

Domain 3: Configure and manage virtual networks for Azure administrators

Configure virtual networks

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/1-introduction>

Introduction

Completed

Azure virtual networks are an essential component for creating private networks in Azure. They allow different Azure resources to securely communicate with each other, the internet, and on-premises networks.

Suppose you work for a company in the healthcare industry. Your company is looking to migrate their on-premises infrastructure to Azure. They want to ensure secure communication between their resources in Azure and their on-premises network. The company is also concerned about scalability and availability. By using Azure virtual networks, they can create a private network in Azure and securely connect their resources.

The topics covered in this module include subnetting, creating virtual networks, and using private and public IP addresses. You also learn about the different scenarios where virtual networks can be used. These scenarios include creating a dedicated private cloud-only network or extending a data center securely. The module provides a detailed explanation of subnets and their benefits, and how to plan IP addressing for Azure resources.

By the end of this module, you have a clear understanding of how to create and configure AI-ready virtual networks in Azure. You're able to effectively use subnets, assign IP addresses, and ensure secure communication between your Azure resources and on-premises network.

Learning objectives

In this module, you learn how to:

- Describe Azure virtual network features and components.
- Identify features and usage cases for subnets and subnetting.
- Identify usage cases for private and public IP addresses.
- Create a virtual network and assign an IP address.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Basic knowledge of virtual networking in cloud environments.
- Familiarity with IP address formats and subnetting.

2. Plan virtual networks

Plan virtual networks

Completed

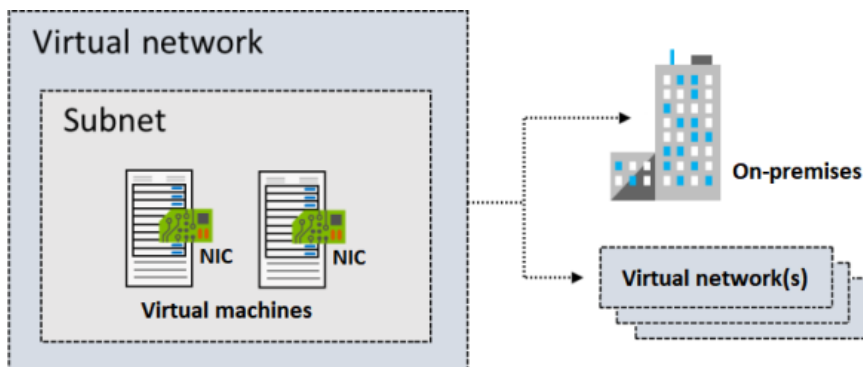
A major incentive for adopting cloud solutions like Azure is to enable information technology departments to transition server resources to the cloud. Moving resources to the cloud can save money and simplify administrative operations. Relocating resources removes the need to maintain expensive datacenters with uninterruptible power supplies, generators, multiple fail-safes, or clustered database servers. For small and medium-sized companies, which might not have the expertise to maintain their own robust infrastructure, moving to the cloud is particularly appealing.

Things to know about Azure virtual networks

You can implement Azure Virtual Network to create a virtual representation of your network in the cloud. With some [planning](#), you can deploy virtual networks and connect the resources you need more effectively. Let's examine some characteristics of virtual networks in Azure.

- An Azure virtual network is a logical isolation of the Azure cloud resources.
- You can use virtual networks to provision and manage virtual private networks (VPNs) in Azure.
- Each virtual network has its own Classless Inter-Domain Routing (CIDR) block and can be linked to other virtual networks and on-premises networks.
- You can link virtual networks with an on-premises IT infrastructure to create hybrid or cross-premises solutions, when the CIDR blocks of the connecting networks don't overlap.
- You control the DNS server settings for virtual networks, and segmentation of the virtual network into subnets.

The next illustration depicts a virtual network that has a subnet containing two virtual machines. The virtual network has connections to an on-premises infrastructure and a separate virtual network.



Things to consider when using virtual networks

Virtual networks can be used in many ways. As you think about the configuration plan for your virtual networks and subnets, consider the following scenarios.

Scenario	Description
Create a dedicated private cloud-only virtual network	Sometimes you don't require a cross-premises configuration for your solution. When you create a virtual network, your services and virtual machines within your virtual network can communicate directly and securely with each other in the cloud. You can still configure endpoint connections for the virtual machines and services that require internet communication, as part of your solution.
Securely extend your data center with virtual networks	You can build traditional site-to-site VPNs to securely scale your datacenter capacity. Site-to-site VPNs use IPSEC to provide a secure connection between your corporate VPN gateway and Azure.
Enable hybrid cloud scenarios	Virtual networks give you the flexibility to support a range of hybrid cloud scenarios. You can securely connect cloud-based applications to any type of on-premises system, such as mainframes and Unix systems.

3. Create subnets

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/3-create-subnets>

Create subnets

Completed

Azure [Subnets](#) provide a way for you to implement logical divisions within your virtual network. Your network can be segmented into subnets to help improve security, increase performance, and make it easier to manage.

Things to know about subnets

There are certain conditions for the IP addresses in a virtual network when you apply segmentation with subnets.

- Each subnet contains a range of IP addresses that fall within the virtual network address space.
- The address range for a subnet must be unique within the address space for the virtual network.
- The range for one subnet can't overlap with other subnet IP address ranges in the same virtual network.
- The IP address space for a subnet must be specified by using CIDR notation.
- You can segment a virtual network into one or more subnets in the Azure portal. Characteristics about the IP addresses for the subnets are listed.

+ Subnet + Gateway subnet Refresh Manage users Delete				
Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated
subnet0	10.0.0.0/24	-	250	-
subnet1	10.0.1.0/24	-	251	-
subnet2	10.0.2.0/24	-	251	-
AzureBastionSubnet	10.0.30.0/26	-	27	-
GatewaySubnet	10.0.3.0/27	-	availability dependent on dynamic use	-

Reserved addresses

For each subnet, Azure reserves five IP addresses. The first four addresses and the last address are reserved.

Let's examine the reserved addresses in an IP address range of `192.168.1.0/24`.

Reserved address	Reason
<code>192.168.1.0</code>	This value identifies the virtual network address.
<code>192.168.1.1</code>	Azure configures this address as the default gateway.
<code>192.168.1.2</code> and <code>192.168.1.3</code>	Azure maps these Azure DNS IP addresses to the virtual network space.
<code>192.168.1.255</code>	This value supplies the virtual network broadcast address.

Things to consider when using subnets

When you plan for adding subnet segments within your virtual network, there are several factors to consider.

- **Consider service requirements.** Each service directly deployed into a virtual network has specific requirements for routing and the types of traffic that must be allowed into and out of associated subnets. A service might require or create their own subnet. There must be enough unallocated space to meet the service requirements. Suppose you connect a virtual network to an on-premises network by using Azure VPN Gateway. The virtual network must have a dedicated subnet for the gateway.
- **Consider network virtual appliances.** Azure routes network traffic between all subnets in a virtual network, by default. You can override Azure's default routing to prevent Azure routing between subnets. You can also override the default to route traffic between subnets through a network virtual appliance. If you require traffic between resources in the same virtual network to flow through a network virtual appliance, deploy the resources to different subnets.
- **Consider network security groups.** You can associate zero or one network security group to each subnet in a virtual network. You can associate the same or a different network security group to each subnet. Each network security group contains rules that allow or deny traffic to and from sources and destinations.
- **Consider private links.** Azure Private Link provides private connectivity from a virtual network to Azure platform as a service (PaaS), customer-owned, or Microsoft partner services. Private Link simplifies the network architecture and secures the connection between endpoints in Azure. The service eliminates data exposure to the public internet.

4. Create virtual networks

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/4-create-virtual-networks>

Create virtual networks

Completed

You can create new virtual networks at any time. You can also add virtual networks when you create a virtual machine.

Things to know about creating virtual networks

Review these requirements for creating a virtual network.

- When you create a virtual network, you need to define the IP address space for the network.
- Plan to use an IP address space that's not already in use in your organization.
- The address space for the virtual network can be either on-premises or in the cloud, but not both.
- To create a virtual network, you need to define at least one subnet.
 - Each subnet contains a range of IP addresses that fall within the virtual network address space.
 - The address range for each subnet must be unique within the address space for the virtual network.

- The range for one subnet can't overlap with other subnet IP address ranges in the same virtual network.
- You can create a virtual network in the Azure portal. Provide the Azure subscription, resource group, virtual network name, and service region for the network.

[Home](#) > [Virtual networks](#) >

Create virtual network

...

×

[Basics](#) IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

Project details

Subscription * ⓘ

Resource group * ⓘ

[Create new](#)

Instance details

Name *

Region *

Note

Default limits on Azure networking resources can change periodically. Be sure to consult the [Azure networking documentation](#) for the latest information.

5. Plan IP addressing

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/5-plan-addressing>

Plan IP addressing

Completed

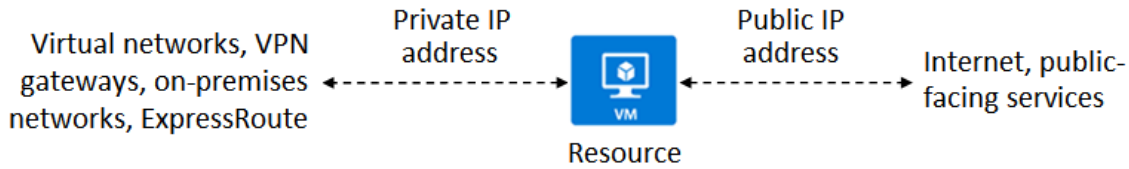
Let's begin with an overview of IP addressing.

You can assign IP addresses to Azure resources to communicate with other Azure resources, your on-premises network, and the internet. There are two types of Azure IP addresses: *private* and *public*.

Private IP addresses enable communication within an Azure virtual network and your on-premises network. You create a private IP address for your resource when you use a VPN gateway or Azure ExpressRoute circuit to extend your network to Azure.

Public IP addresses allow your resource to communicate with the internet. You can create a public IP address to connect with Azure public-facing services.

The next illustration shows a virtual machine resource that has a private IP address and a public IP address.



Things to know about IP addresses

Let's take a closer look at the characteristics of IP addresses.

- IP addresses can be statically assigned or dynamically assigned.
- You can separate dynamically and statically assigned IP resources into different subnets.
- Static IP addresses don't change and are best for certain situations, such as:
 - DNS name resolution, where a change in the IP address requires updating host records.
 - IP address-based security models that require apps or services to have a static IP address.
 - TLS/SSL certificates linked to an IP address.
 - Firewall rules that allow or deny traffic by using IP address ranges.
 - Role-based virtual machines such as Domain Controllers and DNS servers.

6. Create public IP addressing

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/6-create-public-ip-addressing>

Create public IP addressing

Completed

You can create a public IP address for your resource in the Azure portal. For example, you could create a public IP address for a virtual machine.

Create public IP address ...

Basics

DDoS Protection

Tags

Review + create

SKU * ⓘ

☐

Standard V2: For use with Standard V2 NAT Gateway

☒

Standard

Availability zone * ⓘ

Zone-redundant ▼

Tier * ⓘ

☒

Regional

☐

Global

IP address assignment

Static IPs are assigned at the time the resource is created and released when the resource is deleted.

IP address assignment *

Static

Things to consider when creating a public IP address

- To create a public IP address, configure these settings. Notice that public IP addresses are often used with load balancers.
- **IP Version:** Public IP addresses can be attached to a load balancer or to a network interface. IPv4 and IPv6 addresses are charged at the same rate.
 - **SKU:** Select the SKU for the public IP address. A Public IP's SKU must match the SKU of the Load Balancer with which it is used.
 - **Tier:** Must match the load balancer tier. A cross-region load balancer distributes traffic across regional backends. Regional distributes traffic within a virtual network.
 - **IP address assignment:** Static addresses are assigned when a public IP address is created. Static addresses aren't released until a public IP address resource is deleted.

7. Associate public IP addresses

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/7-associate-public-ip-addresses>

Associate public IP addresses

Completed

A [public IP address](#) resource can be associated with virtual machine network interfaces, internet-facing load balancers, VPN gateways, and application gateways. You can associate your resource with both dynamic and static public IP addresses.

Things to consider when associating public IP addresses

The next table summarizes how you can associate public IP addresses for different types of resources.

Top-level resource	IP address configuration
Virtual machine	Network interface configuration
Virtual Network Gateway (VPN), Virtual Network Gateway (ER), NAT Gateway	Gateway IP configuration
Public Load Balancer, Application Gateway, Azure Firewall, Route Server, API Management	Front-end configuration
Bastion host	Public IP configuration

Public IP address SKU features

The next table summarizes the standard SKU features.

Public IP address	Standard SKU
Allocation method	Static
Security	Secure by default model
Available zones	Supported. Standard IPs can be nonzonal, zonal, or zone-redundant.

8. Allocate or assign private IP addresses

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/8-associate-private-ip-addresses>

Allocate or assign private IP addresses

Completed

A [private IP address](#) resource can be associated with virtual machine network interfaces, internal load balancers, and application gateways. Azure can provide an IP address (dynamic assignment) or you can assign the IP address (static assignment).

Things to consider when associating private IP addresses

The next table summarizes how you can associate private IP addresses for different types of resources.

Resource	Private IP address association	Dynamic IP address	Static IP address
Virtual machine	NIC	Yes	Yes
Internal load balancer	Front-end configuration	Yes	Yes
Application gateway	Front-end configuration	Yes	Yes

Private IP address assignment

A private IP address is allocated from the address range of the virtual network subnet that a resource is deployed in. There are two options: dynamic and static.

- **Dynamic:** Azure assigns the next available unassigned or unreserved IP address in the subnet's address range. Dynamic assignment is the default allocation method.

Suppose addresses 10.0.0.4 through 10.0.0.9 are already assigned to other resources. In this case, Azure assigns the address 10.0.0.10 to a new resource.

- **Static:** You select and assign any unassigned or unreserved IP address in the subnet's address range.

Suppose a subnet's address range is 10.0.0.0/16, and addresses 10.0.0.4 through 10.0.0.9 are already assigned to other resources. In this scenario, you can assign any address between 10.0.0.10 and 10.0.255.254.

Tip

For a more complete review of virtual networks, consider the [Introduction to Virtual Networks](#) training module.

9. Exercise: Create and configure virtual networks

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/9-simulation-create-networks>

Exercise: Create and configure virtual networks

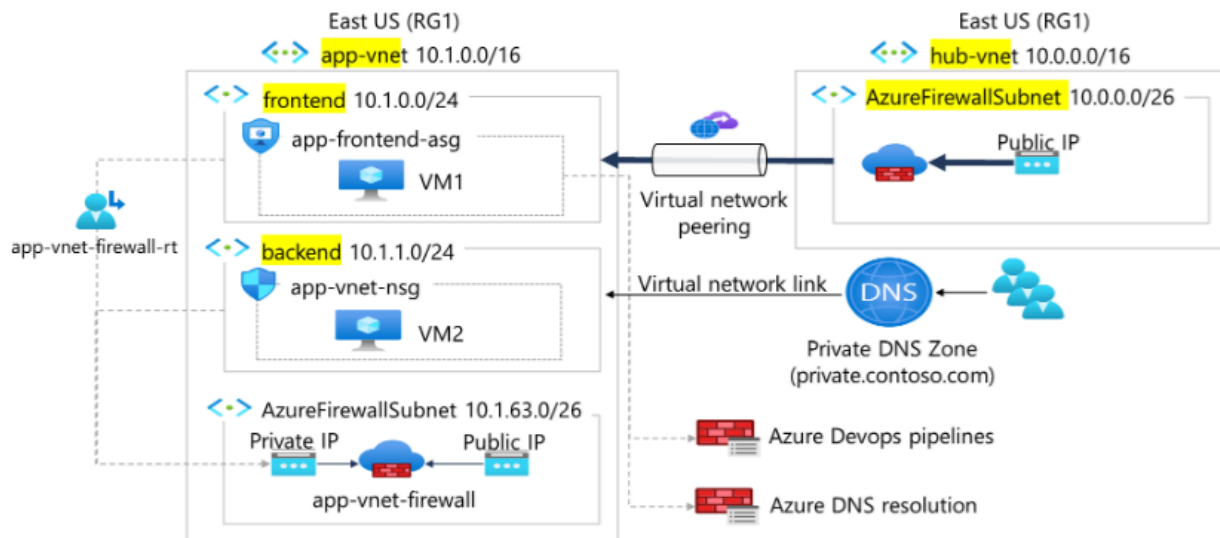
Completed

Exercise scenario

Your organization is migrating a web-based application to Azure. Your first task is to put in place the virtual networks and subnets. You also need to securely peer the virtual networks. You identify these requirements.

- Two virtual networks are required, app-vnet and hub-vnet. The virtual networks simulate a hub and spoke network architecture.
- The app-vnet hosts the application. The app-vnet virtual network requires two subnets. The frontend subnet hosts the web servers. The backend subnet hosts the database servers.
- The hub-vnet only requires a subnet for the firewall.
- The two virtual networks must be able to communicate with each other securely and privately through virtual network peering.
- Both virtual networks should be in the same region.

Architecture diagram



Job skills

- Create a virtual network.
- Create a subnet.
- Configure virtual network peering (optional).

Note

Estimated time: 30 minutes. To complete this exercise, you need an [Azure subscription](#).

Launch the exercise, and follow the instructions. When finished, be sure to return to this page so you can continue learning.

Launch Exercise

10. Module assessment

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/10-knowledge-check>

Module assessment

Completed

11. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-virtual-networks/11-summary-resources>

Summary and resources

Completed

In this module, you learned about Azure virtual networks and their importance in creating private networks in Azure. You explored the benefits of using virtual networks, such as scalability, availability, and isolation. You learned how to create virtual networks with subnetting and how to determine which resources require public or private IP addresses.

The main takeaways from this module are:

- Azure virtual networks allow different Azure resources to securely communicate with each other, the internet, and on-premises networks.
- Subnets within virtual networks provide logical divisions, improving security, performance, and management.
- When creating virtual networks, ensure that the IP address space is unique and doesn't overlap with other subnets.
- IP addresses can provide public or private access to resources.

Learn more with Copilot

Copilot can assist you in designing Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- Explain CIDR for a nontechnical audience. Provide examples.
- What are the basic steps and considerations for creating a virtual network in Azure?
- What types of Azure resources should be assigned a static IP address?

Learn more with documentation

- [What is Azure Virtual Network?](#). This article is your starting point to learn about virtual networks.
- [Public IP addresses](#). This article reviews the basics of when to use public IP addresses.
- [Private IP addresses](#). This article reviews the basics of when to use private IP addresses.

Learn more with self-paced training

- [Introduction to Azure Virtual Networks](#). Learn how to design and implement core Azure Networking infrastructure.
- [Implement Windows Server IaaS virtual machine IP addressing and routing](#). Learn about IP addressing and virtual networks for virtual machines.

Configure network security groups

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/1-introduction>

Introduction

Completed

Network security groups are a way to limit network traffic to resources in your virtual network. Network security groups contain a list of security rules that allow or deny inbound or outbound network traffic.

Suppose your company has several locations and wants to migrate to a cloud based solution. The company only considers moving key systems onto the cloud platform if stringent security requirements can be met. These requirements include tight control over which computers have network access to the app servers. You need to secure both virtual machine networking and Azure services networking. Your goal is to prevent unwanted or unsecured network traffic from being able to reach key systems.

In this module, you learn how to create an AI-ready network security group, configure inbound and outbound port rules, and verify secure connectivity. The goal of this module is to teach you how to control network traffic with network security groups.

Learning objectives

In this module, you learn how to:

- Determine when to use network security groups.
- Create network security groups.
- Implement and evaluate network security group rules.
- Describe the function of application security groups.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Familiarity with Azure virtual networks and resources such as virtual machines.
- Working knowledge of the Azure portal so you can configure the network security groups.
- Basic understanding of traffic routing and traffic control strategies.

2. Implement network security groups

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/2-implement-network-security-groups>

Implement network security groups

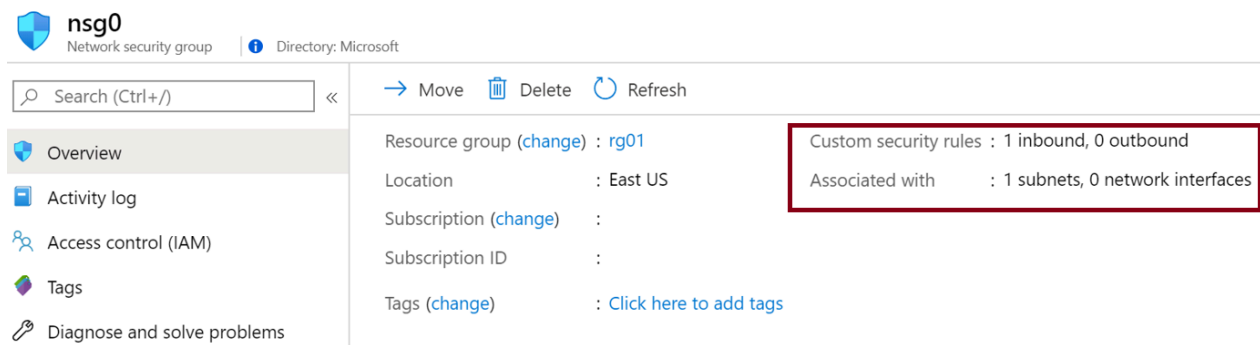
Completed

You can limit network traffic to resources in your virtual network by using a [network security group](#). You can assign a network security group to a subnet or a network interface, and define security rules in the group to control network traffic.

Things to know about network security groups

Let's look at the characteristics of network security groups.

- A network security group contains a list of security rules that allow or deny inbound or outbound network traffic.
- A network security group can be associated to a subnet or a network interface.
- A network security group can be associated multiple times.
- You create a network security group and define security rules in the Azure portal.
- The Overview page for a virtual machine provides information about the associated network security groups. You can see details such as the assigned subnets, assigned network interfaces, and the defined security rules.



nsg0
Network security group | Directory: Microsoft

Search (Ctrl+/) << → Move Delete Refresh

Overview

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

Resource group (change) : rg01

Location : East US

Subscription (change) :

Subscription ID :

Tags (change) : [Click here to add tags](#)

Custom security rules : 1 inbound, 0 outbound

Associated with : 1 subnets, 0 network interfaces

Network security groups and subnets

You can assign network security groups to a subnet and create a protected screened subnet (also referred to as a demilitarized zone or DMZ). A DMZ acts as a buffer between resources within your virtual network and the internet.

- Use the network security group to restrict traffic flow to all machines that reside within the subnet.
- Each subnet can have a maximum of one associated network security group.

Network security groups and network interfaces

You can assign network security groups to a network interface card (NIC).

- Define network security group rules to control all traffic that flows through a network interface.
- Each network interface that exists in a subnet can have zero, or one, associated network security groups.

3. Determine network security group rules

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/3-determine-network-security-groups-rules>

Determine network security group rules

Completed

Security rules in network security groups enable you to filter network traffic. You can define rules to control the traffic flow in and out of virtual network subnets and network interfaces.

Things to know about security rules

Let's review the characteristics of security rules in network security groups.

- Azure creates several default security rules within each network security group, including inbound traffic and outbound traffic. Examples of default rules include `DenyAllInbound` traffic and `AllowInternetOutbound` traffic.
- Azure creates the default security rules in each network security group that you create.
- You can add more security rules to a network security group by [specifying conditions](#). Here's a list of the most common conditions.

Setting	Value
Source	Any, IP Addresses, My IP address, Service Tag, or Application security group
Source port ranges	Specify the ports on which the rule allows or denies traffic
Destination	Any, IP Addresses, Service Tag, or Application security group
Protocol	Restrict the rule to Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), Encapsulating Security Payload (ESP), or Authentication Header (AH). ESP and AH protocols are only available via JSON templates and PowerShell. The default is for the rule to apply to all protocols (Any).
Action	Allow or Deny
Priority	A value between 100 and 4,096 that's unique for all security rules within the NSG

- Each security rule is assigned a Priority value. All security rules for a network security group are processed in priority order. When a rule has a low Priority value, the rule has a higher priority or precedence in terms of order processing.
- You can't remove the default security rules.
- You can override a default security rule by creating another security rule that has a higher Priority setting for your network security group.

Inbound traffic rules

Azure defines three default inbound security rules for your network security group. These rules **deny all inbound traffic** except traffic from your virtual network and Azure load balancers. The next image shows the default inbound security rules for a network security group in the Azure portal.



PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny

Outbound traffic rules

Azure defines three default outbound security rules for your network security group. These rules **only allow outbound traffic** to the internet and your virtual network. The next image shows the default outbound security rules for a network security group in the Azure portal.

 **VM1-nsg - Outbound security rules**
 Network security group

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✓ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✗ Deny

4. Determine network security group effective rules

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/4-determine-network-security-groups-effective-rules>

Determine network security group effective rules

Completed

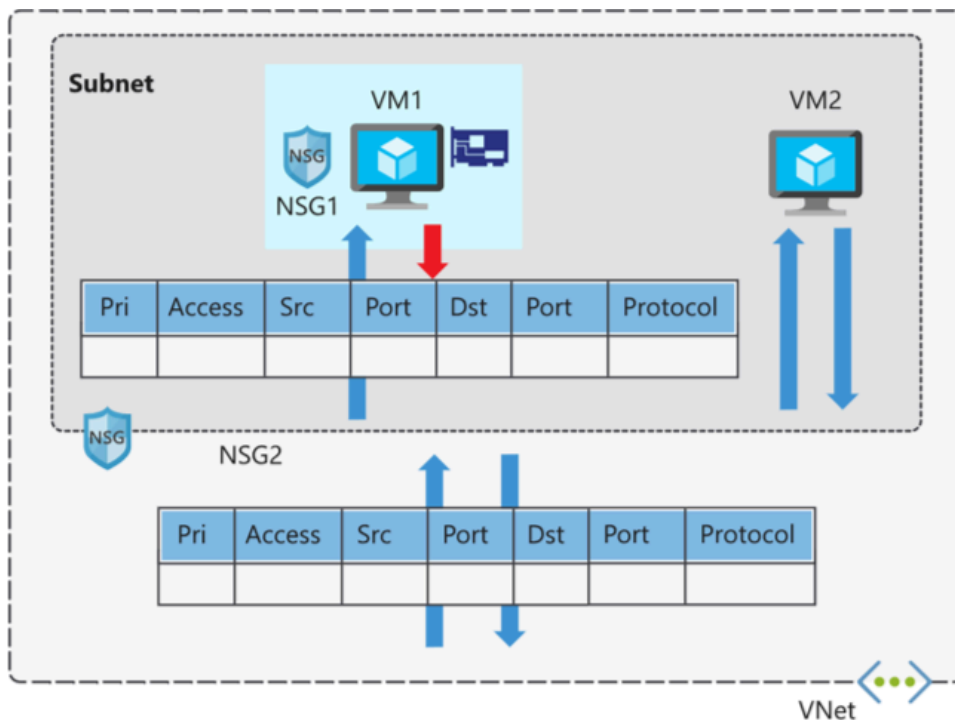
Each network security group and its defined security rules are evaluated independently. Azure processes the conditions in each rule defined for each virtual machine in your configuration.

- For inbound traffic, Azure first processes network security group security rules for any associated subnets and then any associated network interfaces.
- For outbound traffic, the process is reversed. Azure first evaluates network security group security rules for any associated network interfaces followed by any associated subnets.
- For both the inbound and outbound evaluation process, Azure also checks how to apply the rules for intra-subnet traffic. Intra-subnet traffic refers to virtual machines in the same subnet.

How Azure ends up applying your defined security rules for a virtual machine determines the overall *effectiveness* of your rules.

Network security group evaluation

When you apply network security groups to both a subnet and a network interface, each network security group is evaluated independently. Both inbound and outbound rules are considered based on the priority and processing order.



Things to consider when creating effective rules

Review the following considerations regarding creating effective security rules for machines in your virtual network.

- **Consider allowing all traffic.** If you place your virtual machine within a subnet or utilize a network interface, you don't have to associate the subnet or NIC with a network security group. This approach allows all network traffic through the subnet or NIC according to the default Azure security rules. If you're not concerned about controlling traffic to your resource at a specific level, then don't associate your resource at that level to a network security group.
- **Consider importance of allow rules.** When you create a network security group, you must define an **allow** rule for both the subnet and network interface in the group to ensure traffic can get through. If you have a subnet or NIC in your network security group, you must define an allow rule at each level. Otherwise, the traffic is denied for any level that doesn't provide an allow rule definition.
- **Consider intra-subnet traffic.** The security rules for a network security group associated to a subnet can affect traffic between all virtual machines in the subnet. You can prohibit intra-subnet traffic by defining a rule in the network security group to deny all inbound and outbound traffic. This rule prevents all virtual machines in your subnet from communicating with each other.
- **Consider rule priority.** The security rules for a network security group are processed in priority order. To ensure a particular security rule is always processed, assign the lowest possible priority value to the rule. It's a good practice to leave gaps in your priority numbering, such as 100, 200, 300, and so. The gaps in the numbering allow you to add new rules without having to edit existing rules.

View effective security rules

If you have several network security groups and aren't sure which security rules are being applied, you can use the **Effective security rules** link in the Azure portal. You can use the link to verify which security rules are applied to your machines, subnets, and network interfaces.

nsgtest | Effective security rules

Network security group

Search

Download Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Monitoring

Automation

Help

Effective security rules

Support + Troubleshooting

Select a network interface below to see the effective security rules and network security groups associated with it.

Scope

Network security group (nsgtest)

Virtual machine

Network interface

Associated NSGs: ⓘ

None

Note

[Network Watcher](#) provides a consolidated view of your infrastructure rules, including both NSG rules and Azure Virtual Network Manager security admin rules. The IP flow verify feature evaluates traffic against both NSG rules and any security admin rules that may be in effect.

5. Create network security group rules

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/5-create-network-security-groups-rules>

Create network security group rules

Completed

It's easy to add security rules to control inbound and outbound traffic in the Azure portal. You can configure your virtual network security group rule settings, and select from a large variety of communication services, including HTTPS, RDP, FTP, and DNS.

Things to know about configuring security rules

Let's look at some of the properties you need to specify to create your security rules. As you review these settings, think about the traffic rules you need to create and what services can fulfill your network requirements.

17 / 79

Source ⓘ

Any

Source port ranges * ⓘ

*

Destination ⓘ

Any

Service ⓘ

Custom

Destination port ranges * ⓘ

8080

- **Source:** Identifies how the security rule controls **inbound** traffic. The value specifies a specific source IP address range to allow or deny. The source filter can be any resource, an IP address range, an application security group, or a default tag.
- **Destination:** Identifies how the security rule controls **outbound** traffic. The value specifies a specific destination IP address range to allow or deny. The destination filter value is similar to the source filter. The value can be any resource, an IP address range, an application security group, or a default tag.
- **Service:** Specifies the destination protocol and port range for the security rule. You can choose a predefined service like RDP or SSH or provide a custom port range. There are a large number of services to select from.

Service ⓘ

Custom

SSH

RDP

Custom

FTP

- **Priority:** Assigns the priority order value for the security rule. Rules are processed according to the priority order of all rules for a network security group, including a subnet and network interface. The lower the priority value, the higher priority for the rule.

Priority * ⓘ

119

Name *

AllowAnyCustom8080Inbound

When to use augmented security rules

A single network security group rule can contain multiple values in the Source, Destination, and Service fields. This approach, called augmented security rules, reduces the total number of rules needed and simplifies NSG management.

Things to know about augmented security rules

- **Multiple IP addresses:** Combine multiple IP addresses into one rule.
- **Multiple port ranges:** Specify multiple ports and ranges in the Service field.
- **Service tags and ASGs:** Mix service tags, application security groups, and IP addresses within the same rule.
- **Reduced rule count:** Instead of creating separate rules for each IP range or port, combine them into fewer, more manageable rules.

In enterprise environments with many IP ranges or services, augmented rules prevent NSG rule sprawl. For example, instead of creating four separate rules for ports 80, 443, 8080, and 8090, create one rule with all the ports.

Tip

Expand your learning with the [Secure and isolate access to Azure resources by using network security groups and service endpoints](#) training module. This module includes a sandbox where you can practice.

6. Implement application security groups

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/6-implement-asgs>

Implement application security groups

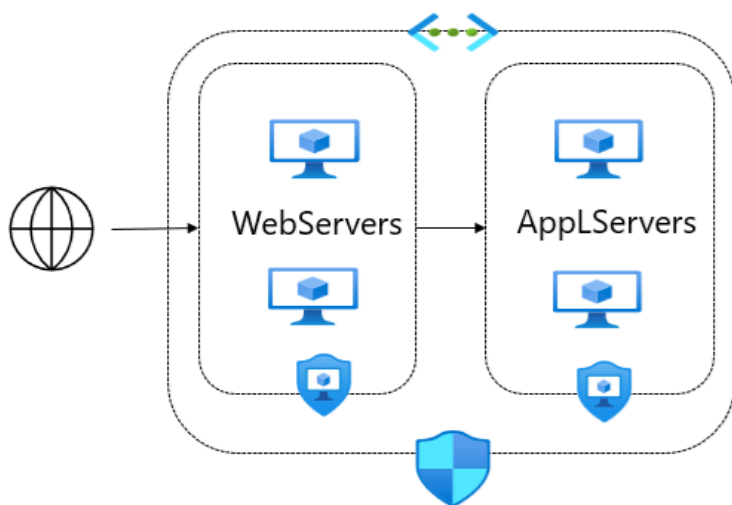
Completed

You can implement [application security groups \(ASGs\)](#) in your Azure virtual network to logically group your virtual machines by workload. You can then define your network security group rules based on your application security groups.

Things to know about using application security groups

Application security groups provide an application-centric way of looking at your infrastructure. You join your virtual machines to an application security group. Then you use the application security group as a source or destination in the network security group rules.

Let's examine how to implement application security groups by creating a configuration for an online retailer. In our example scenario, we need to control network traffic to virtual machines in application security groups.



Source	Destination	Port
Internet	WebServers	80, 443
WebServers	SQLServers	1433

Note

In the diagram, the application servers are handling SQL Server requests.

Scenario requirements

Here are the scenario requirements for our example configuration:

- In this scenario, there are two tiers: web servers and application servers.
- The web servers handle HTTP and HTTPS internet traffic.
- The application servers process SQL requests from the web servers.

Solution

For our scenario, we need to build the following configuration:

1. Create application security groups for each tier.
2. For each virtual machine server, assign its network interface to the appropriate application security group.
3. Create the network security group and security rules.
 - **Rule 1:** Set **Priority** to 100. Allow access from the internet to the web servers machines from HTTP port 80 and HTTPS port 443.

Rule 1 has the lowest priority value, so it has precedence over the other rules in the group. Customer access to our online catalog is paramount in our design.

- **Rule 2:** Set **Priority** to 110. Allow access from the web servers to application servers over SQL port 1433.
- **Rule 3:** Set **Priority** to 120. **Deny** access from anywhere to application server machines on the HTTP and HTTPS ports.

The combination of Rule 2 and Rule 3 ensures that only our web servers can access our database servers. This security configuration protects our inventory databases from outside attack.

Things to consider when using application security groups

There are several advantages to implementing application security groups in your virtual networks.

- **Consider IP address maintenance.** When you control network traffic by using application security groups, you don't need to configure inbound and outbound traffic for specific IP addresses. If you have many virtual machines in your configuration, it can be difficult to specify all of the affected IP addresses. As you maintain your configuration, the number of your servers can change. These changes can require you to modify how you support different IP addresses in your security rules.
- **Consider no subnets.** By organizing your virtual machines into application security groups, you don't need to also distribute your servers across specific subnets. You can arrange your servers by application and purpose to achieve logical groupings.
- **Consider simplified rules.** Application security groups help to eliminate the need for multiple rule sets. You don't need to create a separate rule for each virtual machine. You can dynamically apply new rules to designated application security groups. New security rules are automatically applied to all the virtual machines in the specified application security group.
- **Consider workload support.** A configuration that implements application security groups is easy to maintain and understand because the organization is based on workload usage. Application security groups provide logical arrangements for your applications, services, data storage, and workloads.
- **Consider service tags.** Service Tags represent a group of IP address prefixes from a specific Azure service. They help minimize the complexity of frequent updates on network security rules. While service tags are used to simplify the management of IP addresses for Azure services, ASGs are used to group VMs and manage network security policies based on those groups.

7. Exercise - Implement virtual networking

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/7-simulation-create-network-groups>

Exercise - Implement virtual networking

Completed

Lab scenario

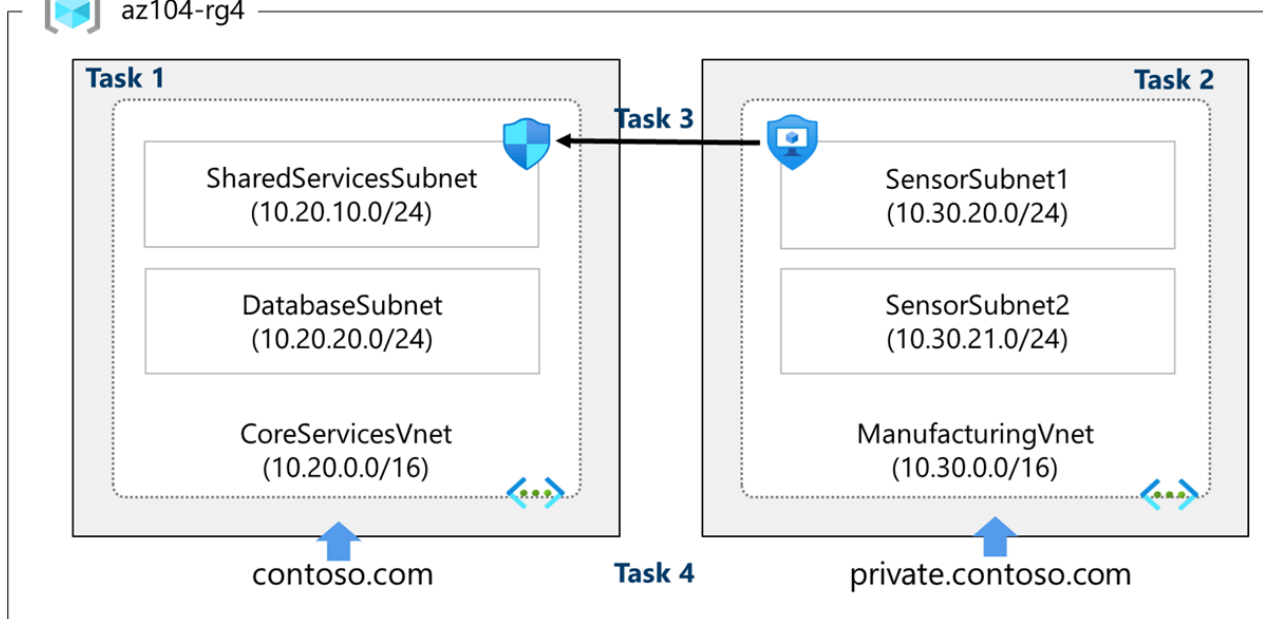
Your organization wants to ensure that access to virtual machines is restricted. As the Azure Administrator, you need to:

- Create and configure network security groups.
- Associate network security groups to virtual machines.
- Deny and allow access to the virtual machines by using network security groups.

Architecture diagram



az104-rg4



Job skills

- Create a virtual network with subnets using the portal.
- Create a virtual network and subnets using a template.
- Create and configure communication between an Application Security Group and a Network Security Group.
- Configure public and private Azure DNS zones. (optional)

Important

Estimated time: 50 minutes. To complete this exercise, you need an [Azure subscription](#).

Launch the exercise, and follow the instructions. When finished, be sure to return to this page so you can continue learning.

[Launch Exercise](#)

8. Module assessment

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/8-knowledge-check>

Module assessment

Completed

9. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-network-security-groups/9-summary-resources>

Summary and resources

Completed

In this module, you learned about network security groups (NSGs) in Azure. NSGs are used to limit network traffic to resources in your virtual network by containing a list of security rules. You can associate NSGs with subnets or network interfaces and define rules to control inbound and outbound traffic.

You also learned how NSG rules are evaluated and processed. Lastly, you learned how application security groups, allow for grouping virtual machines based on workload.

The main takeaways from this module are:

- Network security groups are essential for controlling network traffic in Azure virtual networks.
- NSG rules are evaluated and processed based on priority and can be created for subnets and network interfaces.
- Effective NSG rules can be achieved by considering rule precedence, intra-subnet traffic, and managing rule priority.
- Application security groups provide an application-centric view of infrastructure and simplify rule management.

Learn more with Copilot

Copilot can assist you in designing Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What is the difference between an Azure network security group and an application security group? Provide usage examples.
- Can you explain NSG rules in detail?
- How can I troubleshoot network security group rules?

Learn more with documentation

- [Read about network security groups](#). This article describes the properties of a network security group rule, the default security rules that are applied, and the rule properties that you can modify.
- [Filter network traffic with network security groups in the Azure portal](#). Learn how to create a network security group and an application security group.
- [Create, change, or delete a network security group](#). Learn how to work with network and application security groups.
- [Application security groups](#). Learn about application security groups and traffic control with rules.

Learn more with self-paced training

- [Secure and isolate access to Azure resources with network security groups and service endpoints \(sandbox\)](#). Learn how to secure your virtual machines and Azure services from unauthorized network access.
- [Filter network traffic with a network security group using the Azure portal](#). Learn how to create, configure, and apply NSGs for improved network security.

Host your domain on Azure DNS

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/1-introduction>

Introduction

Completed

Azure DNS lets you host your Domain Name System (DNS) records for your domains on Azure infrastructure. With Azure DNS, you can use the same credentials, APIs, tools, and billing as your other Azure services.

Let's say that your company recently bought the custom domain name `wideworldimporters.com` from a third-party domain-name registrar. The domain name is for a new website that your organization plans to launch. You need a hosting service for DNS domains. This hosting service would resolve the `wideworldimporters.com` domain to your web server's IP address.

You're already using Azure to build your website, so you decide to use Azure DNS to manage your domain.

This module shows you how to configure Azure DNS to host your domain. You also see how to add an alias and other DNS records to resolve your domain name to a website.

Learning objectives

In this module, you will:

- Configure Azure DNS to host your domain.

Prerequisites

- Knowledge of networking concepts like name resolution and IP addresses.

2. What is Azure DNS?

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/2-what-is-azure-dns>

What is Azure DNS?

Completed

3. Configure Azure DNS to host your domain

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/3-configure-azure-dns-host-domain>

Configure Azure DNS to host your domain

Completed

The new company website is in final testing. You're working on the plan to deploy the wideworldimports.com domain by using Azure DNS. You need to understand what steps are involved.

In this unit, you learn how to:

- Create and configure a DNS zone for your domain by using Azure DNS.
- Understand how to link your domain to an Azure DNS zone.
- Create and configure a private DNS zone.

Configure a public DNS zone

You use a DNS zone to host the DNS records for a domain, such as wideworldimports.com.

Step 1: Create a DNS zone in Azure

You used a third-party domain-name registrar to register the wideworldimports.com domain. The domain doesn't point to your organization's website yet.

To host the domain name with Azure DNS, you first need to create a DNS zone for that domain. A DNS zone holds all the DNS entries for your domain.

When creating a DNS zone, you need to supply the following details:

- **Subscription:** The subscription to be used.
- **Resource group:** The name of the resource group to hold your domains. If one doesn't exist, create one to allow for better control and management.
- **Name:** Your domain name, which in this case is wideworldimports.com.
- **Resource group location:** The location defaults to the location of the resource group.

[Home](#) > [DNS zones](#) >

Create DNS zone ...

Basics Tags Review + create

A DNS zone is used to host the DNS records for a particular domain. For example, the domain 'contoso.com' may contain a number of DNS records such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site). Azure DNS allows you to host your DNS zone and manage your DNS records, and provides name servers that will respond to DNS queries from end users with the DNS records that you create. [Learn more.](#)

Project details

Subscription *

Resource group *

[Create new](#)

Instance details

☐ This zone is a child of an existing zone already hosted in Azure DNS ⓘ

Name *

Resource group location ⓘ

Step 2: Get your Azure DNS name servers

After you create a DNS zone for the domain, you need to get the name server details from the name servers (NS) record. You use these details to update your domain registrar's information and point to the Azure DNS zone.

The screenshot shows the Azure portal interface for a DNS zone named 'worldwideimports.com'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Properties, Locks, Monitoring, Alerts, Metrics, Automation, Tasks (preview), Export template, Help, Resource health, and Support + Troubleshooting. The main content area is divided into 'Essentials' and 'Record sets'. The 'Essentials' section displays the resource group 'myresourcegroup', subscription, and subscription ID. A red box highlights the four name servers: Name server 1 : ns1-35.azure-dns.com, Name server 2 : ns2-35.azure-dns.net, Name server 3 : ns3-35.azure-dns.org, and Name server 4 : ns4-35.azure-dns.info. The 'Record sets' section shows a table with columns: Name, Type, TTL, Value, Alias resource type, and Alias target. The table contains two records: an NS record for '@' with a TTL of 172800 and a SOA record for '@' with a TTL of 3600.

Name	Type	TTL	Value	Alias resource type	Alias target
@	NS	172800	ns1-35.azure-dns.com. ns2-35.azure-dns.net. ns3-35.azure-dns.org. ns4-35.azure-dns.info.		...
@	SOA	3600	Email: azure-dns-hostmaster... Host: ns1-35.azure-dns.co... Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1		...

Step 3: Update the domain registrar setting

As the domain owner, you need to sign in to the domain-management application provided by your domain registrar. In the management application, edit the NS record and change the NS details to match your Azure DNS name server details.

Changing the NS details is called *domain delegation*. When you delegate the domain, you must use all four name servers provided by Azure DNS.

Step 4: Verify delegation of domain name services

The next step is to verify that the delegated domain now points to the Azure DNS zone you created for the domain. This process can take as few as 10 minutes, but might take longer.

To verify the success of the domain delegation, query the start of authority (SOA) record. The SOA record is automatically created when the Azure DNS zone is set up. You can verify the SOA record using a tool like nslookup.

The SOA record represents your domain and becomes the reference point when other DNS servers are searching for your domain on the internet.

To verify the delegation, use nslookup like this:

```
nslookup -type=SOA wideworldimports.com
```

Step 5: Configure your custom DNS settings

The domain name is wideworldimports.com. When it's used in a browser, the domain resolves to your website. But what if you want to add in web servers or load balancers? These resources need to have their own custom settings in the DNS zone, either as an A record or a CNAME.

A record

Each A record requires the following details:

- **Name:** The name of the custom domain, for example *webserver1*.
- **Type:** In this instance, it's A.
- **TTL:** Represents the "time-to-live" as a whole unit, where 1 is one second. This value indicates how long the A record lives in a DNS cache before it expires.
- **IP address:** The IP address of the server to which this A record should resolve.

CNAME record

The CNAME is the canonical name, or the alias for an A record. Use CNAME when you have different domain names that all access the same website. For example, you might need a CNAME in the *wideworldimports* zone if you want both www.wideworldimports.com and wideworldimports.com to resolve to the same IP address.

You'd create the CNAME record in the *wideworldimports* zone with the following information:

- NAME: www
- TTL: 600 seconds
- Record type: CNAME

If you exposed a web function, you'd create a CNAME record that resolves to the Azure function.

Configure a private DNS zone

Another type of DNS zone that you can configure and host in Azure is a private DNS zone. Private DNS zones aren't visible on the Internet, and don't require that you use a domain registrar. You can use private DNS zones to assign DNS names to virtual machines (VMs) in your Azure virtual networks.

Step 1: Create a private DNS zone

In the Azure portal, search for *private DNS zones*. To create the private zone, you need enter a resource group and the name of the zone. For example, the name might be something like private.wideworldimports.com.

[Home](#) > [Private DNS zones](#) >

Create Private DNS zone

Basics

Tags

Review create

A Private DNS zone provides name resolution services within virtual networks. A Private DNS zone is accessible only from the virtual networks that it is linked to and can't be accessed over internet. For example you can create a Private DNS zone named [contoso.com](#) and then create DNS records like [www.contoso.com](#) in this zone. You can then link the zone to a one or more virtual networks. [Learn more.](#)

Project details

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

Subscription *

Resource group *

[Create new](#)

Instance details

Name * ⓘ

Resource group location ⓘ

 You can link virtual networks to this Private DNS zone after zone has been created.

Step 2: Identify virtual networks

Let's assume that your organization already created your VMs and virtual networks in a production environment. Identify the virtual networks associated with VMs that need name-resolution support. To link the virtual networks to the private zone, you need the virtual

network names.

Step 3: Link your virtual network to a private DNS zone

To link the private DNS zone to a virtual network, you create a virtual network link. In the Azure portal, go to the private zone, and select **Virtual network links**.

Home > NoMarketplace-20230920225936 | Overview > private.wideworldimports.com

private.wideworldimports.com | Virtual network links

Private DNS zone

Search

«

+ Add

Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Virtual network links

Properties

Locks

Monitoring

Alerts

Metrics

Automation

Tasks (preview)

Export template

Help

Support + Troubleshooting

Search virtual network links

Link Name	Link status	Virtual network	Auto-Registration
No results.			

Select **Add** to pick the virtual network you want to link to the private zone.

Home > NoMarketplace-20230920225936 | Overview > private.wideworldimports.com

Add virtual network link

private.wideworldimports.com

Link name *

Virtual network details

Only virtual networks with Resource Manager deployment model are supported for linking with Private DNS zones. Virtual networks with Classic deployment model are not supported.

☐ I know the resource ID of virtual network

Subscription *

Virtual network *

Configuration

☐ Enable auto registration

You add a virtual network link record for each virtual network that needs private name-resolution support.

In the next unit, you learn how to create a public DNS zone.

4. Exercise - Create a DNS zone and an A record by using Azure DNS

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/4-exercise-create-dns-zone-a-record>

Exercise - Create a DNS zone and an A record by using Azure DNS

Completed

In the previous unit, we described setting up and configuring the wideworldimports.com domain to point to your Azure hosting on Azure DNS.

In this unit, you'll:

- Set up an Azure DNS and create a public DNS zone.
- Create an A record.
- Verify that the A record resolves to an IP address.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Note

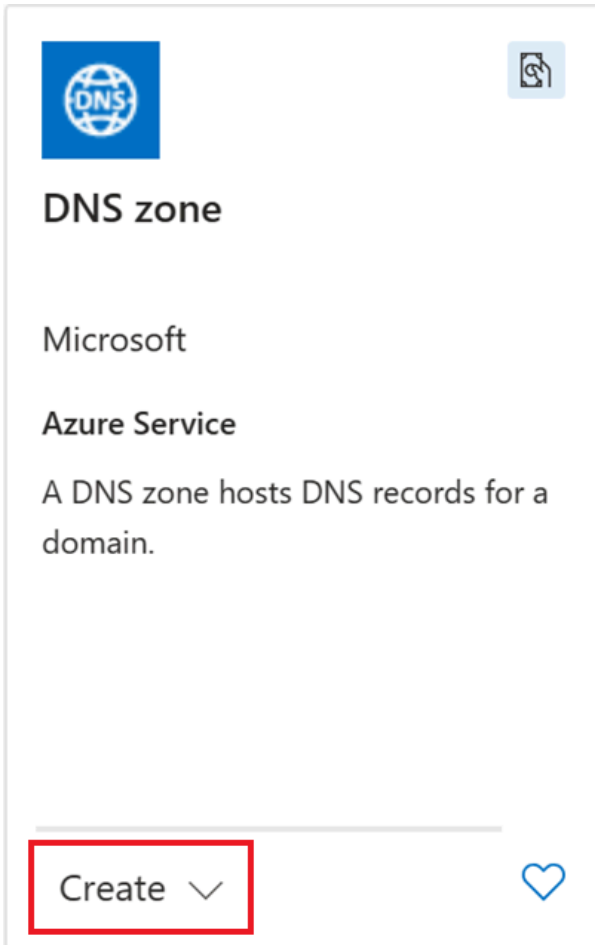
You need to use a resource group to complete the steps in this exercise. You can use a resource group that you already created, or you can create a new resource group specifically for this exercise. If you choose to create a new resource group, that will make it easier to clean up any resources that you create as you complete the exercise. If you don't have an existing resource group or you want to create a new one specifically for this exercise, you can follow the steps in [Use the Azure portal and Azure Resource Manager to manage resource groups](#) to create a resource group by using the Azure portal, or you can follow the steps in [Manage Azure resource groups by using Azure CLI](#) to create a resource group by using the the Azure CLI.

Create a DNS zone in Azure DNS

Before you can host the wideworldimports.com domain on your servers, you need to create a DNS zone. The DNS zone holds all the configuration records associated with your domain.

To create your DNS zone:

1. Sign in to the [Azure portal](#) with the account you used to activate the sandbox.
2. On the Azure **home** page, under **Azure services**, select **Create a resource**. The **Create a resource** pane appears.
3. In the *Search services and marketplace* search box, search for and select **DNS zone** by Microsoft. The **DNS zone** pane appears.
4. Select **Create > DNS zone**.



The **Create DNS zone** pane appears.

5. On the **Basics** tab, enter the following values for each setting.

Setting	Value
Project details	
Subscription	Choose your subscription
Resource group	From the dropdown list, select the resource group you want to use for this exercise.
Instance details	
Name	The name needs to be unique in the sandbox. Use <code>wideworldimportsXXXX.com</code> , and replace the "Xs" with letters or numbers.

Create DNS zone ...

Basics Tags Review + create

A DNS zone is used to host the DNS records for a particular domain. For example, the domain 'contoso.com' may contain a number of DNS records such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site). Azure DNS allows you to host your DNS zone and manage your DNS records, and provides name servers that will respond to DNS queries from end users with the DNS records that you create. [Learn more.](#)

Project details

Subscription * Concierge Subscription

Resource group * learn-a7287f0c-710a-45c2-b032-eb3430eab5c0 [Create new](#)

Instance details

☐ This zone is a child of an existing zone already hosted in Azure DNS ⓘ

Name * worldwideimportsXXXX.com ✓

Resource group location ⓘ West US

6. Select **Review + create**.

7. After validation passes, select **Create**. It takes a few minutes to create the DNS zone.

8. When deployment is complete, select **Go to resource**. The **Overview** pane for your **DNS zone** appears.

9. Select **Record Sets** from the top menu bar.

By default, the NS and SOA record sets are automatically created whenever a DNS zone is created, and automatically deleted whenever a DNS zone is deleted. The NS record set defines the Azure DNS namespaces and contains the four Azure DNS records. You use all four records when you update the registrar.

The SOA record represents your domain, and is used when other DNS servers are searching for your domain.

10. Make a note of the NS record values. You need them in the next section.

Create a DNS record

Now that the DNS zone exists, you need to create the necessary records to support the domain.

The primary record set to create is the A record. The A record set is used to point traffic from a logical domain name to the hosting server's IP address. An A record set can have multiple records. In a record set, the domain name remains constant, while the IP addresses differ.

1. If you're not already on the **Record Sets** screen, then open the **DNS zone** pane for *wideworldimportsXXXX.com*. In the top menu bar, select **Record sets**.

2. On the **Record Sets** pane, select **+ Add** in the top menu bar.

3. Select **Add** at the top of the Record sets page.

worldwideimportsXXXX.com | Recordsets ☆ ...

DNS zone

Search

+ Add Refresh Delete

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
DNS Management
Recordsets
Monitoring
Automation
Help

A record set is a collection of records in a zone that have the same name and are the same type. You can search for record sets that have been loaded on this page. If you don't see what you're looking for, you can try scrolling to allow more record sets to load. [Learn more](#)

Search

Fetches 2 record set(s).

Name	Type	TTL	Value	Alias resource type	Alias target
@	NS	172800	ns1-01.azure-dns.com. ns2-01.azure-dns.net. ns3-01.azure-dns.org. ns4-01.azure-dns.info.		
@	SOA	3600	Email: azuredns-hostmaster.microsoft.com Host: ns1-01.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1		

The **Add record set** pane appears.

4. Enter the following values for each setting.

Setting	Value	Description
Name	www	The host name that you want to resolve to an IP address.
Type	A	The A record is the most commonly used. If you're using IPv6, select the AAAA type.
Alias record set	No	This setting can only be applied to A, AAAA, and CNAME record types.
TTL	1	The time to live, which specifies the period of time each DNS server caches the resolution before being purged.
TTL unit	Hours	This value can be seconds, minutes, hours, days, or weeks. Here, you're selecting hours.
IP Address	10.10.10.10	The IP address the record name resolves to. In a real-world scenario, you'd enter the public IP address for your web server.

5. Select **Add** to add the record to your zone.

worldwideimportsXXXX.com | Recordsets ☆ ...

DNS zone

Search

+ Add Refresh Delete

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
DNS Management
Recordsets
Monitoring
Automation
Help

A record set is a collection of records in a zone that have the same name and are the same type. You can search for record sets that have been loaded on this page. If you don't see what you're looking for, you can try scrolling to allow more record sets to load. [Learn more](#)

Search

Fetches 3 record set(s).

Name	Type	TTL	Value	Alias resource type	Alias target
@	NS	172800	ns1-01.azure-dns.com. ns2-01.azure-dns.net. ns3-01.azure-dns.org. ns4-01.azure-dns.info.		
@	SOA	3600	Email: azuredns-hostmaster.microsoft.com Host: ns1-01.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1		
www	A	3600	10.10.10.10		

It's possible to have more than one IP address set up for your web server. In that case, you add all the associated IP addresses as records in the A record set. After the record set is created, you can update it with more IP addresses.

Verify your global Azure DNS

In a real-world scenario, after you create the public DNS zone, you update the NS records of the domain-name registrar to delegate the domain to Azure.

Even though we don't have a registered domain, it's still possible to verify that the DNS zone works as expected by using the `nslookup` tool.

Use nslookup to verify the configuration

Here's how to use `nslookup` to verify the DNS zone configuration.

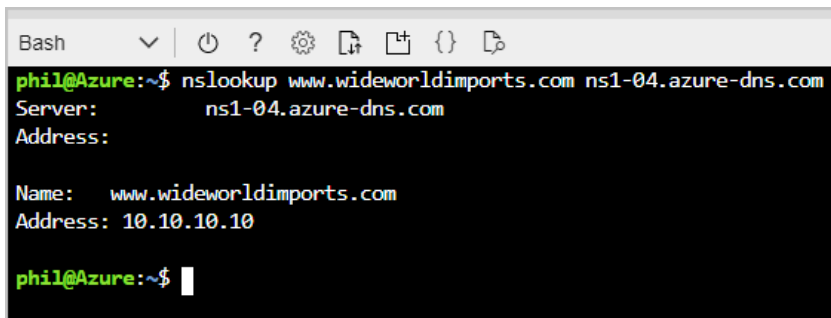
1. Use Cloud Shell to run the following command. Replace the DNS zone name with the zone you created, and replace `<name server address>` with one of the NS values you copied after you created the DNS zone.

```
nslookup www.wideworldimportsXXXX.com <name server address>
```

The command should look something like the following example:

```
nslookup www.wideworldimportsXXXX.com ns1-04.azure-dns.com
```

2. You should see that your host name `www.wideworldimportsXXXX.com` resolves to 10.10.10.10.



```
Bash
phil@Azure:~$ nslookup www.wideworldimports.com ns1-04.azure-dns.com
Server:      ns1-04.azure-dns.com
Address:

Name:   www.wideworldimports.com
Address: 10.10.10.10

phil@Azure:~$
```

Congratulations! You successfully set up a DNS zone and created an A record.

5. Dynamically resolve resource name by using alias record

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/5-resolve-name-alias-record>

Dynamically resolve resource name by using alias record

Completed

In the previous exercise, you successfully delegated the domain from the domain registrar to your Azure DNS and configured an A record to link the domain to your web server.

The next phase of the deployment is to improve resiliency by using a load balancer. Load balancers distribute inbound data requests and traffic across one or more servers. They reduce the load on any one server and improve performance. This technology is well established. You can use it throughout your on-premises network.

You know that the A record and CNAME record don't support direct connection to Azure resources like your load balancers. You're tasked with finding out how to link the apex domain with a load balancer.

What is an apex domain?

The apex domain is your domain's highest level. In our case, that's `wideworldimports.com`. The apex domain is also sometimes referred to as the *zone apex* or *root apex*. The `@` symbol often represents the apex domain in your DNS zone records.

If you check the DNS zone for `wideworldimports.com`, you see that there are two apex domain records: NS and SOA. The NS and SOA records are automatically created when you created the DNS zone.

CNAME records that you might need for an Azure Traffic Manager profile or Azure Content Delivery Network endpoints aren't supported at the zone apex level. However, other *alias records* are supported at the zone apex level.

What are alias records?

Azure alias records enable a zone apex domain to reference other Azure resources from the DNS zone. You don't need to create complex redirection policies. You can also use an Azure alias to route all traffic through Traffic Manager.

The Azure alias record can point to the following Azure resources:

- A Traffic Manager profile
- Azure Content Delivery Network endpoints
- A public IP resource
- A front-door profile

Alias records provide lifecycle tracking of target resources, ensuring that changes to any target resource are automatically applied to the DNS zone. Alias records also provide support for load-balanced applications in the zone apex.

The alias record set supports the following DNS zone record types:

- **A:** The IPv4 domain name-mapping record.
- **AAAA:** The IPv6 domain name-mapping record.
- **CNAME:** The alias for your domain, which links to the A record.

Uses for alias records

The following are some of the advantages of using alias records:

- **Prevents dangling DNS records:** A dangling DNS record occurs when the DNS zone records aren't up to date with changes to IP addresses. Alias records prevent dangling references by tightly coupling the lifecycle of a DNS record with an Azure resource.
- **Updates DNS record set automatically when IP addresses change:** When the underlying IP address of a resource, service, or application is changed, the alias record ensures that any associated DNS records are automatically refreshed.
- **Hosts load-balanced applications at the zone apex:** Alias records allow for zone apex resource routing to Traffic Manager.
- **Points zone apex to Azure Content Delivery Network endpoints:** With alias records, you can now directly reference your Azure Content Delivery Network instance.

An alias record allows you to link the zone apex (`wideworldimports.com`) to a load balancer. It creates a link to the Azure resource rather than a direct IP-based connection. So, if the IP address of your load balancer changes, the zone apex record continues to work.

6. Exercise - Create alias records for Azure DNS

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/6-exercise-create-alias-records>

Exercise - Create alias records for Azure DNS

Completed

Your new website's deployment was a huge success. Usage volumes are higher than anticipated. The single web server on which the website runs is showing signs of strain. Your organization wants to increase the number of servers and distribute the load using a load balancer.

You now know you can use an Azure alias record to provide a dynamic, automatically refreshing link between the zone apex and the load balancer.

In this unit, you'll:

- Set up a virtual network with two VMs and a load balancer.
- Learn how to configure an Azure alias at the zone apex to direct to the load balancer.
- Verify that the domain name resolves to one or either of the VMs on your virtual network.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Set up a virtual network, load balancer, and VMs in Azure

When you manually create a virtual network, load balancer, and two VMs, it takes some time. To reduce this time, you can use a Bash setup script that's available on GitHub. Follow these instructions to create a test environment for your alias record.

1. In Azure Cloud Shell, run the following setup script:

```
git clone https://github.com/MicrosoftDocs/mslearn-host-domain-azure-dns.git
```

2. To run the setup script, run the following commands:

```
cd mslearn-host-domain-azure-dns
chmod +x setup.sh
./setup.sh
```

The setup script takes a few minutes to run. The script:

- Creates a network security group.
- Creates two network interface controllers (NICs) and two VMs.
- Creates a virtual network and assigns the VMs.
- Creates a public IP address and updates the configuration of the VMs.
- Creates a load balancer that references the VMs, including rules for the load balancer.
- Links the NICs to the load balancer.

After the script completes, it shows you the public IP address for the load balancer. Copy the IP address to use it later.

Create an alias record in your zone apex

Now that you created a test environment, you're ready to set up the Azure alias record in your zone apex.

1. In the [Azure portal](#), select **Resource groups**. The **Resource groups** pane appears.
2. Select your resource group. The **Resource group** pane appears.
3. In the list of resources, select the DNS zone you created in the previous exercise, `wideworldimportsXXXX.com`. The **wideworldimportsXXXX.com DNS zone** pane appears.
4. In the menu bar, select **+ Record set**. The **Add record set** pane appears.
5. Enter the following values for each setting to create an alias record.

Setting	Value
Name	Leave the name blank. Blank indicates the DNS zone for <code>wideworldimportsXXXX.com</code> .
Type	A. Even though we're creating an alias, the base record type must still be either A, AAAA, or CNAME.
Alias record set	Yes
Alias type	Azure resource
Azure resource	From the list of resources, select myPublicIP . It can take up to 15 minutes for the deployments to propagate. If this resource isn't listed, wait several minutes, refresh the portal, and try again.

Add record set

×

worldwideimportsXXXX.com

Name

.worldwideimportsXXXX.com

Type

A – Address record

Alias record set ⓘ

Yes

Alias type

Azure resource

Choose a subscription *

Concierge Subscription

Azure resource *

myPublicIP

TTL *

1

TTL unit

Hours

Add

Cancel

6. Select **OK** to add the record to your zone.

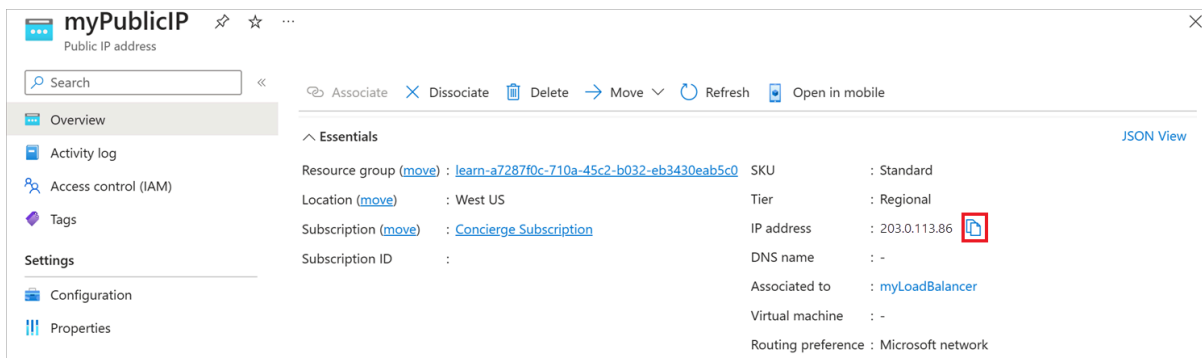
When the new alias record is created, it should look something like this:

NAME	TYPE	TTL	VALUE	ALIAS RESOURCE TYPE	ALIAS TARGET
@	A	3600	-	Public IP Address	myPublicIP

Verify that the alias resolves to the load balancer

Now, you need to verify that the alias record is set up correctly. In a real-world scenario, you'd have an actual domain, and would complete the domain delegation to Azure DNS. You'd use the registered domain name for this exercise. Because this unit assumes there's no registered domain, you use the public IP address.

1. In the Azure portal, go to the resource group, select **myPublicIP**, then select the **Copy** icon next to the IP address.



2. In a web browser, paste the Public IP address as the URL.
3. You see a basic web page that shows the name of the virtual machine (VM) to which the load balancer sent the request.

7. Summary

<https://learn.microsoft.com/en-us/training/modules/host-domain-azure-dns/7-summary>

Summary

Completed

Your company recently bought the custom domain name `wideworldimporters.com` from a third-party domain-name registrar. The domain name is for a new website your organization plans to launch. You need a hosting service for DNS domains. This hosting service would resolve the `wideworldimporters.com` domain to your Azure-based web server's IP address.

Your company wanted to manage all their infrastructure and related domain name information in one place. You saw how easy it was to manage Domain Name System (DNS) information by using an Azure DNS zone. First, you created an Azure DNS zone, and then you updated the NS records at your domain registrar to point at it.

You learned the uses of the different record sets, A, AAAA, CNAME, NS, and SOA. You also learned how you can use Azure aliases to override the static A/AAAA/CNAME record to provide a dynamic reference to your resources. Using an Azure DNS zone improved your company's administration of resources, because your staff only needed one place to manage DNS-related tasks.

The Azure DNS zone allows better control and integration with your Azure resources. It's possible to achieve some of the more basic record set functions by using the domain registrar's management console. However, linking to any of your Azure resources becomes difficult or impossible without a high degree of complex redirection.

By using an Azure DNS zone to host your domain, your organization benefits by having all the resources managed through a single, common interface. This solution provides better integration with existing Azure resources, improved security, and monitoring tools.

Important

In the optional exercises for this module, you created resources by using your own Azure subscription. Clean up these resources so that you won't continue to be charged for them.

Learn more

- [Quickstart: Create an Azure private DNS zone by using the Azure portal](#)
- [Overview of DNS zones and records](#)

Configure Azure Virtual Network peering

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/1-introduction>

Introduction

Completed

Azure Virtual Network peering lets you connect virtual networks in the same or different regions. Azure Virtual Network peering provides secure communication between resources in the peered networks.

Suppose your engineering company is migrating services to Azure. The company is deploying services into separate Azure virtual networks. Private connectivity between the virtual networks isn't yet configured. Several business units identified services in the virtual networks that need to communicate with each other.

You're responsible for implementing an Azure Virtual Network peering solution and enabling connectivity between the virtual networks. Two of your strategy goals include preventing exposure of the services to the internet, and keeping the integration as simple as possible. Your solution should address transit and connectivity concerns.

The goal of this module is to successfully implement Azure Virtual Network peering.

Learning objectives

In this module, you learn how to:

- Identify usage cases and product features of Azure Virtual Network peering.
- Configure your network to implement Azure VPN Gateway for transit connectivity.
- Extend peering by using a hub and spoke network with user-defined routes and service chaining.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Basic understanding of cloud networking including virtual networks and virtual machines.
- Familiarity with the command line connectivity testing tools.

2. Determine Azure Virtual Network peering uses

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/2-determine-uses>

Determine Azure Virtual Network peering uses

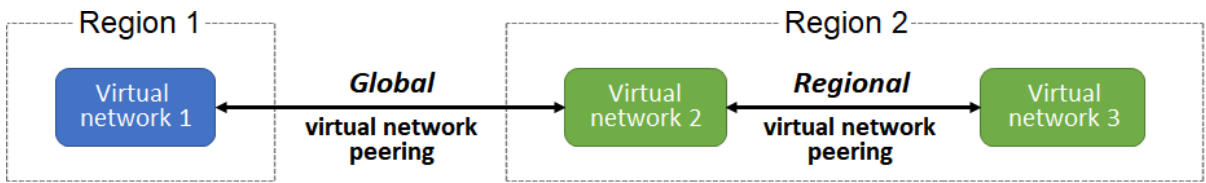
Completed

Perhaps the simplest and quickest way to connect your virtual networks is to use Azure Virtual Network peering. Virtual Network peering enables you to seamlessly connect two Azure virtual networks. After the networks are peered, the two virtual networks operate as a single network, for connectivity purposes.

Things to know about Azure Virtual Network peering

Let's examine some prominent characteristics of Azure Virtual Network peering.

- There are two types of Azure Virtual Network peering: *regional* and *global*.



- **Regional virtual network peering** connects Azure virtual networks that exist in the same region.
- **Global virtual network peering** connects Azure virtual networks that exist in different regions.
- You can create a regional peering of virtual networks in the same Azure public cloud region, or in the same China cloud region, or in the same Microsoft Azure Government cloud region.
- You can create a global peering of virtual networks in any Azure public cloud region, or in any China cloud region.
- Global peering of virtual networks in different Azure Government cloud regions isn't permitted.
- After you create a peering between virtual networks, the individual virtual networks are still managed as separate resources.
- Virtual networks can be peered across subscriptions and tenants.

Things to consider when using Azure Virtual Network peering

Consider the benefits of using Azure Virtual Network peering.

Benefit	Description
Private network connections	When you implement Azure Virtual Network peering, network traffic between peered virtual networks is private. Traffic between the virtual networks is kept on the Microsoft Azure backbone network. No public internet, gateways, or encryption is required in the communication between the virtual networks.
Strong performance	Because Azure Virtual Network peering utilizes the Azure infrastructure, you gain a low-latency, high-bandwidth connection between resources in different virtual networks.
Simplified communication	Azure Virtual Network peering lets resources in one virtual network communicate with resources in a different virtual network, after the virtual networks are peered.
Seamless data transfer	You can create an Azure Virtual Network peering configuration to transfer data across Azure subscriptions, deployment models, and across Azure regions.
No resource disruptions	Azure Virtual Network peering doesn't require downtime for resources in either virtual network when creating the peering, or after the peering is created.

Things to know about peering requirements and limitations

While VNet peering provides many benefits, there are important constraints to understand.

Requirements/Limitations	Description
Nonoverlapping address spaces	Peered virtual networks must have non-overlapping IP address spaces. Peering creation fails if address ranges overlap.
Address space modification restrictions	If you want to change a VNet's address range, you need to delete the peering first, update the address space, and then set up the peering again.
Basic Load Balancer limitations	Resources in one VNet can't communicate with Basic Internal Load Balancer IPs in VNets peered across regions. Use the Standard Load Balancer for cross-region connections.
DNS resolution boundaries	Azure's built-in name resolution does not work across peered VNets. Configure Azure Private DNS zones or custom DNS servers for cross-VNet name resolution.

3. Determine gateway transit and connectivity

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/3-determine-gateway-transit-connectivity>

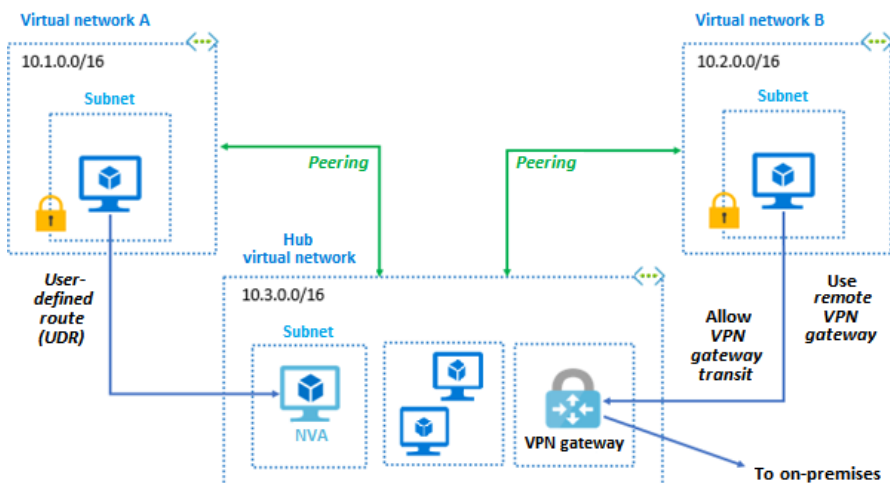
Determine gateway transit and connectivity

Completed

When virtual networks are peered, you can configure Azure VPN Gateway in the peered virtual network as a *transit point*. In this scenario, a peered virtual network uses the remote VPN gateway to gain access to other resources.

Transit and connectivity use

Consider a scenario where three virtual networks in the same region are connected by virtual network peering. Virtual network A and virtual network B are each peered with a hub virtual network. The hub virtual network contains several resources, including a gateway subnet and an Azure VPN gateway. The VPN gateway is configured to allow VPN gateway transit. Virtual network B accesses resources in the hub, including the gateway subnet, by using a remote VPN gateway.



The Azure portal provides four key settings when configuring virtual network peering.

Add peering

...



vnet-1

Virtual network peering enables you to seamlessly connect two or more virtual networks in Azure. This will allow resources in either virtual network to directly connect and communicate with resources in the peered virtual network.

Remote virtual network summary

Peering link name *

vnet-2-to-vnet-1

Remote virtual network peering settings

Allow 'vnet-2' to access 'vnet-1' ⓘ



Allow 'vnet-2' to receive forwarded traffic from 'vnet-1' ⓘ



Allow gateway or route server in 'vnet-2' to forward traffic to 'vnet-1' ⓘ



Enable 'vnet-2' to use 'vnet-1's' remote gateway or route server ⓘ



- **Traffic to remote virtual network.** Controls whether traffic flows from this VNet to the remote VNet.
- **Traffic forwarded from remote virtual network.** Controls whether forwarded (non-originating) traffic is accepted from the peered VNet.
- **Virtual network gateway or Route Server.** Enables gateway transit. Lets peered VNets use this VNet's VPN Gateway or Azure Route Server.
- **Remote virtual network gateway or Route Server.** Enables this VNet to use the remote VNet's VPN Gateway or Route Server.

Things to know about Azure VPN Gateway

Let's take a closer look at how Azure VPN Gateway is implemented with Azure Virtual Network peering.

- A virtual network can have only one VPN gateway.
- Gateway transit is supported for both regional and global virtual network peering.
- When you allow VPN gateway transit, the virtual network can communicate to resources outside the peering. In our sample illustration, the gateway subnet gateway within the hub virtual network can complete tasks such as:
 - 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.
- Gateway transit allows peered virtual networks to share the gateway and get access to external resources. With this implementation, you don't need to deploy a VPN gateway in the peered virtual network.
- You can apply network security groups in a virtual network to block or allow access to other virtual networks or subnets. When you configure virtual network peering, you can choose to open or close the network security group rules between the virtual networks.

4. Create virtual network peering

Create virtual network peering

Completed

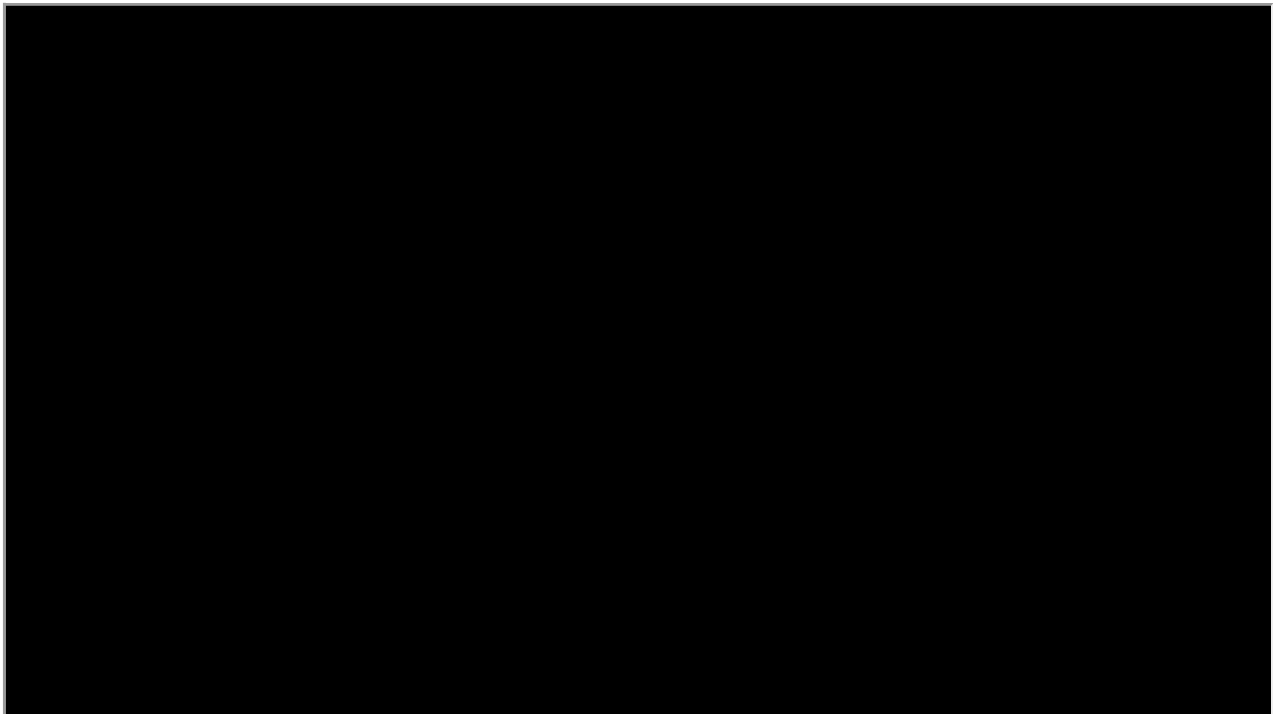
Azure Virtual Network peering can be configured for virtual networks by using PowerShell, the Azure CLI, and in the Azure portal. In this module, we review the steps to create the peering in the Azure portal for virtual networks deployed through Azure Resource Manager.

Things to know about creating virtual network peering

There are a few points to review before we look at how to create the peering in the Azure portal.

- To implement virtual network peering, your Azure account must be assigned to the **Network Contributor** role. Alternatively, your Azure account can be assigned to a custom role that can complete the necessary peering actions. For details, see [Permissions](#).
- To create a peering, you need two virtual networks.
- The second virtual network in the peering is referred to as the *remote network*.
- Initially, the virtual machines in your virtual networks can't communicate with each other. After the peering is established, the machines can communicate within the peered network based on your configuration settings.

How to connect virtual networks across Azure regions with Azure Global virtual network peering



How to check your peering status

In the Azure portal, you can check the connectivity status of the virtual networks in your virtual network peering. The status conditions depend on how your virtual networks are deployed.

Important

Your peering isn't successfully established until both virtual networks in the peering have a status of **Connected**.

- The two peering status conditions are **Initiated** and **Connected**.
- When you create the initial peering to the second (remote) virtual network from the first virtual network, the peering status for the first virtual network is **Initiated**.

5. Extend peering with user-defined routes and service chaining

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/5-determine-service-chaining-uses>

Extend peering with user-defined routes and service chaining

Completed

Virtual network peering is nontransitive. The communication capabilities in a peering are available to only the virtual networks and resources in the peering. Other mechanisms have to be used to enable traffic to and from resources and networks outside the private peering network.

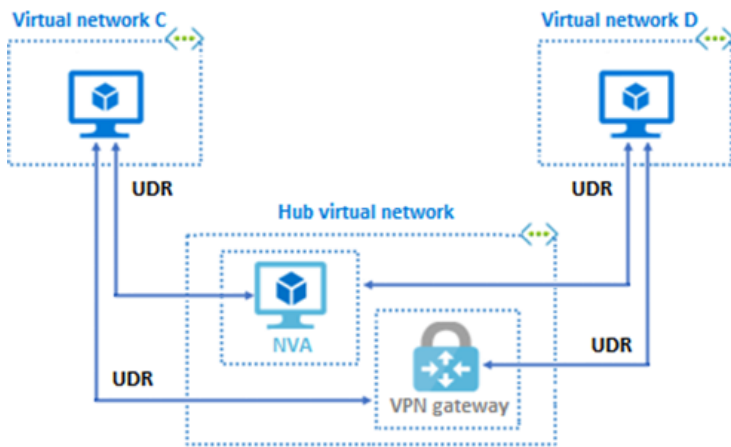
Suppose you have three virtual networks: A, B, and C. You establish virtual network peering between networks A and B, and also between networks B and C. You don't set up peering between networks A and C. The virtual network peering capabilities that you set up between networks B and C don't automatically enable peering communication capabilities between networks A and C.

Things to know about extending peering

There are a few ways to extend the capabilities of your peering for resources and virtual networks outside your peering network.

Mechanism	Description
Hub and spoke network	When you deploy a hub-and-spoke network, the hub virtual network can host infrastructure components like a network virtual appliance (NVA) or Azure VPN gateway. All the spoke virtual networks can then peer with the hub virtual network. Traffic can flow through NVAs or VPN gateways in the hub virtual network.
User-defined route (UDR)	Virtual network peering enables the next hop in a user-defined route to be the IP address of a virtual machine in the peered virtual network, or a VPN gateway.
Service chaining	Service chaining is used to direct traffic from one virtual network to a virtual appliance or gateway. To enable service chaining, configure UDRs that point to virtual machines in peered virtual networks as the next hop IP address. UDRs could also point to virtual network gateways to enable service chaining.
Azure Virtual Network Manager	Centrally manages hub-and-spoke or mesh peering topologies at scale. Automates peering creation without manual per-VNet configuration.

The following diagram shows a hub and spoke virtual network with an NVA and VPN gateway. The hub and spoke network is accessible to other virtual networks via user-defined routes and service chaining.



Tip

Use the **Ask Learn** icon (top right) to learn more about *service chaining and user-defined routes*.

6. Exercise - Implement Intersite Connectivity

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/6-simulation-peering>

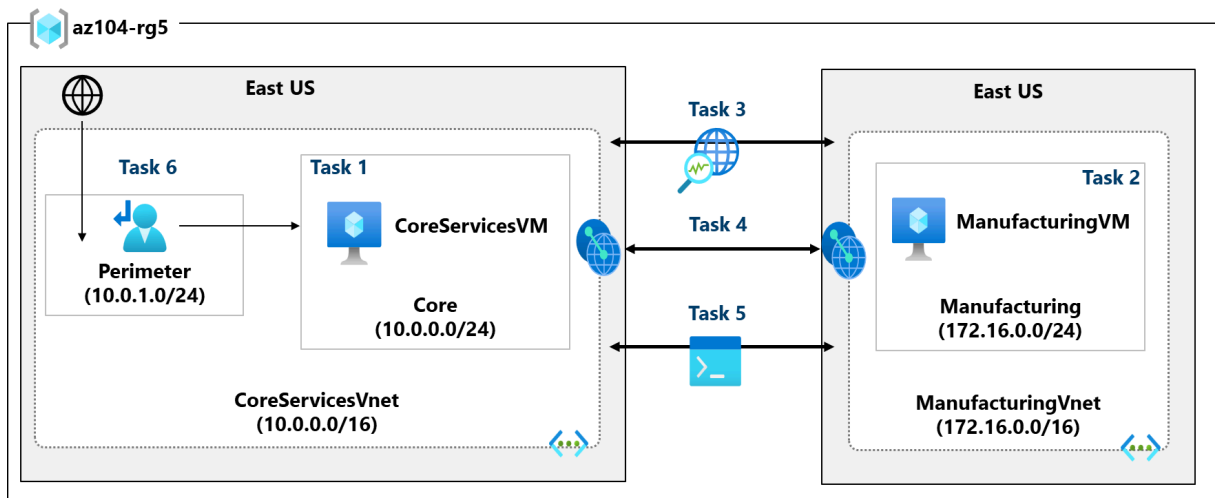
Exercise - Implement Intersite Connectivity

Completed

Lab scenario

Your organization segments core IT apps and services (such as DNS and security services) from other parts of the business, including your manufacturing department. However, in some scenarios, apps and services in the core area need to communicate with apps and services in the manufacturing area. In this lab, you configure connectivity between the segmented areas. Separating production from development or separating one subsidiary from another is a common networking scenario.

Architecture diagram



Job skills

- Create a virtual machine in a virtual network.
- Create a virtual machine in a different virtual network.
- Use Network Watcher to test the connection between virtual machines.
- Configure virtual network peerings between different virtual networks.
- Use Azure PowerShell to test the connection between virtual machines.
- Create a custom route. (optional)

Important

Estimated time: 50 minutes. To complete the exercise, you need an [Azure subscription](#).

Launch the exercise, and follow the instructions. When finished, be sure to return to this page so you can continue learning.

Launch Exercise

7. Module assessment

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/7-knowledge-check>

Module assessment

Completed

8. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-vnet-peering/8-summary-resources>

Summary and resources

Completed

In this module, you learned Azure Virtual Network peering lets you connect virtual networks in a hub and spoke topology. You learned how to configure your virtual networks with Azure VPN Gateway for transit connectivity. You explored how to extend peering with user-defined routes and service chaining.

The main takeaways from this module are:

- Azure Virtual Network peering allows for the connection of virtual networks in a hub and spoke topology.
- There are two types of peering: regional and global. Regional peering connects virtual networks in the same region. Global peering connects virtual networks in different regions.
- Network traffic between peered virtual networks is private and kept on the Azure backbone network.
- You can configure Azure VPN Gateway in the peered virtual network as a transit point to access resources in another network.
- Network security groups can be applied to block or allow access between virtual networks when configuring virtual network peering.

Learn more with Copilot

Copilot can assist you in designing Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What is Azure virtual network peering and what are the advantages of this feature?
- What are some of the configurations settings for Azure virtual network peering?

Learn more with documentation

- [Azure Virtual Network peering](#). This article is your starting point for learning about virtual network peering.
- [Create, change, or delete a virtual network peering](#). This article reviews how to create a virtual network peering and what each setting means.

Learn more with self-paced training

- [Introduction to Azure Virtual Networks](#). Learn how to design and implement core Azure networking infrastructure such as virtual networks, and virtual network peering.

Manage and control traffic flow in your Azure deployment with routes

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/1-introduction>

Introduction

Completed

A virtual network lets you implement a security perimeter around your resources in the cloud. You can control the information that flows in and out of a virtual network. You can also restrict access to allow only the traffic that originates from trusted sources.

Suppose that you're the solution architect for a retail organization. Also suppose that your organization recently suffered a security incident that exposed customer information such as names, addresses, and credit card numbers. Malicious actors infiltrated vulnerabilities in your retailer's network infrastructure, which resulted in the loss of customers' confidential information.

As part of a remediation plan, the security team recommends adding network protections in the form of network virtual appliances. The cloud infrastructure team must ensure traffic gets properly routed through the virtual appliances and gets inspected for malicious activity.

In this module, you'll learn about Azure routing, and you'll create custom routes to control the traffic flow. You'll also learn to redirect the traffic through the network virtual appliance so you can inspect the traffic before it's allowed through.

Learning objectives

In this module, you'll:

- Identify the routing capabilities of an Azure virtual network.
- Configure routing within a virtual network.
- Deploy a basic network virtual appliance.
- Configure routing to send traffic through a network virtual appliance.

Prerequisites

- Knowledge of basic networking concepts, including subnets and IP addressing
- Familiarity with Azure virtual networking

2. Identify routing capabilities of an Azure virtual network

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/2-azure-virtual-network-route>

Identify routing capabilities of an Azure virtual network

Completed

3. Exercise - Create custom routes

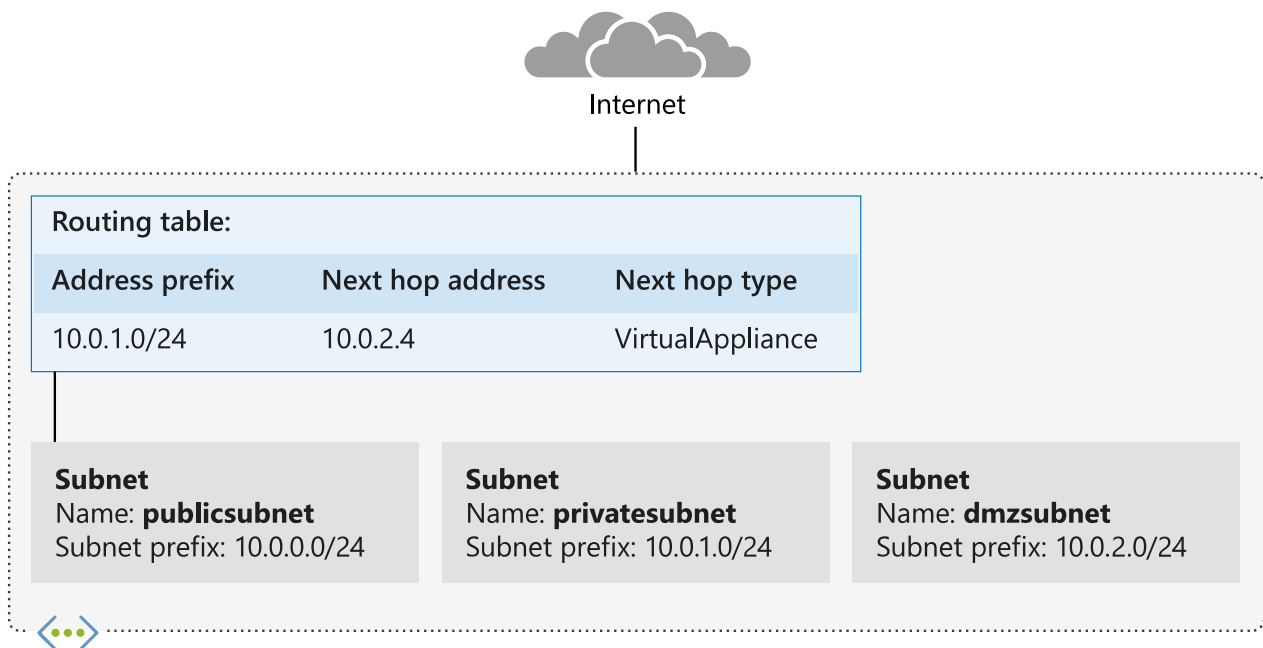
Exercise - Create custom routes

Completed

As you implement your security strategy, you want to control how network traffic is routed across your Azure infrastructure.

In the following exercise, you use a network virtual appliance (NVA) to help secure and monitor traffic. You want to ensure communication between front-end public servers and internal private servers is always routed through the appliance.

You configure the network so that all traffic flowing from a public subnet to a private subnet will be routed through the NVA. To make this flow happen, you create a custom route for the public subnet to route this traffic to a perimeter-network subnet. Later, you deploy an NVA to the perimeter-network subnet.



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

In this exercise, you create the route table, custom route, and subnets. You'll then associate the route table with a subnet.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Note

You need to use a resource group to complete the steps in this exercise. You can use a resource group that you already created, or you can create a new resource group specifically for this exercise. If you choose to create a new resource group, that will make it easier to clean up any resources that you create as you complete the exercise. If you don't have an existing resource group or you want to create a new one specifically for this exercise, you can follow the steps in [Use the Azure portal and Azure Resource Manager to manage resource groups](#) to create a resource group by using the Azure portal, or you can follow the steps in [Manage Azure resource groups by using Azure CLI](#) to create a resource group by using the the Azure CLI.

Note

Throughout this exercise, replace **myResourceGroupName** in the examples with the name of an existing resource group, or the name of the resource group that you created for this exercise.

Create a route table and custom route

The first task is to create a new routing table and then add a custom route for all traffic intended for the private subnet.

Note

You might get an error that reads: *This command is implicitly deprecated.* Please ignore this error for this learning module. We're working on it!

1. Open the [Azure Cloud Shell](#), select **Settings**, then select **Go to Classic version**.
2. In Azure Cloud Shell, run the following command to create a route table. Replace **myResourceGroupName** with the name of your resource group.

```
az network route-table create \  
  --name publictable \  
  --resource-group "myResourceGroupName" \  
  --disable-bgp-route-propagation false
```

3. Run the following command in Cloud Shell to create a custom route:

```
az network route-table route create \  
  --route-table-name publictable \  
  --resource-group "myResourceGroupName" \  
  --name productionsubnet \  
  --address-prefix 10.0.1.0/24 \  
  --next-hop-type VirtualAppliance \  
  --next-hop-ip-address 10.0.2.4
```

Create a virtual network and subnets

The next task is to create the **vnet** virtual network and the three subnets you need: **publicsubnet**, **privatesubnet**, and **dmzsubnet**.

1. Run the following command to create the **vnet** virtual network and the **publicsubnet** subnet:

```
az network vnet create \  
  --name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.0.0/16 \  
  --subnet-name publicsubnet \  
  --subnet-prefixes 10.0.0.0/24
```

2. Run the following command in Cloud Shell to create the **privatesubnet** subnet:

```
az network vnet subnet create \  
  --name privatesubnet \  
  --vnet-name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.1.0/24
```

3. Run the following command to create the **dmzsubnet** subnet:

```
az network vnet subnet create \  
  --name dmzsubnet \  
  --vnet-name vnet \  
  --resource-group "myResourceGroupName" \  
  --address-prefixes 10.0.2.0/24
```

4. You should now have three subnets. Run the following command to show all of the subnets in the **vnet** virtual network:

```
az network vnet subnet list \
  --resource-group "myResourceGroupName" \
  --vnet-name vnet \
  --output table
```

Associate the route table with the public subnet

The final task in this exercise is to associate the route table with the **publicsubnet** subnet.

Run the following command to associate the route table with the public subnet.

```
az network vnet subnet update \
  --name publicsubnet \
  --vnet-name vnet \
  --resource-group "myResourceGroupName" \
  --route-table publictable
```

4. What is an NVA?

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/4-network-virtual-appliances>

What is an NVA?

Completed

5. Exercise - Create an NVA and virtual machines

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/5-exercise-create-nva-vm>

Exercise - Create an NVA and virtual machines

Completed

In the next stage of your security implementation, you'll deploy a network virtual appliance (NVA) to secure and monitor traffic between your front-end public servers and internal private servers.

First, configure the appliance to forward IP traffic. If IP forwarding isn't enabled, traffic that is routed through your appliance will never be received by its intended destination servers.

In this exercise, you deploy the **nva** network appliance to the **dmzsubnet** subnet. Then you enable IP forwarding so that traffic from * and traffic that uses the custom route is sent to the **privatesubnet** subnet.



Internet

Routing table:

Address prefix	Next hop address	Next hop type
10.0.1.0/24	10.0.2.4	VirtualAppliance

VM: NVA
P address: 10.0.2.4
IP forwarding **ON**

Subnet

Name: **publicsubnet**
Subnet prefix: 10.0.0.0/24

Subnet

Name: **privatesubnet**
Subnet prefix: 10.0.1.0/24

Subnet

Name: **dmzsubnet**
Subnet prefix: 10.0.2.0/24



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

In the following steps, you'll deploy an NVA. You'll then update the Azure virtual NIC and the network settings within the appliance to enable IP forwarding.

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Deploy the network virtual appliance

To build the NVA, deploy an Ubuntu LTS instance.

1. In [Azure Cloud Shell](#), run the following command to deploy the appliance. Replace `<password>` with a suitable password of your choice for the **azureuser** admin account, and **myResourceGroupName** with the name of your resource group.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name nva \
  --vnet-name vnet \
  --subnet dmzsubnet \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --admin-password <password>
```

Enable IP forwarding for the Azure network interface

In the next steps, you enable IP forwarding for the **nva** network appliance. When traffic flows to the NVA but is meant for another target, the NVA will route that traffic to its correct destination.

1. Run the following command to get the NVA network interface's ID:

```
NICID=$(az vm nic list \
  --resource-group "myResourceGroupName" \
  --vm-name nva \
  --query "[0].id" --output tsv)
```

```
echo $NICID
```

2. Run the following command to get the NVA network interface's name:

```
NICNAME=$(az vm nic show \
  --resource-group "myResourceGroupName" \
  --vm-name nva \
  --nic $NICID \
  --query "{name:name}" --output tsv)

echo $NICNAME
```

3. Run the following command to enable IP forwarding for the network interface:

```
az network nic update --name $NICNAME \
  --resource-group "myResourceGroupName" \
  --ip-forwarding true
```

Enable IP forwarding in the appliance

1. Run the following command to save the NVA virtual machine's public IP address to the variable `NVAIP`:

```
NVAIP="$(az vm list-ip-addresses \
  --resource-group "myResourceGroupName" \
  --name nva \
  --query "[].virtualMachine.network.publicIpAddresses[*].ipAddress" \
  --output tsv)"

echo $NVAIP
```

2. Run the following command to enable IP forwarding within the NVA:

```
ssh -t -o StrictHostKeyChecking=no azureuser@$NVAIP 'sudo sysctl -w net.ipv4.ip_forward=1; exit;'
```

When prompted, enter the password you used when you created the virtual machine.

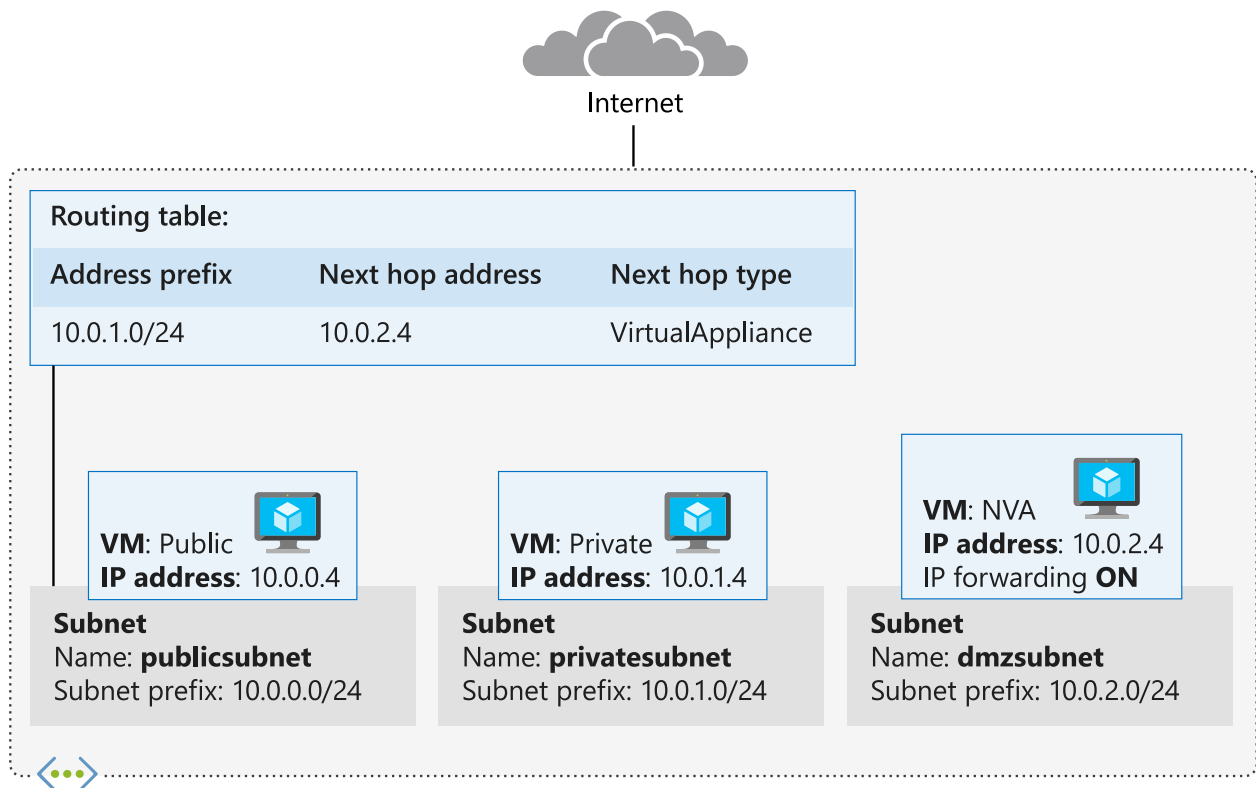
6. Exercise - Route traffic through the NVA

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/6-exercise-route-traffic-through-nva>

Exercise - Route traffic through the NVA

Completed

Now that you've created the network virtual appliance (NVA) and virtual machines (VMs), you'll route the traffic through the NVA.



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

Note

This exercise is optional. If you want to complete this exercise, you'll need to create an Azure subscription before you begin. If you don't have an Azure account or you don't want to create one at this time, you can read through the instructions so you understand the information that's being presented.

Create public and private virtual machines

The next steps deploy a VM into the public and private subnets.

1. Open the [Azure Cloud Shell](#) editor and create a file named **cloud-init.txt**.

```
code cloud-init.txt
```

2. Add the following configuration information to the file. With this configuration, the `inetutils-traceroute` package is installed when you create a new VM. This package contains the `traceroute` utility that you'll use later in this exercise.

```
#cloud-config
package_upgrade: true
packages:
  - inetutils-traceroute
```

3. Press **Ctrl+S** to save the file, and then press **Ctrl+Q** to close the editor.
4. In Cloud Shell, run the following command to create the **public** VM. Replace `<password>` with a suitable password for the **azureuser** account.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name public \
  --vnet-name vnet \
```

```
--subnet publicsubnet \
--image Ubuntu2204 \
--admin-username azureuser \
--no-wait \
--custom-data cloud-init.txt \
--admin-password <password>
```

5. Run the following command to create the **private** VM. Replace `<password>` with a suitable password.

```
az vm create \
  --resource-group "myResourceGroupName" \
  --name private \
  --vnet-name vnet \
  --subnet privatesubnet \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --no-wait \
  --custom-data cloud-init.txt \
  --admin-password <password>
```

6. Run the following Linux `watch` command to check that the VMs are running. The `watch` command periodically runs the `az vm list` command so that you can monitor the progress of the VMs.

```
watch -d -n 5 "az vm list \
  --resource-group "myResourceGroupName" \
  --show-details \
  --query '[*].{Name:name, ProvisioningState:provisioningState, PowerState:powerState}' \
  --output table"
```

A **ProvisioningState** value of "Succeeded" and a **PowerState** value of "VM running" indicate a successful deployment. When all three VMs are running, you're ready to move on. Press Ctrl-C to stop the command and continue with the exercise.

7. Run the following command to save the public IP address of the **public** VM to a variable named `PUBLICIP`:

```
PUBLICIP="$(az vm list-ip-addresses \
  --resource-group "myResourceGroupName" \
  --name public \
  --query "[].virtualMachine.network.publicIpAddresses[*].ipAddress" \
  --output tsv)"

echo $PUBLICIP
```

8. Run the following command to save the public IP address of the **private** VM to a variable named `PRIVATEIP`:

```
PRIVATEIP="$(az vm list-ip-addresses \
  --resource-group "myResourceGroupName" \
  --name private \
  --query "[].virtualMachine.network.publicIpAddresses[*].ipAddress" \
  --output tsv)"

echo $PRIVATEIP
```

Test traffic routing through the network virtual appliance

The final steps use the Linux `traceroute` utility to show how traffic is routed. You'll use the `ssh` command to run `traceroute` on each VM. The first test shows the route taken by ICMP packets sent from the **public** VM to the **private** VM. The second test shows the route taken by ICMP packets sent from the **private** VM to the **public** VM.

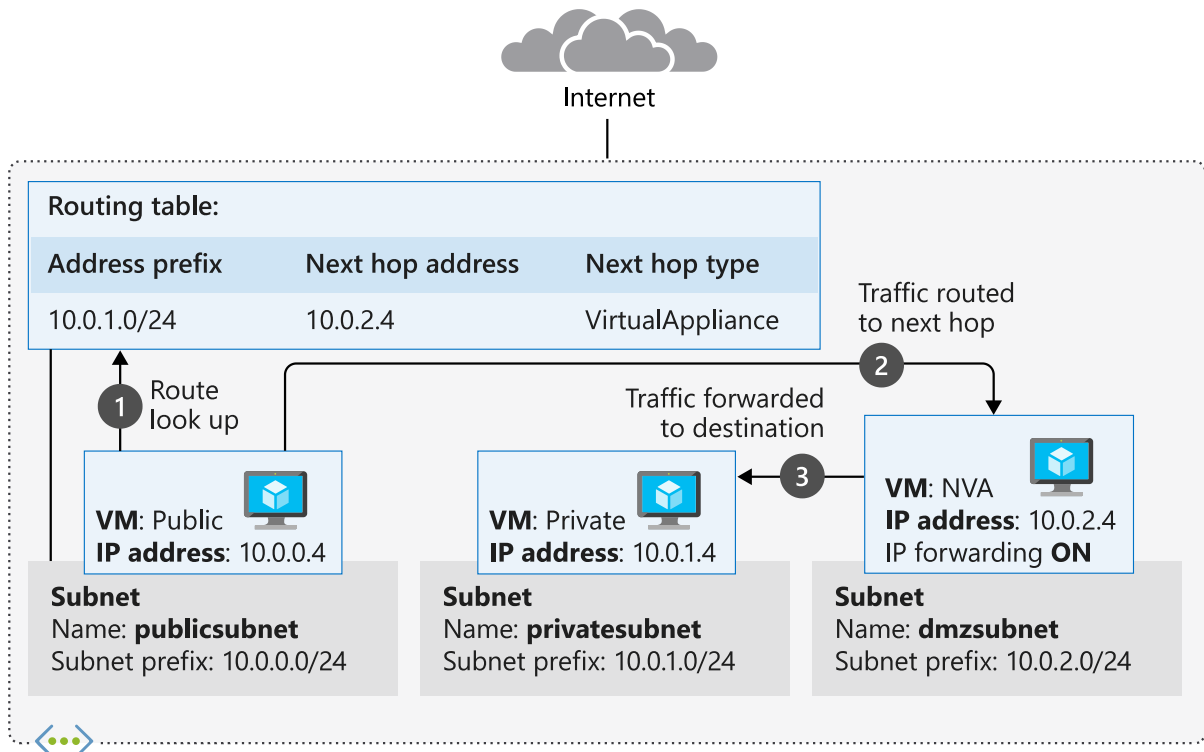
1. Run the following command to trace the route from **public** to **private**. When prompted, enter the password for the **azureuser** account that you specified earlier.

```
ssh -t -o StrictHostKeyChecking=no azureuser@$PUBLICIP 'traceroute private --type=icmp; exit'
```

If you receive the error message `bash: traceroute: command not found`, wait a minute and retry the command. The automated installation of `traceroute` can take a minute or two after VM deployment. After the command succeeds, the output should look similar to the following example:

```
tracert to private.kzffavtrkpeulburui2lgywxwg.gx.internal.cloudapp.net (10.0.1.4), 64 hops max
1  10.0.2.4  0.710ms  0.410ms  0.536ms
2  10.0.1.4  0.966ms  0.981ms  1.268ms
Connection to 52.165.151.216 closed.
```

Notice that the first hop is to 10.0.2.4. This address is the private IP address of **nva**. The second hop is to 10.0.1.4, the address of **private**. In the first exercise, you added this route to the route table and linked the table to the **publicsubnet** subnet. So now all traffic from **public** to **private** is routed through the NVA.



Virtual network

Name: **vnet**

Address prefix: 10.0.0.0/16

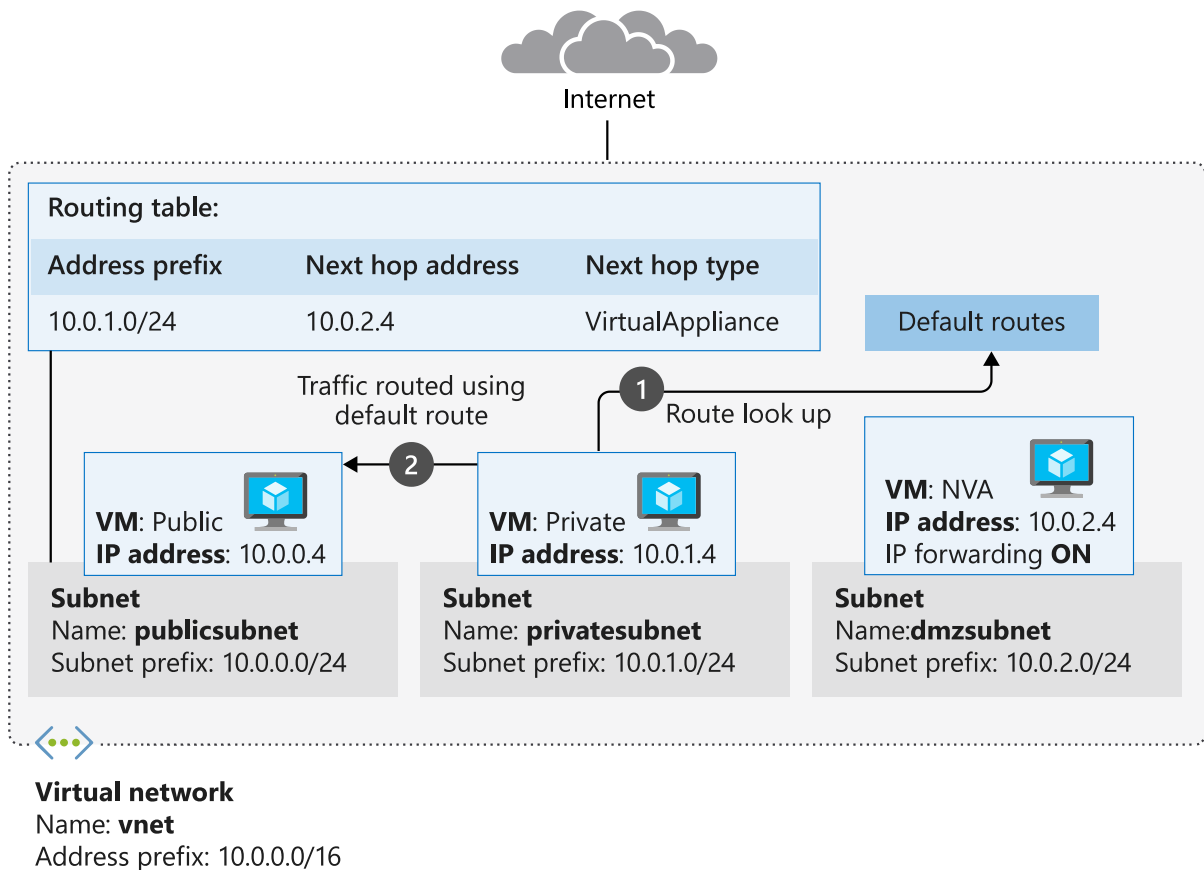
- Run the following command to trace the route from **private** to **public**. When prompted, enter the password for the **azureuser** account.

```
ssh -t -o StrictHostKeyChecking=no azureuser@$PRIVATEIP 'tracert public --type=icmp; exit'
```

You should see the traffic go directly to **public** (10.0.0.4) and not through the NVA, as shown in the following command output.

```
tracert to public.kzffavtrkpeulburui2lgywxwg.gx.internal.cloudapp.net (10.0.0.4), 64 hops max
1  10.0.0.4  1.095ms  1.610ms  0.812ms
Connection to 52.173.21.188 closed.
```

The **private** VM is using default routes, and traffic is routed directly between the subnets.



You've now configured routing between subnets to direct traffic from the public internet through the **dmzsubnet** subnet before it reaches the private subnet. In the **dmzsubnet** subnet, you added a VM that acts as an NVA. You can configure this NVA to detect potentially malicious requests and block them before they reach their intended targets.

7. Summary

<https://learn.microsoft.com/en-us/training/modules/control-network-traffic-flow-with-routes/7-summary>

Summary

Completed

In this module, you learned how to customize routes in an Azure virtual network and how to redirect the traffic flow through a network virtual appliance. You also learned how to create your own custom network virtual appliance by deploying an Azure virtual machine.

Important

In the optional exercises for this module, you created resources by using your own Azure subscription. Clean up these resources so that you won't continue to be charged for them.

Learn more

For more information on using routes in your network infrastructure, see the following articles:

- [Virtual network traffic routing](#)
- [Tutorial: Route network traffic with a route table using the Azure portal](#)
- [Deploy highly available network virtual appliances](#)

- Implement a secure hybrid network

Introduction to Azure Load Balancer

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-load-balancer/1-introduction>

Introduction

Completed

You're responsible for networking at Adatum, a new and expanding online commerce store. Adatum has several three-tier applications that run on virtual machines (VMs) that were migrated from on-premises datacenters. These VMs are hosted across various virtual networks. Some of the applications are available to the public internet and others should be accessible only to users at Adatum's main office location in Sydney.

When the applications were hosted on-premises, various hardware devices used Open Systems Interconnection (OSI) model Layer 4 load balancing to distribute incoming traffic across the web-tier VMs and across the middle-tier VMs that perform data analysis and transformation tasks. These Layer 4 devices were configured so that you could use remote desktop protocol to connect to individual VMs to perform administrative tasks. The hardware devices would also stop forwarding traffic to any VM that experienced a failure and ensured that client traffic for a session only occurred with one VM in the back-end pool. Now that the VMs are migrated to Azure, you would like to replicate the functionality provided by the Layer 4 hardware devices using native Azure services. You believe you can accomplish this goal with Load Balancer.

Azure Load Balancer distributes inbound traffic across a set of VMs in a back-end pool. The back-end pool can be made up of Azure infrastructure as a service (IaaS) VMs or instances in a Virtual Machine Scale Set. You can configure how incoming traffic is distributed across the back-end pool using load-balancing rules. You can ensure that traffic isn't directed to unresponsive nodes using health probes.

This module explains what Azure Load Balancer does, how it works, and when you should choose to use Load Balancer as a solution to meet your organization's needs.

Learning objectives

In this module, you'll:

- Learn what Azure Load Balancer is and the functionality it provides.
- Determine whether Load Balancer meets the needs of your organization.

Prerequisites

- Understanding of basic networking concepts

2. What is Azure Load Balancer?

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-load-balancer/2-what-is-azure-load-balancer>

What is Azure Load Balancer?

Completed

Some applications have so much incoming traffic that the single server hosting them becomes overwhelmed and can't respond to client requests in a timely manner. Instead of continuously adding network capacity, processors, disk resources, and RAM, you can address this traffic by implementing load balancing. Load balancing is a process in which you distribute incoming traffic equitably across multiple computers. A pool of computers that have lower levels of resources often responds to traffic more effectively than a single server with higher performance.

Azure Load Balancer is an Azure service that allows you to evenly distribute incoming network traffic across a group of Azure VMs, or across instances in a Virtual Machine Scale Set. Load Balancer delivers high availability and network performance in the following ways:

- Load-balancing rules determine how traffic is distributed to instances that comprise the back end.
- Health probes ensure the resources in the back end are healthy and that traffic isn't directed to unhealthy back-end instances.

You can deploy **public** load balancers and **internal** (or *private*) load balancers in Azure:

- *Public load balancers* are used to load balance internet traffic to your virtual machines (VMs). A public load balancer maps the public IP address and port number of incoming traffic to the private IP address and port number of the back-end pool VMs. For example, you can spread the load of incoming web-request traffic from the internet across multiple web servers. A public load balancer can also provide outbound connections for VMs inside your virtual network.
- An *internal load balancer* directs traffic to resources that are inside a virtual network or that use a VPN to access Azure infrastructure. Internal load balancer front-end IP addresses and virtual networks are never directly exposed to an internet endpoint. Internal line-of-business (LOB) applications run in Azure and are accessed from within Azure or from on-premises resources. An internal load balancer is used where private IPs are needed at the front end only. Internal load balancers are often used to balance traffic from the front-end web tier infrastructure as a service (IaaS) VMs across a set of secondary VMs that perform tasks such as performing calculations or data processing.

An internal load balancer enables the following types of load balancing:

- **Within a virtual network:** Load balancing from VMs in the virtual network to a set of VMs that reside within the same virtual network.
- **For a cross-premises virtual network:** Load balancing from on-premises computers to a set of VMs that reside within the same virtual network.
- **For multi-tier applications:** Load balancing for internet-facing multi-tier applications where the back-end tiers aren't internet-facing. The back-end tiers require traffic load balancing from the internet-facing tier.
- **For LOB applications:** Load balancing for LOB applications that are hosted in Azure without added load balancer hardware or software. This scenario includes on-premises servers that are in the set of computers whose traffic is load balanced.

Each Load Balancer type can be used for inbound and outbound scenarios and scale up to millions of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) application flows.

3. How Azure Load Balancer works

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-load-balancer/3-how-azure-load-balancer-works>

How Azure Load Balancer works

Completed

Azure Load Balancer operates at the transport layer of the Open Systems Interconnection (OSI) model. This Layer 4 functionality allows traffic management based on specific properties of the traffic. Properties including source and destination address, TCP or UDP protocol type, and port number.

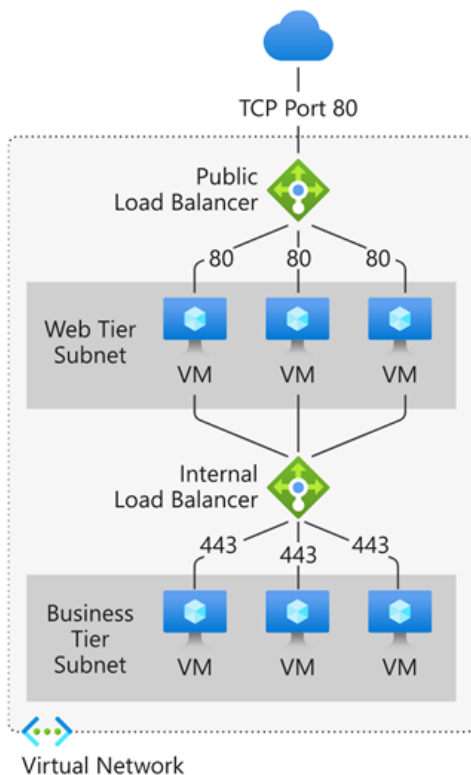
Load Balancer has several elements that work together to ensure an application's high availability and performance:

- Front-end IP
- Load balancer rules
- Back-end pool
- Health probes
- Inbound NAT rules
- High availability ports
- Outbound rules

Front-end IP

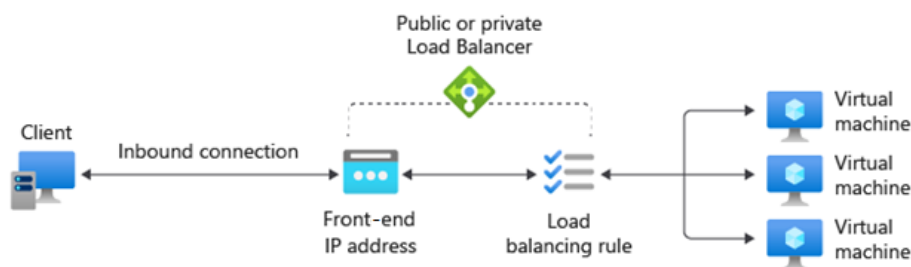
The front-end IP address is the address clients use to connect to your web application. A front-end IP address can be either a public or a private IP address. Azure load balancers can have multiple front-end IPs. The selection of a public or a private IP address determines which type of load balancer to create:

- **Public IP address: A public load balancer:** A public load balancer maps the public IP and port of incoming traffic to the private IP and port of the virtual machine (VM). You can distribute specific types of traffic across multiple VMs or services by applying load-balancing rules. For example, you can spread the load of web request traffic across multiple web servers. The load balancer maps the response traffic from the private IP and port of the VM to the public IP and port of the load balancer. Then, it transmits the response back to the requesting client.
- **Private IP address: An internal load balancer:** An internal load balancer distributes traffic to resources that are inside a virtual network. Azure restricts access to the front-end IP addresses of a virtual network that are load balanced. Front-end IP addresses and virtual networks are never directly exposed to an internet endpoint. Internal line-of-business applications run in Azure and are accessed from within Azure or from on-premises resources through a VPN or ExpressRoute connection.



Load Balancer rules

A load balancer rule defines how traffic is distributed to the back-end pool. The rule maps a given front-end IP and port combination to a set of back-end IP addresses and port combination.



Traffic is managed using a five-tuple hash made from the following elements:

- **Source IP:** The IP address of the requesting client.
- **Source port:** The port of the requesting client.

- **Destination IP:** The destination IP address of the request.
- **Destination port:** The destination port of the request.
- **Protocol type:** The specified protocol type, Transmission Control Protocol (TCP), or User Datagram Protocol (UDP).
- **Session affinity:** Ensures that the same pool node always handles traffic for a client.

Load Balancer allows you to load balance services on multiple ports, multiple IP addresses, or both. You can configure different load balancing rules for each front-end IP. Multiple front-end configurations are only supported with IaaS VMs.

Load Balancer can't apply different rules based on internal traffic content because it operates at Layer 4 (transport layer) of the OSI model. If you need to manage traffic based on its Layer 7 (application layer) properties, you need to deploy a solution like Azure Application Gateway.

Back-end pool

The back-end pool is a group of VMs or instances in a Virtual Machine Scale Set that responds to the incoming request. To scale cost-effectively to meet high volumes of incoming traffic, computing guidelines generally recommend adding more instances to the back-end pool.

Load Balancer implements automatic reconfiguration to redistribute load across the altered number of instances when you scale instances up or down. For example, if you added two more VMs instances to the back-end pool, Load Balancer would reconfigure itself to start balancing traffic to those instances based on the already configured load balancing rules.

Health probes

A health probe is used to determine the health status of the instances in the back-end pool. This health probe determines if an instance is healthy and can receive traffic. You can define the unhealthy threshold for your health probes. When a probe fails to respond, the load balancer stops sending new connections to the unhealthy instances. A probe failure doesn't affect existing connections. The connection continues until:

- The application ends the flow.
- Idle timeout occurs.
- The VM shuts down.

Load Balancer allows you to configure different health probe types for endpoints: TCP, HTTP, and HTTPS.

- **TCP custom probe:** This probe relies on establishing a successful TCP session to a defined probe port. If the specified listener on the VM exists, the probe succeeds. If the connection is refused, the probe fails. You can specify the Port, Interval, and Unhealthy threshold.
- **HTTP or HTTPS custom probe:** The load balancer regularly probes your endpoint (every 15 seconds, by default). The instance is healthy if it responds with an HTTP 200 within the timeout period (default of 31 seconds). Any status other than HTTP 200 causes the probe to fail. You can specify the port (Port), the URI for requesting the health status from the back end (URI), amount of time between probe attempts (Interval), and the number of failures that must occur for the instance to be considered unhealthy (Unhealthy threshold).

Session persistence

By default, Load Balancer distributes network traffic equally among multiple VM instances. It provides stickiness only within a transport session. Session persistence specifies how traffic from a client should be handled. The default behavior (None) is that any healthy VM can handle successive requests from a client.

Session persistence is also known as session affinity, source IP affinity, or client IP affinity. This distribution mode uses a two-tuple (source IP and destination IP) or three-tuple (source IP, destination IP, and protocol type) hash to route to back-end instances. When you use session persistence, connections from the same client go to the same back-end instance within the back-end pool. You can configure one of the following session persistence options:

- **None (default):** Specifies that any healthy VM can handle the request.
- **Client IP (2-tuple):** Specifies that the same back-end instance can handle successive requests from the same client IP address.
- **Client IP and protocol (3-tuple):** Specifies that the same back-end instance can handle successive requests from the same client IP address and protocol combination.

You can change this behavior by configuring one of the options that are described in the following sections.

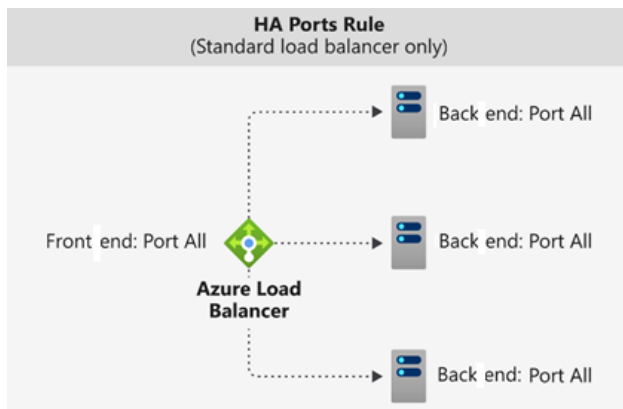
High availability ports

A load balancer rule configured with `protocol - all` and `port - 0` is known as a *high availability (HA) port rule*. This rule enables a single rule to load balance all TCP and UDP flows that arrive on all ports of an internal standard load balancer.

The load-balancing decision is made per flow. This action is based on the following five-tuple connection:

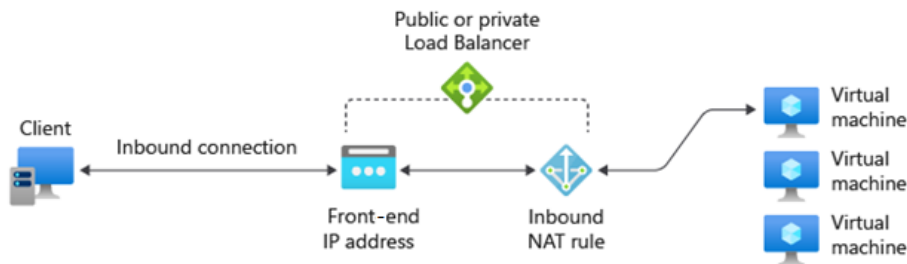
- Source IP address
- Source port
- Destination IP address
- Destination port
- Protocol

HA ports load-balancing rules help you with critical scenarios, such as high availability and scale for network virtual appliances (NVAs) inside virtual networks. The feature can help when a large number of ports must be load balanced.



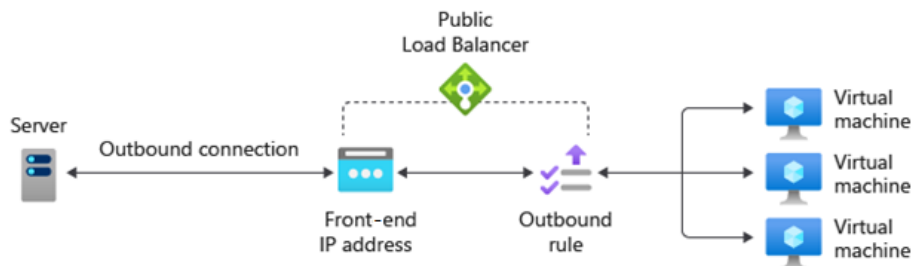
Inbound NAT rules

You can use load balancing rules in combination with Network Address Translation (NAT) rules. For example, you could use NAT from the load balancer's public address to TCP 3389 on a specific VM. This rule combination allows remote desktop access from outside of Azure.



Outbound rules

An outbound rule configures Source Network Address Translation (SNAT) for all VMs or instances identified by the back-end pool. This rule enables instances in the back end to communicate (outbound) to the internet or other public endpoints.



4. When to use Azure Load Balancer

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-load-balancer/4-when-to-use-azure-load-balancer>

When to use Azure Load Balancer

Completed

Azure Load Balancer is best suited for applications that require ultra-low latency and high performance. Load Balancer is suitable for your organization's needs because you're replacing existing network hardware devices that load balance traffic across applications. The applications used multiple virtual machine (VM) tiers when the applications were on-premises with an Azure service that has the same functionality.

Because Load Balancer operates at Layer 4 like hardware devices that were used on-premises before the organization migrated to Azure, you can use Load Balancer to replicate that hardware device functionality. This functionality includes using health probes to ensure that Load Balancer doesn't forward traffic to failed VM nodes. It also includes using session persistence to ensure that clients only communicate with a single VM during a session.

You can configure public load balancers for front-end traffic to web tiers of applications. You can also configure internal load balancers to balance traffic between the web tier and the tier that performs data analysis and transformation tasks.

You can configure inbound NAT rules to allow remote desktop protocol access to a VM instance to perform administrative tasks.

When not to use Azure Load Balancer

Azure Load Balancer isn't appropriate if you have a web application that doesn't require load balancing running on a single IaaS VM instance. For example, if your web application only receives a small amount of traffic and the existing infrastructure already competently deals with the existing load, there's no need to deploy a back-end pool of VMs and no need to use Load Balancer.

Azure provides other load-balancing solutions as alternatives to Azure Load Balancer, including Azure Front Door, Azure Traffic Manager, and Azure Application Gateway:

- **Azure Front Door** is an application-delivery network that provides a global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like TLS/SSL offload, path-based routing, fast failover, a web application firewall, and caching to improve performance and high availability of your applications. Choose this option in scenarios such as load balancing a web app deployed across multiple Azure regions.
- **Azure Traffic Manager** is a DNS-based traffic load balancer that allows 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 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 TLS/SSL termination to the gateway. Application Gateway works within a region rather than globally.

In comparison to these solutions:

- **Azure Load Balancer** is a 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. If Adatum had applications that required web application firewall functionality, Azure Load Balancer wouldn't be an appropriate solution for the company.

5. Module assessment

Module assessment

Completed

6. Summary

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-load-balancer/6-summary>

Summary

Completed

In this module, you learned about Azure Load Balancer. An Azure service that allows you to evenly distribute incoming network traffic across a group of Azure virtual machines, or across instances in a Virtual Machine Scale Set. You also learned how Load Balancer delivers high availability and network performance to your applications. You learned about the types of scenarios in which Load Balancer is an appropriate solution for your organization, and how Load Balancer is likely able to meet Adatum's networking needs. You also learned about the difference between Azure Load Balancer and other traffic management technologies, including Azure Application Gateway and Azure Traffic Manager.

Learn more

- [What is Azure Load Balancer?](#)
- [Load-balancing options](#)
- [Azure Load Balancer components](#)
- [Module: Introduction to Azure Load Balancer](#)

Introduction to Azure Application Gateway

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/1-introduction>

Introduction

Completed

Azure Application Gateway is an Azure service that processes traffic to web apps that are hosted on a pool of web servers. The processing performed by Azure Application Gateway includes load balancing HTTP traffic and inspecting traffic using web application firewall. It also includes encrypting traffic between users and an application gateway, and encrypting traffic between application servers and an application gateway.

Adatum is a new and expanding online commerce store that sells industrial drones. You're responsible for networking at the company. Adatum has several web applications that are hosted on computers in its on-premises datacenter. At the moment, a special but aging hardware device is deployed on the Adatum perimeter network that manages traffic to the web applications hosted on these computers. You want to retire this device and have traffic mediated by an Azure service.

To meet your goals, you need to ensure that the Azure service replicates the functionality that the special hardware currently provides. Important functionality that must be present in the replacement service includes:

- Detect if one of the on-premises servers becomes unavailable so that traffic is no longer directed to it.
- TLS (Transport Layer Security) termination functionality to reduce the amount of CPU capacity consumed by encryption and decryption operations.
- Session affinity to ensure that the same back-end pool host always serves a client connection to a web application.
- Security filtering of malicious traffic such as SQL injection and cross site scripting attacks.

This module explains what Azure Application Gateway does, how it works, and when you should choose to use Azure Application Gateway as a solution to meet your organization's needs.

Learning objectives

In this module, you'll:

- Learn what Azure Application Gateway is and the functionality it provides.
- Determine whether Azure Application Gateway meets the needs of your organization.

Prerequisites

- Understanding of basic networking concepts
- Familiarity with Azure virtual machines and Azure App Service
- Familiarity with Azure virtual networking

2. What is Azure Application Gateway?

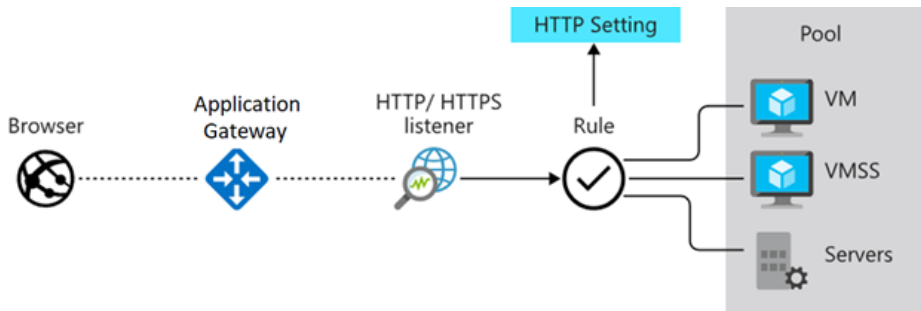
<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/2-what-is-azure-application-gateway>

What is Azure Application Gateway?

Completed

Azure Application Gateway manages the requests that client applications send to web apps that are hosted on a pool of web servers. The pool of web servers can be Azure virtual machines, Azure Virtual Machine Scale Sets, Azure App Service, and even on-premises servers.

Application Gateway provides features such as load balancing HTTP traffic and web application firewall. It provides support for TLS/SSL encryption of traffic between users and an application gateway and between application servers and an application gateway.



Application Gateway uses a round-robin process to load balance requests to the servers in each back-end pool. Session stickiness ensures client requests in the same session are routed to the same back-end server. Session stickiness is especially important with e-commerce applications where you don't want a transaction to be disrupted because the load balancer bounces it around between back-end servers.

Azure Application Gateway includes the following 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.
- Connection draining allowing graceful removal of back-end pool members during planned service updates.

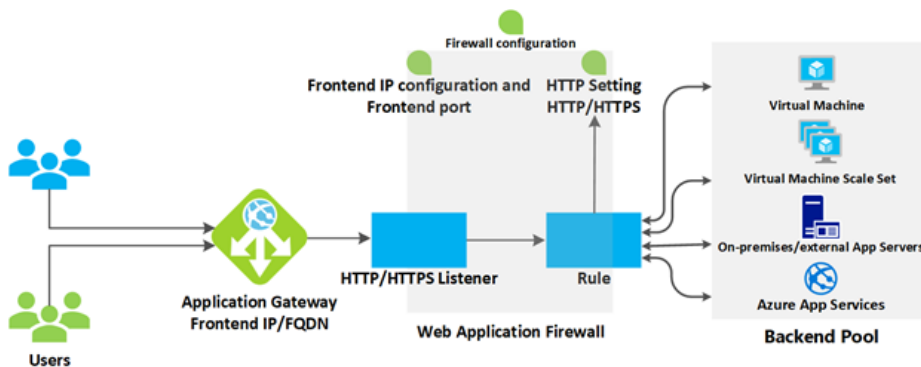
3. How Azure Application Gateway works

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/3-how-azure-application-gateway-works>

How Azure Application Gateway works

Completed

Azure Application Gateway has a series of components that combine to securely route and load balance requests across a pool of web servers. Application Gateway includes the following components:



- **Front-end IP address:** Client requests are received through a front-end IP address. You can configure Application Gateway to have a public IP address, a private IP address, or both. Application Gateway can't have more than one public IP address and one private IP address.

- **Listeners:** Application Gateway uses one or more listeners to receive incoming requests. A listener accepts traffic arriving on a specified combination of protocol, port, host, and IP address. Each listener routes requests to a back-end pool of servers following routing rules that you specify. A listener can be Basic or Multi-site. A *Basic* listener only routes a request based on the path in the URL. A *Multi-site* listener can also route requests using the hostname element of the URL. Listeners also handle TLS/SSL certificates for securing your application between the user and Application Gateway.
- **Routing rules:** A routing rule binds a listener to the back-end pools. A rule specifies how to interpret the hostname and path elements in the URL of a request and direct the request to the appropriate back-end pool. A routing rule also has an associated set of HTTP settings. These HTTP settings indicate whether (and how) traffic is encrypted between Application Gateway and the back-end servers. Other configuration information includes Protocol, Session stickiness, Connection draining, Request timeout period, and Health probes.

Load balancing in Application Gateway

Application Gateway uses a round-robin mechanism to automatically load balance the requests sent to the servers in each back-end pool. Load-balancing works with the Open Systems Interconnection (OSI) Layer 7 routing implemented by Application Gateway routing, which means that it load balances requests based on the routing parameters (host names and paths) used by the Application Gateway rules. In comparison, other load balancers, such as Azure Load Balancer, function at the OSI Layer 4 level and distribute traffic based on the IP address of the target of a request.

You can configure session stickiness if you need to ensure that all requests for a client in the same session are routed to the same server in a back-end pool.

Web application firewall

The web application firewall (WAF) is an optional component that handles incoming requests before they reach a listener. The web application firewall checks each request for many common threats based on the Open Web Application Security Project (OWASP). Common threats include: SQL-injection, Cross-site scripting, Command injection, HTTP request smuggling, HTTP response splitting, Remote file inclusion, Bots, crawlers, and scanners, and HTTP protocol violations and anomalies.

OWASP defines a set of generic rules for detecting attacks. These rules are referred to as the Core Rule Set (CRS). The rule sets are under continuous review as attacks evolve in sophistication. WAF supports four rule sets: CRS 3.2, 3.1, 3.0 and 2.2.9. CRS 3.1 is the default. If necessary, you can opt to select only specific rules in a rule set, targeting certain threats. Additionally, you can customize the firewall to specify which elements in a request to examine, and limit the size of messages to prevent massive uploads from overwhelming your servers.

Back-end pools

A back-end pool is a collection of web servers that can be made up of: a fixed set of virtual machines, a virtual machine scale-set, an app hosted by Azure App Services, or a collection of on-premises servers.

Each back-end pool has an associated load balancer that distributes work across the pool. When configuring the pool, you provide the IP address or name of each web server. All the servers in the back-end pool should be configured in the same way, including their security settings.

If you're using TLS/SSL, the back-end pool has an HTTP setting that references a certificate used to authenticate the back-end servers. The gateway re-encrypts the traffic by using this certificate before sending it to one of your servers in the back-end pool.

If you're using Azure App Service to host the back-end application, you don't need to install any certificates in Application Gateway to connect to the back-end pool. All communications are automatically encrypted. Application Gateway trusts the servers because Azure manages them.

Application Gateway uses a rule to specify how to direct the messages that it receives on its incoming port to the servers in the back-end pool. If the servers are using TLS/SSL, you must configure the rule to indicate:

- That your servers expect traffic through the HTTPS protocol.
- Which certificate to use to encrypt traffic and authenticate the connection to a server.

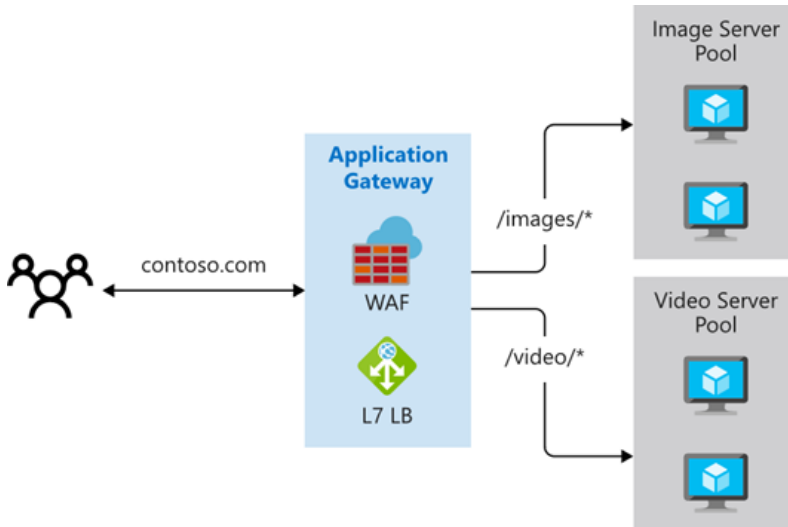
Application Gateway routing

When the gateway routes a client request to a web server in the back-end pool, it uses a set of rules configured for the gateway to determine where the request should go. There are two primary methods of routing this client request traffic: path-based routing and

multiple-site routing.

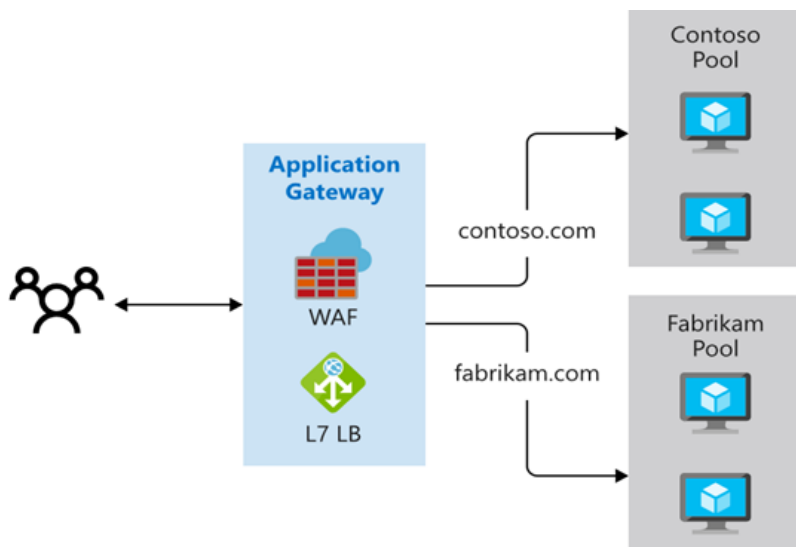
Path-based routing

Path-based routing sends requests with different URL paths to 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 domain name system 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 `http://contoso.com` to servers in one back-end pool, and requests for `http://fabrikam.com` to another back-end pool. The following diagram shows this configuration:



Multi-site configurations are useful for supporting multitenant applications, where each tenant has its own set of virtual machines or other resources hosting a web application.

Application Gateway routing also includes these features:

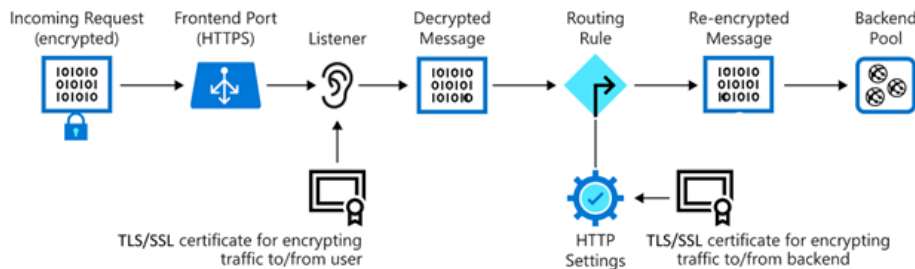
- **Redirection.** Redirection can be used to another site, or from HTTP to HTTPS.
- **Rewrite HTTP headers.** HTTP headers allow the client and server to pass parameter information with the request or the response.
- **Custom error pages.** Application Gateway allows you to create custom error pages instead of displaying default error pages. You can use your own branding and layout using a custom error page.

TLS/SSL termination

When you terminate the TLS/SSL connection at the application gateway, it offloads the CPU-intensive TLS/SSL termination workload from your servers. Also, you don't need to install certificates and configure TLS/SSL on your servers.

If you need end-to-end encryption, Application Gateway can decrypt the traffic on the gateway by using your private key, then re-encrypt again with the public key of the service running in the back-end pool.

Traffic enters the gateway through a front-end port. You can open many ports, and Application Gateway can receive messages on any of these ports. A listener is the first thing that your traffic meets when entering the gateway through a port. The listener is set up to listen for a specific host name, and a specific port on a specific IP address. The listener can use an TLS/SSL certificate to decrypt the traffic that enters the gateway. The listener then uses a rule that you define to direct the incoming requests to a back-end pool.



Exposing your website or web application through the application gateway also means that you don't directly connect your servers to the web. You're exposing only port 80 or port 443 on the application gateway, which is then forwarded to the back-end pool server. In this configuration, your web servers aren't directly accessible from the internet, which reduces the attack surface of your infrastructure.

Health probes

Health probes determine which servers are available for load-balancing in a back-end pool. The Application Gateway uses a health probe to send a request to a server. When the server returns an HTTP response with a status code between 200 and 399, the server is considered healthy. If you don't configure a health probe, Application Gateway creates a default probe that waits for 30 seconds before deciding that a server is unavailable. Health probes ensure that traffic isn't directed to a nonresponsive or failed web endpoint in the back-end pool.

Autoscaling

Application Gateway supports autoscaling, and can scale up or down based on changing traffic load patterns. Autoscaling also removes the requirement to choose a deployment size or instance count during provisioning.

WebSocket and HTTP/2 traffic

Application Gateway provides native support for the WebSocket and HTTP/2 protocols. The WebSocket and HTTP/2 protocols enable full duplex communication between a server and a client over a long-running Transmission Control Protocol (TCP) connection. This type of communication is more interactive between the web server and the client, and can be bidirectional without the need for polling as required in HTTP-based implementations. These protocols have low overhead (unlike HTTP) and can reuse the same TCP connection for multiple request/responses resulting in a more efficient resource utilization. These protocols are designed to work over traditional HTTP ports of 80 and 443.

4. When to use Azure Application Gateway

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/4-when-to-use-azure-application-gateway>

When to use Azure Application Gateway

Completed

Azure Application Gateway can meet your organization's needs for the following reasons:

- Azure Application Gateway routing allows traffic to be directed from an endpoint in Azure to a back-end pool made up of servers running in Adatum's on-premises datacenter. The health-probe functionality of Azure Application Gateway ensures that traffic isn't being directed to any server that becomes unavailable.
- Azure Application Gateway TLS termination functionality reduces the amount of CPU capacity that servers in the back-end pool allocate to encryption and decryption operations.
- Azure Application Gateway allows Adatum to use a web application firewall to block cross-site scripting and SQL injection traffic before it reaches servers in the back-end pool.
- Azure Application Gateway supports session affinity. This support is required because the several web applications deployed by Adatum use user session state information stored locally on individual servers in the back-end pool.

When not to use Azure Application Gateway

Azure Application Gateway isn't appropriate if you have a web application that doesn't require load balancing. For example, if you have a web application that only receives a small amount of traffic and the existing infrastructure already competently deals with the existing load, there's no need to deploy a back-end pool of web apps or virtual machines and no need for Application Gateway.

Azure provides other load balancing solutions, including Azure Front Door, Azure Traffic Manager, and Azure Load Balancer. The following list describes the differences between these services:

- **Front Door** is an application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like TLS/SSL offload, path-based routing, fast failover, web application firewall, and caching to improve performance and high-availability of your applications. Choose this option in scenarios such as load balancing a web app deployed across multiple Azure regions.
- **Traffic Manager** is a 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 TTLs.
- **Azure Load Balancer** is a high-performance, ultra low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. Azure Load Balancer is built to handle millions of requests per second while ensuring that your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across availability zones. Azure Load Balancer works within a region rather than globally.

5. Module assessment

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/5-knowledge-check>

Module assessment

Completed

6. Summary

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-application-gateway/6-summary>

Summary

Completed

In this module, you learned about Azure Application Gateway, a service that allows you to manage the requests that client applications can send to a web app. You learned that Application Gateway routes traffic to a pool of web servers based on the URL of a request. The pool of web servers can be Azure virtual machines, Azure virtual machine scale sets, Azure App Service, and even on-premises servers. You learned that Application Gateway provides features such as load balancing HTTP traffic, a web application firewall, and support for TLS/SSL encryption of your data. You also learned that Application Gateway supports encrypting traffic between users and an application gateway, and between application servers and an application gateway.

Learn more

- [What is Azure Application Gateway?](#)
- [Azure Application Gateway features](#)
- [How an application gateway works](#)
- [Application gateway components](#)
- [Load balance your web service traffic with Application Gateway](#)

Introduction to Azure Network Watcher

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/1-introduction>

Introduction

Completed

Organizations can use Azure Network Watcher to detect and monitor issues related to the network performance of infrastructure as a service (IaaS) resources in Microsoft Azure.

Adatum is a new and expanding online commerce store. You work in DevOps at Adatum, and you're responsible for networking. Adatum has several three-tier applications running on virtual machines that were migrated from on-premises datacenters to Azure. The applications are hosted across multiple Azure virtual networks. Adatum also has several hybrid applications that have compute tiers in both on-premises and cloud locations.

This module explains what Azure Network Watcher does, how it works, and when you should choose to use Azure Network Watcher as a solution to meet your organization's needs.

Learning objectives

In this module, you'll:

- Learn what Azure Network Watcher is and the functionality it provides.
- Determine whether Azure Network Watcher meets the needs of your organization.

Prerequisites

- Understanding of basic networking concepts such as IP addressing, subnetting, routing, and network security groups.
- Basic familiarity with Azure network integration concepts such as VPNs, Azure ExpressRoute, and peering.

2. What is Azure Network Watcher?

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/2-what-is-azure-network-watcher>

What is Azure Network Watcher?

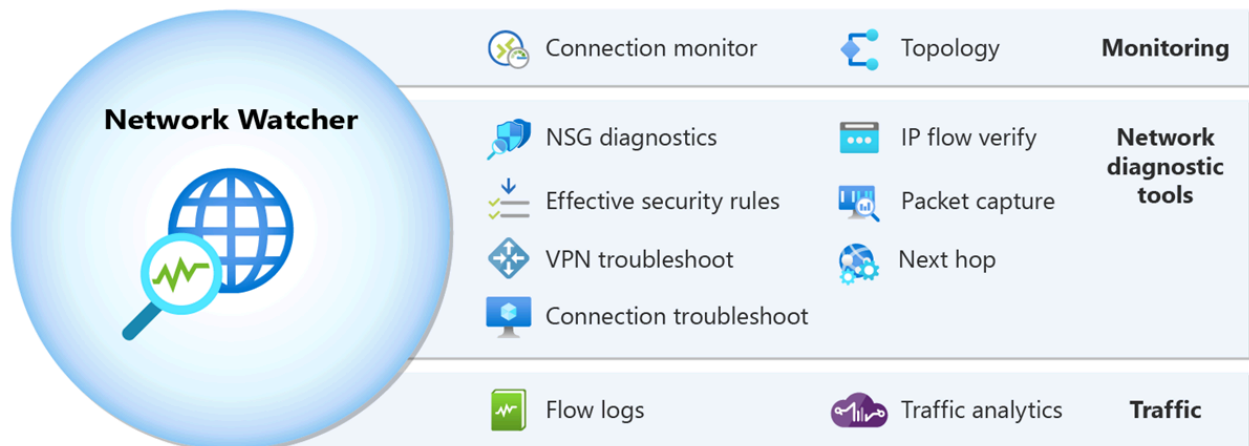
Completed

Azure Network Watcher provides a suite of tools to monitor, diagnose, view metrics, and enable or disable logs for Azure IaaS (Infrastructure-as-a-Service) resources. Network Watcher enables you to monitor and repair the network health of IaaS products like virtual machines (VMs), virtual networks (VNETs), application gateways, load balancers, etc. Network Watcher isn't designed or intended for PaaS monitoring or Web analytics.

Network Watcher consists of three major sets of tools and capabilities:

- Monitoring
 - Topology

- Connection monitor
- Network diagnostic
 - IP flow verify
 - NSG diagnostics
 - Next hop
 - Effective security rules
 - Connection troubleshoot
 - Packet capture
 - VPN troubleshoot
- Traffic
 - Flow logs
 - Traffic analytics



Monitoring

Network Watcher offers two monitoring tools that help you view and monitor resources:

- Topology
- Connection monitor

Topology

The **Topology** tool provides a visualization of the entire network for understanding network configuration. It provides an interactive interface to view resources and their relationships in Azure spanning across multiple subscriptions, resource groups, and locations. At the beginning of the troubleshooting process, this tool helps you visualize all of the elements involved in the problem, allowing you to find something that isn't apparent by looking at the contents of resource groups.

Connection monitor

Connection monitor provides end-to-end connection monitoring for Azure and hybrid endpoints. It helps you understand network performance between various endpoints in your network infrastructure. You can use connection monitor to verify that two IaaS VMs that host the components of a multi-tier application can communicate with each other. You can also use it to verify connectivity in hybrid scenarios.

Network diagnostic tools

Network Watcher offers seven network diagnostic tools that help troubleshoot and diagnose network issues:

- IP flow verify
- NSG diagnostics
- Next hop
- Effective security rules
- Connection troubleshoot
- Packet capture
- VPN troubleshoot

IP flow verify

IP flow verify allows you to detect traffic filtering issues at a virtual machine level. It checks if a packet is allowed or denied to or from an IP address (IPv4 or IPv6 address). It also tells you which security rule allowed or denied the traffic.

NSG diagnostics

NSG diagnostics allows you to detect traffic filtering issues at a virtual machine, virtual machine scale set, or application gateway level. It checks if a packet is allowed or denied to or from an IP address, IP prefix, or a service tag. It tells you which security rule allowed or denied the traffic. It also allows you to add a new security rule with a higher priority to allow or deny the traffic.

Next hop

Next hop allows you to detect routing issues. It checks if traffic is routed correctly to the intended destination. It provides you with information about the Next hop type, IP address, and Route table ID for a specific destination IP address.

Effective security rules

Effective security rules allows you to view the effective security rules applied to a network interface. It shows you all security rules applied to the network interface, the subnet the network interface is in, and the aggregate of both.

Connection troubleshoot

Connection troubleshoot enables you to test a connection between a virtual machine, a virtual machine scale set, an application gateway, or a Bastion host and a virtual machine, an FQDN, a URI, or an IPv4 address. The test returns similar information returned when using the connection monitor tool, but tests the connection at a point in time instead of monitoring it over time, as connection monitor does.

Packet capture

Packet capture allows you to remotely create packet capture sessions to record all network traffic to and from a virtual machine (VM) or a virtual machine scale set.

VPN troubleshoot

VPN troubleshoot enables you to troubleshoot virtual network gateways and their connections.

Traffic

Network Watcher offers two traffic tools that help you log and visualize network traffic:

- Flow logs
- Traffic analytics

Flow logs

Flow logs allows you to log information about your Azure IP traffic and stores the data in Azure storage. You can log IP traffic flowing through a network security group or Azure virtual network.

Traffic analytics

Traffic analytics provides rich visualizations of flow logs data.

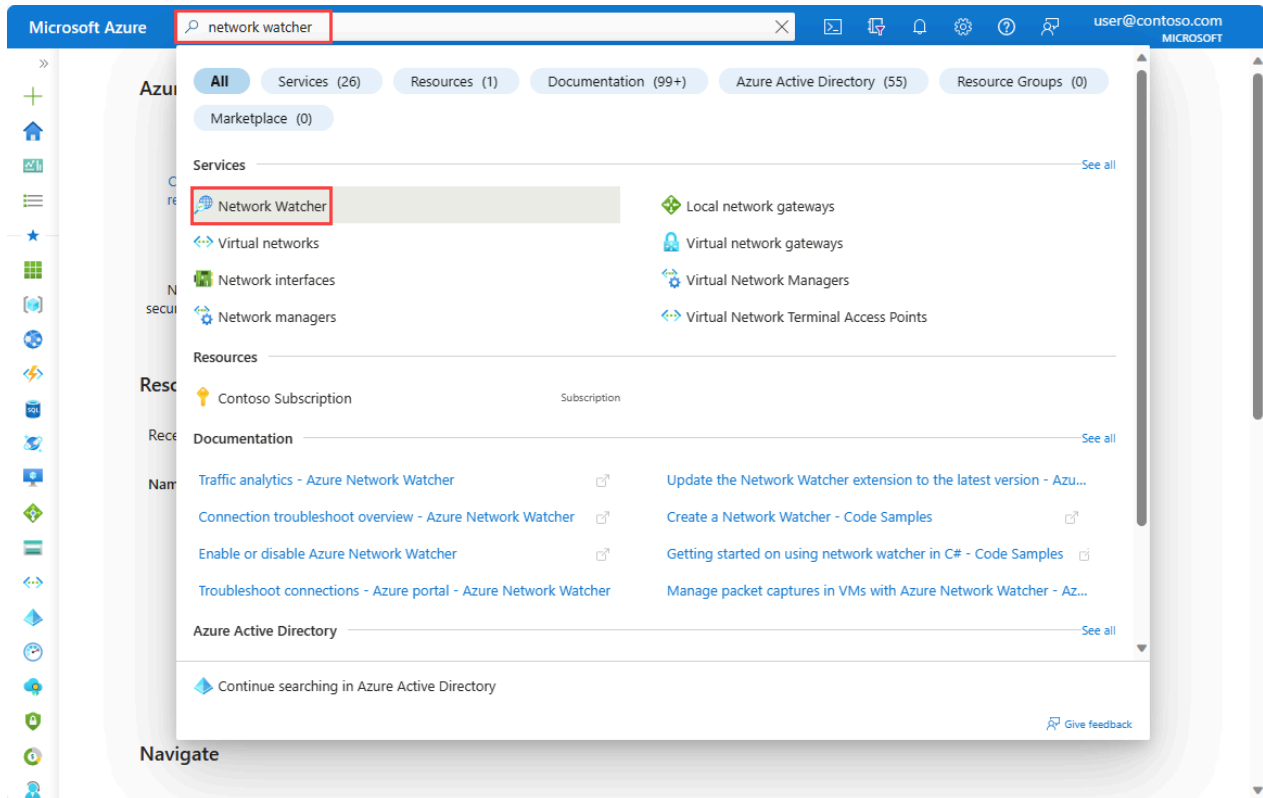
3. How Azure Network Watcher works

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/3-how-azure-network-watcher-works>

How Azure Network Watcher works

Completed

Network Watcher becomes automatically available when you create a virtual network in an Azure region in your subscription. You can access Network Watcher directly in the Azure portal by typing **Network Watcher** in the **Search** bar.



Network Watcher Topology tool

The topology capability of Azure Network Watcher allows you to view all of the following resources in a virtual network. Including, the resources associated to resources in a virtual network and the relationships between the resources.

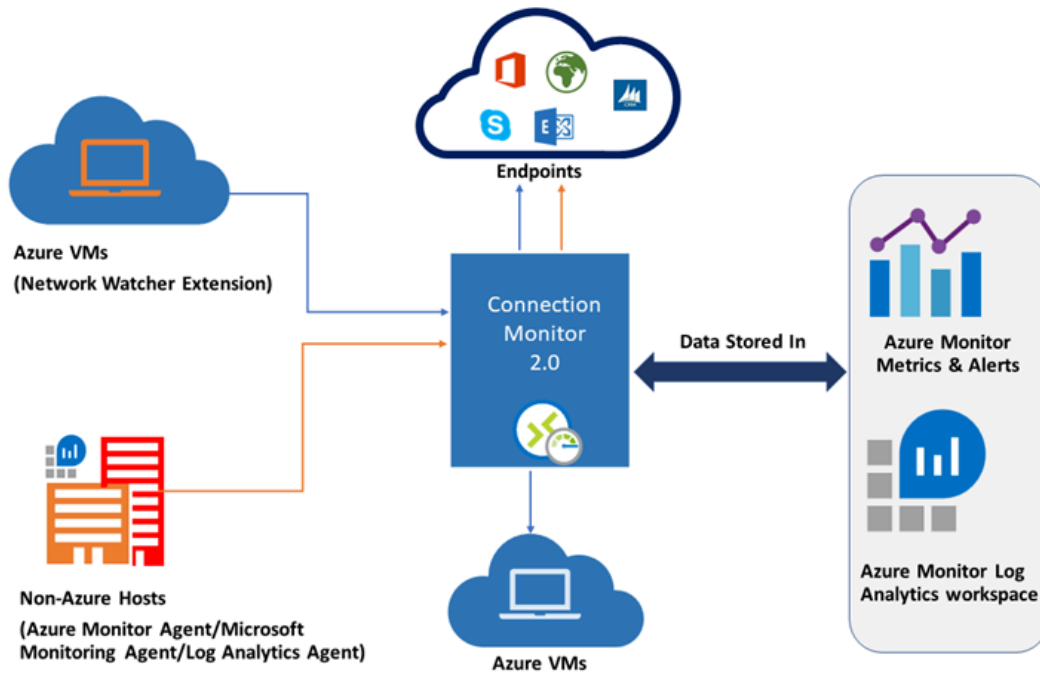
- Subnets
- Network interfaces
- Network security groups
- Load balancer
- Load balancer health probes
- Public IP addresses
- Virtual network peering
- Virtual network gateways
- VPN gateway connections
- Virtual machines
- Virtual Machine Scale Sets

All resources returned in a topology have the following properties:

- **Name:** The name of the resource.
- **Id:** The URI of the resource.
- **Location:** The Azure region the resource is in.
- **Associations:** A list of associations to the referenced object. Each association has the following properties:
 - **AssociationType:** References the relationship between the child object and the parent. Valid values are **Contains** and **Associated**.
 - **Name:** The name of the referenced resource.
 - **ResourceId:** The URI of the resource referenced in the association.

Connection Monitor tool

Connection Monitor provides unified, end-to-end connection monitoring in Azure Network Watcher. Connection Monitor supports both hybrid and Azure cloud deployments. You can use the Connection Monitor tool to measure the latency between resources. Connection Monitor can detect changes that affect connectivity, such as network configuration changes or modifications to NSG rules. You can configure Connection Monitor to probe VMs at regular intervals to look for failures or changes. Connection Monitor can diagnose problems and provide explanations about why the issue occurred and the steps that you can take to fix an issue.



To use Connection Monitor for monitoring, you need to install monitoring agents on the hosts that you monitor. Connection Monitor uses lightweight executable files to run connectivity checks, whether a host is located in an Azure virtual network or in an on-premises network. With Azure VMs, you can install the Network Watcher Agent VM, also known as the Network Watcher extension.

IP flow verify

The IP flow verify tool uses a 5-tuple packet parameter-based verification mechanism to detect whether packets that are inbound or outbound are allowed or denied from a VM. Within the tool, you can specify a local and remote port, the protocol (TCP or UDP), the local IP, the remote IP, the VM, and the VM's network adapter.

Next hop

Traffic from an IaaS VM is sent to a destination based on the effective routes associated with a network interface (NIC). Next hop gets the next hop type and IP address of a packet from a specific VM and NIC. Knowing the next hop helps you determine whether traffic is being directed to the intended destination or whether the traffic is being sent nowhere. An improper configuration of routes, in which traffic is directed to an on-premises location or to a virtual appliance, might lead to connectivity issues. 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`.

Effective security rules

Network security groups (NSGs) filter packets based on their source and destination IP address and port numbers. More than one NSG can apply to an IaaS resource on an Azure virtual network. By taking into account all rules that are applied across all NSGs for a resource, the Effective Security Rules tool allows you to determine why some traffic might be denied or allowed.

Packet capture

Packet capture is a virtual machine extension that is remotely started through Network Watcher. This capability eases the burden of running a packet capture manually on a specific virtual machine by using operating system tools or third-party utilities. Packet capture

can be triggered through the portal, PowerShell, the Azure CLI, or REST API. Network Watcher allows you to configure filters for the capture session to ensure you capture traffic you want to monitor. Filters are based on 5-tuple (protocol, local IP address, remote IP address, local port, and remote port) information. The captured data is stored on the local disk or in a storage blob.

Connection troubleshoot

The connection troubleshoot tool checks TCP connectivity between a source and a destination VM. You can specify the destination VM by using an FQDN, a URI, or an IP address. If the connection is successful, information about the communication appears, including:

- The latency in milliseconds.
- The number of probe packets sent.
- The number of hops in the complete route to the destination.

If the connection is unsuccessful, the tool displays details about the fault. You might see the following fault types:

- **CPU**: The connection failed because of high CPU utilization.
- **Memory**: The connection failed because of high memory utilization.
- **GuestFirewall**: A firewall outside of Azure blocked the connection.
- **DNSResolution**: The destination IP address couldn't be resolved.
- **NetworkSecurityRule**: An NSG blocked the connection.
- **UserDefinedRoute**: There's an incorrect user route in a routing table.

VPN troubleshoot

Network Watcher provides the capability to troubleshoot gateways and connections. The capability can be called through the portal, PowerShell, the Azure CLI, or REST API. When called, Network Watcher diagnoses the health of the gateway or connection, and then it returns the appropriate results. The request is a long-running transaction. The preliminary results that are returned give an overall picture of the health of the resource.

The following list describes the values that are returned by calling the VPN troubleshoot API:

- **startTime**: The time the troubleshooting started.
- **endTime**: The time the troubleshooting ended.
- **code**: This value is `UnHealthy` if there's a single diagnosis failure.
- **results**: A collection of results returned on the connection or the virtual network gateway.
 - **id**: The fault type.
 - **summary**: A summary of the fault.
 - **detailed**: A detailed description of the fault.
 - **recommendedActions**: A collection of recommended actions to take.
 - **actionText**: Text that describes what action to take.
 - **actionUri**: The URI for documentation that describes what action to take.
 - **actionUriText**: A short description of the action text.

4. When to use Azure Network Watcher

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/4-when-to-use-azure-network-watcher>

When to use Azure Network Watcher

Completed

Azure Network Watcher is useful when you're attempting to troubleshoot networking issues related to Azure IaaS products. For example, you can use the tools included in Azure Network Watcher in the following scenarios:

- Resolve connectivity issues related to IaaS VMs.
- Troubleshoot VPN connections.
- Determine cross region network latencies.

Resolve connectivity issues related to IaaS VMs

Recently, a Windows Server IaaS VM was deployed to Azure. The developers who deployed the VM are unable to establish a remote PowerShell session to this IaaS VM from another IaaS VM on the same virtual network.

You can troubleshoot this issue using the IP flow verify tool. This tool lets you specify a local and remote port, the protocol (TCP/UDP), the local IP, and the remote IP to check the connection status. It also lets you specify the direction of the connection (inbound or outbound). IP flow verify runs a logical test on the rules in place on your network.

In this case, you can use IP flow verify to specify the VM's IP address and the TCP port 5986 (used by PowerShell when using HTTPS). Then, specify the remote VM's IP address and port. Choose the TCP protocol, then select **Check**. If an NSG rule is blocking traffic, the IP flow verify rule reports which rule is responsible for the dropped traffic.

Troubleshoot VPN connections

An IaaS VM is deployed on an Azure virtual network. Connections to this IaaS VM from on-premises hosts are made through a site-to-site VPN connection.

You can troubleshoot this VPN connection using the Azure VPN troubleshoot tool. This tool runs diagnostics on a virtual network gateway connection, and returns a health diagnosis. You can run this tool from the Azure portal, PowerShell, or the Azure CLI.

When you run the tool, it checks the gateway for common issues and returns the health diagnosis. You can also view the log file to get more information. The diagnosis shows whether the VPN connection is working. If the VPN connection isn't working, the Azure VPN troubleshoot tool suggests ways to resolve the issue.

Determine cross-region network latency

You can use Network Watcher tools to determine the best location to place IaaS resources based on network latencies. For example, you can use Network Watcher to have IaaS VMs in different regions regularly ping each other to determine cross region network latency. This information can allow you to determine whether all your IaaS VMs need to be located in a single region. Or, if they can be spread across different regions to support specific application architectures.

Let's say you have an on-premises hybrid application and an application running in an Azure IaaS VM that connects to the same storage account endpoint. You could use Network Watcher to perform a comparison of latencies for the two applications. If the latency for the on-premises application is too high, it might strengthen a case for migrating that application to Azure. If the latency for the Azure IaaS VM is too high, it might strengthen a case for migrating the VM to another region with lower latency.

When not to use Network Watcher

Network Watcher tools provide intermediate levels of network diagnostic functionality. These tools don't provide some of the advanced features available in some third-party tools. If your organization needs access to this advanced functionality, you might need to deploy a third-party tool that includes this advanced functionality to accomplish your diagnostic goals.

It's important to realize that Network Watcher is mostly used for IaaS resources on Azure virtual networks. You can't use Azure Network Watcher to diagnose connectivity issues related to PaaS services or Web analytics. If you're encountering problems related to these services, you should check the Azure status or service health dashboard.

5. Module assessment

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/5-knowledge-check>

Module assessment

Completed

6. Summary

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-network-watcher/6-summary>

Summary

Completed

In this module, you learned about Azure Network Watcher, a service you can use to perform monitoring and diagnostic tasks on IaaS resources deployed on Azure virtual networks. You learned about the monitoring and diagnostic tools that are available in Network Watcher, and which tool is appropriate for specific troubleshooting scenarios.

Learn more

- [What is Azure Network Watcher?](#)
- [Enable or disable Azure Network Watcher](#)
- [Monitor and troubleshoot your end-to-end Azure network infrastructure by using network monitoring tools](#)
- [Configure Network Watcher](#)
- [Design and implement network monitoring](#)