDP connection, thereby verifying that you can connect because port 3389 is allowing inbound connections from the Internet to the management servers application security group (for example, MyAsgMgmtServers).
 - While connected to the RDP session on the management server (for example, MyVMMgmt), you would then test an RDP connection from the management server virtual machine (for example, MyVMMgmt) to the web server virtual machine (for example, MyVMWeb), which again should succeed because virtual machines in the same network can communicate with each other over any port by default.
 - However, you'll not be able to create an RDP connection to the web server virtual machine (for example, MyVMWeb) from the internet, because the security rule for the web servers application security group (for example, MyAsgWebServers) prevents connections to port 3389 inbound from the Internet. Inbound traffic from the Internet is denied to all resources by default.
 - While connected to the RDP session on the web server (for example, MyVMWeb), you could then install IIS on the web server, then disconnect from the web server virtual machine RDP session, and disconnect from the management server virtual machine RDP session. In the Azure portal, you would then determine the Public IP address of the web server virtual machine (for example, MyVMWeb), and confirm you can access the web server virtual machine from the Internet by opening a web browser on your computer and navigating to <http://23.96.39.113>. You should see the standard IIS welcome screen, because port 80 is allowed inbound access from the Internet to the web servers application security group (for example, MyAsgWebServers). The network interface attached to the web server virtual machine (for example, MyVMWeb) is associated with the web servers application security group (for example, MyAsgWebServers) and therefore allows the connection.

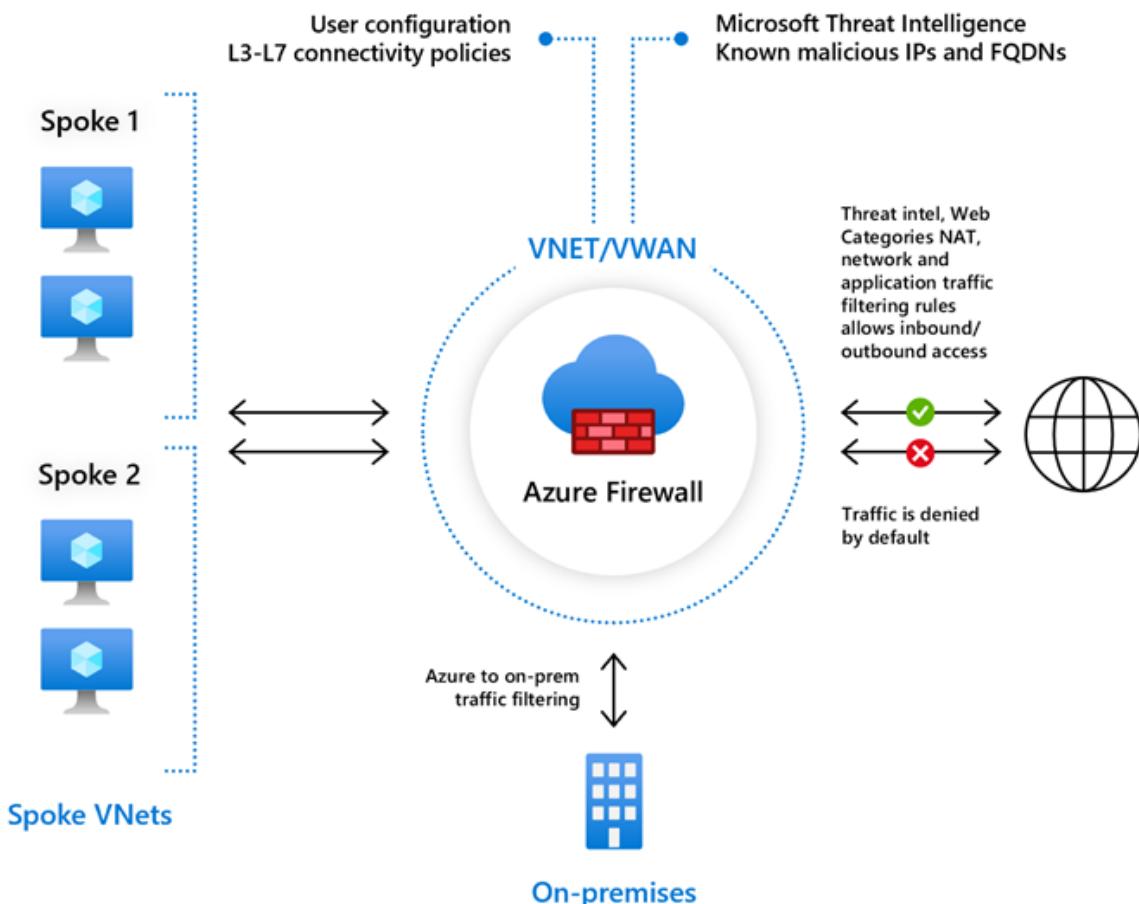
To view the detailed steps for all these tasks, see [Tutorial: Filter network traffic with a network security group using the Azure portal](#).

Design and implement Azure Firewall

200 XP

- 10 minutes

Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources. It is a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability.



Azure Firewall features

Azure Firewall includes the following features:

- **Built-in high availability** - High availability is built in, so no extra load balancers are required and there's nothing you need to configure.
- **Unrestricted cloud scalability** - Azure Firewall can scale out as much as you need to accommodate changing network traffic flows, so you do not need to budget for your peak traffic.

- **Application FQDN filtering rules** - You can limit outbound HTTP/S traffic or Azure SQL traffic to a specified list of fully qualified domain names (FQDN) including wild cards. This feature does not require TLS termination.
- **Network traffic filtering rules** - You can centrally create allow or deny network filtering rules by source and destination IP address, port, and protocol. Azure Firewall is fully stateful, so it can distinguish legitimate packets for different types of connections. Rules are enforced and logged across multiple subscriptions and virtual networks.
- **FQDN tags** - These tags make it easy for you to allow well-known Azure service network traffic through your firewall. For example, say you want to allow Windows Update network traffic through your firewall. You create an application rule and include the Windows Update tag. Now network traffic from Windows Update can flow through your firewall.
- **Service tags** - A service tag represents a group of IP address prefixes to help minimize complexity for security rule creation. You cannot create your own service tag, nor specify which IP addresses are included within a tag. Microsoft manages the address prefixes encompassed by the service tag, and automatically updates the service tag as addresses change.
- **Threat intelligence** - Threat intelligence-based filtering (IDPS) can be enabled for your firewall to alert and deny traffic from/to known malicious IP addresses and domains. The IP addresses and domains are sourced from the Microsoft Threat Intelligence feed.
- **TLS inspection** - The firewall can decrypt outbound traffic, processes the data, then encrypt the data and sends it to the destination.
- **Outbound SNAT support** - All outbound virtual network traffic IP addresses are translated to the Azure Firewall public IP (Source Network Address Translation (SNAT)). You can identify and allow traffic originating from your virtual network to remote Internet destinations.
- **Inbound DNAT support** - Inbound Internet network traffic to your firewall public IP address is translated (Destination Network Address Translation) and filtered to the private IP addresses on your virtual networks.
- **Multiple public IP addresses** - You can associate multiple public IP addresses (up to 250) with your firewall, to enable specific DNAT and SNAT scenarios.
- **Azure Monitor logging** - All events are integrated with Azure Monitor, allowing you to archive logs to a storage account, stream events to your Event Hubs, or send them to Azure Monitor logs.
- **Forced tunneling** - You can configure Azure Firewall to route all Internet-bound traffic to a designated next hop instead of going directly to the Internet. For example, you may have an on-premises edge firewall or other network virtual appliance (NVA) to process network traffic before it is passed to the Internet.
- **Web categories** - Web categories let administrators allow or deny user access to web site categories such as gambling websites, social media websites, and others. Web categories are included in Azure Firewall Standard, but it is more fine-tuned in Azure Firewall Premium Preview. As opposed to the Web categories capability in

- the Standard SKU that matches the category based on an FQDN, the Premium SKU matches the category according to the entire URL for both HTTP and HTTPS traffic.
- **Certifications** - Azure Firewall is Payment Card Industry (PCI), Service Organization Controls (SOC), International Organization for Standardization (ISO), and ICSA Labs compliant.

Rule processing in Azure Firewall

In the Azure Firewall, you can configure NAT rules, network rules, and applications rules, and this can be done either by using classic rules or Firewall Policy. An Azure Firewall denies all traffic by default, until rules are manually configured to allow traffic.

Rule processing with classic rules

With classic rules, rule collections are processed according to the rule type in priority order, lower numbers to higher numbers from 100 to 65,000. A rule collection name can have only letters, numbers, underscores, periods, or hyphens. It must also begin with either a letter or a number, and it must end with a letter, a number, or an underscore. The maximum name length is 80 characters. It is best practice to initially space your rule collection priority numbers in increments of 100 (i.e., 100, 200, 300, and so on) so that you give yourself space to add more rule collections when needed.

Rule processing with Firewall Policy

With Firewall Policy, rules are organized inside Rule Collections which are contained in Rule Collection Groups. Rule Collections can be of the following types:

- DNAT (Destination Network Address Translation)
- Network
- Application

You can define multiple Rule Collection types within a single Rule Collection Group, and you can define zero or more Rules in a Rule Collection, but the rules within a Rule Collection must be of the same type (i.e., DNAT, Network, or Application).

With Firewall Policy, rules are processed based on Rule Collection Group Priority and Rule Collection priority. Priority is any number between 100 (highest priority) and 65,000 (lowest priority). Highest priority Rule Collection Groups are processed first, and inside a Rule Collection Group, Rule Collections with the highest priority (i.e., the lowest number) are processed first.

In the case of a Firewall Policy being inherited from a parent policy, Rule Collection Groups in the parent policy always takes precedence regardless of the priority of the child policy.

Application rules are always processed after network rules, which are themselves always processed after DNAT rules regardless of Rule Collection Group or Rule Collection priority and policy inheritance.

Outbound connectivity using network rules and application rules

If you configure both network rules and application rules, then network rules are applied in priority order before application rules. Additionally, all rules are terminating, therefore, if a match is found in a network rule, no other rules are processed thereafter.

If there is no network rule match, and if the protocol is either HTTP, HTTPS, or MSSQL, then the packet is then evaluated by the application rules in priority order. For HTTP, Azure Firewall looks for an application rule match according to the Host Header, whereas for HTTPS, Azure Firewall looks for an application rule match according to Server Name Indication (SNI) only.

Inbound connectivity using DNAT rules and network rules

Inbound Internet connectivity can be enabled by configuring DNAT. As mentioned previously, DNAT rules are applied in priority before network rules. If a match is found, an implicit corresponding network rule to allow the translated traffic is added. For security reasons, the recommended approach is to add a specific Internet source to allow DNAT access to the network and avoid using wildcards.

Application rules aren't applied for inbound connections. So, if you want to filter inbound HTTP/S traffic, you should use Web Application Firewall (WAF).

For enhanced security, if you modify a rule to deny access to traffic that had previously been allowed, any relevant existing sessions are dropped.

Deploying and configuring Azure Firewall

Be aware of the following when deploying Azure Firewall:

- It can centrally create, enforce, and log application and network connectivity policies across subscriptions and virtual networks.

- It uses a static, public IP address for your virtual network resources. This allows outside firewalls to identify traffic originating from your virtual network.
- It is fully integrated with Azure Monitor for logging and analytics.
- When creating firewall rules, it is best to use the FQDN tags.

The key stages of deploying and configuring Azure Firewall are as follows:

- Create a resource group
- Create a virtual network and subnets
- Create a workload VM in a subnet
- Deploy the firewall and policy to the virtual network
- Create a default outbound route
- Configure an application rule
- Configure a network rule
- Configure a Destination NAT (DNAT) rule
- Test the firewall

Deploying Azure Firewall with Availability Zones

One of the major features of Azure Firewall is Availability Zones.

When deploying Azure Firewall, you can configure it to span multiple Availability Zones for increased availability. When you configure Azure Firewall in this way your availability increases to 99.99% uptime. The 99.99% uptime SLA is offered when two or more Availability Zones are selected.

You can also associate Azure Firewall to a specific zone just for proximity reasons, using the service standard 99.95% SLA.

For more information, see the [Azure Firewall Service Level Agreement \(SLA\)](#).

There is no additional cost for a firewall deployed in an Availability Zone. However, there are added costs for inbound and outbound data transfers associated with Availability Zones.

For more information, see [Bandwidth pricing details](#).

Azure Firewall Availability Zones are only available in regions that support Availability Zones.

Availability Zones can only be configured during firewall deployment. You cannot configure an existing firewall to include Availability Zones.

Methods for deploying an Azure Firewall with Availability Zones

You can use several methods for deploying your Azure Firewall using Availability Zones.

- Azure portal
- Azure PowerShell - see [Deploy an Azure Firewall with Availability Zones using Azure PowerShell](#)
- Azure Resource Manager template - see [Quickstart: Deploy Azure Firewall with Availability Zones - Azure Resource Manager template](#)

Exercise: Deploy and configure Azure Firewall using the Azure portal

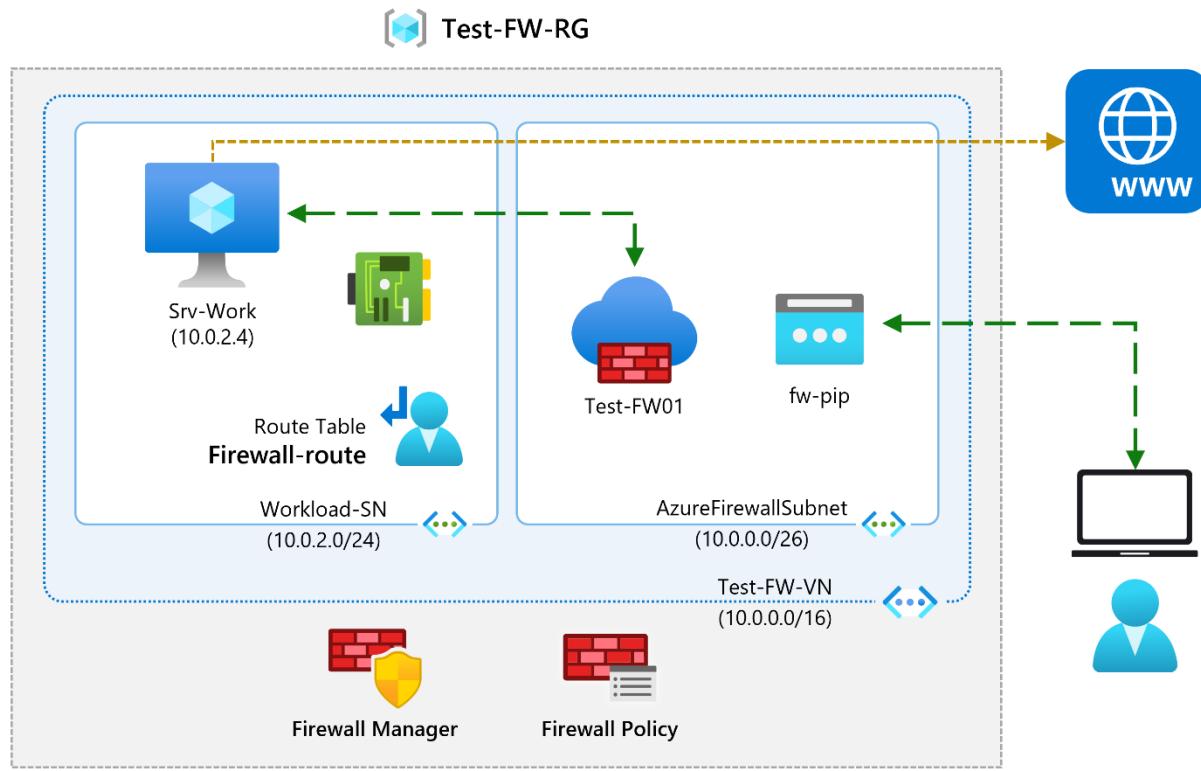
Completed 100 XP

- 9 minutes

Lab scenario

Being part of the Network Security team at Contoso, your next task is to create firewall rules to allow/deny access to certain websites. The following steps walk you through creating a resource group, a virtual network and subnets, and a virtual machine as environment preparation tasks, and then deploying a firewall and firewall policy, configuring default routes and application, network and DNAT rules, and finally testing the firewall.

Architecture diagram



Objectives

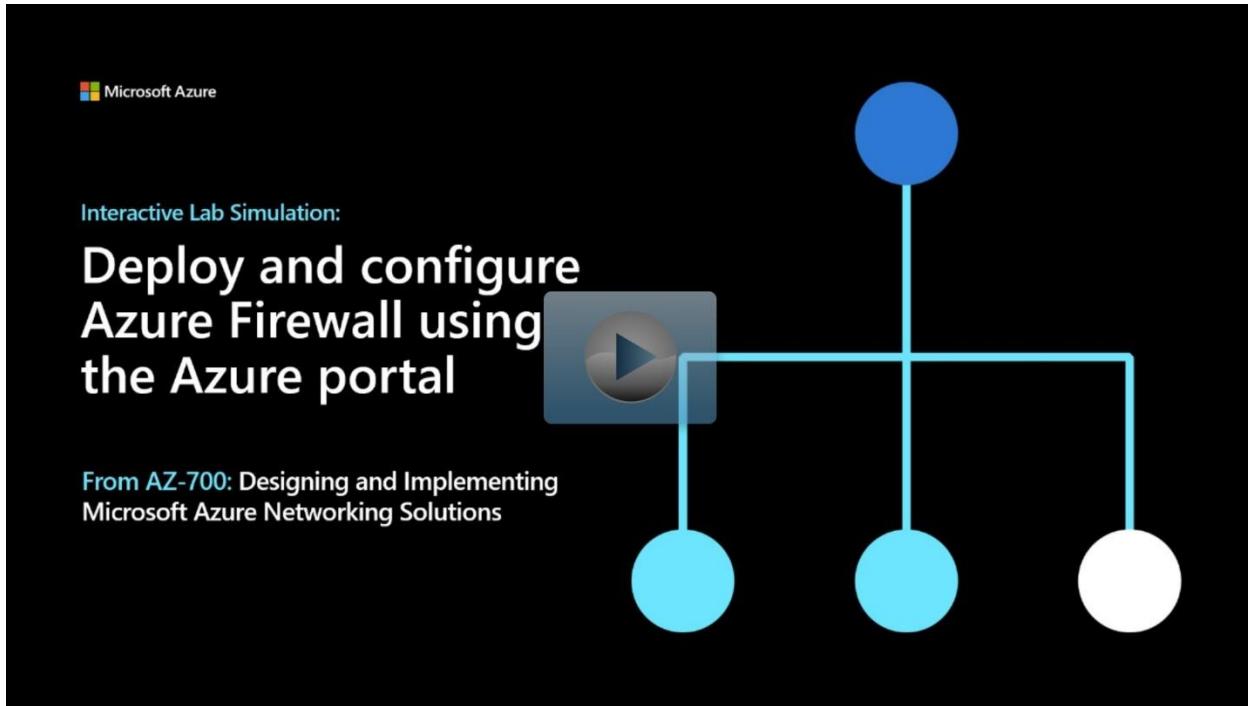
- **Task 1:** Create a virtual network and subnets
- **Task 2:** Create a virtual machine
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 3:** Deploy the firewall and firewall policy
- **Task 4:** Create a default route
- **Task 5:** Configure an application rule
- **Task 6:** Configure a network rule
- **Task 7:** Configure a Destination NAT (DNAT) rule
- **Task 8:** Change the primary and secondary DNS address for the server's network interface
- **Task 9:** Test the firewall

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Deploy%20and%20configure%20Azure%20Firewall%20using%20the%20Azure%20portal>

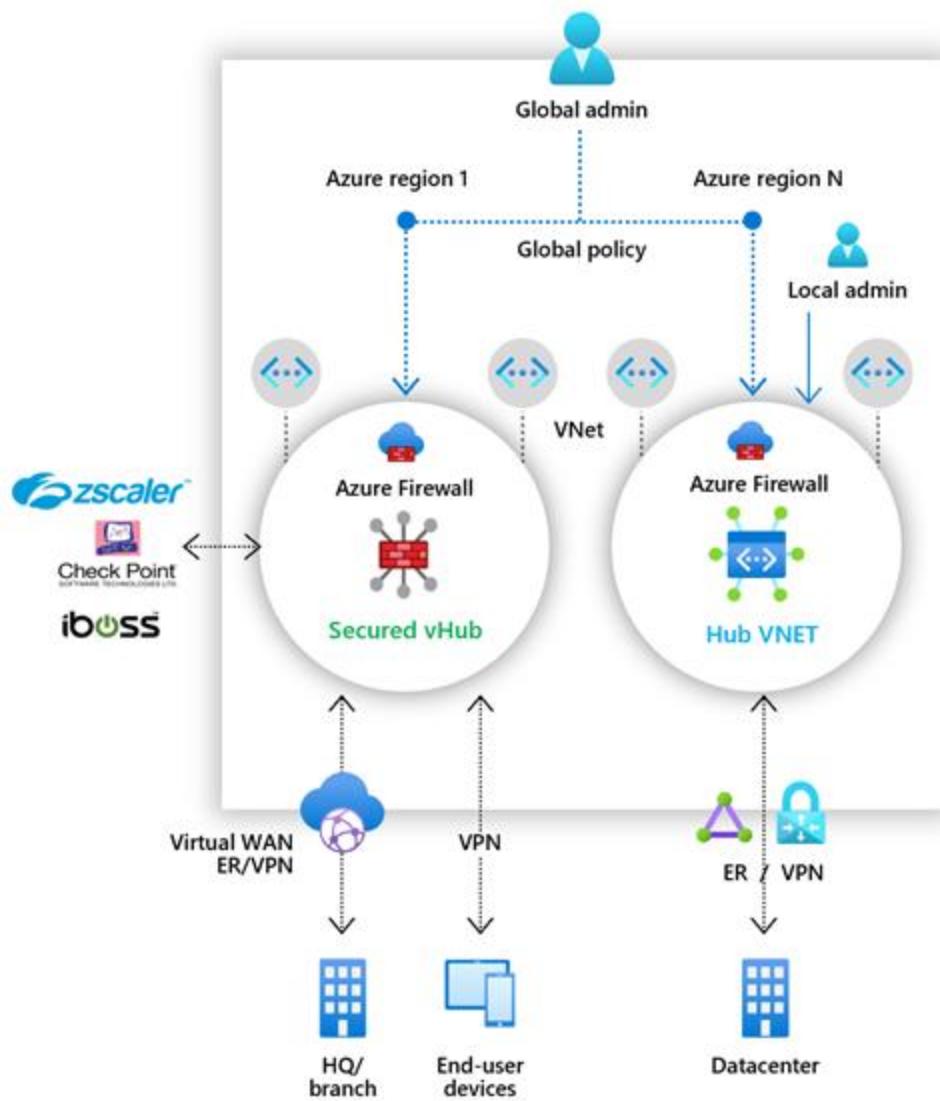
Secure your networks with Azure Firewall Manager

Completed 100 XP

- 7 minutes

Working with Azure Firewall Manager

Azure Firewall Manager is a security management service that provides central security policy and route management for cloud-based security perimeters.



Azure Firewall Manager simplifies the process of centrally defining network and application-level rules for traffic filtering across multiple Azure Firewall instances. You can span different Azure regions and subscriptions in hub and spoke architectures for traffic governance and protection.

If you manage multiple firewalls, you know that continuously changing firewall rules make it difficult to keep them in sync. Central IT teams need a way to define base firewall policies and enforce them across multiple business units. At the same time, DevOps teams want to create their own local derived firewall policies that are implemented across organizations. Azure Firewall Manager can help solve these problems.

Firewall Manager can provide security management for two network architecture types:

- **Secured Virtual Hub** - This is the name given to any Azure Virtual WAN Hub when security and routing policies have been associated with it. An Azure Virtual WAN Hub is a Microsoft-managed resource that lets you easily create hub and spoke architectures.
- **Hub Virtual Network** - This is the name given to any standard Azure virtual network when security policies are associated with it. A standard Azure virtual network is a resource that you create and manage yourself. At this time, only Azure Firewall Policy is supported. You can peer spoke virtual networks that contain your workload servers and services. You can also manage firewalls in standalone virtual networks that are not peered to any spoke.

Azure Firewall Manager features

The key features offered by Azure Firewall Manager are:

- **Central Azure Firewall deployment and configuration**

You can centrally deploy and configure multiple Azure Firewall instances that span different Azure regions and subscriptions.
- **Hierarchical policies (global and local)**

You can use Azure Firewall Manager to centrally manage Azure Firewall policies across multiple secured virtual hubs. Your central IT teams can author global firewall policies to enforce organization wide firewall policy across teams. Locally authored firewall policies allow a DevOps self-service model for better agility.
- **Integrated with third-party security-as-a-service for advanced security**

In addition to Azure Firewall, you can integrate third-party security-as-a-service providers to provide additional network protection for your VNet and branch Internet connections. This feature is available only with secured virtual hub deployments (see above).
- **Centralized route management**

You can easily route traffic to your secured hub for filtering and logging without the need to manually set up User Defined Routes (UDR) on spoke

virtual networks. This feature is available only with secured virtual hub deployments (see above).

- **Region availability**

You can use Azure Firewall Policies across regions. For example, you can create a policy in the West US region, and still use it in the East US region.

- **DDoS protection plan**

You can associate your virtual networks with a DDoS protection plan within Azure Firewall Manager.

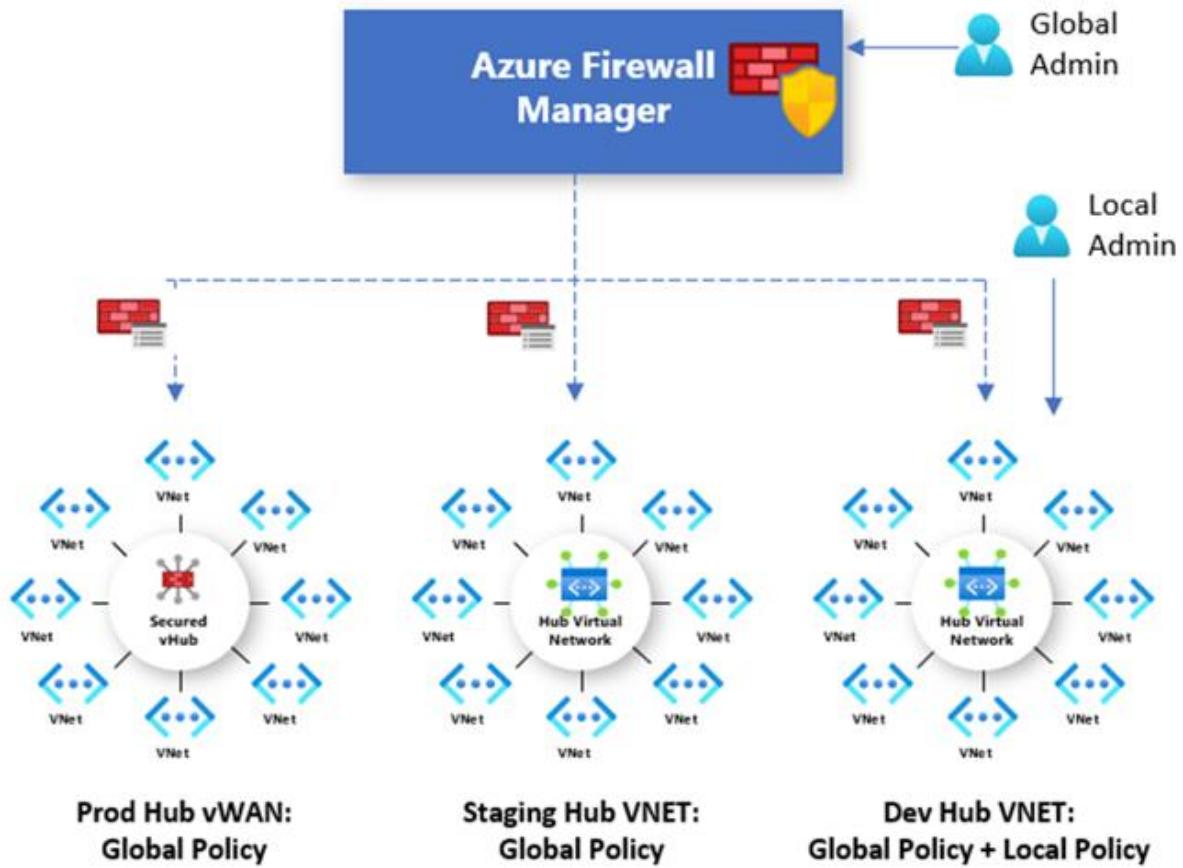
- **Manage Web Application Firewall policies**

You can centrally create and associate Web Application Firewall (WAF) policies for your application delivery platforms, including Azure Front Door and Azure Application Gateway.

Azure Firewall Manager policies

A Firewall policy is an Azure resource that contains NAT, network, and application rule collections and Threat Intelligence settings. It is a global resource that can be used across multiple Azure Firewall instances in Secured Virtual Hubs and Hub Virtual Networks. New policies can be created from scratch or inherited from existing policies. Inheritance allows DevOps to create local firewall policies on top of organization mandated base policy. Policies work across regions and subscriptions.

You can create Firewall Policy and associations with Azure Firewall Manager. However, you can also create and manage a policy using REST API, templates, Azure PowerShell, and the Azure CLI. Once you create a policy, you can associate it with a firewall in a virtual WAN hub making it a Secured Virtual Hub and/or associate it with a firewall in a standard Azure virtual network making it a Hub Virtual Network.



Deploying Azure Firewall Manager for Hub Virtual Networks

The recommended process to deploy Azure Firewall Manager for Hub Virtual Networks is as follows:

- 1. Create a firewall policy**

You can either create a new policy, derive a base policy, and customize a local policy, or import rules from an existing Azure Firewall. Ensure you remove NAT rules from policies that should be applied across multiple firewalls.

- 2. Create your hub and spoke architecture**

Do this either by creating a Hub Virtual Network using Azure Firewall Manager and peering spoke virtual networks to it using virtual network

peering, or by creating a virtual network and adding virtual network connections and peering spoke virtual networks to it using virtual network peering.

3. Select security providers and associate firewall policy

Currently, only Azure Firewall is a supported provider. This can be done while creating a Hub Virtual Network, or by converting an existing virtual network to a Hub Virtual Network. It is also possible to convert multiple virtual networks.

4. Configure User Defined Routes to route traffic to your Hub Virtual Network firewall

Deploying Azure Firewall Manager for Secured Virtual Hubs

The recommended process to deploy Azure Firewall Manager for Secured Virtual Hubs is as follows:

1. Create your hub and spoke architecture

Do this either by creating a Secured Virtual Hub using Azure Firewall Manager and adding virtual network connections, or by creating a Virtual WAN Hub and adding virtual network connections.

2. Select security providers

This can be done while creating a Secured Virtual Hub, or by converting an existing Virtual WAN Hub to a Secure Virtual Hub.

3. Create a firewall policy and associate it with your hub

This is applicable only if you are using Azure Firewall. Third-party security-as-a-service policies are configured via the partners management experience.

4. Configure route settings to route traffic to your Secured Virtual Hub

You can easily route traffic to your secured hub for filtering and logging without User Defined Routes (UDR) on spoke Virtual Networks by using the Secured Virtual Hub Route Setting page.

You cannot have more than one hub per virtual WAN per region, however you can add multiple virtual WANs in the region to achieve this.

You cannot have overlapping IP spaces for hubs in a vWAN.

Your hub VNet connections must be in the same region as the hub.

Exercise: secure your virtual hub using Azure Firewall Manager

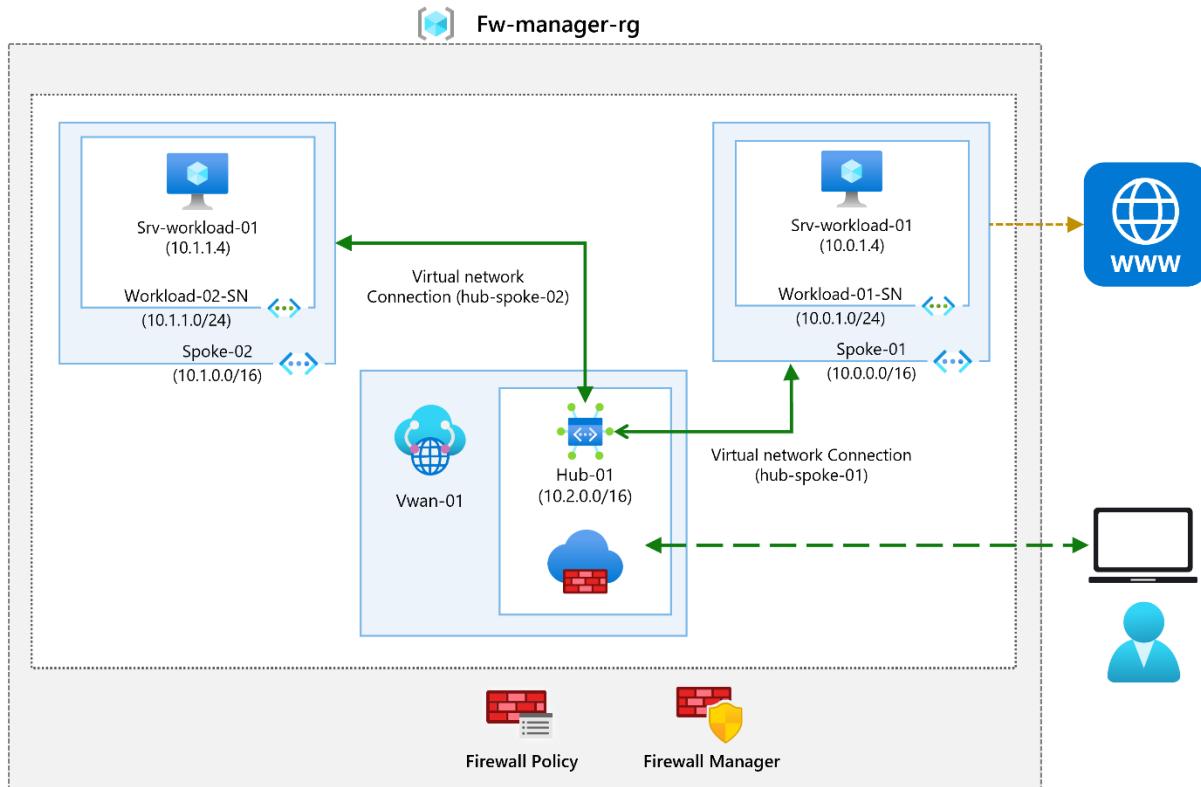
Completed 100 XP

- 11 minutes

Lab scenario

In this lab, you will secure your virtual hub with Azure Firewall Manager.

Architecture diagram



Objectives

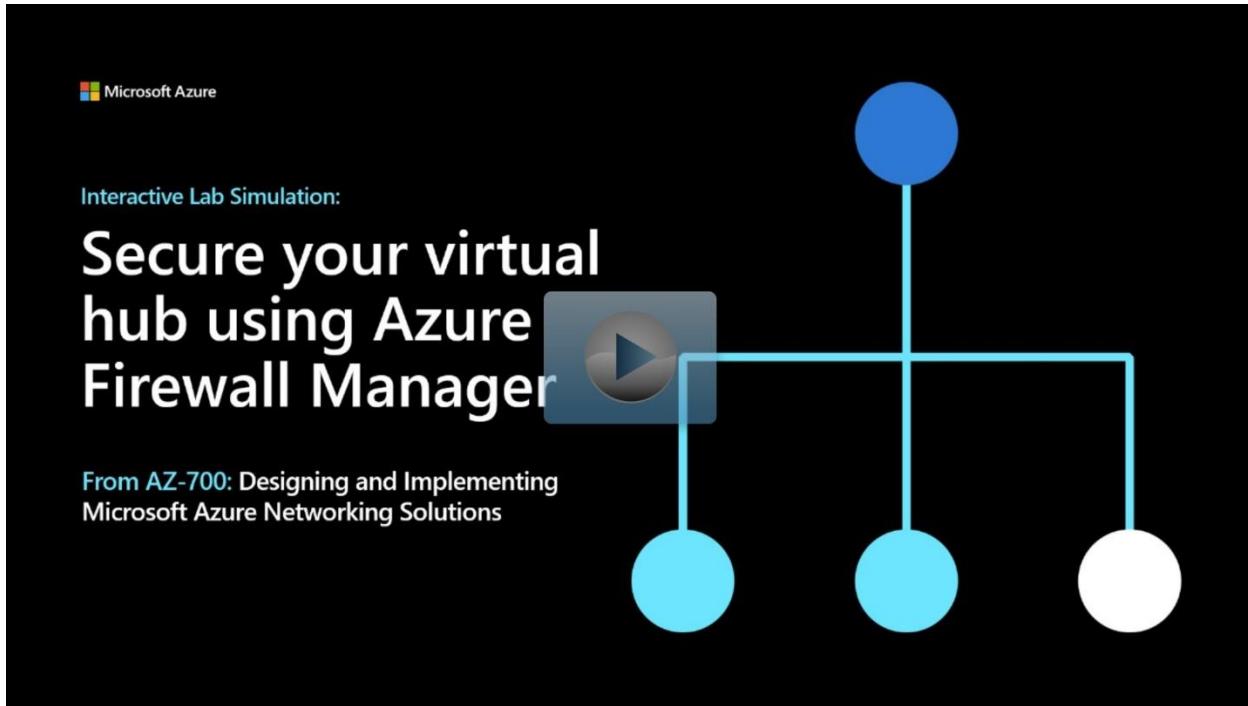
- **Task 1:** Create two spoke virtual networks and subnets
- **Task 2:** Create the secured virtual hub
- **Task 3:** Connect the hub and spoke virtual networks
- **Task 4:** Deploy the servers
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 5:** Create a firewall policy and secure your hub
- **Task 6:** Associate the firewall policy
- **Task 7:** Route traffic to your hub
- **Task 8:** Test the application rule
- **Task 9:** Test the network rule

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



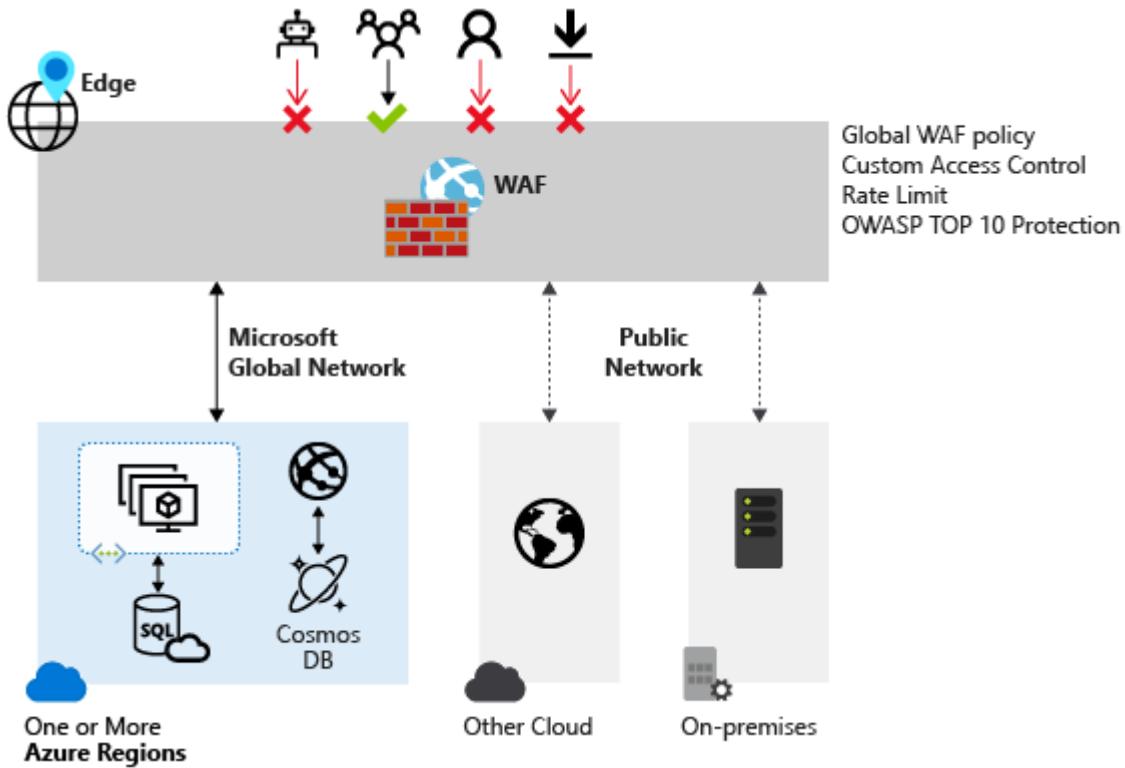
<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-Secure%20your%20virtual%20hub%20using%20Azure%20Firewall%20Manager>

Implement a Web Application Firewall on Azure Front Door

200 XP

- 6 minutes

Web Application Firewall (WAF) provides centralized protection of your web applications from common exploits and vulnerabilities. Web applications are increasingly targeted by malicious attacks that exploit commonly known vulnerabilities. SQL injection and cross-site scripting are among the most common attacks.



Preventing such attacks in application code is challenging. It can require rigorous maintenance, patching, and monitoring at multiple layers of the application topology. A centralized web application firewall helps make security management much simpler. A WAF also gives application administrators better assurance of protection against threats and intrusions.

A WAF solution can react to a security threat faster by centrally patching a known vulnerability, instead of securing each individual web application.

Web Application Firewall policy modes

When you create a Web Application Firewall (WAF) policy, by default the WAF policy is in Detection mode. In Detection mode, WAF does not block any requests; instead, requests matching the WAF rules are logged at WAF logs. To see WAF in action, you can change the mode settings from Detection to Prevention. In Prevention mode, requests that match rules that are defined in Default Rule Set (DRS) are blocked and logged at WAF logs.

A screenshot of the Azure WAF policy settings page for 'wafpolicy1'. The 'Policy settings' tab is selected. The 'Mode' section shows 'Prevention' is selected (radio button is unselected) and 'Detection' is selected (radio button is selected). Below it are fields for 'Redirect URL' (empty), 'Block response status code' (set to 403), and 'Block response body' (with a placeholder message). The left sidebar shows other tabs like Overview, Activity log, Access control (IAM), Tags, and Managed rules.

Web Application Firewall Default Rule Set rule groups and rules

Azure Front Door web application firewall (WAF) protects web applications from common vulnerabilities and exploits. Azure-managed rule sets provide an easy way to deploy protection against a common set of security threats. Since such rule sets are managed by Azure, the rules are updated as needed to protect against new attack signatures.

Managed rules

Azure-managed Default Rule Set includes rules against the following threat categories:

- Cross-site scripting
- Java attacks
- Local file inclusion
- PHP injection attacks
- Remote command execution
- Remote file inclusion
- Session fixation
- SQL injection protection
- Protocol attackers

Azure-managed Default Rule Set is enabled by default. The current default version is DefaultRuleSet_2.1. Other rule sets are available in the drop-down box.

To disable an individual rule, select the checkbox in front of the rule number, and select **Disable** at the top of the page. To change action types for individual rules within

the rule set, select the checkbox in front of the rule number, and then select **Change action** at the top of the page.

Rule Id	Description	Action	Status
921110	HTTP Request Smuggling Attack	Block	Enabled
921120	HTTP Response Splitting Attack	Block	Enabled
921130	HTTP Response Splitting Attack	Block	Enabled
921140	HTTP Header Injection Attack via headers	Block	Enabled
921150	HTTP Header Injection Attack via payload (CR/LF detected)	Block	Enabled
921160	HTTP Header Injection Attack via payload (CR/LF and header-name detected)	Block	Enabled
921151	HTTP Header Injection Attack via payload (CR/LF detected)	Block	Enabled
930100	Path Traversal Attack (./..)	Block	Enabled
930110	Path Traversal Attack (./..)	Block	Enabled
930120	OS File Access Attempt	Block	Enabled
930130	Restricted File Access Attempt	Block	Enabled

Custom rules

Azure WAF with Front Door allows you to control access to your web applications based on the conditions you define. A custom WAF rule consists of a priority number, rule type, match conditions, and an action. There are two types of custom rules: match rules and rate limit rules. A match rule controls access based on a set of matching conditions while a rate limit rule controls access based on matching conditions and the rates of incoming requests. You may disable a custom rule to prevent it from being evaluated, but still keep the configuration.

When creating a WAF policy, you can create a custom rule by selecting **Add custom rule** under the **Custom rules** section. This launches the custom rule configuration page.

The screenshot shows the 'Custom rules' page for a WAF policy named 'wafpolicy1'. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Settings (Policy settings, Managed rules, Custom rules, Associations), and a search bar. The main content area displays a table for custom rules, with a red box highlighting the 'Add custom rule' button. A descriptive text at the top explains how custom rules work.

wafpolicy1 | Custom rules

Front Door WAF policy

Search (Ctrl+ /) Save Discard Refresh

Overview Activity log Access control (IAM) Tags

Policy settings

Managed rules

Custom rules

Associations

Configure a policy with custom-authored rules. Once a rule is matched, the corresponding action defined in the rule is applied to the request. Once such a match is processed, rules with lower priorities are not processed further. A smaller integer value for a rule denotes a higher priority. [Learn more](#)

Add custom rule

Priority	Name	Rule type	Action	Status
No custom rules to display.				

The example screenshot below shows the configuration of a custom rule to block a request if the query string contains **blockme**.

Add custom rule

X

A custom rule is made up of one or more conditions followed by an action. All custom rules for a WAF policy are match rules. [Learn more about custom rules](#)

Custom rule name *	blockQSexample
Status ⓘ	<input checked="" type="button"/> Enabled <input type="button"/> Disabled
Rule type ⓘ	<input checked="" type="button"/> Match <input type="button"/> Rate limit
Priority * ⓘ	4

Conditions

If

Match type ⓘ	String
Match variable *	QueryString
Operation	<input checked="" type="radio"/> is <input type="radio"/> is not
Operator *	Contains
Transformation ⓘ	Select a transformation
Match values	blockme
	Enter a match value
+ Add new condition	
Then	Deny traffic

Add Cancel

Create a Web Application Firewall policy on Azure Front Door

This section describes how to create a basic Azure Web Application Firewall (WAF) policy and apply it to a profile in Azure Front Door.

The key stages to create a WAF policy on Azure Front Door using the Azure portal are:

1. Create a Web Application Firewall policy

This is where you create a basic WAF policy with managed Default Rule Set (DRS).

2. Associate the WAF policy with a Front Door profile

This is where you associate the WAF policy created in stage 1 with a Front Door profile. This association can be done during the creation of the WAF policy, or it can be done on a previously created WAF policy. During the association you specify the Front Door profile and the domain/s within the Front Door profile you want the WAF policy to be applied to. During this stage if the domain is associated to a WAF policy, it is shown as grayed out. You must first remove the domain from the associated policy, and then re-associate the domain to a new WAF policy.

3. Configure WAF policy settings and rules

This is an optional stage, where you can configure policy settings such as the **Mode** (Prevention or Detection) and configure managed rules and custom rules.

To view the detailed steps for all these tasks, see [Tutorial: Create a Web Application Firewall policy on Azure Front Door using the Azure portal](#).

Summary and resources

Completed 100 XP

- 1 minute

As your organization moves to Azure, you must design your network to protect resources from unauthorized access or attack by applying controls to network traffic, allowing only legitimate traffic/requests. In this module, you saw a range of network

security solutions that you could implement to meet your organization's network security requirements.

You now have the fundamental knowledge required to design and implement network security in Azure.

Design and implement private access to Azure Services

Explain virtual network service endpoints

Completed 100 XP

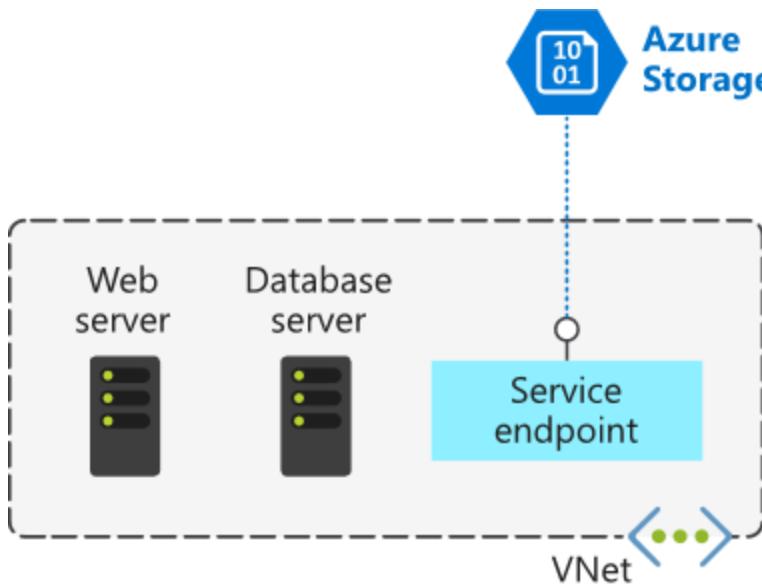
- 22 minutes

You've migrated your existing app and database servers for your ERP system to Azure as VMs. Now, to reduce your costs and administrative requirements, you're considering using some Azure platform as a service (PaaS) services. Storage services will hold certain large file assets, such as engineering diagrams. These engineering diagrams have proprietary information, and must remain secure from unauthorized access. These files must only be accessible from specific systems.

In this unit, you'll look at how to use virtual network service endpoints for securing supported Azure services.

What is a virtual network service endpoint?

Use virtual network service endpoints to extend your private address space in Azure by providing a direct connection to your Azure services. Service endpoints let you secure your Azure resources to only your virtual network. Service traffic will remain on the Azure backbone, and doesn't go out to the internet.



By default, Azure services are all designed for direct internet access. All Azure resources have public IP addresses, including PaaS services such as Azure SQL Database and Azure Storage. Because these services are exposed to the internet, anyone can potentially access your Azure services.

Service endpoints can connect certain PaaS services directly to your private address space in Azure, so they act like they're on the same virtual network. Use your private address space to access the PaaS services directly. Adding service endpoints doesn't remove the public endpoint. It simply provides a redirection of traffic.

Preparing to Implement Service Endpoints

To enable a Service Endpoint, you must do the following two things:

- Turn off public access to the service.
- Add the Service Endpoint to a virtual network.

When you enable a Service Endpoint, you restrict the flow of traffic, and enable your Azure VMs to access the service directly from your private address space. Devices cannot access the service from a public network. On a deployed VM vNIC, if you look at Effective routes, you'll notice the Service Endpoint as the Next Hop Type.

This is an example route table, before enabling a Service Endpoint:

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE
Default	Active	10.1.1.0/24	VNet
Default	Active	0.0.0.0/0	Internet
Default	Active	10.0.0.0/8	None
Default	Active	100.64.0.0/	None
Default	Active	192.168.0.0/16	None

And here's an example route table after you've added two Service Endpoints to the virtual network:

[Expand table](#)

SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE
Default	Active	10.1.1.0/24	VNet
Default	Active	0.0.0.0/0	Internet
Default	Active	10.0.0.0/8	None
Default	Active	100.64.0.0/	None
Default	Active	192.168.0.0/16	None
Default	Active	20.38.106.0/23, 10 more	VirtualNetworkServiceEndpoint
Default	Active	20.150.2.0/23, 9 more	VirtualNetworkServiceEndpoint

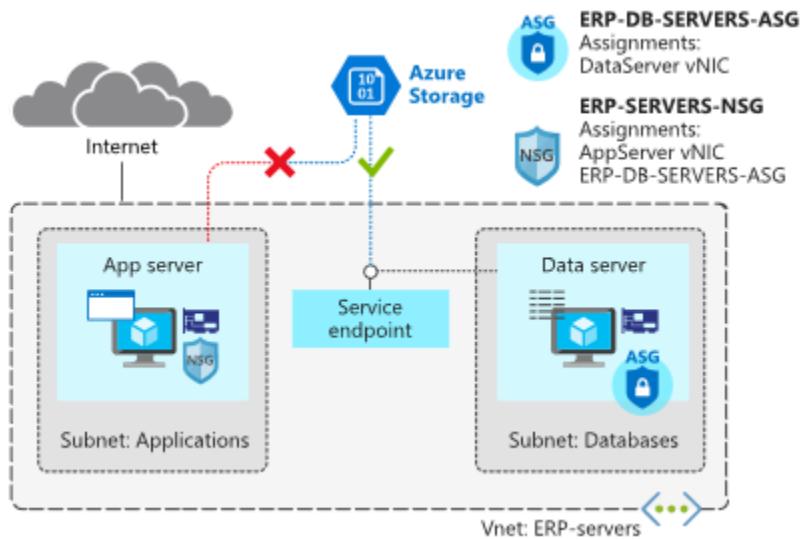
All traffic for the service now is routed to the Virtual Network Service Endpoint and remains internal to Azure.

Create Service Endpoints

As the network engineer, you're planning to move sensitive engineering diagram files into Azure Storage. The files must only be accessible from computers inside the corporate network. You want to create a virtual network Service Endpoint for Azure Storage to secure the connectivity to your storage accounts.

In the [service endpoint tutorial](#) you will learn how to:

- Enable a service endpoint on a subnet
- Use network rules to restrict access to Azure Storage
- Create a virtual network service endpoint for Azure Storage
- Verify that access is denied appropriately



Configure service tags

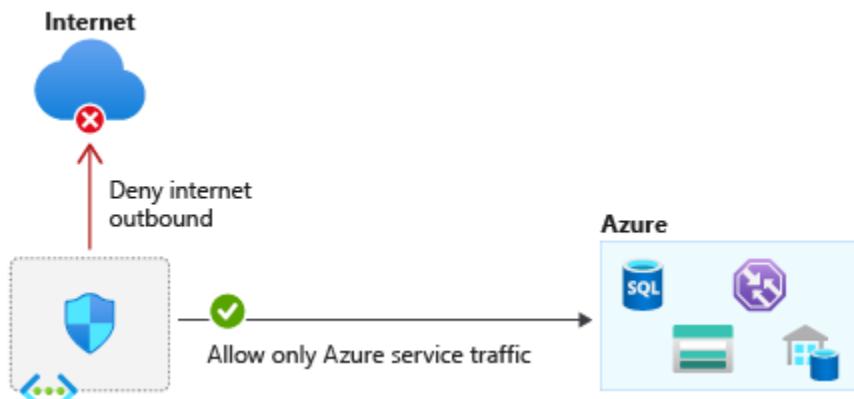
A service tag represents a group of IP address prefixes from a given Azure service. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change, minimizing the complexity of frequent updates to network security rules.

You can use service tags to define network access controls on network security groups or Azure Firewall. Use service tags in place of specific IP addresses when you create security rules. By specifying the service tag name, such as API Management, in the appropriate source or destination field of a rule, you can allow or deny the traffic for the corresponding service.

As of March 2021, you can also use Service Tags in place of explicit IP ranges in user defined routes. This feature is currently in Public Preview.

You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. Create inbound/outbound network security group rules to deny traffic to/from Internet

and allow traffic to/from AzureCloud or other available service tags of specific Azure services.



Network Security Group (NSG)				
Action	Name	Source	Destination	Port
Allow	AllowStorage	VirtualNetwork	Storage	Any
Allow	AllowSQL	VirtualNetwork	Sql.EastUS	Any
Deny	DenyAllOutBound	Any	Any	Any

Available service tags

The following table includes all the service tags available for use in network security group rules.

The columns indicate whether the tag:

- Is suitable for rules that cover inbound or outbound traffic.
- Supports regional scope.
- Is usable in Azure Firewall rules.

By default, service tags reflect the ranges for the entire cloud. Some service tags also allow more granular control by restricting the corresponding IP ranges to a specified region. For example, the service tag Storage represents Azure Storage for the entire cloud, but Storage.WestUS narrows the range to only the storage IP address ranges from the WestUS region. The [following table](#) indicates whether each service tag supports such regional scope.

Service tags of Azure services denote the address prefixes from the specific cloud being used. For example, the underlying IP ranges that correspond to the SQL tag value on the Azure Public cloud will be different from the underlying ranges on the Azure China cloud.

If you implement a virtual network Service Endpoint for a service, such as Azure Storage or Azure SQL Database, Azure adds a route to a virtual network subnet for the service. The address prefixes in the route are the same address prefixes, or CIDR ranges, as those of the corresponding service tag.

Define Private Link Service and private endpoint

200 XP

- 7 minutes

What is Azure Private Link?

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a Private Endpoint in your virtual network.

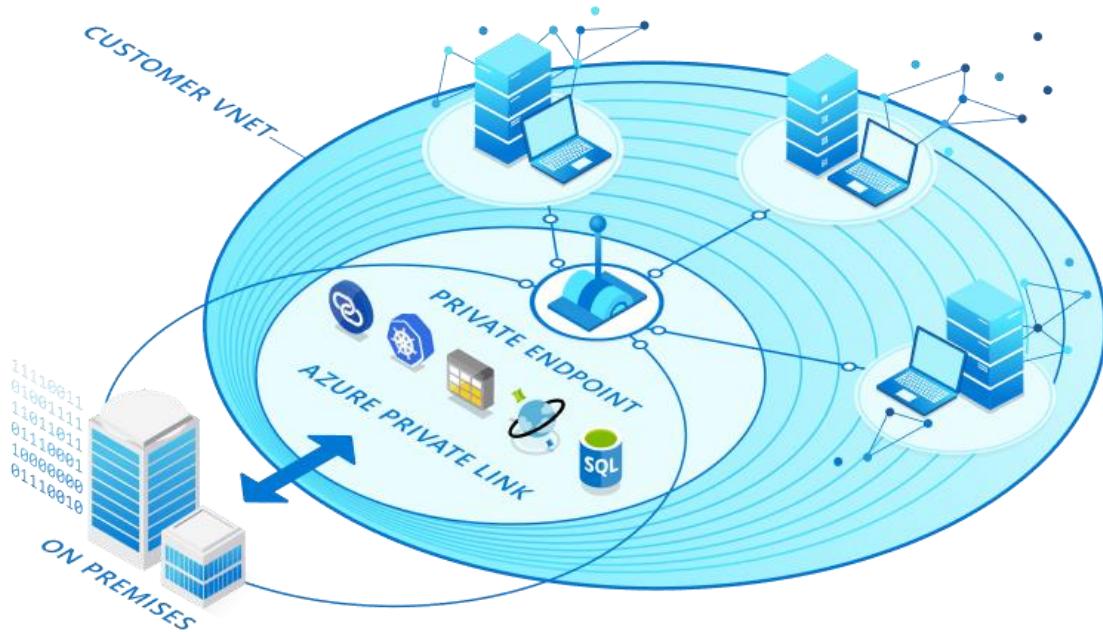
Before you learn about Azure Private Link and its features and benefits, let's examine the problem that Private Link is designed to solve.

Contoso has an Azure virtual network, and you want to connect to a PaaS resource such as an Azure SQL database. When you create such resources, you normally specify a public endpoint as the connectivity method.

Having a public endpoint means that the resource is assigned a public IP address. So, even though both your virtual network and the Azure SQL database are located within the Azure cloud, the connection between them takes place over the internet.

The concern here is that your Azure SQL database is exposed to the internet via its public IP address. That exposure creates multiple security risks. The same security risks are present when an Azure resource is accessed via a public IP address from the following locations:

- A peered Azure virtual network
- An on-premises network that connects to Azure using ExpressRoute and Microsoft peering
- A customer's Azure virtual network that connects to an Azure service offered by your company



Private Link is designed to eliminate these security risks by removing the public part of the connection.

Private Link provides secure access to Azure services. Private Link achieves that security by replacing a resource's public endpoint with a private network interface. There are three key points to consider with this new architecture:

- The Azure resource becomes, in a sense, a part of your virtual network.
- The connection to the resource now uses the Microsoft Azure backbone network instead of the public internet.
- You can configure the Azure resource to no longer expose its public IP address, which eliminates that potential security risk.

What is Azure Private Endpoint?

Private Endpoint is the key technology behind Private Link. Private Endpoint is a network interface that enables a private and secure connection between your virtual network and

an Azure service. In other words, Private Endpoint is the network interface that replaces the resource's public endpoint.

Private Link provides secure access to Azure services. Private Link achieves that security by replacing a resource's public endpoint with a private network interface. Private Endpoint uses a private IP address from the VNet to bring the service into the VNet.

How is Azure Private Endpoint different from a service endpoint?

Private Endpoints grant network access to specific resources behind a given service providing granular segmentation. Traffic can reach the service resource from on premises without using public endpoints.

A service endpoint remains a publicly routable IP address. A private endpoint is a private IP in the address space of the virtual network where the private endpoint is configured.

Note

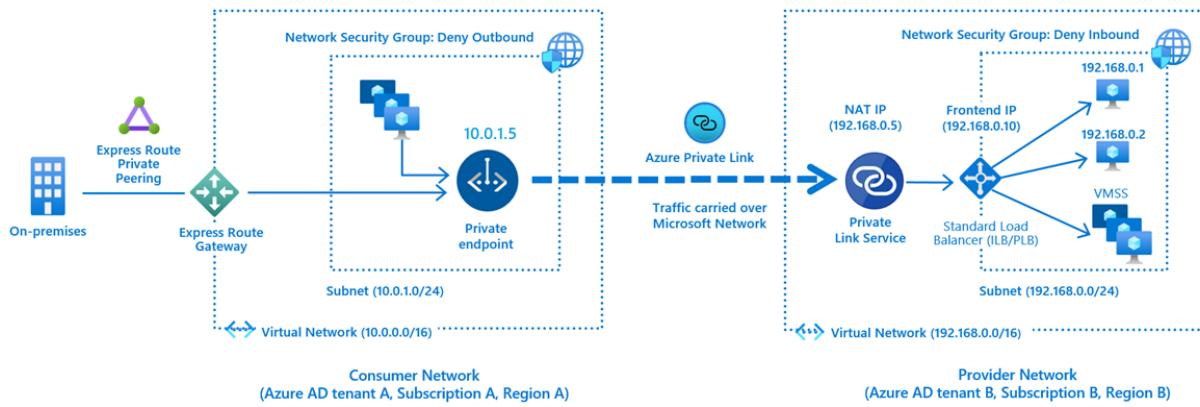
Microsoft recommends use of Azure Private Link for secure and private access to services hosted on Azure platform.

What is Azure Private Link Service?

Private Link gives you private access from your Azure virtual network to PaaS services and Microsoft Partner services in Azure. However, what if your company has created its own Azure services that are consumed by your company's customers? Is it possible to offer those customers a private connection to your company's services?

Yes, by using Azure Private Link Service. This service lets you offer Private Link connections to your custom Azure services. Consumers of your custom services can then access those services privately—that is, without using the internet—from their own Azure virtual networks.

Azure Private Link service is the reference to your own service that is powered by Azure Private Link. Your service that is running behind Azure standard load balancer can be enabled for Private Link access so that consumers to your service can access it privately from their own VNets. Your customers can create a private endpoint inside their VNet and map it to this service. A Private Link service receives connections from multiple private endpoints. A private endpoint connects to one Private Link service.



Private Endpoint properties

Before creating a Private Endpoint, you should consider the Private Endpoint properties and collect data about specific needs to be addressed. These include:

- A unique name with a resource group
- A subnet to deploy and allocate private IP addresses from a virtual network
- The Private Link resource to connect using resource ID or alias, from the list of available types. A unique network identifier will be generated for all traffic sent to this resource.
- The subresource to connect. Each Private Link resource type has different options to select based on preference.
- An automatic or manual connection approval method. Based on Azure role-based access control (Azure RBAC) permissions, your Private Endpoint can be approved automatically. If you try to connect to a Private Link resource without Azure RBAC, use the manual method to allow the owner of the resource to approve the connection.
- A specific request message for requested connections to be approved manually. This message can be used to identify a specific request.
- Connection status, A read-only property that specifies if the Private Endpoint is active. Only Private Endpoints in an approved state can be used to send traffic.

Also consider the following details:

- Private Endpoint enables connectivity between the consumers from the same VNet, regionally peered VNets, globally peered VNets and on premises using VPN or Express Route and services powered by Private Link.

- Network connections can only be initiated by clients connecting to the Private Endpoint. Service providers do not have any routing configuration to initiate connections into service consumers. Connections can only be established in a single direction.
- When creating a Private Endpoint, a read-only network interface is also created for the lifecycle of the resource. The interface is assigned dynamically private IP addresses from the subnet that maps to the Private Link resource. The value of the private IP address remains unchanged for the entire lifecycle of the Private Endpoint.
- The Private Endpoint must be deployed in the same region and subscription as the virtual network.
- The Private Link resource can be deployed in a different region than the virtual network and Private Endpoint.
- Multiple Private Endpoints can be created using the same Private Link resource. For a single network using a common DNS server configuration, the recommended practice is to use a single Private Endpoint for a given Private Link resource to avoid duplicate entries or conflicts in DNS resolution.
- Multiple Private Endpoints can be created on the same or different subnets within the same virtual network. There are limits to the number of Private Endpoints you can create in a subscription. For details, see Azure limits.
- The subscription from the Private Link resource must also be registered with Microsoft.

Integrate private endpoint with DNS

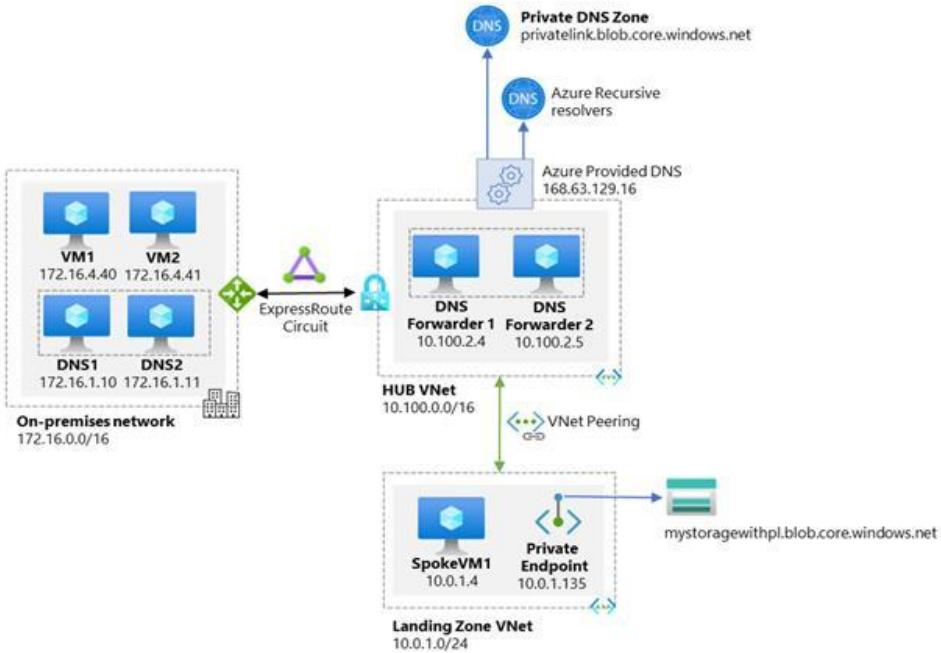
200 XP

- 17 minutes

Private DNS zones are typically hosted centrally in the same Azure subscription where the hub VNet is deployed. This central hosting practice is driven by cross-premises DNS name resolution and other needs for central DNS resolution such as Active Directory. In most cases, only networking/identity admins have permissions to manage DNS records in these zones.

Azure Private Endpoint DNS configuration

The following diagram shows a typical high-level architecture for enterprise environments with central DNS resolution and where name resolution for Private Link resources is done via Azure Private DNS:



From the previous diagram, it is important to highlight that:

- On-premises DNS servers have conditional forwarders configured for each Private Endpoint public DNS zone forwarder pointing to the DNS forwarders (10.100.2.4 and 10.100.2.5) hosted in the hub VNet.
- The DNS servers 10.100.2.4 and 10.100.2.5 hosted in the hub VNet use the Azure-provided DNS resolver (168.63.129.16) as a forwarder.
- All Azure VNets have the DNS forwarders (10.100.2.4 and 10.100.2.5) configured as the primary and secondary DNS servers.
- There are two conditions that must be true to allow application teams the freedom to create any required Azure PaaS resources in their subscription:
 - Central networking and/or central platform team must ensure that application teams can only deploy and access Azure PaaS services via Private Endpoints.
 - Central networking and/or central platform teams must ensure that whenever Private Endpoints are created, the corresponding records are automatically created in the centralized private DNS zone that matches the service created.
- DNS record needs to follow the lifecycle of the Private Endpoint and automatically remove the DNS record when the Private Endpoint is deleted.

Significance of IP address 168.63.129.16

IP address 168.63.129.16 is a virtual public IP address that is used to facilitate a communication channel to Azure platform resources. Customers can define any address

space for their private virtual network in Azure. The Azure platform resources must be presented as a unique public IP address. This virtual public IP address facilitates the following things:

- Enables the VM Agent to communicate with the Azure platform to signal that it is in a "Ready" state
- Enables communication with the DNS virtual server to provide filtered name resolution to the resources (such as VM) that do not have a custom DNS server. This filtering makes sure that customers can resolve only the hostnames of their resources
- Enables health probes from Azure load balancer to determine the health state of VMs
- Enables the VM to obtain a dynamic IP address from the DHCP service in Azure
- Enables Guest Agent heartbeat messages for the PaaS role

Azure services Private DNS zone configuration examples

Azure creates a canonical name DNS record (CNAME) on the public DNS. The CNAME record redirects the resolution to the private domain name. You can override the resolution with the private IP address of your Private Endpoints.

Your applications don't need to change the connection URL. When resolving to a public DNS service, the DNS server will resolve to your Private Endpoints. The process doesn't affect your existing applications.

Private networks already using the private DNS zone for a given type, can only connect to public resources if they don't have any Private Endpoint connections, otherwise a corresponding DNS configuration is required on the private DNS zone in order to complete the DNS resolution sequence.

For Azure services, use the [recommended zone names](#) found in the documentation.

DNS configuration scenarios

The FQDN of the services resolves automatically to a public IP address. To resolve to the private IP address of the Private Endpoint, change your DNS configuration.

DNS is a critical component to make the application work correctly by successfully resolving the Private Endpoint IP address.

Based on your preferences, the following scenarios are available with DNS resolution integrated:

- [Virtual network workloads without custom DNS server](#)
- [On-premises workloads using a DNS forwarder](#)
- [Virtual network and on-premises workloads using a DNS forwarder](#)
- [Private DNS zone group](#)

On-premises workloads using a DNS forwarder

For on-premises workloads to resolve the FQDN of a Private Endpoint, use a DNS forwarder to resolve the Azure service public DNS zone in Azure. A DNS forwarder is a Virtual Machine running on the Virtual Network linked to the Private DNS Zone that can proxy DNS queries coming from other Virtual Networks or from on-premises. This is required as the query must be originated from the Virtual Network to Azure DNS. A few options for DNS proxies are: Windows running DNS services, Linux running DNS services, Azure Firewall.

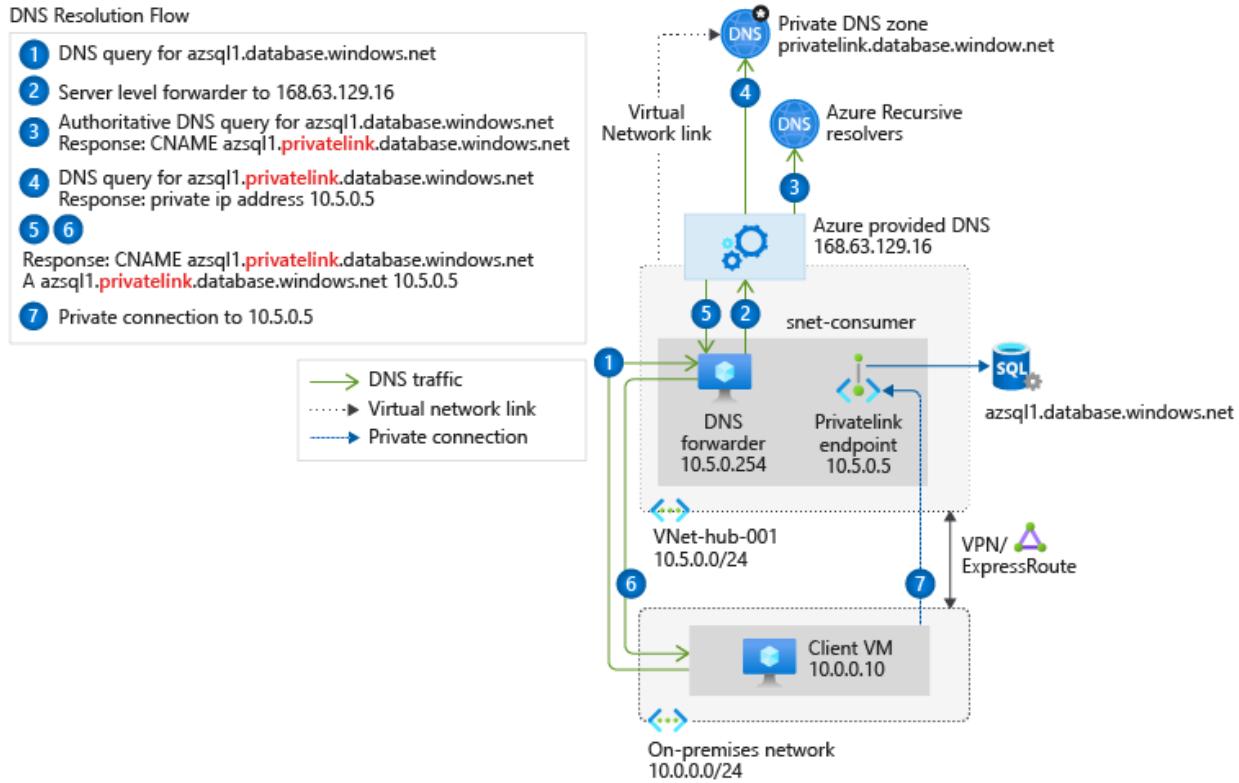
The following scenario is for an on-premises network that has a DNS forwarder in Azure. This forwarder resolves DNS queries via a server-level forwarder to the Azure provided DNS 168.63.129.16.

This scenario uses the Azure SQL Database-recommended private DNS zone. For other services, you can adjust the model using the following reference: Azure services DNS zone configuration.

To configure properly, you need the following resources:

- On-premises network
- Virtual network connected to on-premises
- DNS forwarder deployed in Azure
- Private DNS zones privatelink.database.windows.net with type A record
- Private Endpoint information (FQDN record name and private IP address)

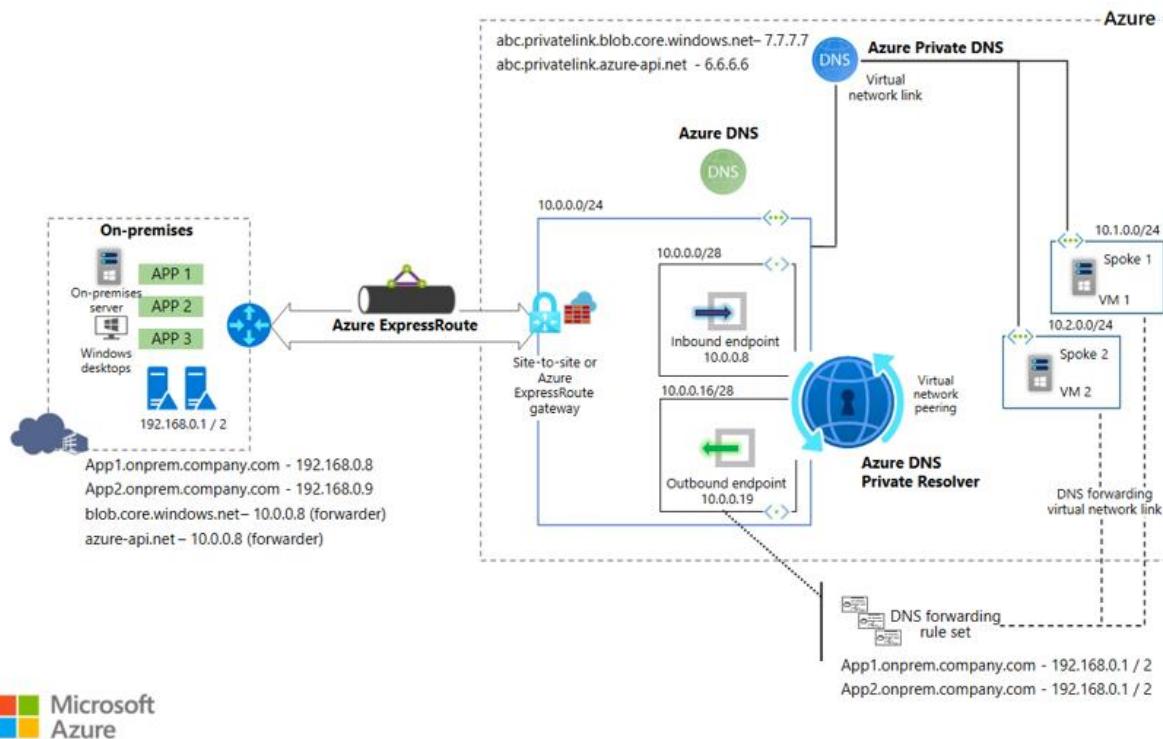
The following diagram illustrates the DNS resolution sequence from an on-premises network. The configuration uses a DNS forwarder deployed in Azure. The resolution is made by a private DNS zone linked to a virtual network:



Virtual network and on-premises workloads using Azure DNS Private Resolver

When you use DNS Private Resolver, you don't need a DNS forwarder VM, and Azure DNS is able to resolve on-premises domain names.

The following diagram uses DNS Private Resolver in a hub-spoke network topology. As a best practice, the Azure landing zone design pattern recommends using this type of topology. A hybrid network connection is established by using Azure ExpressRoute and Azure Firewall. This setup provides a secure hybrid network. DNS Private Resolver is deployed in the hub network.



- Review the [DNS Private Resolver solution components](#)
- Review the [traffic flow for an on-premises DNS query](#)
- Review the [traffic flow for a VM DNS query](#)
- Review the [traffic flow for a VM DNS query via DNS Private Resolver](#)
- Review the [traffic flow for a VM DNS query via an on-premises DNS server](#)

Exercise: Restrict network access to PaaS resources with virtual network service endpoints using the Azure portal

Completed 100 XP

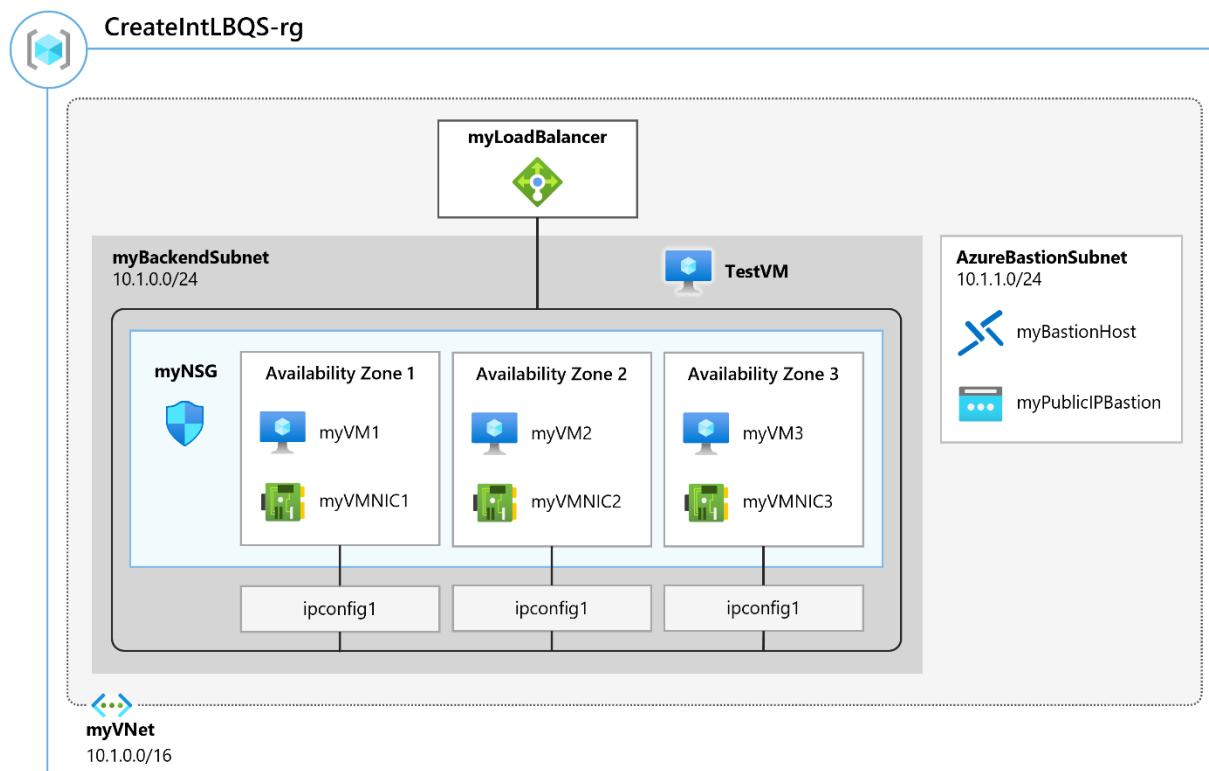
- 20 minutes

Lab scenario

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address.

space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network.

Architecture diagram



Objectives

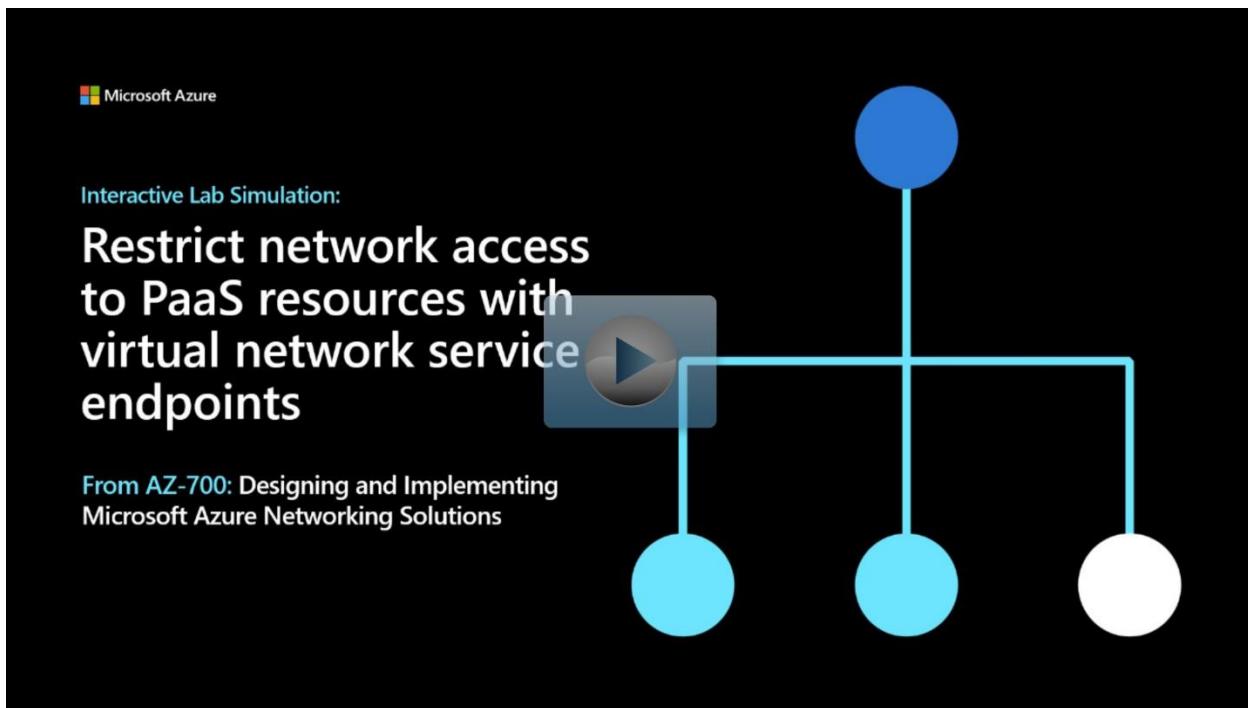
- **Task 1:** Create a virtual network
- **Task 2:** Enable a service endpoint
- **Task 3:** Restrict network access for a subnet
- **Task 4:** Add additional outbound rules
- **Task 5:** Allow access for RDP connections
- **Task 6:** Restrict network access to a resource
- **Task 7:** Create a file share in the storage account
- **Task 8:** Restrict network access to a subnet
- **Task 9:** Create virtual machines
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 10:** Confirm access to storage account

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Restrict%20network%20access%20to%20PaaS%20resources%20with%20virtual%20network%20service%20endpoints>

Exercise: Create an Azure private endpoint using Azure PowerShell

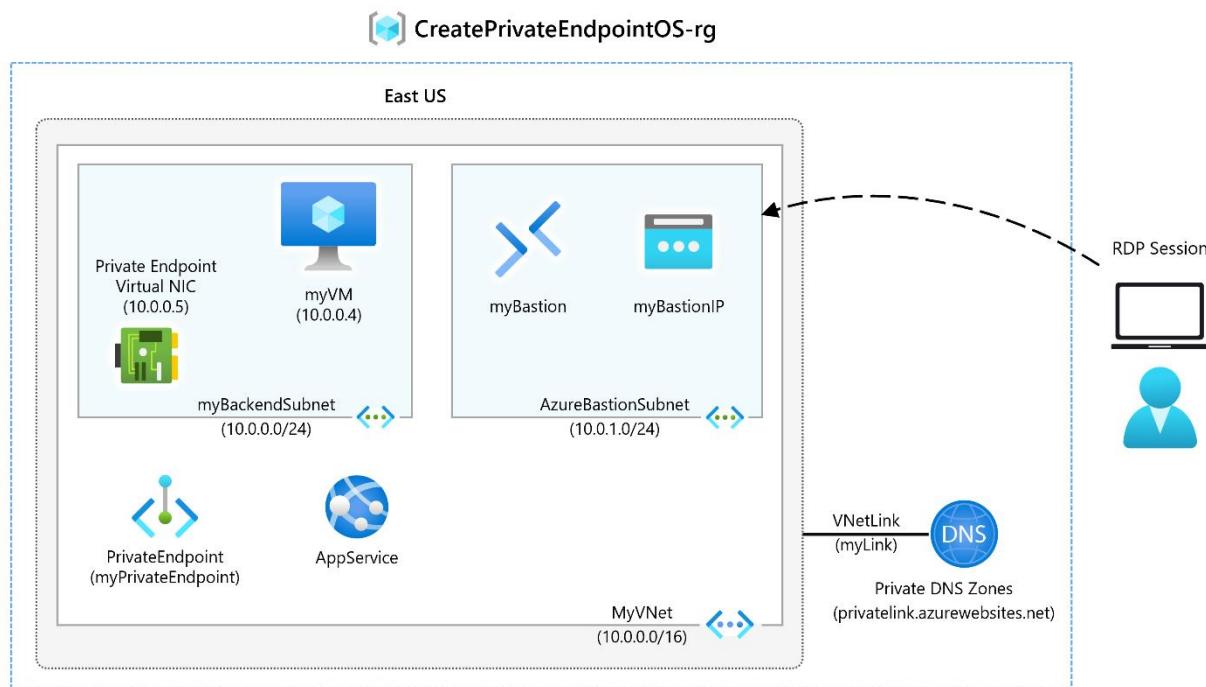
Completed 100 XP

- 8 minutes

Lab scenario

In this lab, you'll create a Private Endpoint for an Azure web app and deploy a virtual machine to test the private connection.

Architecture diagram



Objectives

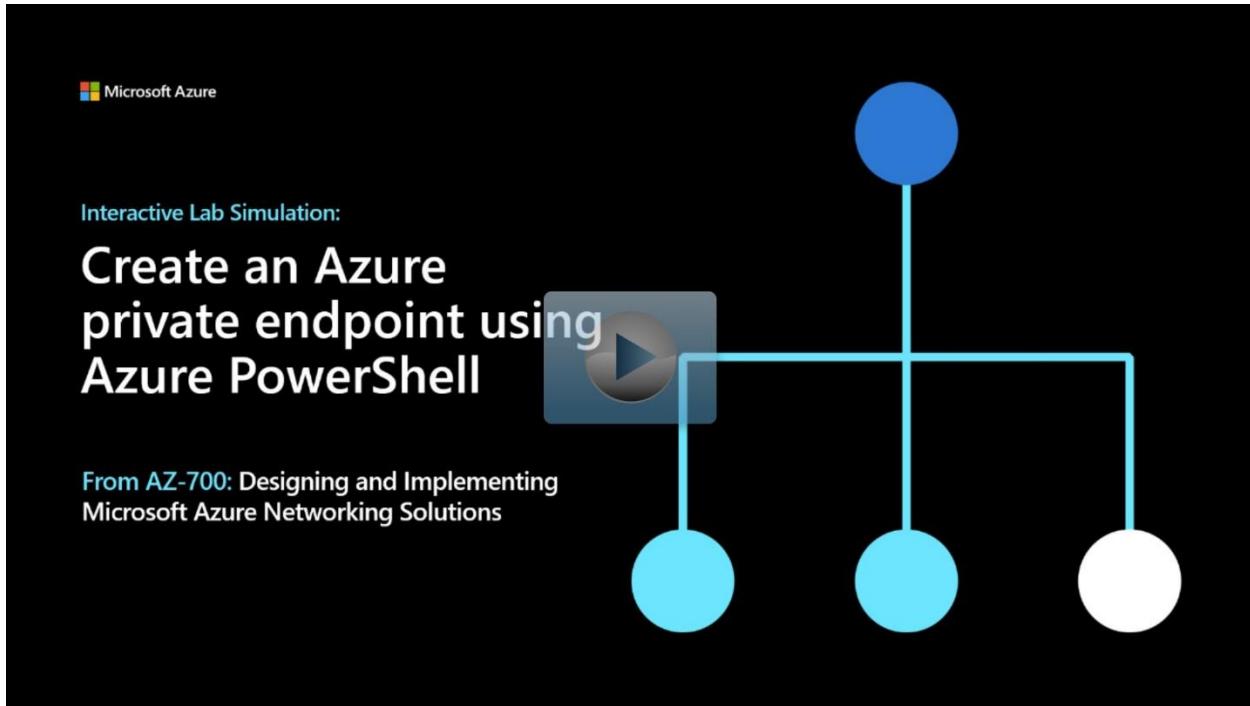
- **Task 1:** Create a resource group and deploy the Prerequisite web app
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 2:** Create a virtual network and bastion host
- **Task 3:** Create a test virtual machine
- **Task 4:** Create a private endpoint
- **Task 5:** Configure the private DNS zone
- **Task 6:** Test connectivity across the private endpoint

Note

Click on the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-%20Create%20an%20Azure%20private%20endpoint%20using%20Azure%20PowerShell>

Summary

Completed 100 XP

- 1 minute

Now that you have reviewed this module, you should be able to:

- Plan, create, configure, and create access to Private Endpoints
- Integrate a Private Link with DNS and on-premises clients
- Create, configure, and provide access to Service Endpoints

Design and implement network monitoring

Monitor your networks using Azure monitor

200 XP

- 19 minutes

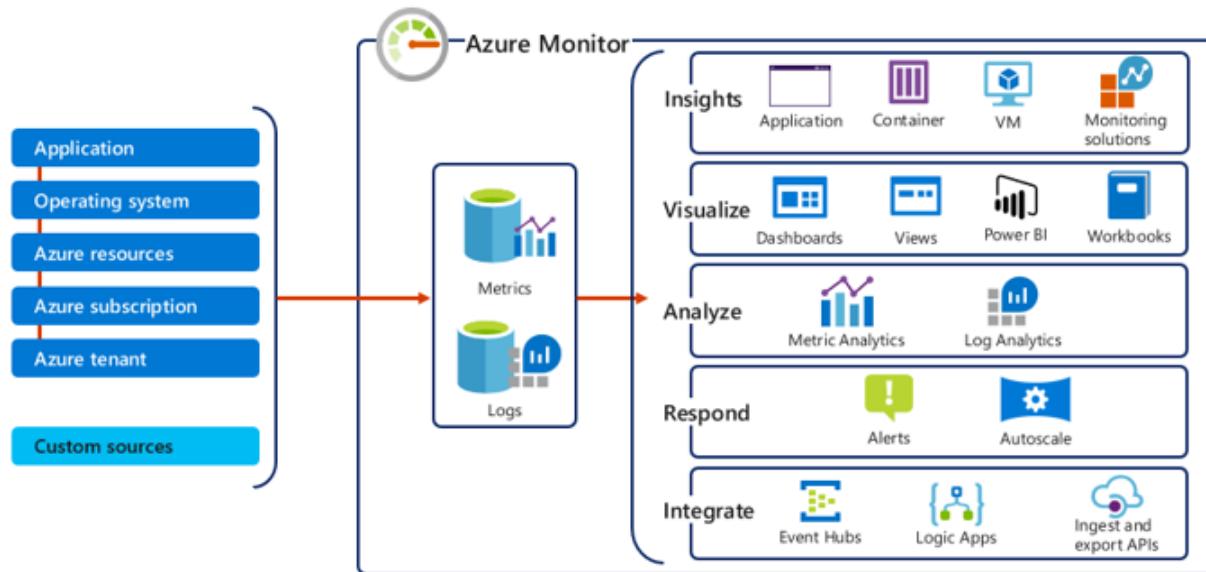
What is Azure Monitor?

Azure Monitor helps you maximize the availability and performance of your applications and services. It delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. This information helps you understand how your applications are performing and proactively identify issues affecting them and the resources they depend on.

Just a few examples of what you can do with Azure Monitor include:

- Detect and diagnose issues across applications and dependencies with Application Insights.
- Correlate infrastructure issues with VM insights and Container insights.
- Drill into your monitoring data with Log Analytics for troubleshooting and deep diagnostics.
- Support operations at scale with smart alerts and automated actions.
- Create visualizations with Azure dashboards and workbooks.
- Collect data from monitored resources using Azure Monitor Metrics.

The diagram below offers a high-level view of Azure Monitor. At the center of the diagram are the data stores for metrics and logs, which are the two fundamental types of data used by Azure Monitor. On the left are the sources of monitoring data that populate these data stores. On the right are the different functions that Azure Monitor performs with this collected data. This includes such actions as analysis, alerting, and streaming to external systems.



Monitor data types in Azure Monitor

The data collected by Azure Monitor fits into one of two fundamental types:

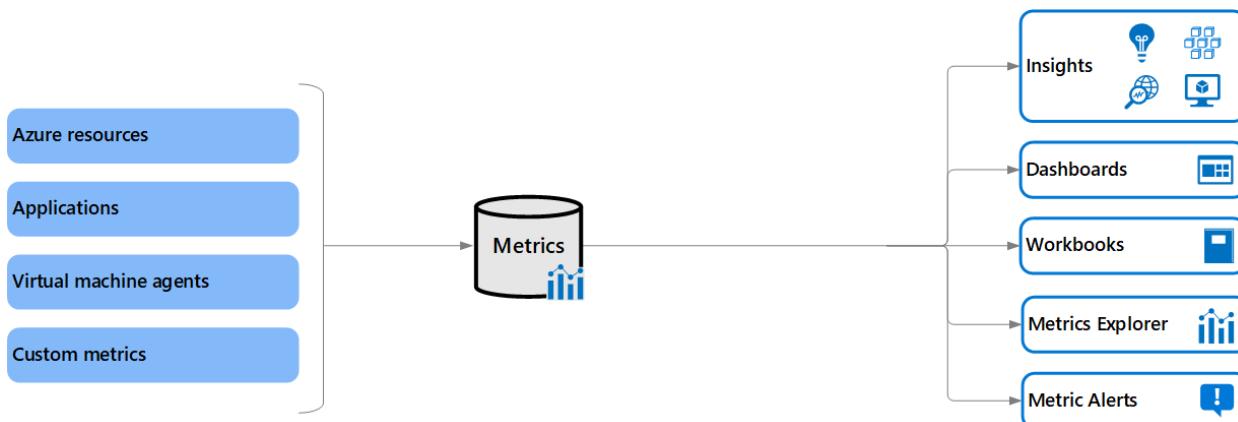
- **Metrics** - Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios.
- **Logs** - Logs contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

Azure Monitor metrics

Azure Monitor Metrics is a feature of Azure Monitor that collects numeric data from monitored resources into a time series database. Metrics are numerical values that are collected at regular intervals and describe some aspect of a system at a particular time. Metrics in Azure Monitor are lightweight and capable of supporting near real-time scenarios making them particularly useful for alerting and fast detection of issues. You can analyze them interactively with metrics explorer, be proactively notified with an alert when a value crosses a threshold or visualize them in a workbook or dashboard.

The table below provides a summary of the various types of tasks you can perform by utilizing metrics in Azure Monitor:

Task	Description
Analyze	Use metrics explorer to analyze collected metrics on a chart and compare metrics from different resources.
Alert	Configure a metric alert rule that sends a notification or takes automated action when the metric value crosses a threshold.
Visualize	Pin a chart from metrics explorer to an Azure dashboard. Create a workbook to combine with multiple sets of data in an interactive report. Export the results of a query to Grafana to leverage its dashboarding and combine with other data sources.
Automate	Use Autoscale to increase or decrease resources based on a metric value crossing a threshold.
Retrieve	Access metric values from a command line using PowerShell cmdlets. Access metric values from custom application using REST API. Access metric values from a command line using CLI.
Export	Route Metrics to Logs to analyze data in Azure Monitor Metrics together with data in Azure Monitor Logs and to store metric values for longer than 93 days Stream Metrics to an event hub to route them to external systems.
Archive	Archive the performance or health history of your resource for compliance, auditing, or offline reporting purposes.



Azure Monitor metrics sources

There are three fundamental sources of metrics collected by Azure Monitor. Once these metrics are collected in the Azure Monitor metric database, they can be evaluated together regardless of their source.

- **Azure resources** - Platform metrics are created by Azure resources and give you visibility into their health and performance. Each type of resource creates a distinct set of metrics without any configuration required. Platform metrics are collected

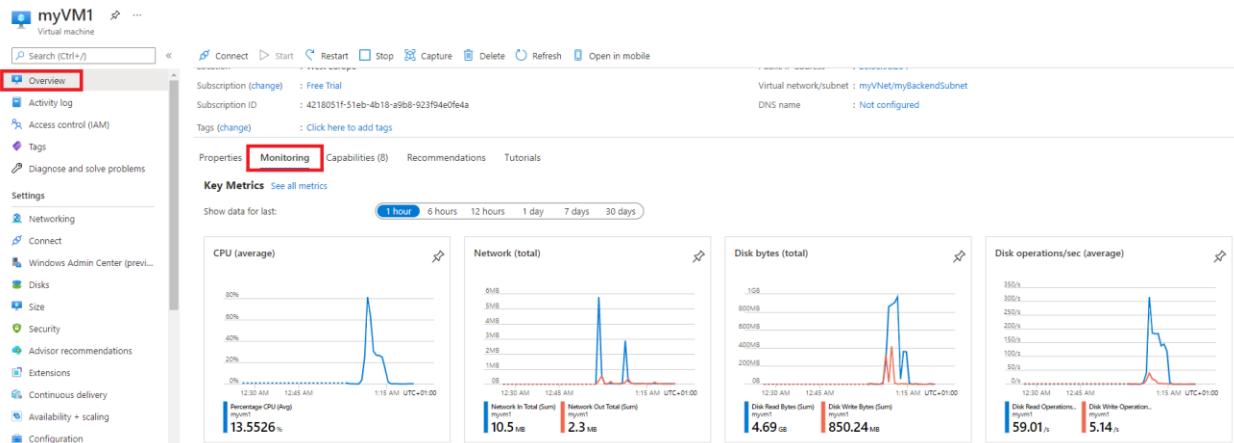
from Azure resources at one-minute frequency unless specified otherwise in the metric's definition.

- **Applications** - Metrics are created by Application Insights for your monitored applications and help you detect performance issues and track trends in how your application is being used. This includes such values as Server response time and Browser exceptions.
- **Virtual machine agents** - Metrics are collected from the guest operating system of a virtual machine. Enable guest OS metrics for Windows virtual machines with Windows Diagnostic Extension (WAD) and for Linux virtual machines with InfluxData Telegraf Agent.
- **Custom metrics** - You can define metrics in addition to the standard metrics that are automatically available. You can define custom metrics in your application that is monitored by Application Insights or create custom metrics for an Azure service using the custom metrics API.

Metrics Explorer

For several of your resources in Azure, you will see the data collected by Azure Monitor illustrated directly in the Azure portal on the **Monitoring** tab of a resource's **Overview** page.

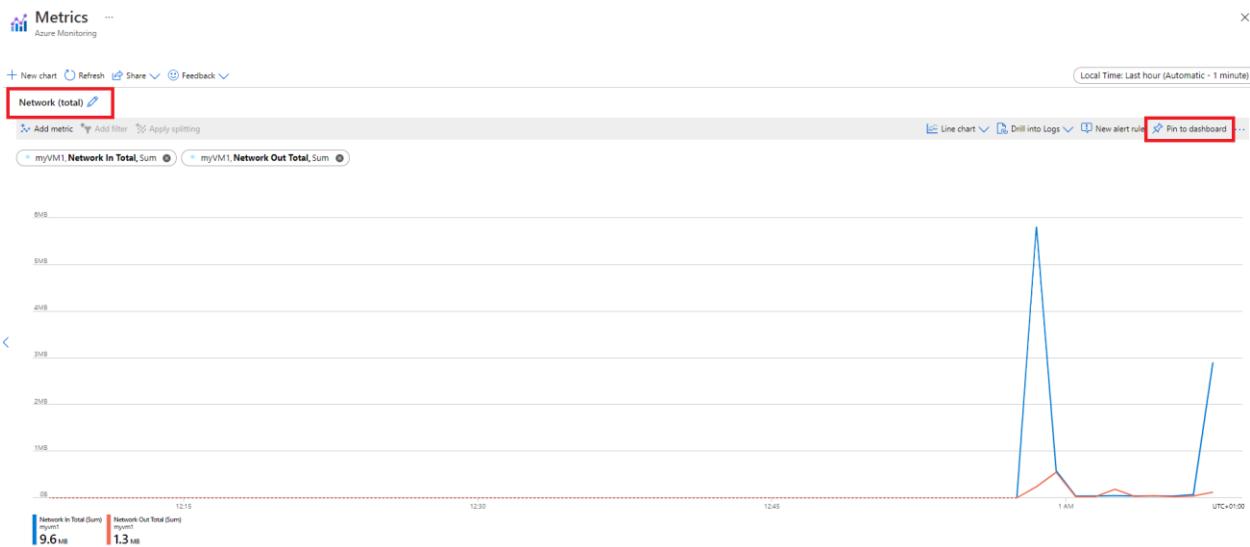
In the screenshot below for example, you can see the Monitoring tab from the Overview page of a virtual machine.



Note the various charts displaying several key performance metrics for system components such as **CPU**, **Network**, and **Disk**.

You can click on these graphs to open the data in **Metrics Explorer** in the Azure portal, which allows you to interactively analyze the data in your metric database and chart the values of multiple metrics over time. You can also pin the charts to a dashboard to view

them with other visualizations later. You can also retrieve metrics by using the Azure monitoring REST API.



The data collected by Azure Monitor Metrics is stored in a time-series database which is optimized for analyzing time-stamped data. Each set of metric values is a time series with the following properties:

- The time the value was collected
- The resource the value is associated with
- A namespace that acts like a category for the metric
- A metric name
- The value itself

Some metrics may have multiple dimensions, and custom metrics can have up to 10 dimensions.

Access Metrics in the Azure portal

You can access metrics from the **Metrics** option in the Azure Monitor menu.

You can also access metrics from the **Metrics** menu of most other services and resources in the Azure portal. The screenshot below for example, displays the **Metrics** page for a virtual network resource.

Create metric charts with metrics explorer

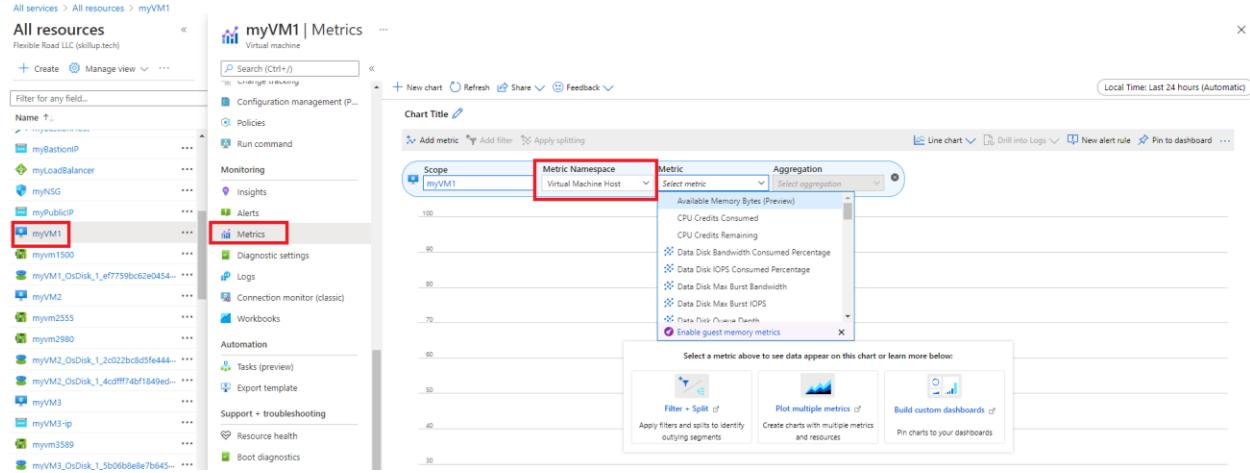
Azure Monitor **Metrics Explorer** is a component of the Microsoft Azure portal that allows plotting charts, visually correlating trends, and investigating spikes and dips in metrics' values. Use the metrics explorer to investigate the health and utilization of your resources.

Start in the following order:

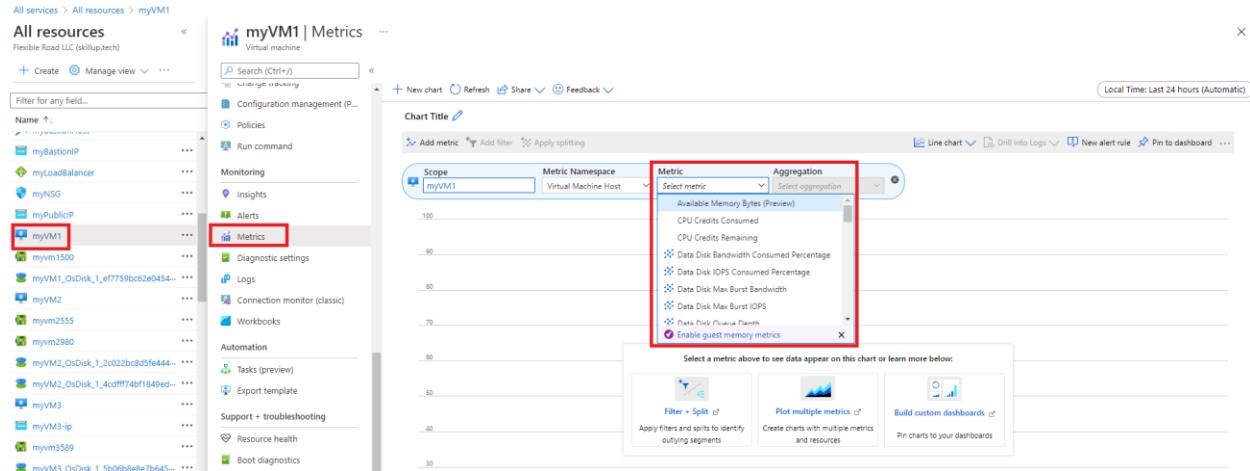
1. Pick a resource and a metric and you see a basic chart. Then select a time range that is relevant for your investigation.
2. Try applying dimension filters and splitting. The filters and splitting allow you to analyze which segments of the metric contribute to the overall metric value and identify possible outliers.
3. Use advanced settings to customize the chart before pinning to dashboards. Configure alerts to receive notifications when the metric value exceeds or drops below a threshold.
4. To create a metric chart, from your resource, resource group, subscription, or Azure Monitor view, open the **Metrics** tab and follow these steps:
5. Click on the "Select a scope" button to open the resource scope picker. This will allow you to select the resource(s) you want to see metrics for. If you opened metrics explorer from the resource's menu, the resource should already be populated.

The screenshot shows the Azure Metrics Explorer interface. On the left, there is a sidebar titled 'All resources' listing various Azure resources. One item, 'myVM1', is highlighted with a red box. On the right, the main pane is titled 'myVM1 | Metrics' and shows a chart area. A dropdown menu is open over the chart, also with 'myVM1' highlighted by a red box. The dropdown lists several metrics available for this scope, such as CPU Credits Consumed, CPU Credits Remaining, Data Disk Bandwidth Consumed Percentage, Data Disk IOPS Consumed Percentage, Data Disk Max Burst Bandwidth, Data Disk Max Burst IOPS, Data Disk Queue Depth, and Enable guest memory metrics. At the bottom of the dropdown, there are three buttons: 'Filter + Split', 'Plot multiple metrics', and 'Build custom dashboards'.

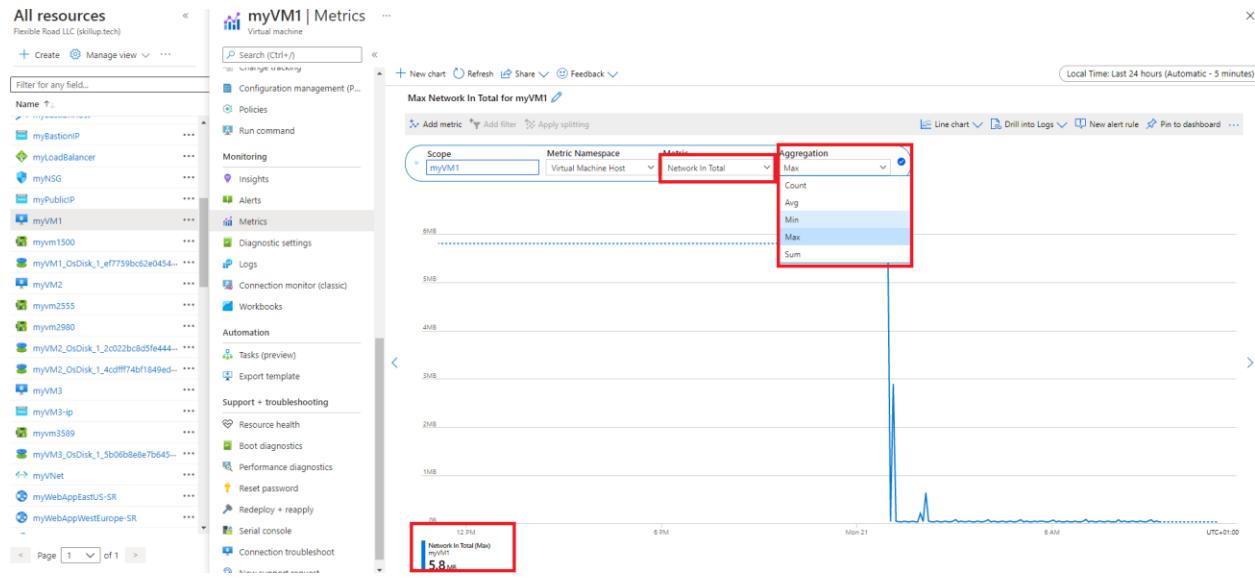
6. For some resources, you must pick a namespace. The namespace is just a way to organize metrics so that you can easily find them. For example, storage accounts have separate namespaces for storing Files, Tables, Blobs, and Queues metrics. Many resource types only have one namespace.



7. Select a metric from the list of available metrics. This list will vary depending on what resource and scope you select.



8. Optionally, you can change the metric aggregation. For example, you might want your chart to show minimum, maximum, or average values of the metric.



Monitor network resources with Azure Monitor Network Insights

You can use the **Insights>Networks** section in **Azure Monitor** to obtain a broad view of health and metrics for all your deployed network resources, without requiring any configuration. It also provides access to network monitoring features such as Connection Monitor, flow logging for network security groups (NSG) flow logs, and Traffic Analytics, and it provides other network diagnostic features.

Azure Monitor Network Insights is structured around these key components of monitoring:

- Network health and metrics
- Connectivity
- Traffic
- Diagnostic Toolkit

The screenshot shows the Azure Monitor Networks interface. On the left, there's a sidebar with various monitoring services like Overview, Activity log, Alerts, Metrics, Logs, Service Health, Workbooks, and Networks (which is selected). The main area has tabs for Network health, Connectivity, and Traffic, with Network health being the active tab (highlighted with a red box). There's a search bar at the top, followed by filters for Subscription (All), Resource Group (All), Type (All), and Sort By (Sort by name A-Z). Below that, it says 'No resources found. Select more subscriptions'. It includes a legend for resource status (Available, Degraded, Unavailable, Unknown, Health not supported) and an alert section for 'Alert - All resource type' with 0 total alerts, 0 smart groups, and 0 total alert rules. The alert time interval is set to 'Last 24 hours'. At the bottom, there's a table for alert severity with 0 entries.

Severity	Total alerts	New	Acknowled...	Closed
Sev 0	0	0	0	0
Sev 1	0	0	0	0
Sev 2	0	0	0	0
Sev 3	0	0	0	0

Network health and metrics

The **Network health** tab of Azure Monitor Network Insights offers a simple method for visualizing an inventory of your networking resources, together with resource health and alerts. It is divided into four key functionality areas: search and filtering, resource health and metrics, alerts, and dependency view.

Search and filtering

You can customize the resource health and alerts view by using filters such as **Subscription**, **Resource Group**, and **Type**.

You can use the search box to search for network resources and their associated resources. For example, a public IP is associated with an application gateway, so a search for the public IP's DNS name would return both the public IP and the associated application gateway.

The screenshot shows the Azure Monitor Networks - Network health page. At the top, there are navigation links: Home > Monitor, Refresh, Feedback, Useful Links, Diagnostic Toolkit, and a Microsoft logo. Below these are tabs: Network health (highlighted with a red box), Connectivity, and Traffic. A search bar and filter options (Subscription == 5 Selected, Resource Group == All, Type == All, Sort By == Sort by name A-Z) are present. A note says 'Fetched metrics data'. Under 'Public IPs', there are filters for Show health (Available, Degraded, Unavailable, Unknown, Health not supported) and a 'Show all resources' button. Three resource tiles are displayed: 'Public IPs(551)' with 551 items, 'Application gateways(7)' with 7 items, and 'Azure firewalls(10)' with 10 items. Each tile shows its count and resource health status. To the right, an 'Alert - Public IPs' section shows 0 total alerts, 0 smart groups, and 0 total alert rules, with a note 'Since 11/17/20, 02:17 PM'. Below the tiles, there are buttons for Select columns, Time interval (Last 24 hours), Group by (Top 100 - No grouping), and Load more (100). A detailed table follows, showing metrics for two specific resources: 'webserver-ip' and 'corporatepublicip'. The table includes columns for Name, Health, Alert, SKU, Net, West, TCP Packets Dropped DDoS, UDP Packets Dropped DDoS, TCP Bytes Dropped DDoS, and UDP Bytes Dropped DDoS.

Network resource health and metrics

You can use the health and metrics information to get an overview of the health status of your various network resources.

In the example screenshot below, each tile represents a particular type of network resource. The tile displays the number of instances of that resource type that are deployed across all your selected subscriptions. It also displays the health status of the resource. Here you can see that there are 19 **Load balancers** deployed, 17 of which are healthy, 1 is degraded, and 1 is unavailable.

The screenshot shows the Azure Monitor Networks - Network Health page. At the top, there are navigation links: Home > Monitor, Refresh, Feedback, Useful Links, Diagnostic Toolkit, and a Microsoft logo. Below these are tabs: Network Health (highlighted with a red box), Connectivity, and Traffic. A search bar and filter options (Subscription == All, Resource Group == All, Type == All, Sort By == Sort by name A-Z) are present. A note says 'Fetched metrics data'. Under 'Network Health', there are filters for Show Health (Available, Degraded, Unavailable, Unknown, Health not supported) and a 'Show all resources' button. Five resource tiles are displayed: 'Application gateways(49)', 'ER and VPN connections(105)', 'Load balancers(19)' (highlighted with a red box), 'Traffic manager profiles(30)', and 'Virtual network gateways(118)'. Each tile shows its count and resource health status. To the right, an 'Alert - All resource type' section shows 0 total alerts, 0 smart groups, and 0 total alert rules, with a note 'Since 11/4/19, 02:48 PM'. Below the tiles, there are buttons for Select columns, Time interval (Last 24 hours), Group by (Top 100 - No grouping), and Load more (100).

If you select one of the tiles, you get a view of the metrics for that network resource. In the example screenshot below, you can see the metrics for the **ER and VPN connections** resource.

Search direct and dependent reso... Subscription == All Resource Group == All Type == All Sort By == Sort by name A-Z

Fetched metrics data

Network Health/ER and VPN connections Show all resources

Show Health Available (green) Degraded (yellow) Unavailable (red) Unknown (grey) Health not supported (light grey)

ER and VPN conne... (105) Resource health 103 2

Application gateways(49) Resource health 49

Load balancers(19) Resource health 17 1 1

Alert time interval * Last 24 hours + New Alert

Alert - ER and VPN connections

Total alerts	Smart groups	Total alert rules
0	0	0

Since 11/4/19, 02:48 PM

Time interval * Last 24 hours Group by Top 100 - No grouping Load more 100

Search to filter items...

Name	Health	Alert	Location	Bits In Per Second	Bits Out Per Second	...
vnet1vpn1tovpn2vpn1		0	N West Central US			...
vent2vp1toent1vpn1		0	N West Central US			...

You can select any item in this grid view. For example, you could select the icon in the **Health** column to get resource health for that connection, or select the value in the **Alert** column to go to the alerts and metrics page for the connection.

Alerts

The **Alert** box on the right side of the page provides a view of all alerts generated for the selected resources across all your subscriptions. If there is a value for the alerts on an item, simply select the alert count for that item to go to a detailed alerts page for it.

Dependency view

Dependency view helps you visualize how a resource is configured. Dependency view is currently available for **Azure Application Gateway**, **Azure Virtual WAN**, and **Azure Load Balancer**. For example, for Application Gateway, you can access dependency view by selecting the Application Gateway resource name in the metrics grid view. You can do the same thing for Virtual WAN and Load Balancer.

The screenshot shows the Network Health section of the Azure Monitor interface. At the top, there are three resource health tiles: Application gateways (1), Network interfaces (8), and Network security groups (7). Below these are alert configuration options for Application gateways, including an alert time interval set to 'Last 24 hours' and a '+ New Alert' button. A summary table shows 0 total alerts, 0 smart groups, and 0 total alert rules, with a timestamp of 'Since 11/5/19, 09:06 AM'. The main content area displays a table of items loaded, with 'Last 24 hours' as the time interval, grouped by 'Top 100 - No grouping', and a limit of 100 items. A search bar and a table header with columns like Name, Health, Alert, Location, Throughput, Response Status, and Failed Requests are shown. A specific row for 'appgwv2' is selected, showing its status as 'Reachable' (green), 0 alerts, located in 'S East US 2', with throughput and response status graphs, and failed requests data.

Connectivity

The **Connectivity** tab of Azure Monitor Network Insights provides an easy way to visualize all tests configured via Connection Monitor and Connection Monitor (classic) for the selected set of subscriptions.

Tests are grouped by **Sources** and **Destinations** tiles and display the reachability status for each test. Reachable settings provide easy access to configurations for your reachability criteria, based on **Checks failed(%)** and **RTT(ms)**.

The screenshot shows the Connectivity page in Azure Monitor. At the top, there are navigation links for Home, Monitor, and a search bar. Below the search bar are filter buttons for Subscription (5 Selected), Resource Group (All), Source type (All), Destination type (All), and Sort By (Sort by name A-Z). A note says 'Fetched both alerts and health for Connection Monitors'. The main area has two sections: 'Sources' and 'Destinations'. The 'Sources' section contains tiles for Virtual machines (2440), Virtual networks (51), and Subnet (8). The 'Destinations' section contains tiles for IP (215), Virtual machines (282), Application gateways (1), Load balancers (10), Virtual networks (27), and Subnet (3). A red box highlights a filter bar at the top of the sources section with checkboxes for 'Reachable' (blue), 'RTT(ms) : 0 - 300 , Checks failed(%) : 0 - 100' (light blue), 'Unreachable' (red), and 'Stopped' (grey). To the right, there is an 'Alert - All resource type' summary with 2 total alerts, 1 smart group, and 1 total alert rule, with a timestamp of 'Since 11/17/20, 02:17 PM'. Below this is a table of alert severity counts: Sev 0 (0), Sev 1 (0), Sev 2 (0), Sev 3 (2), and Sev 4 (0).

After you set the values, the status for each test updates based on the selection criteria.

Home > Monitor

Monitor | Networks

Microsoft

Refresh Feedback Useful Links Diagnostic Toolkit

Network health Connectivity Traffic

Search direct and dependent reso... Subscription == 5 Selected Resource Group == All Source type == All Destination type == All Sort By == Sort by name A-Z

Fetched both alerts and health for Connection Monitors

Reachable Unreachable Stopped

+ Connection Monitor

Alert time interval * (Last 24 hours) + New Alert

Criteria

RTT(ms) : 0 - 300 , Checks failed(%): 0 - 100

Sources

- Virtual machine: Resource health (1008)
- On-premises: Resource health (25)

Destinations

- IP(215): Resource health (35 180)
- Virtual machines(282): Resource health (64 218)
- Application gateways(1): Resource health (1)
- Load balancers(10): Resource health (10)
- Virtual networks(27): Resource health (16 11)
- Subnet(3): Resource health (2 1)

Alert - All resource type

Total alerts	Smart groups	Total alert rules		
2	1	1		
Since 11/17/20, 02:17 PM				
Severity	Total alerts	New	Acknowled...	Closed
Sev 0	0	0	0	0
Sev 1	0	0	0	0
Sev 2	0	0	0	0
Sev 3	2	2	0	0
Sev 4	0	0	0	0

From here, you can then select any source or destination tile to open it up in metric view. In the example screenshot below, the metrics for the **Destinations>Virtual machines** tile are being displayed.

Home > Monitor

Monitor | Networks

Microsoft

Refresh Feedback Useful Links Diagnostic Toolkit

Sources

- Virtual machines(2440): Resource health (1008 3 1429)
- Virtual networks(51): Resource health (26 25)
- Subnet(8): Resource health (2 6)

Destinations

- Virtual machines(282): Resource health (64 218)
- IP(215): Resource health (35 180)
- Application gateways(1): Resource health (1)

Alert - Virtual machines

Total alerts	Smart groups	Total alert rules
0	0	0
Since 11/17/20, 02:17 PM		

100 of 282 items loaded Select columns Time interval * (Last 24 hours) Group by Load more

Search to filter items...

Source	Destination	Connection Monit...	Reacha...	Test...	Pro...	Port	Alert	RTT...	Che...	Checks Failed Percent	Round-Trip Time (ms)
vakarana-EU52EUAP(WIN appvm3	AppGwTopoTest3	✓	-	HTTP	80	0	0	100			...
vakarana-EU52EUAP(WIN appvm3	AppGwTopoTest3	✓	-	TCP	80	0	0	100			...
ubuntu1804bugbash	win2016bugBash	BugBashEndpoint	✓	-	ICMP	-	0	0	100		...
ubuntu1804bugbash	win2016bugBash	BugBashEndpoint	✓	-	TCP	80	0	0	100		...
win2016bugBash	ubuntu1804bugbash	BugBashEndpoint	✓	-	ICMP	-	0	0	100		...
win2016bugBash	ubuntu1804bugbash	BugBashEndpoint	✓	-	TCP	80	0	0	100		...
VM01	WindowsDB-VM	CM1	✓	-	HTTP	80	0	0	100		...
VM01	WindowsDB-VM	CM1	✓	-	HTTP	80	0	0	100		...

Traffic

The **Traffic** tab of Azure Monitor Network Insights provides access to all NSGs configured for **NSG flow logs** and **Traffic Analytics** for the selected set of subscriptions, grouped by location. The search functionality provided on this tab enables you to identify the NSGs configured for the searched IP address. You can search for any IP address in your environment. The tiled regional view will display all NSGs along with the NSG flow logs and Traffic Analytics configuration status.

The screenshot shows the Azure Monitor Networks interface with the 'Traffic' tab selected. At the top, there are filters for 'Search IP Address or Address prefix', 'Subscription' (5 Selected), 'Resource Group' (All), 'Location' (All), and 'Sort By' (Resource count descending). A note says 'Fetched traffic data. Click on the Resource tiles below to view details'. Below this, there are three filter checkboxes: 'Flow Logs and Traffic Analytics Enabled' (selected), 'Flow Logs Enabled' (selected), and 'Flow Logs Not Configured' (selected). The main area displays a grid of nine region tiles, each showing NSG counts for different log types. To the right, there's a sidebar for 'Alert - Traffic Analytics' with an alert time interval set to 'Last 24 hours', a '+ New Alert' button, and a table showing total alerts (0), smart groups (0), and total alert rules (0) since November 17, 2020, at 02:17 PM. The table also includes columns for Severity, Total alerts, New, Acknowledged..., and Closed.

Severity	Total alerts	New	Acknowledged...	Closed
Sev 0	0	0	0	0
Sev 1	0	0	0	0
Sev 2	0	0	0	0
Sev 3	0	0	0	0
Sev 4	0	0	0	0

If you select any region tile, a grid view will appear which shows NSG flow logs and Traffic Analytics in a view that is simple to interpret and configure.

The screenshot shows the Azure Monitor Networks - Traffic page. At the top, there are tabs for Network health, Connectivity, and Traffic, with Traffic selected. Below the tabs are filters for Search IP Address or Address prefix, Subscription (5 Selected), Resource Group (All), Location (All), and Sort By (Resource count descending). A note says 'Fetched grid data' and 'East US 2'. There are checkboxes for Flow Logs and Traffic Analytics Enabled, Flow Logs Enabled, and Flow Logs Not Configured. On the right, there's an alert section for 'Alert - Traffic Analytics (East US 2)' with options for Last 24 hours and + New Alert. Below the alert is a summary of total alerts, smart groups, and total alert rules (all 0). The main area displays a grid of NSGs grouped by location: East US 2 (42), Southeast Asia (43), and West US 2 (41). Each row in the grid includes columns for NSG name, Flowlog Configuration Status (with icons for green checkmark, yellow warning, and red error), Alert (with a count of 0), Subscription, Storage Account, Traffic Analytics Configuration (with icons for green checkmark, yellow warning, and red error), and Traffic Analytics Workspace (with a list of workspaces like testworkspaceus2, dummyws, canaryws1, canaryws1). The 'Flowlog Configuration Status' and 'Alert' columns are highlighted with red boxes.

In this grid view you can select an icon in the **Flow log Configuration Status** column to edit the NSG flow log and Traffic Analytics configuration. Or you can select a value in the **Alert** column to go to the traffic alerts configured for that NSG, and you can navigate to the Traffic Analytics view by selecting the **Traffic Analytics Workspace**.

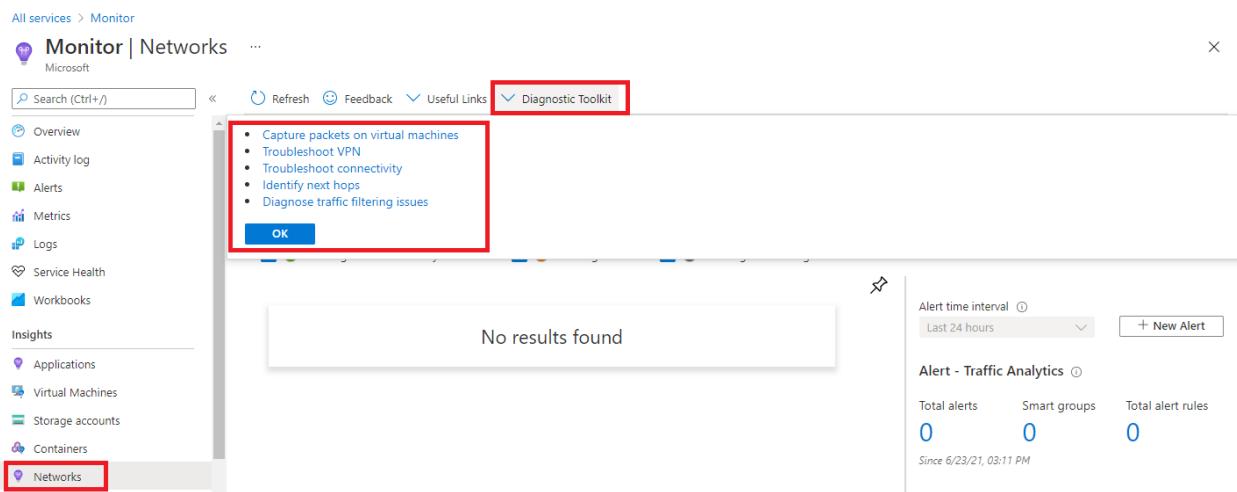
Diagnostic Toolkit

The Diagnostic Toolkit feature in Azure Monitor Network Insights provides access to all the diagnostic features available for troubleshooting your networks and their components.

The **Diagnostic Toolkit** drop-down list provides access to the following network monitoring features:

- Capture packets on virtual machines - opens the **Network Watcher packet capture** network diagnostic tool to enable you create capture sessions to track traffic to and from a virtual machine. Filters are provided for the capture session to ensure you capture only the traffic you want. Packet capture helps to diagnose network anomalies, both reactively, and proactively. Packet capture is a virtual machine extension that is remotely started through Network Watcher.
- Troubleshoot VPN - opens the **Network Watcher VPN Troubleshoot** tool to diagnose the health of a virtual network gateway or connection.

- Troubleshoot connectivity - opens the **Network Watcher Connection Troubleshoot** tool to check a direct TCP connection from a virtual machine (VM) to a VM, fully qualified domain name (FQDN), URI, or IPv4 address.
- Identify next hops - opens the **Network Watcher Next hop** network diagnostic tool to obtain the next hop type and IP address of a packet from a specific VM and NIC. Knowing the next hop can help you establish if traffic is being directed to the expected destination, or whether the traffic is being sent nowhere.
- Diagnose traffic filtering issues - opens the **Network Watcher IP flow verify** network diagnostic tool to verify if a packet is allowed or denied, to or from a virtual machine, based on 5-tuple information. The security group decision and the name of the rule that denied the packet is returned.



Exercise: monitor a load balancer resource using Azure monitor

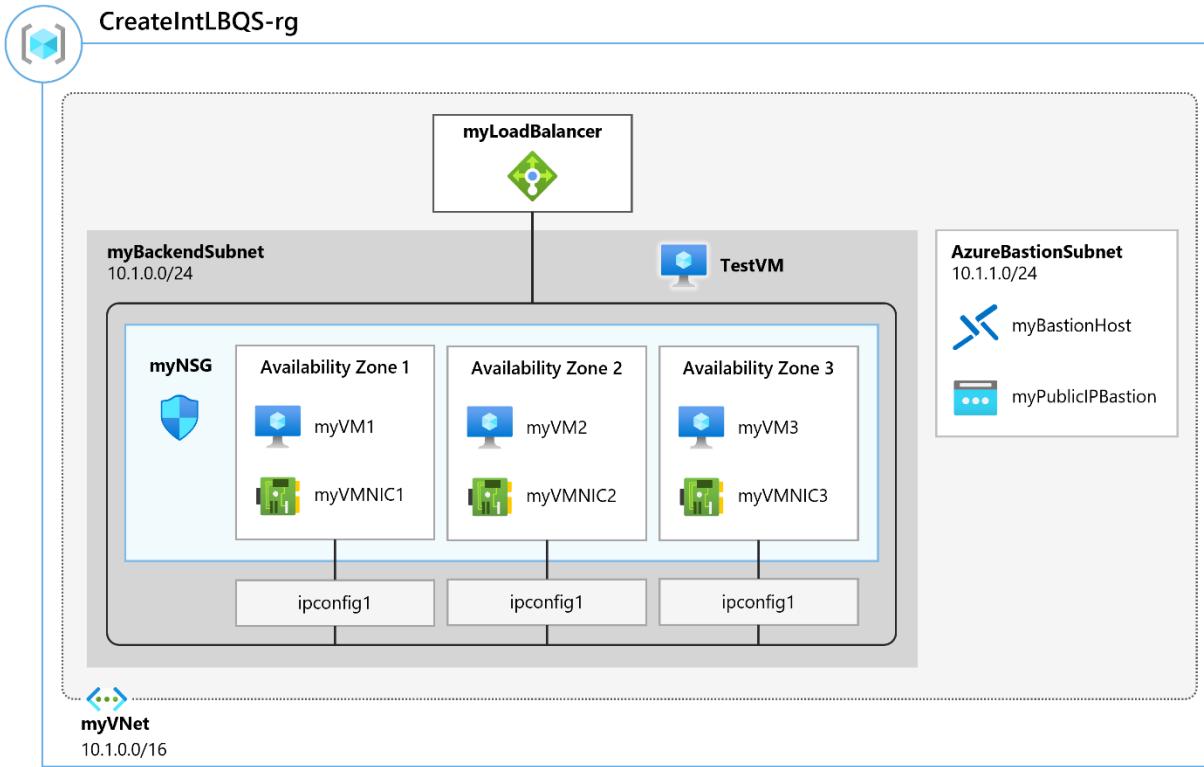
Completed 100 XP

- 3 minutes

Lab scenario

In this exercise, you create an internal load balancer for the fictional Contoso Ltd organization. Then you create a Log Analytics workspace, and use Azure Monitor Insights to view information about your internal load balancer. Finally, you configure the load balancer's diagnostic settings to send metrics to the Log Analytics workspace you created.

Architecture diagram



Objectives

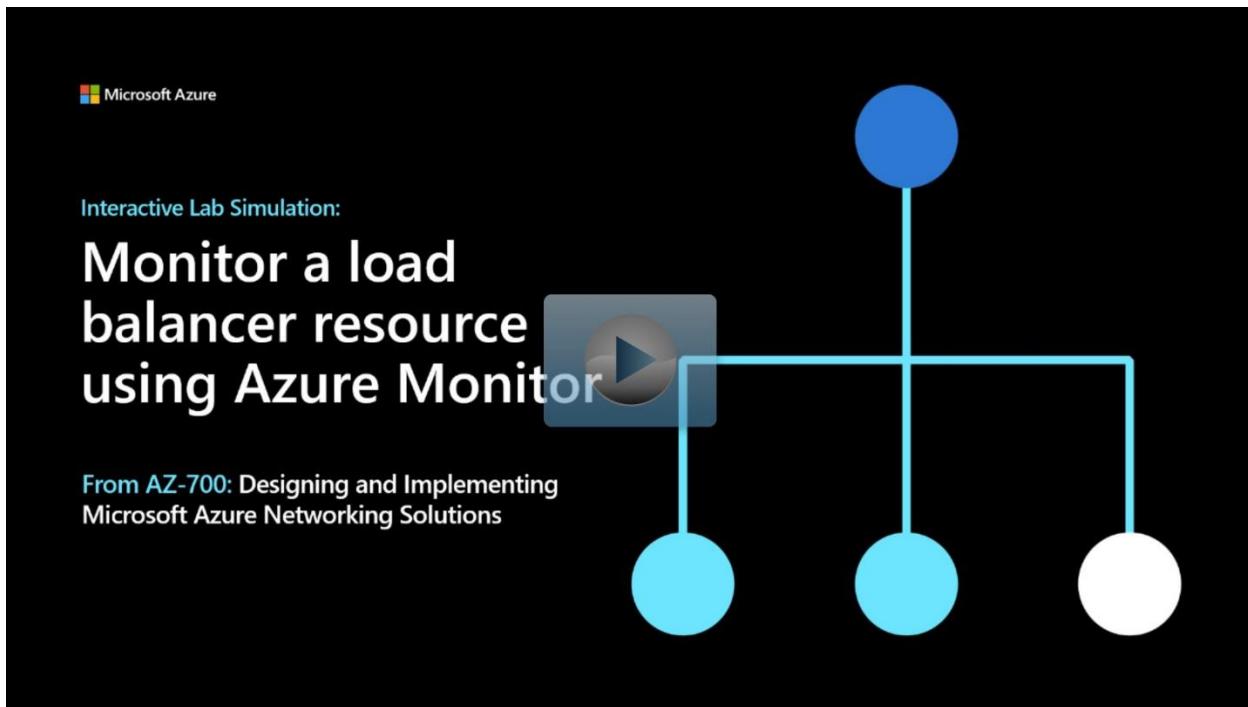
- **Task 1:** Create the virtual network
- **Task 2:** Create the load balancer
- **Task 3:** Create a backend pool
- **Task 4:** Create a health probe
- **Task 5:** Create a load balancer rule
- **Task 6:** Create backend servers
 - Use a template to create the virtual machines. You can review the [lab template](#).
 - Use Azure PowerShell to deploy the template.
- **Task 7:** Add VMs to the backend pool
- **Task 8:** Install IIS on the VMs
- **Task 9:** Test the load balancer
- **Task 10:** Create a Log Analytics Workspace
- **Task 11:** Use Functional Dependency View
- **Task 12:** View detailed metrics
- **Task 13:** View resource health
- **Task 14:** Configure diagnostic settings
- **Task 15:** Clean up resources

Note

Select the thumbnail image to start the lab simulation. When you're done, be sure to return to this page so you can continue learning.

Note

You may find slight differences between the interactive simulation and the hosted lab, but the core concepts and ideas being demonstrated are the same.



<https://mslabs.cloudguides.com/guides/AZ-700%20Lab%20Simulation%20-Monitor%20a%20load%20balancer%20resource%20using%20Azure%20Monitor>

Monitor your networks using Azure network watcher

200 XP

- 20 minutes

Azure Network Watcher

Azure Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level in, to, and from Azure. Scenario level monitoring enables you to diagnose problems at an end-to-end network level view. Network diagnostic and visualization tools available with Network Watcher help you understand, diagnose, and gain insights to your network in Azure. Network Watcher is enabled through the creation of a Network Watcher resource, which allows you to utilize Network Watcher capabilities. Network Watcher is designed to monitor and repair the network health of IaaS products which includes Virtual Machines, Virtual Networks, Application Gateways, and Load Balancers.

- **Automate remote network monitoring with packet capture.** Monitor and diagnose networking issues without logging in to your virtual machines (VMs) using Network Watcher. Trigger packet capture by setting alerts, and gain access to real-time performance information at the packet level. When you observe an issue, you can investigate in detail for better diagnoses.
- **Gain insight into your network traffic using flow logs. Build a deeper understanding of your network traffic pattern using Network Security Group flow logs.** Information provided by flow logs helps you gather data for compliance, auditing and monitoring your network security profile.
- **Diagnose VPN connectivity issues. Network Watcher provides you the ability to diagnose your most common VPN Gateway and Connections issues.** Allowing you, not only, to identify the issue but also to use the detailed logs created to help further investigate.

Network Topology: The topology capability enables you to generate a visual diagram of the resources in a virtual network, and the relationships between the resources.

Verify IP Flow: Quickly diagnose connectivity issues from or to the internet and from or to the on-premises environment. For example, confirming if a security rule is blocking ingress or egress traffic to or from a virtual machine. IP flow verify is ideal for making sure security rules are being correctly applied. When used for troubleshooting, if IP flow verify doesn't show a problem, you will need to explore other areas such as firewall restrictions.

Next Hop: To determine if traffic is being directed to the intended destination by showing the next hop. This will help determine if networking routing is correctly configured. Next hop also returns the route table associated with the next hop. If the route is defined as a user-defined route, that route is returned. Otherwise, next hop returns System Route. Depending on your situation the next hop could be Internet, Virtual Appliance, Virtual Network Gateway, VNet Local, VNet Peering, or None. None

lets you know that while there may be a valid system route to the destination, there is no next hop to route the traffic to the destination. When you create a virtual network, Azure creates several default outbound routes for network traffic. The outbound traffic from all resources, such as VMs, deployed in a virtual network, are routed based on Azure's default routes. You might override Azure's default routes or create additional routes.

Effective security rules: Network Security groups are associated at a subnet level or at a NIC level. When associated at a subnet level, it applies to all the VM instances in the subnet. Effective security rules view returns all the configured NSGs and rules that are associated at a NIC and subnet level for a virtual machine providing insight into the configuration. In addition, the effective security rules are returned for each of the NICs in a VM. Using Effective security rules view, you can assess a VM for network vulnerabilities such as open ports.

VPN Diagnostics: Troubleshoot gateways and connections. VPN Diagnostics returns a wealth of information. Summary information is available in the portal and more detailed information is provided in log files. The log files are stored in a storage account and include things like connection statistics, CPU and memory information, IKE security errors, packet drops, and buffers and events.

Packet Capture: Network Watcher variable packet capture allows you to create packet capture sessions to track traffic to and from a virtual machine. Packet capture helps to diagnose network anomalies both reactively and proactively. Other uses include gathering network statistics, gaining information on network intrusions, to debug client-server communications and much more.

Connection Troubleshoot: Azure Network Watcher Connection Troubleshoot is a more recent addition to the Network Watcher suite of networking tools and capabilities. Connection Troubleshoot enables you to troubleshoot network performance and connectivity issues in Azure.

NSG Flow Logs: NSG Flow Logs maps IP traffic through a network security group. These capabilities can be used in security compliance and auditing. You can define a prescriptive set of security rules as a model for security governance in your organization. A periodic compliance audit can be implemented in a programmatic way by comparing the prescriptive rules with the effective rules for each of the VMs in your network.

configure Network Watcher

When you create or update a virtual network in your subscription, Network Watcher will be enabled automatically in your Virtual Network's region. There is no impact to your resources or associated charge for automatically enabling Network Watcher.

To create a Network Watcher in the Azure portal:

1. Navigate to **All services > Networking > Network Watcher**.

The screenshot shows the Azure portal's 'All services' blade. On the left, there's a sidebar with categories like Overview, Categories (All), General, Compute, **Networking**, Storage, Web, Mobile, Containers, Databases, Analytics, Blockchain, AI + machine learning, Internet of things, Mixed reality, Integration, Identity, and Security. The 'Networking' category is selected and highlighted with a red box. In the main pane, under 'NETWORKING (38)', there are several service icons: Virtual networks, Load balancers, CDN profiles, Network Watcher (which is also highlighted with a red box and has a tooltip), Network interfaces, Public IP Prefixes, Route tables, Application security groups, Service endpoint policies, Web Application Firewall policies (WAF), Virtual WANs, DNS zones, Application gateways, and Front Doors. To the right of the main pane, there's a 'Network Watcher' service card with a star icon, a 'View' button, and a 'Useful links' section containing 'Overview', 'Get started', and 'Documentation'.

2. Right-click your subscription and choose **Enable network watcher in all regions**.

All services >

Network Watcher

Microsoft

Search (Ctrl+ /)

Overview

Monitoring

- Topology
- Connection monitor (classic)
- Connection monitor
- Network Performance Monitor

Network diagnostic tools

- IP flow verify
- NSG diagnostic
- Next hop
- Effective security rules
- VPN troubleshoot
- Packet capture
- Connection troubleshoot

Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level. [Learn more.](#)

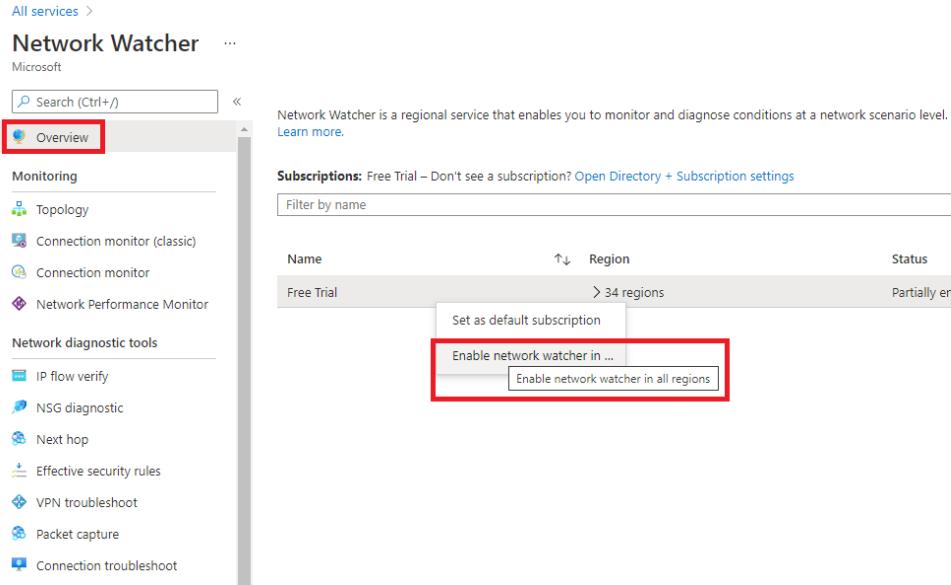
Subscriptions: Free Trial – Don't see a subscription? [Open Directory + Subscription settings](#)

Name	Region	Status	...
Free Trial	> 34 regions	Partially enabled	...

Set as default subscription

Enable network watcher in ...

Enable network watcher in all regions



3. Note that the status is now showing as **Enabled**.

All services >

Network Watcher

Microsoft

Search (Ctrl+ /)

Overview

Monitoring

- Topology
- Connection monitor (classic)
- Connection monitor
- Network Performance Monitor

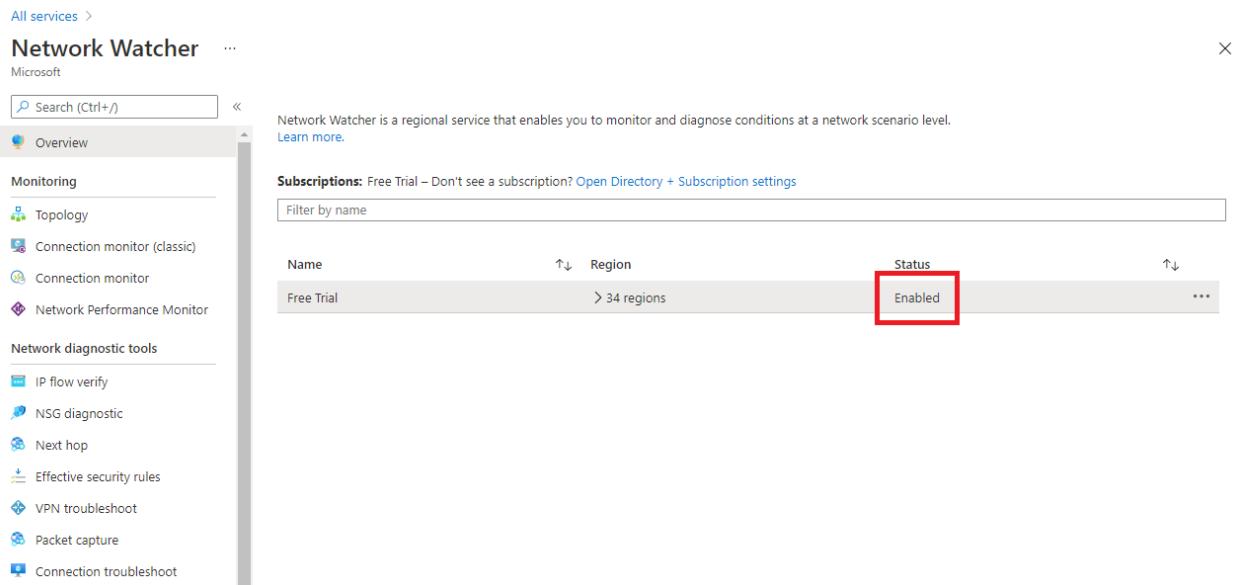
Network diagnostic tools

- IP flow verify
- NSG diagnostic
- Next hop
- Effective security rules
- VPN troubleshoot
- Packet capture
- Connection troubleshoot

Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level. [Learn more.](#)

Subscriptions: Free Trial – Don't see a subscription? [Open Directory + Subscription settings](#)

Name	Region	Status	...
Free Trial	> 34 regions	Enabled	...



4. If you expand the regions, you will see that all regions within this subscription are enabled.

Name	Region	Status
Free Trial	West US	Enabled
	East US	Enabled
	North Europe	Enabled
	West Europe	Enabled
	East Asia	Enabled
	Southeast Asia	Enabled
	North Central US	Enabled
	South Central US	Enabled
	Central US	Enabled
	East US 2	Enabled
	Japan East	Enabled

5. When you enable Network Watcher using the portal, the name of the Network Watcher instance is automatically set to **NetworkWatcher_region_name** where **region_name** corresponds to the Azure region where the instance is enabled. For example, a Network Watcher enabled in the West US region is named **NetworkWatcher_westus**.
6. The Network Watcher instance is automatically created in a resource group named **NetworkWatcherRG**. The resource group is created if it does not already exist.

Name	Subscription	Location
Contoso-RG-TM1	Free Trial	East US
Contoso-RG-TM2	Free Trial	West Europe
DefaultResourceGroup-EUS	Free Trial	East US
DefaultResourceGroup-WEU	Free Trial	West Europe
IntLB-RG	Free Trial	West US
NetworkWatcherRG	Free Trial	North Europe
ResourceMoverRG-uksouth-westus-eus2	Free Trial	East US 2

7. To disable a Network Watcher for a region in the Azure portal, expand the regions section, right click the name of the region you wish to disable the Network Watcher on, and click **Disable network watcher**.

Subscriptions: Free Trial – Don't see a subscription? [Open Directory + Subscription settings](#)

Name	Region	Status
Free Trial	34 regions	Enabled
	West US	Enabled
	East US	Enabled
	North Europe	Enabled
	West Europe	Enabled
	East Asia	Enabled
	Southeast Asia	Enabled
	North Central US	Enabled
	South Central US	Enabled
	Central US	Enabled

Configure NSG Flow Logs

Network security groups (NSG) allow or deny inbound or outbound traffic to a network interface in a VM.

NSG flow logs is a feature of Azure Network Watcher that allows you to log information about IP traffic flowing through an NSG. The NSG flow log capability allows you to log the source and destination IP address, port, protocol, and whether traffic was allowed or denied by an NSG. You can analyze logs using a variety of tools, such as Power BI and the Traffic Analytics feature in Azure Network Watcher.

Common use cases for NSG flow logs are:

- **Network Monitoring** - Identify unknown or undesired traffic. Monitor traffic levels and bandwidth consumption. Filter flow logs by IP and port to understand application behavior. Export Flow Logs to analytics and visualization tools of your choice to set up monitoring dashboards.
- **Usage monitoring and optimization** - Identify top talkers in your network. Combine with GeoIP data to identify cross-region traffic. Understand traffic growth for capacity forecasting. Use data to remove overtly restrictive traffic rules.
- **Compliance** - Use flow data to verify network isolation and compliance with enterprise access rules.
- **Network forensics and security analysis** - Analyze network flows from compromised IPs and network interfaces. Export flow logs to any SIEM or IDS tool of your choice.

You can enable NSG flow logs from any of the following:

- [Azure portal](#)
- [PowerShell](#)
- [Azure CLI](#)
- [REST](#)
- [Azure Resource Manager](#)

1. To configure the parameters of NSG flow logs in the Azure portal, navigate to the **NSG Flow Logs** section in **Network Watcher**.
2. Click the name of the NSG to bring up the **Settings** pane for the Flow log.

Home > Network Watcher >

Flow logs settings

...

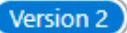
 Save  Discard

Flow logs

Status

 Off  On

Flow Logs version 

 Version 1  Version 2

Version 1 logs ingress and egress IP traffic flows for both allowed and denied traffic. Version 2 provides additional throughput information (bytes and packets) per flow.

[Learn more.](#)

contosostorageaccount

Select storage account

Retention (days) 

 5

Traffic Analytics

 Traffic Analytics provides rich analytics and visualization derived from NSG flow logs and other Azure resources' data. Drill through geo-map, easily figure out traffic hotspots and get insights into optimization possibilities.

[Learn about all features](#)

To use this feature, choose an Log Analytics workspace. To minimize data egress costs, we recommend that you choose a workspace in the same region your flow logs storage account is located. Network Performance Monitor solution will be installed on the workspace.

We also advise that you use the same workspace for all NSGs as much as possible. Additional metadata is added to your flow logs data, to provide enhanced analytics.

Traffic Analytics status

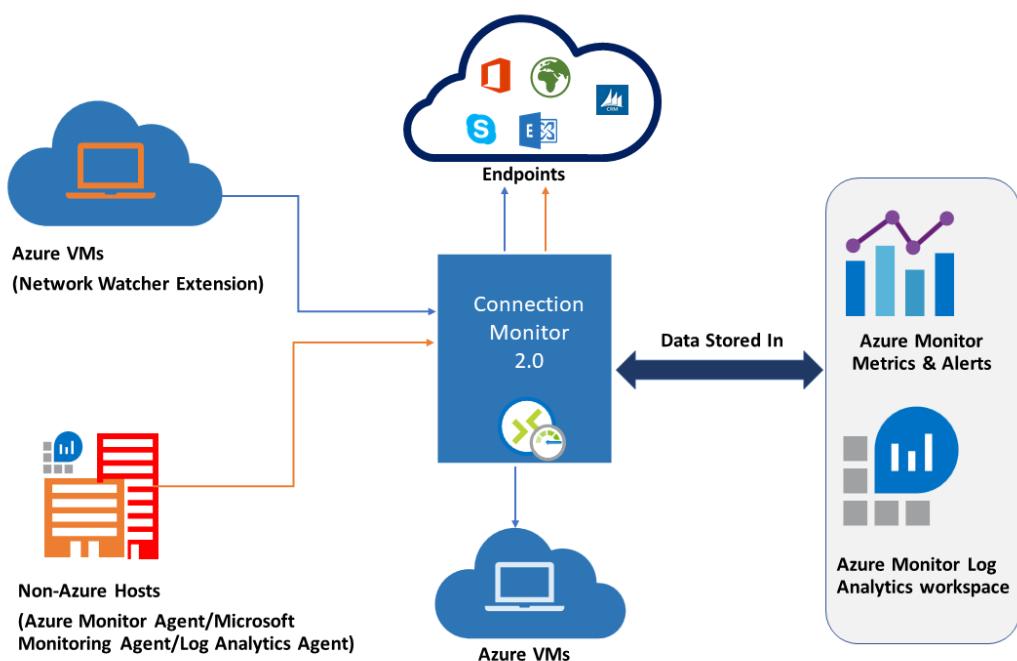
 Off  On

3. Change the parameters you want and click **Save** to deploy the changes.

Connection Monitor

Connection Monitor overview

Connection Monitor provides unified end-to-end connection monitoring in Azure Network Watcher. The Connection Monitor feature supports hybrid and Azure cloud deployments. Network Watcher provides tools to monitor, diagnose, and view connectivity-related metrics for your Azure deployments.



Here are some use cases for Connection Monitor:

- Your front-end web server VM communicates with a database server VM in a multi-tier application. You want to check network connectivity between the two VMs.
- You want VMs in the East US region to ping VMs in the Central US region, and you want to compare cross-region network latencies.
- You have multiple on-premises office sites in Seattle, Washington, and in Ashburn, Virginia. Your office sites connect to Microsoft 365 URLs. For your users of Microsoft 365 URLs, compare the latencies between Seattle and Ashburn.
- Your hybrid application needs connectivity to an Azure Storage endpoint. Your on-premises site and your Azure application connect to the same Azure Storage endpoint. You want to compare the latencies of the on-premises site to the latencies of the Azure application.

- You want to check the connectivity between your on-premises setups and the Azure VMs that host your cloud application.

Connection Monitor combines the best of two features: the Network Watcher Connection Monitor (Classic) feature and the Network Performance Monitor (NPM) Service Connectivity Monitor, ExpressRoute Monitoring, and Performance Monitoring feature.

Here are some benefits of Connection Monitor:

- Unified, intuitive experience for Azure and hybrid monitoring needs
- Cross-region, cross-workspace connectivity monitoring
- Higher probing frequencies and better visibility into network performance
- Faster alerting for your hybrid deployments
- Support for connectivity checks that are based on HTTP, TCP, and ICMP
- Metrics and Log Analytics support for both Azure and non-Azure test setups

Set up Connection Monitor

There are several key steps you need to perform in order to setup Connection Monitor for monitoring:

1. **Install monitoring agents** - Connection Monitor relies on lightweight executable files to run connectivity checks. It supports connectivity checks from both Azure environments and on-premises environments. The executable file that you use depends on whether your VM is hosted on Azure or on-premises. For more information, visit [Install monitoring agents](#).
2. **Enable Network Watcher on your subscription** - All subscriptions that have a virtual network are enabled with Network Watcher. When you create a virtual network in your subscription, Network Watcher is automatically enabled in the virtual network's region and subscription. This automatic enabling doesn't affect your resources or incur a charge. Ensure that Network Watcher isn't explicitly disabled on your subscription.
3. **Create a connection monitor** - Connection Monitor monitors communication at regular intervals. It informs you of changes in reachability and latency. You can also check the current and historical network topology between source agents and destination endpoints. Sources can be Azure VMs or on-premises machines that have an installed monitoring agent. Destination endpoints can be Microsoft 365 URLs, Dynamics 365 URLs, custom URLs, Azure VM resource IDs, IPv4, IPv6, FQDN, or any domain name.
4. **Set up data analysis and alerts** - The data that Connection Monitor collects is stored in the Log Analytics workspace. You set up this workspace when you created the connection monitor. Monitoring data is also available in Azure Monitor Metrics.

You can use Log Analytics to keep your monitoring data for as long as you want. Azure Monitor stores metrics for only 30 days by default. For more information, visit [Data collection, analysis, and alerts](#).

5. **Diagnose issues in your network** - Connection Monitor helps you diagnose issues in your connection monitor and your network. Issues in your hybrid network are detected by the Log Analytics agents that you installed earlier. Issues in Azure are detected by the Network Watcher extension. You can view issues in the Azure network in the network topology. For more information, visit [Diagnose issues in your network](#).

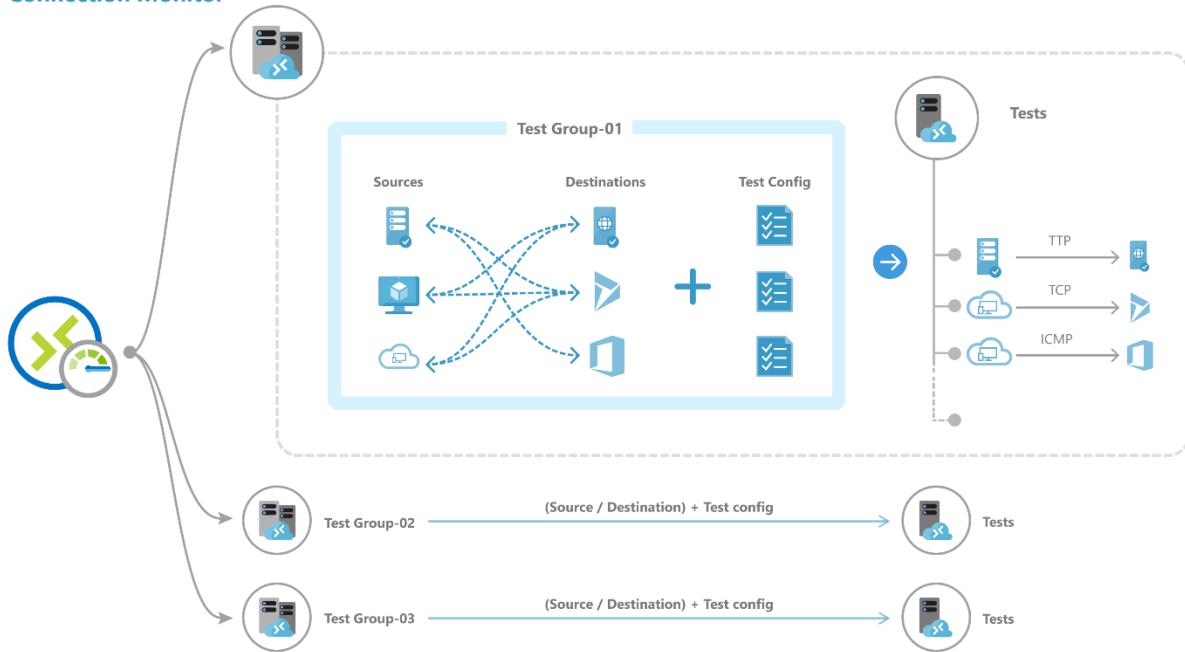
Create a Connection Monitor

In connection monitors that you create by using Connection Monitor, you can add both on-premises machines and Azure VMs as sources. These connection monitors can also monitor connectivity to endpoints. The endpoints can be on Azure or on any other URL or IP.

Connection Monitor includes the following entities:

- **Connection monitor resource** – A region-specific Azure resource. All of the following entities are properties of a connection monitor resource.
- **Endpoint** – A source or destination that participates in connectivity checks. Examples of endpoints include Azure VMs, on-premises agents, URLs, and IPs.
- **Test configuration** – A protocol-specific configuration for a test. Based on the protocol you chose, you can define the port, thresholds, test frequency, and other parameters.
- **Test group** – The group that contains source endpoints, destination endpoints, and test configurations. A connection monitor can contain more than one test group.
- **Test** – The combination of a source endpoint, destination endpoint, and test configuration. A test is the most granular level at which monitoring data is available. The monitoring data includes the percentage of checks that failed and the round-trip time (RTT).

Connection Monitor



You can create a connection monitor using Azure portal, ARMClient or PowerShell.

To create a monitor in **Connection Monitor** by using the **Azure portal**:

1. On the Azure portal home page, go to **Network Watcher**.

Azure services

Create a resource	Network Watcher	All resources	Virtual machines	Monitor	Network interfaces	Virtual networks	Virtual networks...	Network security groups	More services
-----------------------------------	---------------------------------	-------------------------------	----------------------------------	-------------------------	------------------------------------	----------------------------------	-------------------------------------	---	-------------------------------

Recent resources

Name	Type	Last Viewed
myVM1	Virtual machine	35 minutes ago
testvm1354	Network interface	59 minutes ago
CoreServicesVnet	Virtual network	an hour ago
myNSG	Network security group	an hour ago
myVNet	Virtual network	7 hours ago
Contoso-TMprofile-SR	Traffic Manager profile	a day ago
Contoso-TMprofile	Traffic Manager profile	a day ago
myPublicIP	Public IP address	a day ago

2. In the left pane, under **Monitoring**, select **Connection monitor**, and then click **Create**.

The screenshot shows the Microsoft Network Watcher interface for the Connection monitor. At the top, there's a search bar labeled 'Search (Ctrl+ /)' and navigation links for 'Overview', 'Get Started', 'Import tests from NPM', and 'Migrate Connection Monitors'. Below these are filter options 'Filter by name' (set to '4 selected'), 'Scope : 1 Subscriptions & 0 Locations', and a timestamp 'Time : Current Time (6/21/2021, 8:47:07 AM)'. A 'View by : Connection Monitor' button is also present. On the left, a sidebar lists 'Monitoring' sections: 'Topology', 'Connection monitor (classic)', 'Connection monitor' (which is selected and highlighted with a red box), and 'Network Performance Monitor'. Under 'Network diagnostic tools', there are links for 'IP flow verify', 'NSG diagnostic', 'Next hop', 'Effective security rules', 'VPN troubleshoot', 'Packet capture', and 'Connection troubleshoot'. The main content area displays a message 'No resources to display' with a note 'Try changing your filters if you don't see what you're looking for. Learn more'.

3. On the **Basics** tab of the **Create Connection Monitor** page, you need to enter the following information for your new connection monitor:

Field	Information
Connection Monitor Name	Enter a name for your connection monitor. Use the standard naming rules for Azure resources.
Subscription	Select your Azure subscription from the list.
Region	Select a region for your connection monitor. You can select only the source VMs that are created in this region.
Workspace configuration	<p>Choose a custom workspace or the default workspace. Your workspace holds your monitoring data.</p> <p>To use the default workspace, select the check box.</p> <p>To choose a custom workspace, clear the check box. Then select the subscription and region for your custom workspace.</p>

Home > Network Watcher >

Create Connection Monitor

Microsoft

Basics Test groups Create alert Review + create

Connection Monitor enables you to monitor connectivity in your Azure and hybrid network. Select your preferred subscription and region from which monitoring will be performed. Use workspace configuration to store monitoring data generated by Connection Monitor tests in Log Analytics workspace. Complete the Basics tab then proceed to Test Groups tab. [Learn more](#)

Connection Monitor Name *

Contoso-CM

Subscription *

Free Trial

Don't see a subscription? [Open Directory + Subscription settings](#)

Region *

North Europe

▲ Workspace configuration

Use workspace created by connection monitor (default)

Workspace ⓘ

DefaultWorkspace-4218051f-51eb-4b18-a9b8-923f94e0fe4a...

Free Trial in northeurope (system generated)

Next : Test groups >>

Review + create

Cancel

4. Click **Next: Test groups >>**.
5. On the next page, you can add sources, test configurations, and destinations in your test groups. Each test group in a connection monitor includes sources and destinations that get tested on network parameters. They are tested for the percentage of checks that fail and the round-trip-time (RTT) over test configurations.

Add test group details ...

A Test group lets you define a logical group that will let you validate a set of tests between a source and destination pair using a defined test configuration. Start by naming your test group and selecting sources and destination based on which you would like to define test for monitoring your network. [Learn more about test groups](#)

Test group name *	<input type="text" value="Contoso-CMTestGroup"/>	<input checked="" type="checkbox"/>																												
Sources ⓘ	1 Items																													
<table border="1"> <tr> <td>Azure endpoints</td> <td>X</td> </tr> <tr> <td>ManufacturingVNet(Contoso-ResourceGr... Subscription: Free Trial Resource group: Contoso-ResourceGroup Edit</td> <td>X</td> </tr> <tr> <td colspan="2">Add sources</td> </tr> </table>			Azure endpoints	X	ManufacturingVNet(Contoso-ResourceGr... Subscription: Free Trial Resource group: Contoso-ResourceGroup Edit	X	Add sources																							
Azure endpoints	X																													
ManufacturingVNet(Contoso-ResourceGr... Subscription: Free Trial Resource group: Contoso-ResourceGroup Edit	X																													
Add sources																														
Test configurations ⓘ	1 Items																													
<table border="1"> <tr> <td>HTTPTest</td> <td>Edit</td> <td>X</td> </tr> <tr> <td colspan="2">Add Test configuration</td> <td></td> </tr> </table>			HTTPTest	Edit	X	Add Test configuration																								
HTTPTest	Edit	X																												
Add Test configuration																														
Destinations ⓘ	13 Items																													
<table border="1"> <tr> <td>External Addresses</td> <td></td> </tr> <tr> <td>teams.microsoft.com</td> <td>X</td> </tr> <tr> <td>URL/IP: teams.microsoft.com</td> <td>X</td> </tr> <tr> <td>Edit</td> <td></td> </tr> <tr> <td>prod.registrar.skype.com</td> <td>X</td> </tr> <tr> <td>URL/IP: prod.registrar.skype.com</td> <td>X</td> </tr> <tr> <td>Edit</td> <td></td> </tr> <tr> <td>prod(tpc.skype.com</td> <td>X</td> </tr> <tr> <td>URL/IP: prod(tpc.skype.com</td> <td>X</td> </tr> <tr> <td>Edit</td> <td></td> </tr> <tr> <td>config.edge.skype.com</td> <td>X</td> </tr> <tr> <td>URL/IP: config.edge.skype.com</td> <td>X</td> </tr> <tr> <td>Edit</td> <td></td> </tr> <tr> <td colspan="2">Add destinations</td> </tr> </table>			External Addresses		teams.microsoft.com	X	URL/IP: teams.microsoft.com	X	Edit		prod.registrar.skype.com	X	URL/IP: prod.registrar.skype.com	X	Edit		prod(tpc.skype.com	X	URL/IP: prod(tpc.skype.com	X	Edit		config.edge.skype.com	X	URL/IP: config.edge.skype.com	X	Edit		Add destinations	
External Addresses																														
teams.microsoft.com	X																													
URL/IP: teams.microsoft.com	X																													
Edit																														
prod.registrar.skype.com	X																													
URL/IP: prod.registrar.skype.com	X																													
Edit																														
prod(tpc.skype.com	X																													
URL/IP: prod(tpc.skype.com	X																													
Edit																														
config.edge.skype.com	X																													
URL/IP: config.edge.skype.com	X																													
Edit																														
Add destinations																														

Disable test group
While creating the Connection Monitor, if you have disabled a test group you will not be charged for it unless you enable it again

[Add Test Group](#) [Cancel](#)

6. Click **Add Test Group**.

The screenshot shows the 'Create Connection Monitor' wizard with the 'Test groups' tab selected. A table lists a single test group named 'Contoso-CMTestGroup'. The table columns are: Name, Sources, Destination, Test Configurations, Status, and an ellipsis column. The 'Name' row contains 'Contoso-CMTestGroup'. The 'Sources' row contains 'ManufacturingVnet(Contoso-Resourc...'. The 'Destination' row contains 'teams.microsoft.com + 12more'. The 'Test Configurations' row contains 'HTTPTest'. The 'Status' row contains 'Enabled'. The 'Next : Create alert >>' button at the bottom is highlighted with a red box.

7. Click **Next: Create Alerts >>**.
8. On the **Create alert** tab, you can set up alerts on tests that are failing based on the thresholds set in test configurations.
9. You need to enter the following information for your alert:
 - Create alert (check box): You can select this check box to create a metric alert in Azure Monitor. When you select this check box, the other fields will be enabled for editing. (Note: Additional charges for the alert will be applicable.)
 - Scope (Resource/Hierarchy): The values here are automatically filled in for you, based on the values you specified on the **Basics** tab.
 - Condition: The alert is created on the Test Result(preview) metric. When the result of the connection monitor test is a failing result, the alert rule will fire.
 - Action group: You can enter your email directly or you can create alerts via action groups. If you enter your email directly, an action group with the name NPM Email ActionGroup is created. The email ID is added to that action group. If you choose to use action groups, you need to select a previously created action group.
 - Alert rule name: This is the name of the connection monitor and is already filled in for you.
 - Enable rule upon creation: Select this check box to enable the alert rule based on the condition (default setting). Disable this check box if you want to create the rule without enabling it - perhaps for evaluation and testing purposes, or because you are just not ready to deploy it yet.

Home > Network Watcher >

Create Connection Monitor

Microsoft

Create alert

Scope

Resource	Hierarchy
NetworkWatcher_northeurope/Contoso-CM	Free Trial > networkwatcherrg

Condition

Condition name

Whenever the count test result is greater than 2 count

Action group

Action group name

stever_CMTest	1 Email
---------------	---------

Select action group

Alert rule

Alert rule name

Enable rule upon creation

<< Previous **Next : Review + create >>** Review + create Cancel

10. Click **Next: Review + create >>**.

Home > Network Watcher >

Create Connection Monitor

Microsoft

Basics Test groups Create alert **Review + create**

This Connection Monitor's estimated monthly cost is \$0.9 [Learn more](#)

This is not your bill. This is an estimated monthly rate for the currently configured criteria. The estimated rate does not take into account any of the discounts you may be eligible for. Currently, you will not be charged for creating Connection Monitors. Hence current cost/month will show \$0. Once pricing is in effect, you will be charged based on published pricing [Learn more](#). Test group wise cost is shown in the estimated cost/month column, which may not include free tests. Connection Monitor cost is shown at the end and this includes free tests if applicable.

Primary details

Essentials

Connection Monitor Name : Contoso-CM	Status : Enabled
Subscription : Free Trial	Workspace : DefaultWorkspace-4218051f-51eb-4b18-a9b8-923f94e0fe4a-NEU
Region : North Europe	

Test groups (1)

Name	Sources	Destination	Test Configurations	Current Cost/Month	Estimated Cost/Month	Status
Contoso-CMTestGroup	ManufacturingVnet(Conto...	teams.microsoft.com + 12...	HTTPTest	\$0.00	\$3.90	Enabled

<< Previous **Create** Cancel Download template

11. Review your information, and then click **Create**.

Traffic Analytics

Traffic Analytics is a cloud-based solution that provides visibility into user and application activity in cloud networks. Traffic Analytics analyzes Network Watcher network security group (NSG) flow logs to provide insights into traffic flow in your Azure cloud and provide rich visualizations of data written to NSG flow logs.

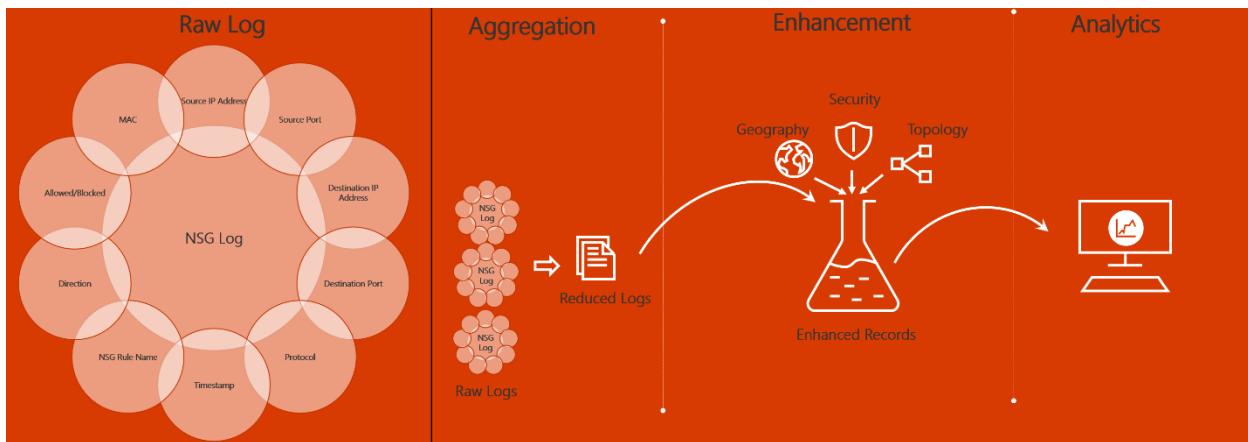
With Traffic Analytics, you can:

- Visualize network activity across your Azure subscriptions and identify hot spots.
- Identify security threats to, and secure your network, with information such as open-ports, applications attempting internet access, and virtual machines (VM) connecting to rogue networks.
- Understand traffic flow patterns across Azure regions and the internet to optimize your network deployment for performance and capacity.
- Pinpoint network misconfigurations leading to failed connections in your network.

How Traffic Analytics works

Traffic analytics examines the raw NSG flow logs and captures reduced logs by aggregating common flows among the same source IP address, destination IP address, destination port, and protocol. For example, Host 1 (IP address: 10.10.10.10) communicating to Host 2 (IP address: 10.10.20.10), 100 times over a period of 1 hour using port (for example, 80) and protocol (for example, http). The reduced log has one entry, that Host 1 & Host 2 communicated 100 times over a period of 1 hour using port 80 and protocol HTTP, instead of having 100 entries. Reduced logs are enhanced with geography, security, and topology information, and then stored in a Log Analytics workspace.

The diagram below illustrates the data flow:



The key components of Traffic Analytics are:

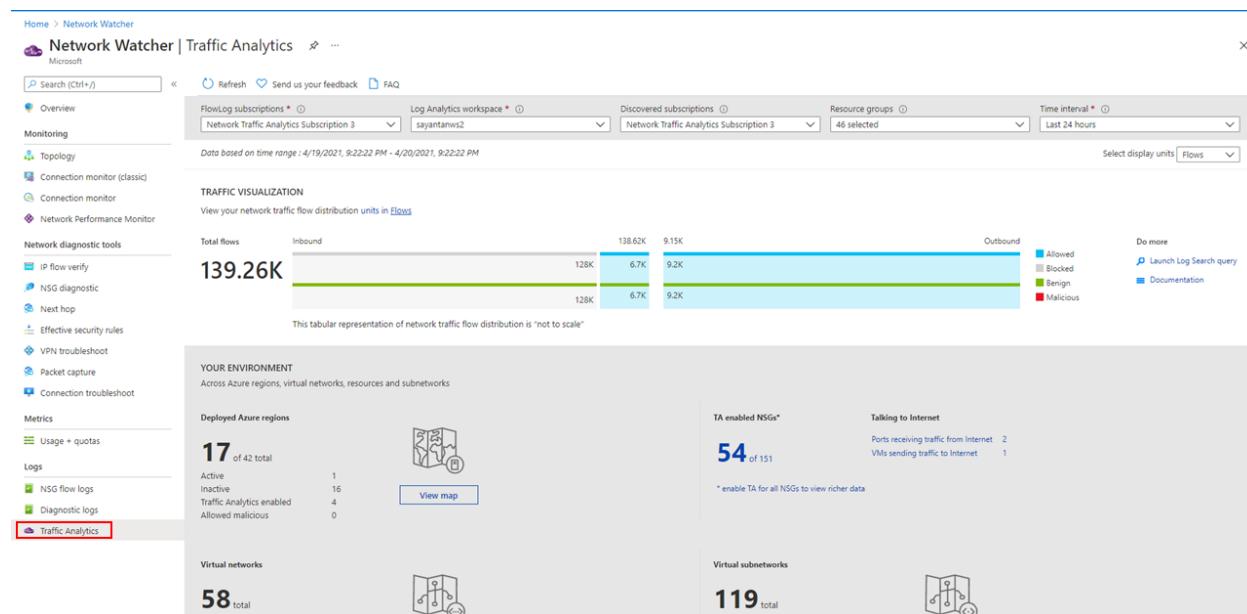
- **Network security group (NSG)** - Contains a list of security rules that allow or deny network traffic to resources connected to an Azure Virtual Network. NSGs can be associated to subnets, individual VMs (classic), or individual network interfaces (NIC) attached to VMs (Resource Manager). For more information, see Network security group overview.
- **Network security group (NSG) flow logs** - Allow you to view information about ingress and egress IP traffic through a network security group. NSG flow logs are written in json format and show outbound and inbound flows on a per rule basis, the NIC the flow applies to, five-tuple information about the flow (source/destination IP address, source/destination port, and protocol), and if the traffic was allowed or denied. For more information about NSG flow logs, see NSG flow logs.
- **Log Analytics** - An Azure service that collects monitoring data and stores the data in a central repository. This data can include events, performance data, or custom data provided through the Azure API. Once collected, the data is available for alerting, analysis, and export. Monitoring applications such as network performance monitor and traffic analytics are built using Azure Monitor logs as a foundation. For more information, see Azure Monitor logs.
- **Log Analytics workspace** - An instance of Azure Monitor logs, where the data pertaining to an Azure account, is stored. For more information about Log Analytics workspaces, see Create a Log Analytics workspace.
- **Network Watcher** - A regional service that enables you to monitor and diagnose conditions at a network scenario level in Azure. You can turn NSG flow logs on and off with Network Watcher. For more information, see Network Watcher.

To analyze traffic, you need to have an existing network watcher, or enable a network watcher in each region that you have NSGs that you want to analyze traffic for. Traffic analytics can be enabled for NSGs hosted in any of the supported regions.

Before enabling NSG flow logging, you must have a network security group to log flows for. If you do not have a network security group, then you must create one using the Azure port, the Azure CLI, or PowerShell.

To view Traffic Analytics, search for **Network Watcher** in the portal search bar. In Network Watcher, to explore traffic analytics and its capabilities, select **Traffic Analytics** from the left menu.

The example screenshot below shows the Traffic Analytics dashboard.



Summary and resources

Completed 100 XP

- 1 minute

Now that you have reviewed this module, you should be able to:

- Configure network health alerts and logging by using Azure Monitor
- Create and configure a Connection Monitor instance
- Configure and use Traffic Analytics
- Configure NSG flow logs
- Enable and configure diagnostic logging
- Configure Azure Network Watcher

Resources

Use these resources to discover more.

[Network monitoring solutions](#)