Introduction

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

What is Azure Load Balancer?

Completed
100 XP
2 minutes

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 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 TCP and UDP application flows.

How Azure Load Balancer works

Azure Load Balancer operates at the transport layer of the 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 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.

Diagram that depicts how public and internal load balancers work in Azure Load Balancer.

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.

Diagram that depicts how load balancer rules work in Azure Load Balancer.

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, TCP or 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 called 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.

Diagram that shows how high availability ports work in Azure Load Balancer.

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.

Diagram that shows how inbound NAT rules work in Azure Load Balancer.

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.

When to use Azure Load Balancer
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 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.
Azure Load Balancer is a high-performance, ultra-low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. Its 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.

Imagine yourself in the role of a network engineer at an organization that is migrating to Azure. As the network engineer you need to ensure line-of-business applications, services, and data are available to end users of your corporate network whenever and wherever possible. You also need to ensure users get access to those network resources in an efficient and timely manner.

Azure provides different flavors of load balancing services that help with the distribution of workloads across your networks. The aim of load balancing is to optimize the use of your resources, while maximizing throughput and minimizing the time it takes for a response. You can create internal and public load balancers in an Azure environment to distribute the network traffic within your network and the network traffic arriving from outside your network. In this module, you learn about using the Azure Load Balancer, and Traffic Manager load balancing services.

Learning objectives
In this module, you:

Understand non-HTTP(S) options for load balancing.
Learn about the Azure Load Balancer.
Learn about Azure Traffic Manager.
Prerequisites
You should have experience with networking concepts, such as IP addressing, Domain Name System (DNS), and routing.

You should have experience with network connectivity methods, such as VPN or WAN.
You should have experience with the Azure portal and Azure PowerShell.

Explore load balancing

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

This video reviews how to select a load balancing solution.

Load Balancing options for Azure

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

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

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

Global versus regional

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

In contrast, Regional load-balancing services distribute traffic within virtual networks across virtual machines (VMs) or zonal and zone-redundant service endpoints within a region. You can

think of them as systems that load balance between VMs, containers, or clusters within a region in a virtual network.

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

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

Important

In this module, we're focusing on the non-HTTP(S) solutions.

This table summarizes these categorizations for each Azure load balancing service.

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

Choosing a load balancing option for Azure
Here are the key factors to decide on a load balancing option.

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

flow chart to help select a load-balancing solution for your application.

Tip

You should use this flowchart and the suggested recommendation only as a starting point. A completed solution can incorporate two or more load-balancing solutions.

Selecting a load balancing solution by using the Azure portal

You can use the Azure Load Balancing page in the Azure portal to help guide you to a load-balancing solution. Search for and select Load balancing - help me choose. The wizard provides an interactive way to select a load balancing solution.

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

This video reviews how to select a load balancer type.

Choosing a load balancer type
Load balancers can be public (external) or internal (private).

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

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

This diagram shows how public and internal load balancers can work together.

Diagram that shows a public and internal load balancer.

Azure load balancer and availability zones
Azure Load Balancer supports availability zones scenarios. A Load Balancer can either be zone redundant, zonal, or nonzonal.

Zone redundant
Diagram that shows Zone redundant load balancers in Azure.

In a region with Availability Zones, a Standard Load Balancer can be zone-redundant. A single frontend IP address survives zone failure. The frontend IP can be used to reach all (nonimpacted) backend pool members no matter the zone. One or more availability zones can fail and the data path survives as long as one zone in the region remains healthy.

Zonal
Diagram that shows Zonal load balancers in Azure.

You can choose to have a frontend guaranteed to a single zone, which is known as a zonal. With this scenario, a single zone in a region serves all inbound or outbound flow. Your frontend shares fate with the health of the zone. The data path is unaffected by failures in zones other than where it was guaranteed.

Nonzonal
Load Balancers can also use a "no-zone" frontend. In these scenarios, a public load balancer would use a public IP or public IP prefix, an internal load balancer would use a private IP. This option doesn't give a guarantee of redundancy.

Selecting an Azure load balancer SKU
There are several load balancer SKUs: Basic, Standard, and Gateway. These SKUs differ in terms of their scenario scope and scale, features, and cost. The Gateway Load Balancer SKU is for high performance and high availability scenarios with Network Virtual Appliances (NVAs). This table compares the Standard and Basic Load Balancer.

Important

On September 30, 2025, Basic Load Balancer will be retired.

Features Standard Load Balancer Basic Load Balancer
Backend pool size Supports up to 1,000 instances. Supports up to 300 instances.
Backend pool endpoints Any virtual machines or virtual machine scale sets in a single virtual network. Virtual machines in a single availability set or virtual machine scale set.
Health probes TCP, HTTP, HTTPS TCP, HTTP
Health probe down behavior TCP connections stay alive on an instance probe down and on all probes down. TCP connections stay alive on an instance probe down. All TCP connections end when all probes are down.
Availability Zones Zone-redundant and zonal frontends for inbound and outbound traffic. Not available.
Diagnostics Azure Monitor multi-dimensional metrics. Azure Monitor logs
Secure by default Closed to inbound flows unless allowed by a network security group. Internal traffic from the virtual network to the internal load balancer is allowed. Open by default. Network security group optional.
Multiple front ends Inbound and outbound Inbound only

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

important point to understand is that Traffic Manager works at the DNS level which is at the Application layer (Layer-7).

This video reviews Traffic Manager features and how the service works.

Key features of Traffic Manager
Traffic Manager offers the several key features.

Feature Description
Increase application availability Traffic Manager delivers high availability for your critical applications by monitoring your endpoints and providing automatic failover when an endpoint goes down.
Improve application performance Azure allows you to run cloud services and websites in datacenters located around the world. Traffic Manager can improve the responsiveness of your website by directing traffic to the endpoint with the lowest latency.
Service maintenance without downtime You can plan maintenance on your applications without downtime. Traffic Manager can direct traffic to alternative endpoints while the maintenance is in progress.
Combine hybrid applications Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including the burst-to-cloud, migrate-to-cloud, and failover-to-cloud scenarios.
Distribute traffic for complex deployments Using nested Traffic Manager profiles, multiple traffic-routing methods are combined to create sophisticated and flexible rules to scale to the needs of larger, more complex deployments.
How Traffic Manager works
Azure Traffic Manager enables you to control how network traffic is distributed to application deployments (endpoints) running in your different datacenters. Azure Traffic Manager uses DNS to direct the client requests to the appropriate service endpoint based on a traffic-routing method. For any profile, Traffic Manager applies the traffic-routing method associated to it to each DNS query it receives. The traffic-routing method determines which endpoint is returned in the DNS response.

Azure Traffic Manager supports different traffic-routing methods to determine how to route network traffic to the various service endpoints. You select the method that best fits your requirements.

This video reviews Traffic Manager routing methods.

Priority routing method
Use the priority routing method for a primary service endpoint for all traffic. You can provide multiple backup endpoints in case the primary or one of the backup endpoints is unavailable.

Diagram that shows the 'Priority' routing method.

Weighted routing method
Use the Weighted routing method when you want to distribute traffic across a set of endpoints based on their importance. Set the weight the same to distribute evenly across all endpoints.

Diagram that shows the 'Weighted' routing method.

Performance routing method
Use the Performance routing method when endpoints are in different geographic locations. Users should use the "closest" endpoint for the lowest network latency.

Diagram that shows the 'Performance' routing method.

Geographic routing method
Use the Geographic routing method to direct users to specific endpoints based on where their DNS queries originate from geographically. Good choice for regional compliance requirements.

Diagram that shows the 'Geographic' routing method.

Tip

Learn more about Traffic Manager check out the Enhance your service availability and data locality by using Azure Traffic Manager module.

Summary

In this module, you learned about the Azure Load Balancer and Azure Traffic Manager.

The main takeaways from this module are:

Load balancing distributes workloads to servers and services.
Azure offers two non-HTTP(S) load balancing solutions: Azure Load Balancer and Azure Traffic Manager.
Azure Load Balancers can distribute workloads globally or regionally.
Azure Load Balancers can be public (external) or internal (private).
Azure Load Balancer has two SKUs: Basic and Standard.
Azure Traffic Manager is a DNS-based network traffic load balancer.
Azure Traffic Manager supports different traffic-routing methods. These methods include performance, weighted, priority, and geographic.
Learn more with Copilot
Copilot can assist you in configuring 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 Load Balancer? Provide benefits, features, and usage cases for the product.
What is Azure Traffic Manager? Provide benefits, features, and usage cases for the product.
Compare and contrast Azure Load Balancer and Azure Traffic Manager. When should you use each product? Provide usage cases.

Skip to main content
Learn

Training
Level 9

23050 /34199 XP
Learn Training Browse Introduction to Azure Load Balancer
Module assessment
200 XP
4 minutes
Choose the best response for each question.

Check your knowledge

1. Which layer of the OSI model does Azure Load Balancer function at?

Layer 4

Layer 7

Layer 5
2. Which of the following components should you configure to ensure that traffic from a specific client computer is always directed to the same server in the back-end pool?

Health probe

High availability port

Session persistence
3. Which of the following components would you configure to ensure that back-end pool instances that are no longer responding to traffic on TCP port 443 no longer have traffic directed to them by Load Balancer?

High availability port

Outbound rule

Health probe
Next unit: Summary
Need help? See our troubleshooting guide or provide specific feedback by reporting an issue.
Feedback
Was this page helpful?

Design Azure Application Gateway

The Azure Application Gateway processes network traffic to web apps hosted on a pool of web servers. The processing performed by Azure Application Gateway includes load balancing HTTP traffic and inspecting traffic using a web application firewall. This type of routing is known as application layer (OSI layer 7) load balancing. Azure Application Gateway includes the following features:

Support for the HTTP, HTTPS, HTTP/2, and WebSocket protocols.
A web application firewall (WAF) 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 backend pool members during planned service updates.
Session stickiness to ensure client requests in the same session are routed to the same backend server.
Path and URL based routing.
How Azure Application Gateway works

Let's review the Azure Application Gateway components.

Front-end IP address. Client requests are received through a front-end IP address. You can configure the Application Gateway to have a public IP address, a private IP address, or both.
Listeners. A listener is a logical entity that checks for incoming connection requests. A listener accepts a request if the protocol, port, hostname, and IP address match the listener's configuration. You must have at least one listener.
Request routing rules. A request routing rule is a key component of an application gateway because it determines how to route traffic on the listener. The rule binds the listener, the

backend server pool, and the backend HTTP settings. When a listener accepts a request, the request routing rule forwards the request to the backend or redirects it elsewhere. If the request is forwarded to the backend, the request routing rule defines which backend server pool to forward it to.
Backend pools. A backend pool is a collection of web servers. Backend targets can include: 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. The backend pool receives and processes requests.
Health probes. Health probes determine which servers are available for load-balancing in a backend pool. Servers are automatically added and removed from the backend pool based on their availability.
Tip

Learn more about Azure Application Gateway check out the Introduction to Azure Application Gateway module.

Check your knowledge

1. What is the primary function of Azure Application Gateway?

The Application Gateway is primarily used as a load balancer and web traffic manager.

The Application Gateway is primarily used for data storage and retrieval.

The Application Gateway is primarily used for machine learning and AI tasks.
2. Which type of routing does the Azure Application Gateway provide?

Transport, layer 4.

Session, layer 5.

Application, layer 7.
3. What is a listener?

A listener is an entity that checks for incoming connection requests.

A listener is an entity that routes traffic based on basic or path-based rules.

A listener is a collection of servers that respond to requests.

Configure Azure Application Gateway
This diagram explains how the Azure Application Gateway components work together.

Routing configuration
One of the most important gateway configuration settings is the routing rules. The Azure Application Gateway has two primary methods of routing client requests: path-based and multiple sites.

Path-based routing
Path-based routing sends requests with different URL paths to different pools of back-end servers. For example, you could direct video requests to a back-end pool optimized to handle video streaming. You could also direct image requests to a pool of servers that handles image retrieval.

Diagram that depicts path-based routing in Azure Application Gateway.

Multiple site routings
Multiple site routing configures more than one web application on the same Application Gateway instance. In a multiple site configuration, you register multiple DNS names (CNAMEs) for the IP address of the application gateway, specifying the name of each site. Application Gateway uses separate listeners to wait for requests for each site. Each listener passes the request to a different rule, which can route the requests to servers in a different back-end pool. For example, you could direct all requests for http://contoso.com to a specific backend pool.

Diagram that depicts multi-site routing in Azure Application Gateway.

This video reviews the routing methods.

Other routing capabilities
Along with path-based routing and multiple site hosting, there are a few other capabilities when routing with Application Gateway.

Redirection. Redirection can be used to another site, or from HTTP to HTTPS. For example, redirecting HTTP requests to a secure HTTPS shopping site.
Rewrite HTTP headers. HTTP headers allow the client and server to pass additional 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.
Tip

Learn more about Azure Application Gateway routing check out the Load balance your web service traffic with Application Gateway module.

Design and configure Azure Front Door

Azure Front Door is Microsoft's modern cloud Content Delivery Network (CDN) that provides fast, reliable, and secure access between your users and your applications. Azure Front Door delivers your content using the Microsoft's global edge network with hundreds of global and local POPs distributed around the world close to both your enterprise and consumer end users.

Azure Front Door tiers
Azure Front Door provides both content delivery and security features. Azure Front Door Standard is content-delivery optimized.

Provide for both static and dynamic content acceleration.
Support global load balancing.
Implement SSL offload.
Implement domain and certificate management.
Benefit from enhanced traffic analytics.
Benefit from basic security capabilities.
Azure Front Door Premium is security optimized.

Extensive security capabilities across Web Application Firewall.
BOT protection.
Private Link support.
Integration with Microsoft Threat Intelligence and security analytics.
Azure Front Door usage cases
Diagram of the Azure Front Door architecture.

This diagram shows a user request processed by Azure Front Door.

A user is requesting www.contoso.com. This request is routed from the client to Azure Front Door. Azure Front Door resides at the edge of the Microsoft Global Network. In Azure, an edge location is a data center that's geographically closer to end-users than traditional Azure regions. These locations are designed to cache content and deliver services with lower latency, improving the speed and responsiveness of applications for users worldwide.
Azure Front Door determines where to direct the client request. The routing process includes the web application firewall, routing rules, rules engine, and caching configuration.
A nonspecific request can be routed to any one of the three regions.
A search request can be routed to a specific region optimized for search.
A request can even be routed to a region with another cloud service.
Other things to know
Routing algorithm. The Azure Front Door routing algorithm first matches based on HTTP protocol, then frontend host, then the Path.
HTTP Protocols (HTTP/HTTPS)
Hosts (for example, www.foo.com, *.bar.com)
Paths (for example, /, /users/, /file.gif)

Response codes. Azure Front Door response codes help clients understand the purpose of the redirect. You can set the protocol used for redirection. The most common use case of the redirect feature is to set HTTP to HTTPS redirection.
Health probes. Front Door periodically sends a synthetic HTTP/HTTPS request to each of your configured backends. Front Door then uses these responses from the probe to determine the "best" backend resources to route your client requests.
Tip

What is Azure Front Door?
Many organizations have applications they want to make available to their customers, their suppliers, and almost certainly their users. The tricky part is making sure those applications are highly available. In addition, they need to be able to quickly respond while being appropriately secured. Azure Front Door provides different tiers (pricing tiers) that meet these requirements. Let's briefly review the features and benefits of these tiers so you can determine which option best suits your requirements.

What is a secure, modern cloud CDN?
A secure, modern cloud CDN provides a distributed platform of servers. This helps minimize latency when users are accessing webpages. Historically, IT staff might have used a CDN and a web-application firewall to control HTTP and HTTPS traffic flowing to and from target applications.

If an organization uses Azure, they might achieve these goals by implementing the products described in the following table:

Product Description
Azure Front Door Enables an entry point to your apps positioned in the Microsoft global edge network. Provides faster, more secure, and scalable access to your web applications.
Azure Content Delivery Network Delivers high-bandwidth content to your users by caching their content at strategically placed physical nodes around the world.
Azure Web Application Firewall Helps provide centralized, greater protection for web applications from common exploits and vulnerabilities.
Azure Front Door definition
Azure Front Door Standard/Premium provides the capabilities of these three products. It offers a fast, reliable, and more secure modern cloud CDN by using the Microsoft global edge network to integrate with intelligent threat protection. Azure Front Door resides in the edge locations and manages user requests to your hosted applications. Users connect to your application through the Microsoft global network. Azure Front Door then routes user requests to the fastest and most available application backend.

Note

An application backend is any internet-facing service that you host, either inside or outside Azure.

The following Azure Front door tiers are available:

Azure Front Door (classic), which is the entry level. Existing Azure customers often bolster these features with Azure Content Delivery Network, and Azure Web Application Firewall.
Azure Front Door Standard, which is optimized for seamless content delivery.
Azure Front Door Premium, which is optimized for improved security.
Let's examine these last two tiers in more detail.

Azure Front Door Standard
Azure Front Door Standard provides the capabilities of Azure Front Door (Classic), Azure Content Delivery Network, and Azure Web Application Firewall. Azure Front Door Standard includes:

Content-delivery optimization
Static and dynamic content acceleration
Global load balancing
Secure Sockets Layer (SSL) offload
Domain and certificate management
Enhanced traffic analytics
Basic security capabilities
Azure Front Door Premium
Azure Front Door Premium provides the same capabilities as Azure Front Door Standard. However, it's security optimized and includes the following extra features:

Extensive security capabilities across Web Application Firewall
Private link support
Integration with Microsoft Threat Intelligence and security analytics
How to improve your cloud app delivery
To improve your cloud application delivery, consider deploying an Azure Front Door solution that best fits your needs. In the following unit, we discuss these choices in more detail.

How Azure Front Door works
In this unit, you learn how Azure Front Door works and how it:

Helps provide fast, secure, and scalable access to your web applications.
Helps protect your cloud-based apps.
Provides high-bandwidth content.
Azure Front Door optimizes access times to content. In the following diagram, users are connecting to content hosted in the custom domain contoso.com. Azure Front Door is implemented at multiple edge locations. Azure Front Door provides CDN features that optimize access to backend content, while the firewall helps to secure that access.

Diagram depicting the Azure Front Door Standard/Premium architecture as previously described.

How Azure Front Door optimizes content delivery
Azure Front Door uses the anycast protocol with split TCP at layer 7 to route HTTP/S client requests to the most available and fastest application backend. The way Azure Front Door routes requests depend on the routing method you select and on backend health. Azure Front Door supports four routing methods, as the following table describes:

Routing method Description
Latency Helps ensure requests are sent to the lowest latency backends, within an acceptable sensitivity range.
Priority Uses administrator-assigned priorities to your backends when you want to configure a primary backend to service all traffic.
Weighted Uses administrator-assigned weights to your backends when you want to distribute traffic across a set of backends.
Session Affinity Allows you to configure session affinity for your frontend hosts or domains. This helps ensure requests from the same end user are sent to the same backend.
Azure Front Door also provides backend health monitoring options. Azure Front Door periodically assesses the health of each of your configured backends. Responses from these backends enable Azure Front Door to determine to which backend resources your client requests can be routed.

Note

Azure Front Door is resilient to failures, including failures of an entire Azure region due to the many edge locations strategically placed around the world.

A CDN is a distributed collection of web servers. These servers deliver web-based content to users. To help minimize latency, CDN's use point-of-presence locations that are next to users to cache content.

Azure Front Door provides the following key CDN features:

Dynamic site acceleration
CDN caching rules
HTTPS custom domain support
Azure diagnostics logs
File compression
Geo-filtering
How Azure Front Door helps secure content
Azure Front Door provides web-application firewall capabilities to help protect your web applications from exploits and vulnerabilities. Managing security for your applications can be challenging because web applications are increasingly targeted.

Azure Front Door operates at the network's edge, close to potential attacks. This helps prevent attacks before they can enter your network. Azure Front Door's web application firewall is based on policies you can associate with one or more instances of Azure Front Door. These firewall policies consist of:

Managed rule sets, which are a collection of preconfigured rules.
Custom rules that you can configure.
Note

If present, custom rules are processed first.

A rule consists of:

A condition, which determines whether a rule applies to traffic.
A priority, which determines the order in which a rule gets processed, based on importance.
An action, which can be Allow, Block, Log, or Redirect.
A mode, which there are two:
Detection: Azure Web Application Firewall only monitors and logs when in this mode. However, it takes no other action.
Prevention: Azure Web Application Firewall takes the defined action while in this mode.
In the next unit, let's consider the factors that will help you determine which Azure Front Door tier is most appropriate for your organizational needs.

When to use Azure Front Door
Now we discuss how to determine which Azure Front Door SKU is the best choice for your needs. Your organization wants to provide efficient, reliable, and optimized access to application content that's hosted in Azure. We review the following criteria:

Scalability
Security
Pricing
Content delivery
It's also important to consider several other Azure products you could use instead of Azure Front Door, including:

Azure Traffic Manager: Provides DNS-based global routing. However, it doesn't provide for Transport Layer Security (TLS) protocol termination, or SSL offload, per-HTTP/HTTPS request, or application-layer processing.
Azure Application Gateway: Can load-balance between your servers in a region at the application layer.
Decision criteria
You can use Azure Front Door to build, operate, and scale out your dynamic web application and static content. Remember, the following tiers are available:

Azure Front Door Standard, which is content-delivery optimized.
Azure Front Door Premium, which is security optimized.
The decision you make depends on whether you require the other features that Azure Front Door Standard and Azure Front Door Premium offer.

Criteria Analysis
Scalability Does your organization scale-out content? Organizations that host scalable content benefit more from using Azure Front Door.
PricingDoes your organization prefer a monthly charge for each policy or hourly billing? Do you want to pay extra charges for custom rules? Review the pricing considerations in the Pricing section later in this unit.
Content delivery Do you require content optimization, without extensive security capabilities? Azure Front Door Standard is a good choice in this case.
Security Do you have enhanced security requirements? Azure Front Door Premium is your best option.
Apply the criteria
To decide which product has the features you need, review the following criteria and the recommendations about which product meets them.

Scalability
Organizations that don't host global, scalable web applications might not benefit from implementing Azure Front Door. However, if it builds, operates, and scales out dynamic web applications and static content, it can benefit from the use of the different Azure Front Door tiers.

Consider using Azure Front Door when you want to:

Define, manage, and monitor your web traffic's global routing.
Optimize for top-tier, end-user performance and reliability through quick global failover.
Pricing
Azure Front Door billing is based on outbound data transfers, inbound data transfers, and routing rules. If you implement Azure Web Application Firewall and Azure Content Delivery Network, pricing includes:

A monthly charge per policy.
Other charges for custom rules and managed rule sets.
Azure Front Door Standard/Premium billing is based on the following criteria:

A fixed charge calculated on hourly basis
Outbound data transfers
Inbound data transfers
Requests incoming from client to Azure Front Door points of presence
Content delivery

Consider using Azure Front Door Standard when you want to:

Optimize your content delivery.
Provide for both static and dynamic content acceleration.
Support global load balancing.
Implement SSL offload.
Implement domain and certificate management.
Benefit from enhanced traffic analytics.
Benefit from basic security capabilities.
Security
Consider using Azure Front Door Premium when you need Azure Front Door Standard features and require:

Extensive security capabilities across Web Application Firewall.
BOT protection.
Private Link support.
Integration with Microsoft Threat Intelligence and security analytics.
Note

Azure Web Application Firewall pricing is included in Premium tier.

1. Contoso is deciding the best way to optimize performance of their distributed web apps. It's important that Contoso's users can access locally cached content to help optimize access times. Which of the following solutions is recommended?

Azure Front Door

Azure Web Application Firewall

Azure Front Door Standard
2. Which of the following Azure Front Door SKUs provides SSL offload support?

Only Azure Front Door Standard

Only Azure Front Door Premium

Both Azure Front Door Standard and Azure Front Door Premium SKUs
3. Security admins at Contoso want to ensure that the Azure Front Door SKU you select integrates with Microsoft Threat Intelligence and security analytics. Which SKU provides this support?

Azure Front Door (classic)

Azure Front Door Premium

Azure Front Door Standard

4. With Azure Front Door, how can you help guarantee that the front-end hosts ensure requests from the same user are forwarded to the same backend?

This isn't a feature of Azure Front Door.

By using weighted routing.

By using session affinity.

5. When an administrator is creating Azure Web Application Firewall rules, which of the following rules is processed first when applied to HTTP or HTTPS traffic?

Custom rules

Actions

Managed rule sets