

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:

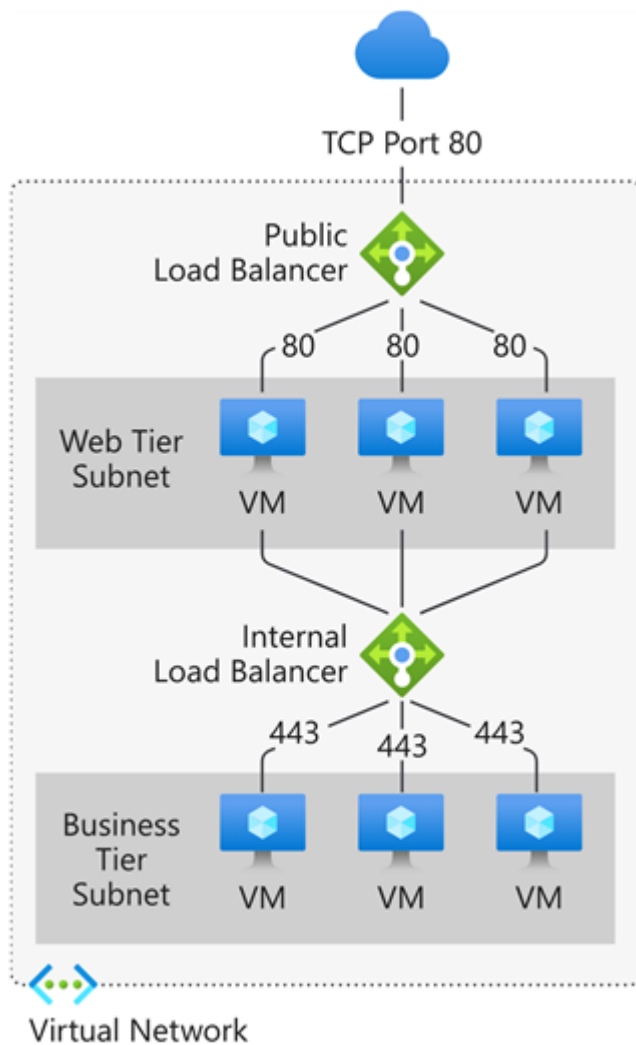
- Front-end IP
- Load balancer rules
- Back-end pool
- Health probes
- Inbound NAT rules

- High availability ports
- Outbound rules

Front-end IP

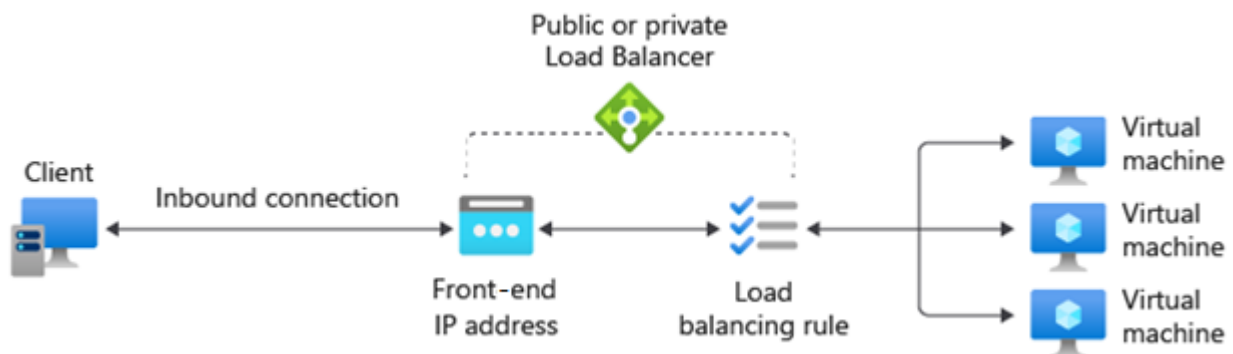
The front-end IP address is the address clients use to connect to your web application. A front-end IP address can be either a public or a private IP address. Azure load balancers can have multiple front-end IPs. The selection of a public or a private IP address determines which type of load balancer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

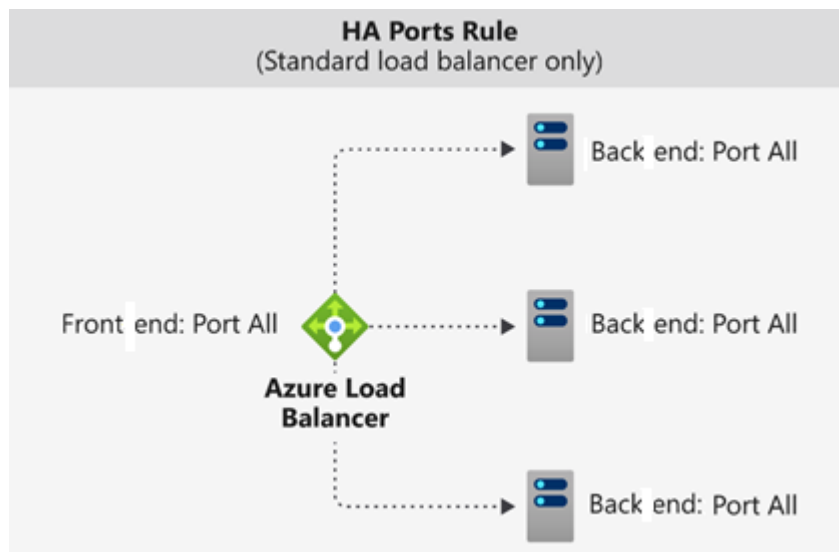
High availability ports

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:

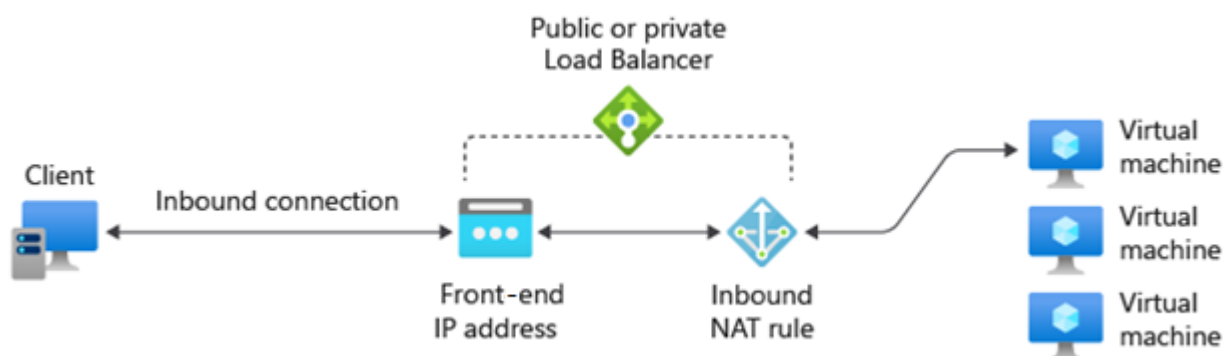
- Source IP address
- Source port
- Destination IP address
- Destination port
- Protocol

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.



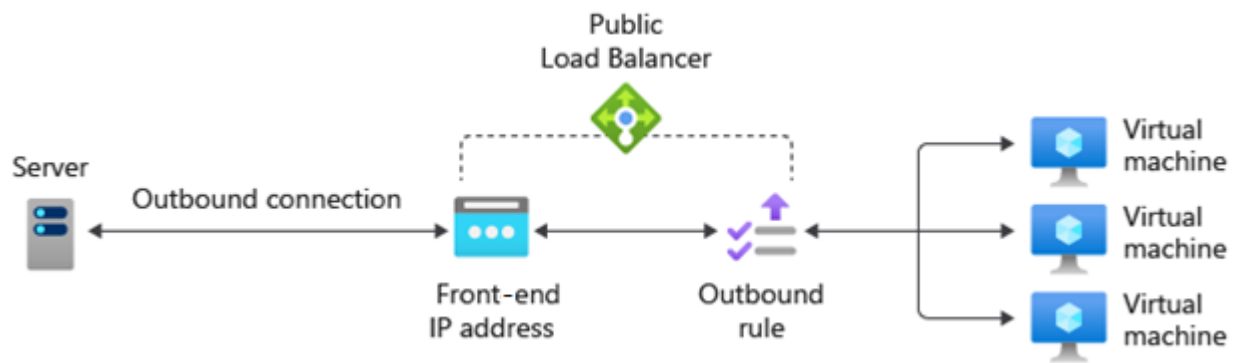
Inbound NAT rules

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

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

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](#)