Elastic Load Balance

Service Overview

Issue 01
Date 2025-08-29

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

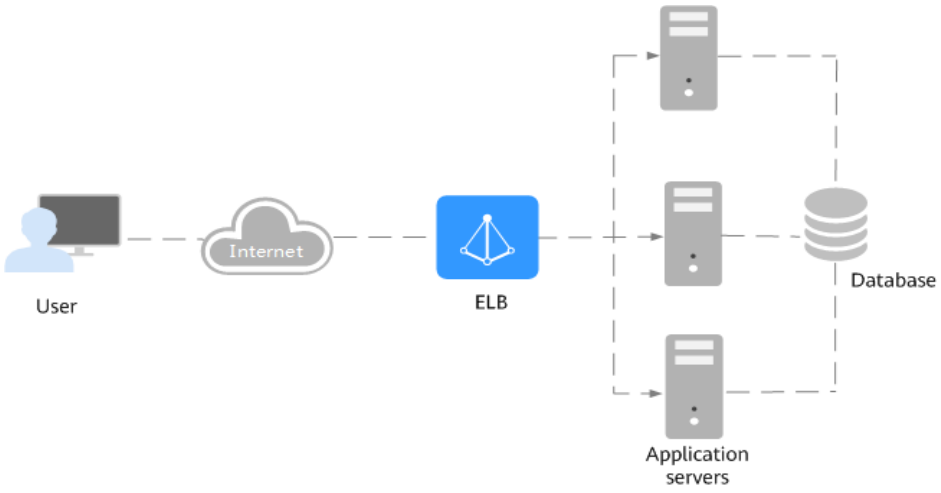
Contents

1 What Is ELB?

Elastic Load Balance (ELB) automatically distributes incoming traffic across multiple backend servers based on the routing policies you configure. ELB expands the service capabilities of your applications and improves their availability by eliminating single points of failure (SPOFs).

In the following figure, ELB distributes incoming traffic to three application servers, and each server processes one third of the requests. ELB checks the health of backend servers and distributes traffic only to servers that are running normally, improving the availability of applications.

Figure 1-1 Using a load balancer



ELB Components

ELB consists of the components as shown in the figure below.

Figure 1-2 ELB components

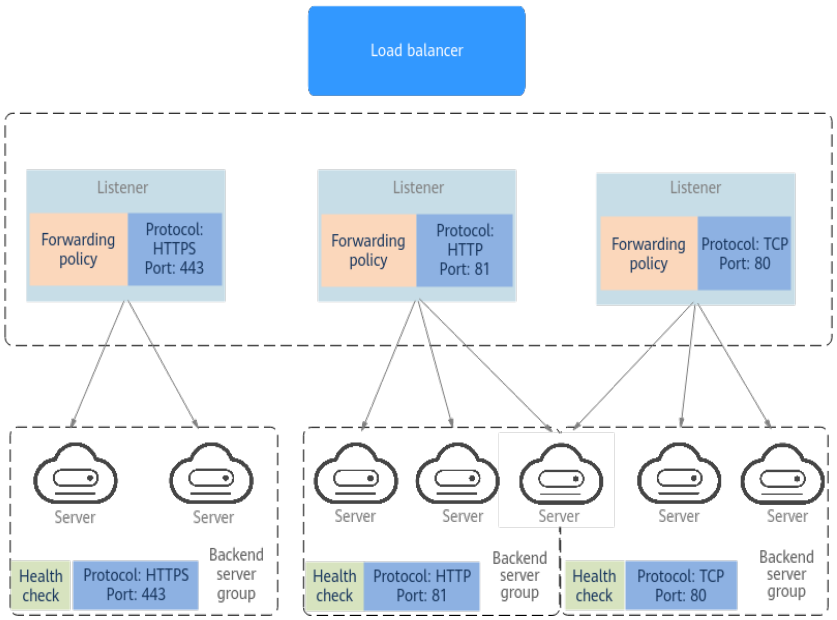


Table 1-1 ELB components

Load balancer	Distributes incoming traffic across backend servers in one or more availability zones (AZs).
Listener	Uses the protocol and port you specify to check requests from clients and route the requests to associated backend servers based on the routing policies and forwarding policies you configure. You can add one or more listeners to a load balancer.
Backend server group	Contains one or more backend servers to receive requests routed by the listener. A backend server can be a cloud server, supplementary network interface, or IP address.
Backend server	Processes requests from the associated load balancer. When you add a listener to a load balancer, you can create or select a backend server group to receive requests from the load balancer by using the port and protocol you specify for the backend server group and the load balancing algorithm you select.

Accessing ELB

You can use either of the following methods to access ELB:

- Management console

  Log in to the management console and choose **Elastic Load Balance (ELB)**.
- APIs

You can call APIs to access ELB. For details, see the **Elastic Load Balance API Reference**.

2 ELB Advantages

ELB Advantages over LVS/Nginx Load Balancing

Table 2-1 Comparison between ELB and LVS/Nginx load balancing

Item	ELB	LVS/Nginx Load Balancing
O&M	Fully managed and O&M-free	Manual installation, upgrade, and maintenance
Billing modes	• Elastic specifications: You are billed for how long each load balancer is running and the number of LCUs you use. • Fixed specifications: Multiple specifications are available for you to select to best meet your needs. You are charged for the total LCUs you use.	You are billed for resources reserved for peak hours.
Deployment	• Deployed in clusters • Multi-AZ	Deployed in VMs or containers
Reliability	• If there are traffic bursts, servers are added automatically. • Node-level/AZ-level DR and 99.99% of SLA	• Sufficient computing resources need to be reserved to handle traffic surges during peak hours. • Layer 7 performance depends on underlying computing resources. There is no SLA commitment.

Item	ELB	LVS/Nginx Load Balancing
Performance	ELB can handle up to tens of millions of concurrent connections and establish millions of new connections.	Only active/standby deployment is supported for Layer 4 load balancing. The performance is restricted by resource specifications.
Configuration change	Dynamic loading is supported.	• A reload process is required for configuration updates, which may interrupt persistent connections. • A reload is required for changing Lua plug-ins.
SSL offloading	SSL encryption/decryption is performed on load balancers. This relieves servers from decrypting or encrypting data.	SSL encryption/decryption is performed on backend servers, compromising server performance.
Related services	• Web Application Firewall (WAF) for protecting apps and websites against attacks • Cloud Eye for monitoring cloud services and resources • Log Tank Service (LTS) for collecting, querying, and storing access logs	Manual deployment is required for additional functions.

Advantages of Dedicated Load Balancers

Table 2-2 Advantage details

Superb performance	Each load balancer has exclusive access to isolated resources, allowing your services to handle a massive number of requests. A single load balancer deployed in an AZ can handle up to 20 million concurrent connections. If multiple AZs are configured for a load balancer, its performance, such as the number of new connections and concurrent connections, will be multiplied by the number of AZs. For example, if you configure two AZs for a dedicated load balancer, it can handle up to 40 million concurrent connections.
Ultra-high security	Dedicated load balancers support TLS 1.3 and can route HTTPS requests to backend servers. You can select or customize security policies that fit your security requirements.

Multiple protocols	Dedicated load balancers support Quick UDP Internet Connection (QUIC), TCP, UDP, HTTP, and HTTPS, so that they can route requests to different types of applications.
High flexibility	Dedicated load balancers can route requests based on their content, such as the request method, header, URL, path, and source IP address. They can also redirect requests to another listener or URL, or return a fixed response to the clients.
No limits	Dedicated load balancers can route requests to both servers on the cloud and on premises, allowing you to leverage cloud resources to handle traffic bursts.
Ease-of-use	Dedicated load balancers provide a diverse set of algorithms that allow you to configure different traffic routing policies to meet your requirements while keeping deployments simple.

Advantages of Shared Load Balancers

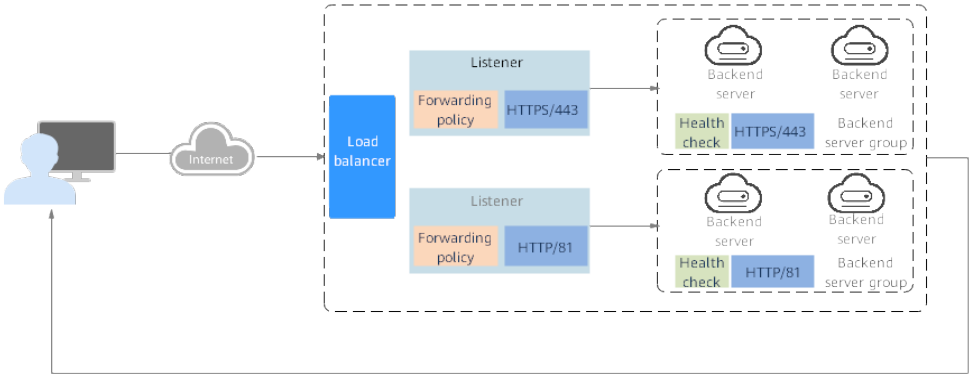
Table 2-3 Advantage details

High availability	Shared load balancers can route traffic across AZs, ensuring that your services are uninterrupted. If servers in an AZ are unhealthy, ELB automatically routes traffic to healthy servers in other AZs. Shared load balancers provide a comprehensive health check system to ensure that incoming traffic is only routed to healthy backend servers, which improves the availability of your applications.
Multiple protocols	Shared load balancers support TCP, UDP, HTTP, and HTTPS protocols to route requests to different types of applications.
Ease-of-use	Shared load balancers provide a diverse set of algorithms that allow you to configure different traffic routing policies to meet your requirements while keeping deployments simple.
High reliability	Load balancers can distribute across AZs more evenly.

3 How ELB Works

Overview

Figure 3-1 How ELB works



The following describes how ELB works:

1. A client sends requests to your application.
2. Each listener added to your load balancer uses the protocol and port you have configured to receive the requests.
3. The load balancer forwards requests.
   a. Each listener forwards the requests to the associated backend server group based on the routing rules you configure.
   b. If there are forwarding policies, each listener evaluates if incoming requests match any forwarding policy. If a match is found, the request is forwarded according to the forwarding action.
4. Healthy backend servers in the backend server group receive the requests based on the load balancing algorithm and the routing rules you specify in the forwarding policy, handle the requests, and return results to the client.

How requests are routed depends on the **load balancing algorithms** configured for each backend server group. If the listener uses HTTP or HTTPS, how requests are routed also depends on the **forwarding policies** configured for the listener.

Load Balancing Algorithms

Dedicated load balancers support four load balancing algorithms: weighted round robin, weighted least connections, source IP hash, and connection ID.

Shared load balancers support weighted round robin, weighted least connections, and source IP hash.

Weighted Round Robin

Figure 3-2 shows an example of how requests are distributed using the weighted round robin algorithm. Two backend servers are in the same AZ and have the same weight, and each server receives the same proportion of requests.

Figure 3-2 Traffic distribution using the weighted round robin algorithm

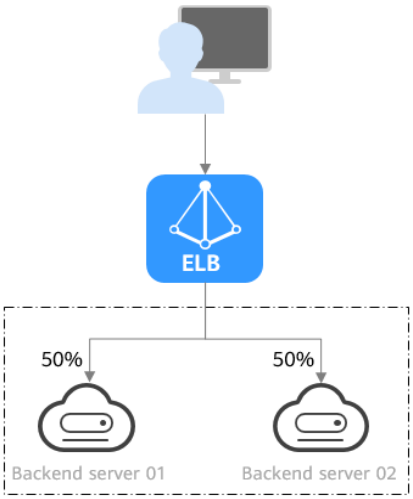


Table 3-1 Weighted round robin

Description	Requests are routed to backend servers in sequence based on their weights. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests.
When to Use	This algorithm is typically used for short connections, such as HTTP connections. • Flexible load balancing: When you need more refined load balancing, you can set a weight for each backend server to specify the percentage of requests to each server. For example, you can set higher weights to backend servers with better performance so that they can process more requests. • Dynamic load balancing: You can adjust the weight of each backend server in real time when the server performance or load fluctuates.

Disadvantages	• You need to set a weight for each backend server. If you have a large number of backend servers or your services require frequent adjustments, setting weights would be time-consuming. • If the weights are inappropriate, the requests processed by each server may be imbalanced. As a result, you may need to frequently adjust server weights.

Weighted Least Connections

Figure 3-3 shows an example of how requests are distributed using the weighted least connections algorithm. Two backend servers are in the same AZ and have the same weight, 100 connections have been established with backend server 01, and 50 connections have been established with backend server 02. New requests are preferentially routed to backend server 02.

Figure 3-3 Traffic distribution using the weighted least connections algorithm

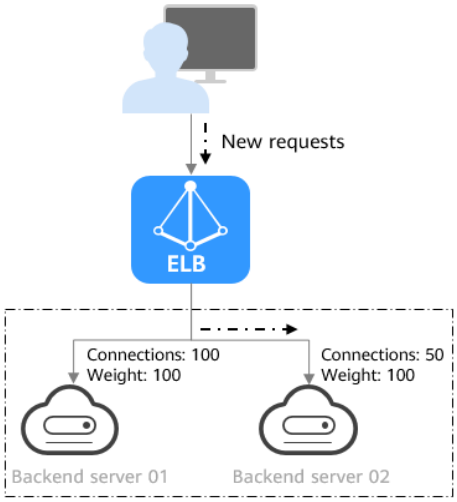


Table 3-2 Weighted least connections

Description	Requests are routed to the server with the lowest connections-to-weight ratio. In addition to the number of connections, each server is assigned a weight based on its capacity. Requests are routed to the server with the lowest connections-to-weight ratio.

When to Use	This algorithm is often used for persistent connections, such as connections to a database. • Flexible load balancing: Load balancers distribute requests based on the number of established connections and the weight of each backend server and route requests to the server with the lowest connections-to-weight ratio. This helps prevent servers from being underloaded or overloaded. • Dynamic load balancing: When the number of connections to and loads on backend servers change, you can use the weighted least connection algorithm to dynamically adjust the requests distributed to each server in real time. • Stable load balancing: You can use this algorithm to reduce the peak loads on each backend server and improve service stability and reliability.
Disadvantages	• Complex calculation: The weighted least connections algorithm needs to calculate and compare the number of connections established with each backend server in real time before selecting a server to route requests. • Dependency on connections to backend servers: The algorithm routes requests based on the number of connections established with each backend server. If monitoring data is inaccurate or outdated, requests may not be distributed evenly across backend servers. The algorithm can only collect statistics on the connections between a given load balancer and a backend server, but cannot obtain the total number of connections to the backend server if it is associated with multiple load balancers. • Too many loads on new servers: If existing backend servers have to handle a large number of requests, new requests will be routed to new backend servers. This may deteriorate new servers or even cause them to fail.

Source IP Hash

Figure 3-4 shows an example of how requests are distributed using the source IP hash algorithm. Two backend servers are in the same AZ and have the same weight. If backend server 01 has processed a request from IP address A, the load balancer will route new requests from IP address A to backend server 01.

Figure 3-4 Traffic distribution using the source IP hash algorithm

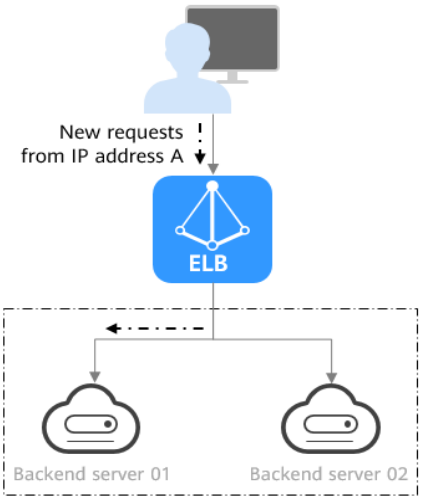


Table 3-3 Source IP hash

Description	The source IP hash algorithm calculates the source IP address of each request and routes requests from the same IP address to the same backend server.
When to Use	This algorithm is often used for applications that need to maintain user sessions or state. • Session persistence: Source IP hash ensures that requests with the same source IP address are distributed to the same backend server. • Data consistency: Requests with the same hash value are distributed to the same backend server. • Load balancing: In scenarios that have high requirements for load balancing, this algorithm can distribute requests to balance loads among servers.
Disadvantages	• Imbalanced loads across servers: This algorithm tries its best to ensure request consistency when backend servers are added or removed. If the number of backend servers decreases, some requests may be redistributed, causing imbalanced loads across servers. • Complex calculation: This algorithm calculates the hash values of requests based on hash factors. If servers are added or removed, some requests may be redistributed, making calculation more difficult.

Connection ID

Figure 3-5 shows an example of how requests are distributed using the connection ID algorithm. Two backend servers are in the same AZ and have the same weight. If backend server 01 has processed a request from client A, the load balancer will route new requests from client A to backend server 01.

Figure 3-5 Traffic distribution using the connection ID algorithm

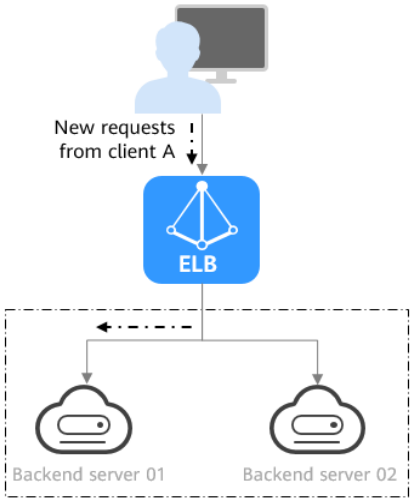


Table 3-4 Connection ID

Description	The connection ID algorithm calculates the QUIC connection ID and routes requests with the same hash value to the same backend server. A QUIC ID identifies a QUIC connection. This algorithm distributes requests by QUIC connection. You can use this algorithm to distribute requests only to QUIC backend server groups.
When to Use	This algorithm is typically used for QUIC requests. • Session persistence: The connection ID algorithm ensures that requests with the same hash value are distributed to the same backend server. • Data consistency: Requests with the same hash value are distributed to the same backend server. • Load balancing: In scenarios that have high requirements for load balancing, this algorithm can distribute requests to balance loads among servers.
Disadvantages	• Imbalanced loads across servers: This algorithm tries its best to ensure request consistency when backend servers are added or removed. If the number of backend servers decreases, some requests may be redistributed, causing imbalanced loads across servers. • Complex calculation: This algorithm calculates the hash values of requests based on hash factors. If servers are added or removed, some requests may be redistributed, making calculation more difficult.

Factors Affecting Load Balancing

In addition to the load balancing algorithms, factors that affect load balancing include the connection type (persistent connection or short connection), session stickiness, and server performance and weights.

Assume that there are two backend servers with the same weight (not zero), the weighted least connections algorithm is selected, sticky sessions are not enabled, and 100 connections have been established with backend server 01, and 50 connections with backend server 02.

When client A wants to access backend server 01, the load balancer establishes a persistent connection with backend server 01 and continuously routes requests from client A to backend server 01 before the persistent connection is disconnected. When other clients access backend servers, the load balancer routes the requests to backend server 02 using the weighted least connects algorithm.

NOTE

If backend servers are declared unhealthy or their weights are set to 0, the load balancer will not route any request to the backend servers.

For details about the load balancing algorithms, see **Load Balancing Algorithms**.

If requests are not evenly routed, troubleshoot the issue by performing the operations described in **How Do I Check Whether Traffic Is Evenly Distributed?**

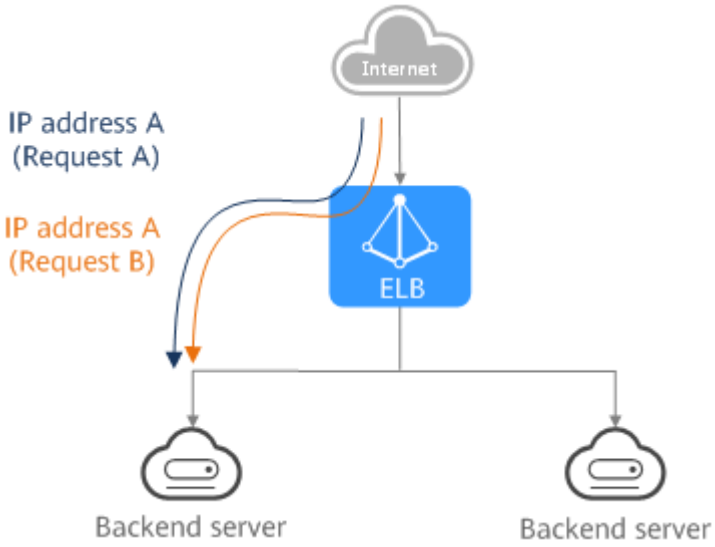
4 Application Scenarios

Heavy-Traffic Applications

For an application with heavy traffic, such as a large portal or mobile app store, ELB evenly distributes incoming traffic across backend servers, balancing the load while ensuring steady performance.

Sticky sessions ensure that requests from one client are always forwarded to the same backend server for fast processing.

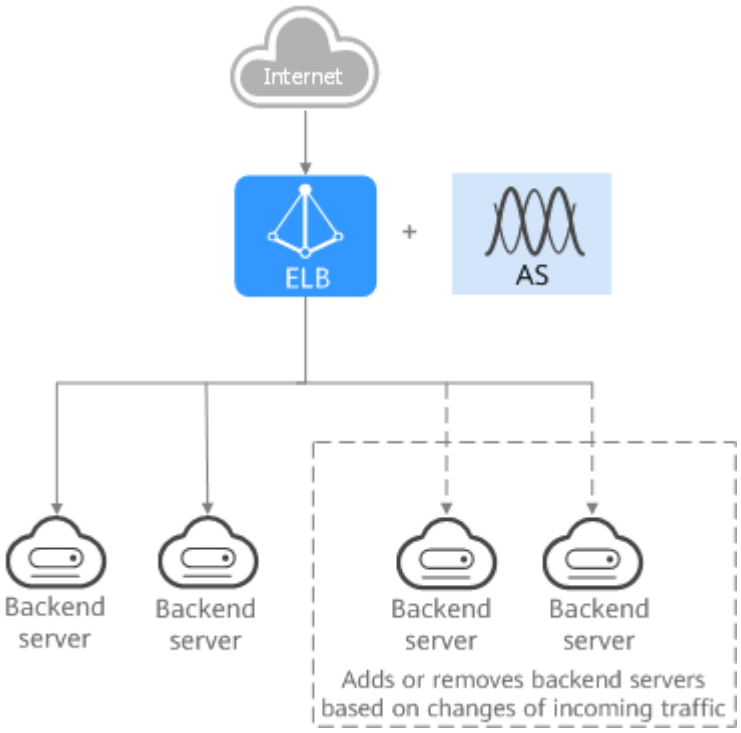
Figure 4-1 Session stickiness



Applications with Predictable Peaks and Troughs in Traffic

For an application that has predictable peaks and troughs in traffic volumes, ELB works with Auto Scaling to automatically add servers during promotions when there are sudden traffic spikes, and then remove them when traffic returns to normal. This helps you improve resource availability and reduce IT costs.

Figure 4-2 Flexible scalability

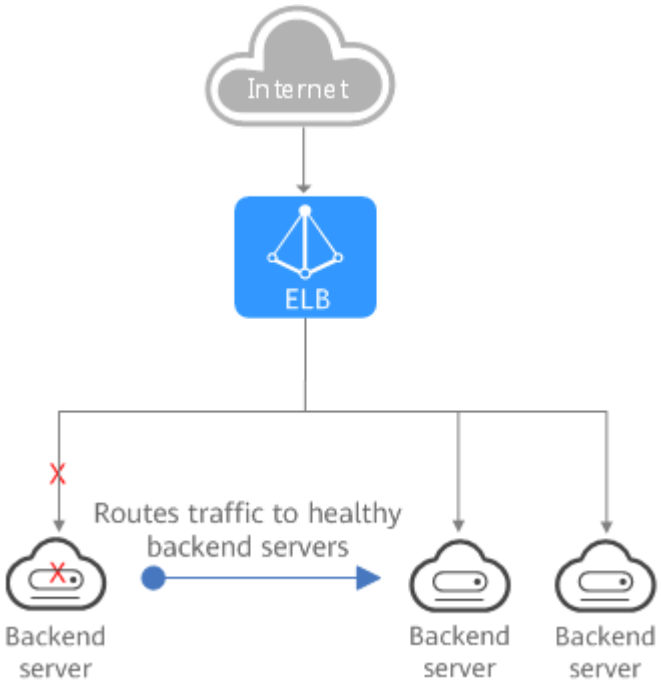


Zero SPOFs

ELB routinely performs health checks on backend servers to monitor their health. If any backend server is detected unhealthy, ELB will not route requests to this server until it recovers.

This makes ELB a good choice for running services that require high reliability, such as websites and toll collection systems.

Figure 4-3 Eliminating SPOFs

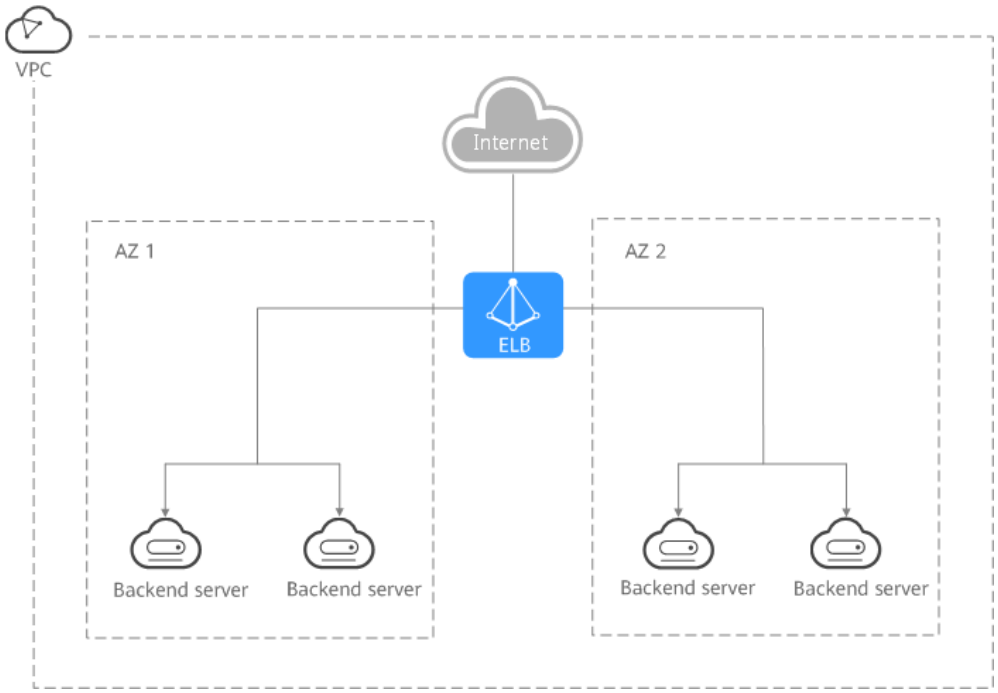


Cross-AZ Load Balancing

ELB can distribute traffic across AZs. When an AZ becomes faulty, ELB distributes traffic across backend servers in other AZs.

ELB is ideal for banking, policing, and large application systems that require high availability.

Figure 4-4 Traffic distribution to servers in one or more AZs



5 Differences Between Dedicated and Shared Load Balancers

5.1 Load Balancer Types

Introduction to ELB

Elastic Load Balance (ELB) automatically distributes incoming traffic across servers to balance their workloads, increasing the service capabilities and fault tolerance of your applications.

Load Balancer Types

ELB provides shared load balancers and dedicated load balancers for you to choose from.

Table 5-1 Load balancer types

Item	Dedicated Load Balancer	Shared Load Balancer
Deployment mode	A dedicated load balancer gets dedicated resources. Its performance is never affected by the loads on other load balancers. In addition, there are a wide range of specifications available for you to choose from.	They are deployed in clusters and share resources with other instances.

Item	Dedicated Load Balancer	Shared Load Balancer
Specifications	• Elastic specifications: You are charged for how long each load balancer is running and the number of LCUs you use. • Fixed specifications: Multiple specifications are available for you to select to best meet your needs. For details, see **Specifications of Dedicated Load Balancers**.	N/A

Item	Dedicated Load Balancer	Shared Load Balancer
AZ	You can select one or more AZs as needed. • If requests are from the Internet, the load balancer in each AZ you select routes the requests based on source IP addresses. If you deploy a load balancer in multiple AZs, the requests the load balancers can handle will be multiplied by the number of AZs. • For requests from a private network: – If clients are in the same AZ as the load balancer, requests are distributed by the load balancer in this AZ. If the load balancer goes down, requests are distributed by the load balancer in another AZ. If the load balancer is healthy but the connections that the load balancer needs to handle exceed the amount defined in the specifications, service may be interrupted. To address this issue, you need upgrade specifications. You can monitor traffic usage on private network by AZ. – If clients are in an AZ that is not selected when you create the load balancer, requests are distributed by the load balancer in each AZ you select based on source IP addresses. • If requests are from a Direct Connect connection, the load balancer in the same AZ as the Direct	N/A

Item	Dedicated Load Balancer	Shared Load Balancer
	Connect connection routes the requests. If the load balancer is unavailable, requests are distributed by the load balancer in another AZ. • If clients are in a VPC that is different from where the load balancer works, the load balancer in the AZ where the original VPC subnet resides routes the requests. If the load balancer is unavailable, requests are distributed by the load balancer in another AZ.	
Billing item	• Fixed specifications: billed by the LCUs based on the specifications you select. • Elastic specifications: billed by how many LCUs you use and how long a load balancer is retained in your account.	You are charged for how long you use each load balancer if guaranteed performance is enabled.

Feature Comparison

Table 5-2 Feature comparison

Item	Dedicated Load Balancer	Shared Load Balancer
Capabilities	Powerful capabilities to process Layer 4 and Layer 7 requests, advanced forwarding policies, and multiple protocols	Basic capabilities to process Layer 4 and Layer 7 requests
Application scenarios	Heavy-traffic and highly concurrent services, such as large websites, cloud-native applications, IoV, and multi-AZ disaster recovery applications	Services with low traffic, such as small websites and common HA applications
Frontend protocols	TCP, UDP, HTTP, and HTTPS	TCP, UDP, HTTP, and HTTPS

Item	Dedicated Load Balancer	Shared Load Balancer
Backend protocols	TCP, UDP, HTTP, HTTPS, and QUIC	TCP, UDP, and HTTP
Forwarding capabilities	Provide powerful Layer 4 and Layer 7 processing capabilities to based on the following: • Forwarding rules: domain name, path, HTTP request method, HTTP header, query string, and CIDR block • Actions: forward to a backend server group, redirect to another listener, redirect to another URL, rewrite, and return a specific response body	Provide basic Layer 4 and Layer 7 processing capabilities to based on the following: • Forwarding rules: domain name and path • Actions: forward to a backend server group and redirect to another listener
Key features of backend server groups	• Health check • Sticky session • Slow start • Support for association with multiple load balancers and listeners	• Health check • Sticky session • Association with only one listener
Load balancing algorithms	• Weighted round robin • Weighted least connections • Source IP hash • Connection ID	• Weighted round robin • Weighted least connections • Source IP hash
Forwarding modes of backend server groups	• Load balancing • Active/Standby forwarding	Load balancing
Backend type	• ECS • IP as backend server • Supplementary network interface • CCE Turbo cluster	• ECS • CCE Turbo cluster

5.2 Feature Comparison Details

Protocols

Table 5-3 Protocols supported by each load balancer type

Protocol	Description	Dedicated Load Balancer	Shared Load Balancer
TCP/UDP (Layer 4)	After receiving TCP or UDP requests from the clients, the load balancer directly routes the requests to backend servers. Load balancing at Layer 4 features high routing efficiency.	Supported	Supported
HTTP/HTTPS (Layer 7)	After receiving an access request, the listener needs to identify the request and forward it based on the header fields. Load balancing at Layer 7 provides some advanced features such as encrypted transmission and cookie-based sticky sessions.	Supported	Supported
HTTPS support	HTTPS can be used as both the frontend and backend protocol.	Supported	Not supported
HTTP/2	Hypertext Transfer Protocol 2.0 (HTTP/2) is a new version of the HTTP protocol. It is compatible with HTTP/1.X and provides improved performance and security. Only HTTPS listeners support this feature.	Supported	Supported
WebSocket	WebSocket is a new HTML5 protocol that provides full-duplex communication between the browser and the server. WebSocket saves server resources and bandwidth, and enables real-time communication.	Supported	Supported

Network Configurations

Table 5-4 Network configuration comparison

Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Public IPv4 network	The load balancer routes requests from the clients to backend servers over the Internet.	Supported	Supported
Private IPv4 network	The load balancer routes requests from the clients to backend servers in a VPC.	Supported	Supported
IPv6 network	Load balancers can route requests from IPv6 clients.	Supported	Not supported
Changing a private IPv4 address	You can change the private IPv4 address into another one in the current subnet or other subnets.	Supported	Not supported
Binding or unbinding an EIP	You can bind an EIP to a load balancer or unbind the EIP from a load balancer based on service requirements.	Supported	Supported
Modifying the bandwidth	You can change the bandwidth of public network load balancers as required.	Supported	Supported

Key Features of Listeners

Table 5-5 Comparison of key features

Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Access Control	You can add IP addresses to a whitelist or blacklist to control access to a listener. • A whitelist allows specified IP addresses to access the listener. • A blacklist denies access from specified IP addresses.	Supported	Supported

Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Mutual Authentication	This feature allows the clients and the load balancer to authenticate each other. Only authenticated clients will be allowed to access the load balancer. Mutual authentication is supported only by HTTPS listeners.	Supported	Supported
SNI	Server Name Indication (SNI) is an extension to TLS and is used when a server uses multiple domain names and certificates. After SNI is enabled, certificates corresponding to the domain names are required. SNI can be enabled only for HTTPS listeners.	Supported	Supported
Transfer Client IP Address	This feature allows backend servers to obtain the real IP addresses of the clients. This feature is enabled for dedicated load balancers by default and cannot be disabled.	Supported	Supported
Advanced features of HTTP/HTTPS listeners			
Default Security Policy	You can select appropriate security policies to improve service security when you add HTTPS listeners. A security policy is a combination of TLS protocols and cipher suites.	Supported	Supported
Custom Security Policy	You can select a TLS protocol and cipher suite to custom a security policy when you add HTTPS listeners.	Supported	Not supported
Transfer Load Balancer EIP	You can store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header and pass it to backend servers.	Supported	Supported

Forwarding Capabilities

You can configure forwarding policies for HTTP or HTTPS listeners to forward requests to different backend server groups. are available only for dedicated load balancers.

Table 5-6 and **Table 5-7** describe the available forwarding rules and actions.

Table 5-6 Forwarding rules supported by each load balancer type

Forwarding Rule	Description	Dedicated Load Balancer	Shared Load Balancer
Domain name	Route requests based on the domain name. The domain name in the request must exactly match that in the forwarding policy.	Supported	Supported
Path	Route requests based on the specified path. There are three matching rules: exact match, prefix match, and regular expression match.	Supported	Supported
HTTP request method	Route requests based on the HTTP method. The options include GET, POST, PUT, DELETE, PATCH, HEAD, and OPTIONS.	Supported	Not supported
HTTP header	Route requests based on the HTTP header. An HTTP header consists of a key and one or more values. You need to configure the key and values separately.	Supported	Not supported
Query string	Route requests based on the query string.	Supported	Not supported
CIDR block	Route requests based on source IP addresses from where the requests originate.	Supported	Not supported

Table 5-7 Actions supported by each load balancer type

Action	Description	Dedicated Load Balancer	Shared Load Balancer
Forward to a backend server group	Forward requests to the specified backend server group.	Supported	Supported
Redirect to another listener	Redirect requests to an HTTPS listener, which then routes the requests to its associated backend server group.	Supported	Supported

Action	Description	Dedicated Load Balancer	Shared Load Balancer
Redirect to another URL	Redirect requests to the configured URL. When clients access website A, the load balancer returns 302 or any other 3xx status code and automatically redirects the clients to website B. You can custom the redirection URL that will be returned to the clients.	Supported	Not supported
Return a specific response body	Return a fixed response to the clients. You can custom the status code and response body that load balancers directly return to the clients without the need to route the requests to backend servers.	Supported	Not supported

Key Features of Backend Server Groups

Table 5-8 Key features supported by each load balancer type

Key Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Backend server group reuse	In an enterprise project, a backend server group can be associated with multiple load balancers and listeners.	Supported	Not supported
Health check	ELB periodically sends requests to backend servers to check their running statuses. This process is called health check. You can perform health checks to determine whether a backend server is available.	Supported	Supported
Sticky session	Requests from the same client will be routed to the same backend server during the session.	Supported	Supported

Key Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Slow start	The load balancer linearly increases the proportion of requests to the new backend servers added to the backend server group. Slow start gives applications time to warm up and respond to requests with optimal performance.	Supported	Not supported
Active/Standby forwarding	The load balancer routes the traffic to the active server if it works normally and to the standby server if the active server becomes unhealthy. You must add two backend servers to the backend server group, one acting as the active server and the other as the standby server.	Supported	Not supported

Load Balancing Algorithms

Table 5-9 Load balancing algorithm comparison

Load Balancing Algorithm	Description	Dedicated Load Balancer	Shared Load Balancer
Weighted round robin	Route requests to backend servers using the round robin algorithm. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests.	Supported	Supported
Weighted least connections	Route requests to backend servers with the smallest ratio (current connections divided by weight).	Supported	Supported
Source IP hash	Route requests from the same client to the same backend server within a period of time.	Supported	Supported

Load Balancing Algorithm	Description	Dedicated Load Balancer	Shared Load Balancer
Connection ID	Calculate the source IP address of each request using the consistent hashing algorithm to obtain a unique hash key and route the requests to the particular server based on the generated key.	Supported	Not supported

Backend Server Type

Table 5-10 Supported backend server types

Backend Server Type	Description	Dedicated Load Balancer	Shared Load Balancer
IP as a backend	You can add servers in a peer VPC, in a VPC that is in another region and connected through a cloud connection, or in an on-premises data center at the other end of a Direct Connect or VPN connection, by using the server IP addresses.	Supported	Not supported
Supplementary network interface	You can attach supplementary network interfaces to backend servers.	Supported	Not supported
ECS	You can use load balancers to distribute incoming traffic across ECSs.	Supported	Supported

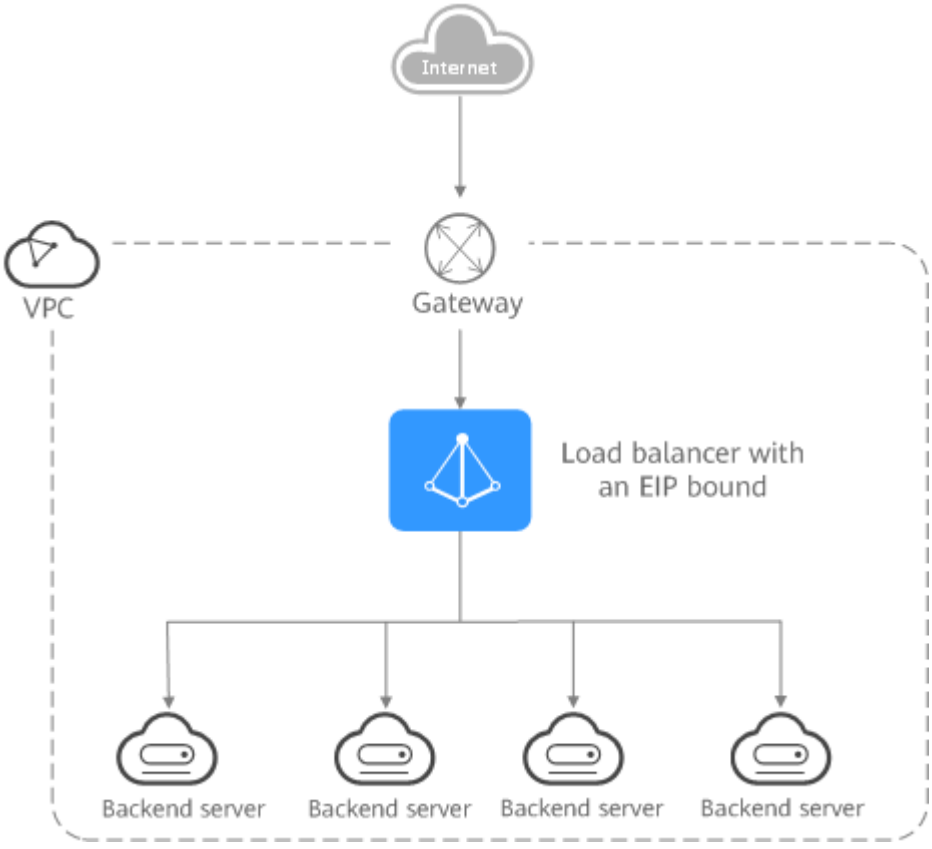
6 Load Balancing on a Public or Private Network

A load balancer can work on either a public or private network.

Load Balancing on a Public Network

You can bind an EIP to a load balancer so that it can receive requests from the Internet and route the requests to backend servers.

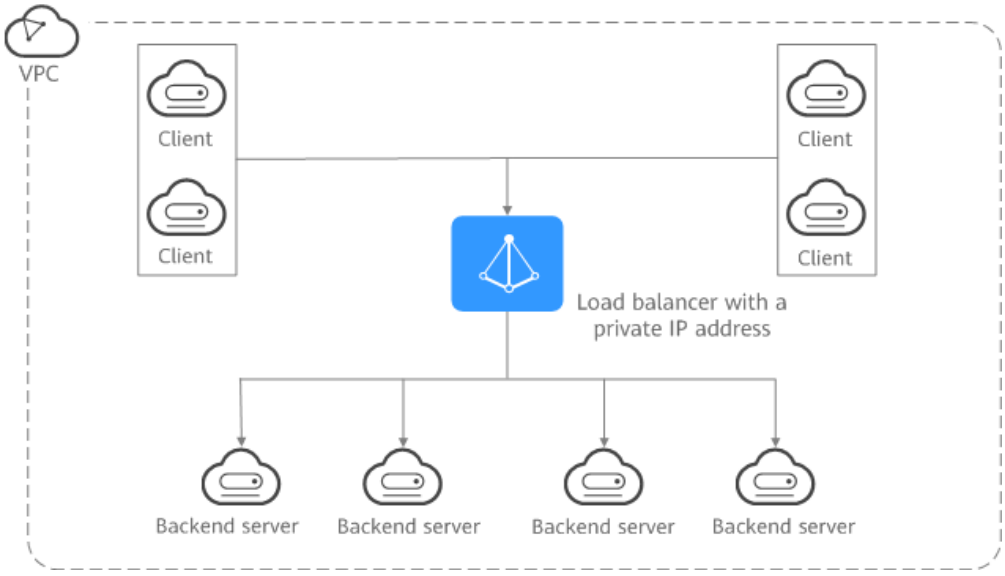
Figure 6-1 Load balancing on a public network



Load Balancing on a Private Network

A load balancer has only a private IP address to receive requests from clients in a VPC and routes the requests to backend servers in the same VPC. This type of load balancer can only be accessed in a VPC.

Figure 6-2 Load balancing on a private network



Network Types and Load Balancers

Table 6-1 Dedicated load balancers and their network types

Load Balancer Type	Network Type	Description
Dedicated load balancers	Public IPv4 network	Each load balancer has an IPv4 EIP bound to enable it to route requests over the Internet.
	Private IPv4 network	Each load balancer has only a private IPv4 address and can route requests in a VPC.
	IPv6 network	Each load balancer has an IPv6 address bound. • If the IPv6 address is added to a shared bandwidth, the load balancer can route requests over the Internet. • If the IPv6 address is not added to a shared bandwidth, the load balancer can route requests only in a VPC.

Table 6-2 Shared load balancers and their network types

Load Balancer Type	Network Type	Description
Shared load balancers	Public IPv4 network	Each load balancer has an EIP bound to enable it to route requests over the Internet.
	Private IPv4 network	Each load balancer has only a private IP address and can route requests in a VPC. **NOTE** Shared load balancers support private IPv4 networks by default. The private IP address of a shared load balancer cannot be changed.

7 Network Traffic Paths

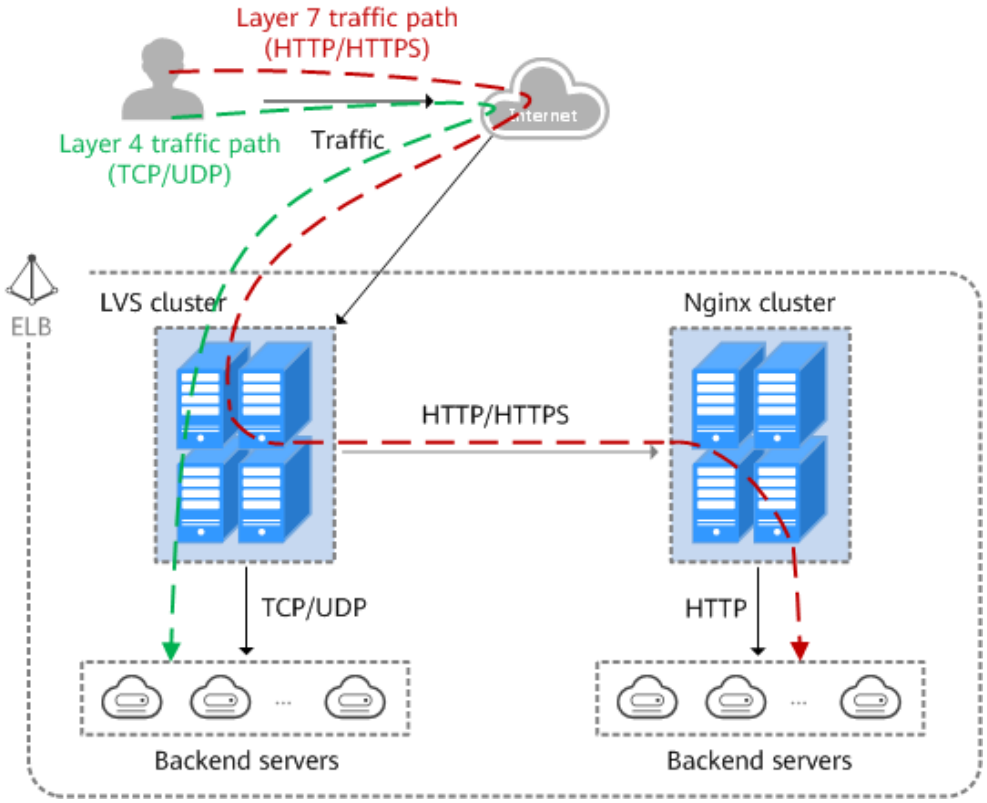
Load balancers communicate with backend servers over a private network.

- If backend servers process only requests routed from load balancers, there is no need to assign EIPs or create NAT gateways.
- If backend servers need to provide Internet-accessible services or access the Internet, you must assign EIPs or create NAT gateways.

Inbound Network Traffic Paths

The listeners' configurations determine how load balancers distribute incoming traffic.

Figure 7-1 Inbound network traffic



When a listener uses TCP or UDP to receive incoming traffic:

- Incoming traffic is routed only through the LVS cluster.
- The LVS cluster directly routes incoming traffic to backend servers using the load balancing algorithm you select when you add the listener.

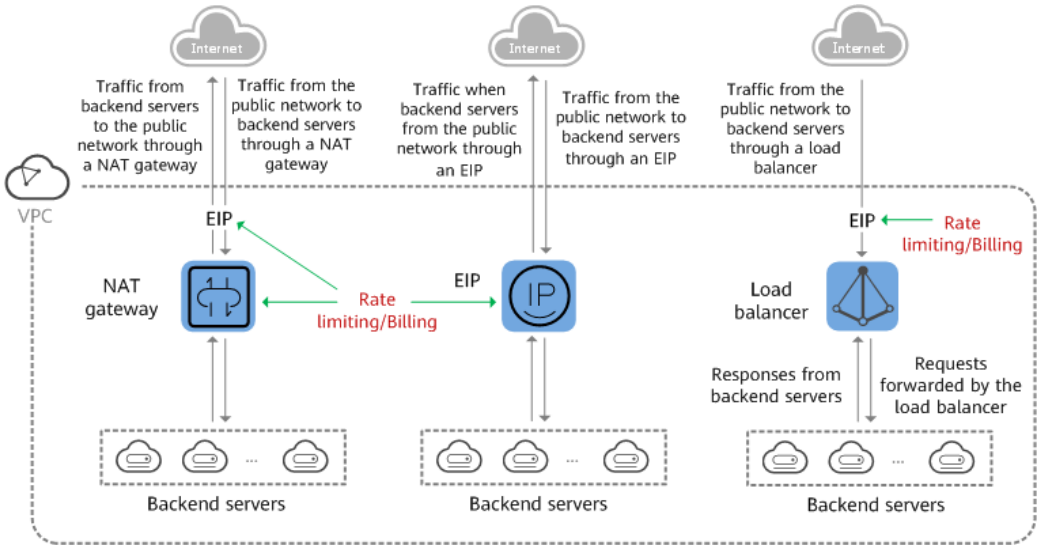
When a listener uses HTTP or HTTPS to receive incoming traffic:

- Incoming traffic is routed first to the LVS cluster, then to the Nginx cluster, and finally across backend servers.
- For HTTPS traffic, the Nginx cluster validates certificates and decrypts data packets before distributing the traffic across backend servers using HTTP.

Outbound Network Traffic Paths

The outbound traffic is routed back the same way the traffic came in.

Figure 7-2 Outbound network traffic



- Because the load balancer receives and responds to requests over the Internet, traffic transmission depends on the bandwidth, which is not limited by ELB. The load balancer communicates with backend servers over a private network.
- If you have a NAT gateway, it receives and responds to incoming traffic. The NAT gateway has an EIP bound, through which backend servers can access the Internet and provide services accessible from the Internet. Although there is a restriction on the connections that can be processed by a NAT gateway, traffic transmission depends on the bandwidth
- If each backend server has an EIP bound, they receive and respond to incoming traffic directly. Traffic transmission depends on the bandwidth.

8 Specifications of Dedicated Load Balancers

When you create a dedicated load balancer, you can select elastic or fixed specifications based on your service requirements. **Table 8-1** lists the differences between the two types of specifications.

Table 8-1 Specifications comparison

Item	Elastic	Fixed
Application scenarios	• For fluctuating traffic • When you need to use resources temporarily or for urgent purposes	• For stable traffic • When you need to use resources for a long term
Network (TCP/UDP) load balancer performance	The performance multiplies as the number of AZs increases. **Table 8-3** shows the maximum performance in an AZ.	The performance multiplies as the number of AZs increases. **Table 8-6** shows the maximum performance in an AZ.
Application (HTTP/HTTPS) load balancer performance	The performance multiplies as the number of AZs increases. **Table 8-3** shows the maximum performance in an AZ.	The performance multiplies as the number of AZs increases. **Table 8-7** shows the maximum performance in an AZ.
Billing items	• LCU • Load balancer	LCU
Capabilities	Same	

Elastic Specifications

If your service traffic fluctuates greatly, you can choose elastic specifications and select either network load balancing (TCP/UDP) or application load balancing (HTTP/HTTPS), or both that best meet your service needs.

NOTE

The listener protocol must match the load balancing type. For example, if you select application load balancing, you can only add HTTP and HTTPS listeners to this load balancer.

Table 8-2 describes the dimensions about elastic specifications. When the traffic exceeds the specifications defined in Table 8-3, new requests will be discarded.

Table 8-2 Elastic specification dimensions

Maximum Connections	Indicates the maximum number of concurrent connections that a load balancer can handle per minute. If the number reaches the maximum connections that is defined in the elastic specifications, new requests will be discarded to ensure the performance of established connections.
Connections Per Second (CPS)	Indicates the number of new connections that a load balancer can establish per second. If the number reaches the CPS that is defined in the elastic specifications, new requests will be discarded to ensure the performance of established connections.
Queries Per Second (QPS)	Indicates the number of HTTP or HTTPS requests sent to a backend server per second. If the QPS reaches that is defined in the elastic specifications, new requests will be discarded to ensure the performance of established connections.
Bandwidth (Mbit/s)	Indicates the maximum amount of data that can be transmitted over a connection per second.

Table 8-3 Maximum elastic specifications

Protocol	Maximum Connections	CPS	QPS	Bandwidth (Mbit/s)
Network load balancing (TCP/UDP)	20,000,000	400,000	N/A	10,000
Network load balancing (TLS)	20,000,000	400,000	N/A	10,000

Protocol	Maximum Connections	CPS	QPS	Bandwidth (Mbit/s)
Application load balancing (HTTP)	8,000,000	80,000	160,000	10,000
Application load balancing (HTTPS)	8,000,000	80,000	160,000	10,000

CAUTION

Available elastic specifications are displayed on the console and may vary depending on regions.

Fixed Specifications

Load balancers are available in different fixed specifications. Choose the specifications that best meet your needs. When your traffic exceeds what defined in your selected specifications, new requests will be discarded. Each specification has the following dimensions.

Table 8-4 Fixed specification dimensions

Maximum Connections	Indicates the maximum number of concurrent connections that a load balancer can handle per minute. If the number reaches the maximum connections that is defined in your selected fixed specifications, new requests will be discarded to ensure the performance of existing connections.
CPS	Indicates the number of new connections that a load balancer can establish per second. If the number reaches the CPS that is defined in your selected fixed specifications, new requests will be discarded to ensure the performance of established connections. HTTPS listeners need to create SSL handshakes to establish connections with clients, and such SSL handshakes occupy more system resources than HTTP listeners. For example, a small I application load balancer can establish 2,000 new HTTP connections per second but only 200 new HTTPS connections per second. For details, see **Table 8-5**.

QPS	Indicates the number of HTTP or HTTPS requests sent to a backend server per second. If the QPS reaches that is defined in your selected fixed specifications, new requests will be discarded to ensure the performance of established connections.
Bandwidth (Mbit/s)	Indicates the maximum amount of data that can be transmitted over a connection per second.

For a small I application load balancer:

- If you only add an HTTP listener, the load balancer can establish up to 2,000 new HTTP connections.
- If you only add an HTTPS listener, the load balancer can establish up to 200 new HTTPS connections.
- If you add an HTTPS listener and an HTTP listener, the new connections are calculated using the following formula:

  New connections = New HTTP connections + New HTTPS connections × Ratio of HTTP connections to HTTPS connections

  For a small I application load balancer, the ratio of HTTP connections to HTTPS connections is 10. **Table 8-5** shows how many new connections a small I application load balancer can establish.

  Table 8-5 New connections that a small I application load balancer can establish

Parameter	Scenario 1	Scenario 2
New HTTP connections	1,000	1,000
New HTTPS connections	50	150
New HTTP and HTTPS connections	1,000 + 50 × 10 = 1,500	1,000 + 150 × 10 = 2,500
Description	• The new connections do not reach the CPS (HTTP) that a small I application load balancer can handle, so new requests can be properly routed.	• The new connections exceed the CPS (HTTP) that a small I application load balancer can handle, so new requests will be discarded.

NOTE

Details in **Table 8-5** are for reference only.

Table 8-6 and Table 8-7 list the fixed specifications of dedicated load balancers.

CAUTION

- **Available fixed specifications on the console may vary depending on the resources in different regions.**
- The listener protocol must match the load balancing type. For example, if you select application load balancing, you can only add HTTP and HTTPS listeners to this load balancer.

Table 8-6 Fixed specifications for a network load balancer (TCP/UDP)

Specification	Maximum Connections (TCP/UDP)	Maximum Connections (TLS)	CPS (TCP/UDP)	CPS (TLS)	Bandwidth (Mbit/s)	LCUs in an AZ
Small I	500,000	30,000	10,000	500	50	10
Small II	1,000,000	60,000	20,000	1,000	100	20
Medium I	2,000,000	120,000	40,000	2,000	200	40
Medium II	4,000,000	240,000	80,000	4,000	400	80
Large I	10,000,000	600,000	200,000	10,000	1,000	200
Large II	20,000,000	1,200,000	400,000	20,000	2,000	400

Table 8-7 Fixed specifications for an application load balancer (HTTP/HTTPS)

Specification	Maximum Connections	CPS (HTTP)	CPS (HTTPS)	QPS (HTTP)	QPS (HTTPS)	Bandwidth (Mbit/s)	LCUs in an AZ
Small I	200,000	2,000	200	4,000	2,000	50	10
Small II	400,000	4,000	400	8,000	4,000	100	20

Specification	Maximum Connections	CPS (HTTP)	CPS (HTTPS)	QPS (HTTP)	QPS (HTTPS)	Bandwidth (Mbit/s)	LCUs in an AZ
Medium I	800,000	8,000	800	16,000	8,000	200	40
Medium II	2,000,000	20,000	2,000	40,000	20,000	400	100
Large I	4,000,000	40,000	4,000	80,000	40,000	1,000	200
Large II	8,000,000	80,000	8,000	160,000	80,000	2,000	400

NOTE

- If you add multiple listeners to a load balancer, the sum of QPS values of all listeners cannot exceed the QPS defined in each specification.
- The bandwidth is the upper limit of the inbound or the outbound traffic. For example, for small I load balancers, the inbound or outbound traffic cannot exceed 50 Mbit/s.
- The bandwidth included in each specification is the maximum bandwidth provided by ELB. If the maximum bandwidth is exceeded, the network performance may be affected.

9 Notes and Constraints

You can create dedicated and shared load balancers on ELB console. This section describes the quotas and restrictions that apply to ELB resources.

ELB Resource Quotas

Quotas put limits on the quantity of resources, such as the maximum number of ECSs or EVS disks that you can create.

Table 9-1 lists the default quotas of ELB resources. You can view your quotas by referring to **How Do I View My Quotas?**

If the existing resource quota cannot meet your service requirements, you can request an increase by referring to **How Do I Apply for a Higher Quota?**

Table 9-1 ELB resource quotas

Resource	Description	Default Quota
Load balancers	Load balancers per account	50
Listeners	Listeners per account	100
Forwarding policies	Forwarding policies per account	500
Backend server groups	Backend server groups per account	500
Certificates	Certificates per account	120
Backend servers	Backend servers per account	500
Listeners per load balancer	Listeners that can be added to a load balancer	50
Forwarding policies per listener	Maximum number of forwarding policies that can be added to a listener	100

NOTE

The quotas apply to a single account.

Other Quotas

In addition to quotas described in **ELB Resource Quotas**, some other resources that you can use are also limited.

You can call APIs to query quotas of the resources described in **Table 9-2** by referring to **Querying Quotas**.

Table 9-2 Other quotas

Resource	Description	Default Quota
Forwarding conditions per forwarding policy	Forwarding conditions that can be added to a forwarding policy	10
Backend server groups per forwarding policy	Maximum number of backend server groups that can be added to a forwarding policy	5
Backend servers per backend server group	Backend servers that can be added to a backend server group	500
Listeners per backend server group	Maximum number of listeners that can be associated with a backend server group	50
IP address group		
IP address groups per load balancer	IP address groups per account	50
Listeners per IP address group	Listeners that can be associated with an IP address group	50
IP addresses per IP address group	IP addresses that can be added to an IP address group	300

Load Balancer

- Before creating a load balancer, you must plan its region, type, protocol, and backend servers. For details, see **Preparations for Creating a Load Balancer**.
- The maximum size of data that a load balancer can forward:
  - Layer 4 listeners: any
  - Layer 7 listeners:
    - 10 GB (file size)
    - 32 KB (the total size of the HTTP request line and HTTP request header)

Listener

- To ensure ELB performance and simplify management, you are advised to properly plan the maximum number of listeners that can be added to a load balancer based on your service requirements. If the default quota of listeners cannot meet your requirements, you can create more load balancers.
- The listener of a dedicated load balancer can be associated with a maximum of 50 backend server groups.
- A certificate can be associated with a maximum of 600 listeners.
- SNI certificates
  - Shared load balancers
    - An HTTPS listener can have up to 30 SNI certificates.
    - A certificate can have a maximum of 30 domain names. By default, all SNI certificates can have up to 30 domain names.
    - A domain name can contain a maximum of 100 characters, and the total length of all domain names cannot exceed 1,024 characters.
  - Dedicated load balancers
    - An HTTPS listener can have up to 30 SNI certificates by default. You can **request an increase to 50**.
    - A certificate can have a maximum of 100 domain names. By default, all SNI certificates can have up to 200 domain names.
    - A domain name can contain a maximum of 100 characters, and the total length of all domain names cannot exceed 10,000 characters.
- Once set, the frontend protocol and port of the listener cannot be modified.

Forwarding Policy

- Forwarding policies can be configured only for HTTP and HTTPS listeners.
- Forwarding policies must be unique.
- A maximum of 100 forwarding policies can be configured for a listener. If the number of forwarding policies exceeds the quota, the excess forwarding policies will not be applied.
- Forwarding conditions:
  - If the advanced forwarding policy is not enabled, each forwarding rule has only one forwarding condition.
  - If the advanced forwarding policy is enabled, each forwarding rule has up to 10 forwarding conditions.

Table 9-3 Restrictions on forwarding policies

Load Balancer Type	Advanced Forwarding	Forwarding Rule	Action	Reference
Shared	Not supported	Domain name and path	**Forward to another backend server group** and **Redirect to another listener**	**Forwarding Policy (Shared Load Balancers)**
Dedicated	Disabled	Domain name and path	**Forward to another backend server group** and **Redirect to another listener**	**Forwarding Policy (Dedicated Load Balancers)**
	Enabled	Domain name, path, HTTP request method, HTTP header, query string, cookie, and CIDR block	**Forward to another backend server group**, **Redirect to another listener**, , and **Return a specific response body**	**Advanced Forwarding (Dedicated Load Balancers)**

Backend Server Group

- The backend server group protocol must match the listener protocol as described in **Table 9-4**.

Table 9-4 The frontend and backend protocol

Load Balancer Specification	Frontend Protocol	Backend Protocol
Network load balancing	TCP	TCP
Network load balancing	UDP	• UDP • QUIC
Application load balancing	HTTP	HTTP
Application load balancing	HTTPS	• HTTP • HTTPS

Backend Server

- If **Transfer Client IP Address** is enabled, a server cannot serve as both a backend server and a client.
- An ECS can be added as a backend server for a maximum of 800 times. If it is added to the same backend server group, the port must be unique.

IP Address Group

You can configure a maximum of five IP address groups for an access control policy. You can add a maximum of 300 entries (including IP addresses and CIDR blocks) to each IP address group.

TLS Security Policy

You can create a maximum of 50 TLS security policies.

10 Permissions

If you need to assign different permissions to personnel in your enterprise to access your ELB resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access your cloud resources.

With IAM, you can create IAM users and assign permissions to control their access to specific resources. For example, if you want some software developers in your enterprise to use ELB resources but do not want them to delete these resources or perform any other high-risk operations, you can grant permission to use ELB resources but not permission to delete them.

Skip this section if your Huawei Cloud account does not require individual IAM users for permissions management.

IAM is a free service. You only pay for the resources in your account. For more information about IAM, see the **IAM Service Overview**.

ELB Permissions

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and attach policies or roles to these groups. The users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

ELB is a project-level service deployed for specific regions. To assign ELB permissions to a user group, specify the scope as region-specific projects and select projects for which you want the permissions to take effect. If you select **All projects**, the permissions will take effect for the user group in all region-specific projects. When accessing ELB, users need to switch to the authorized region.

You can grant permissions by using roles and policies.

- Roles: A coarse-grained authorization strategy provided by IAM to assign permissions based on users' job responsibilities. Only a limited number of service-level roles are available for authorization. When you grant permissions using roles, you also need to attach any existing role dependencies. Roles are not ideal for fine-grained authorization and least privilege access.
- Policies: A fine-grained authorization strategy provided by IAM to assign permissions required to perform operations on specific cloud resources under

certain conditions. This type of authorization is more flexible and is ideal for least privilege access. For example, you can grant ELB users only permissions to manage a certain type of resources. A majority of fine-grained policies contain permissions for specific APIs, and permissions are defined using API actions. For the API actions supported by ELB, see **Permissions Policies and Supported Actions**.

Table 10-1 lists all the system-defined permissions for ELB.

Table 10-1 System-defined permissions for ELB

Role/ Policy Name	Description	Type
ELBFullAccess	Permissions: all permissions on ELB resources Scope: project-level service	System-defined policy
ELBReadOnlyAccess	Permissions: read-only permissions on ELB resources Scope: project-level service	System-defined policy
ELB Administrator	Permissions: all permissions on ELB resources. To be granted this permission, users must also have the **Tenant Administrator**, **VPC Administrator**, **CES Administrator**, **Server Administrator** and **Tenant Guest** permissions. Scope: project-level service **NOTE** If your account has applied for fine-grained permissions, configure fine-grained policies for ELB system permissions, instead of ELB Administrator policies.	System-defined role

Table 10-2 describes common operations supported by each system policy of ELB.

Table 10-2 Common operations supported by system-defined policies

Operation	ELBFullAccess	ELBReadOnlyAccess	ELB Administrator
Creating a load balancer	Supported	Not supported	Supported
Querying a load balancer	Supported	Supported	Supported
Querying a load balancer and associated resources	Supported	Supported	Supported

Operation	ELBFullAccess	ELBReadOnlyAccess	ELB Administrator
Querying load balancers	Supported	Supported	Supported
Modifying a load balancer	Supported	Not supported	Supported
Deleting a load balancer	Supported	Not supported	Supported
Adding a listener	Supported	Not supported	Supported
Querying a listener	Supported	Supported	Supported
Modifying a listener	Supported	Not supported	Supported
Deleting a listener	Supported	Not supported	Supported
Adding a backend server group	Supported	Not supported	Supported
Querying a backend server group	Supported	Supported	Supported
Modifying a backend server group	Supported	Not supported	Supported
Deleting a backend server group	Supported	Not supported	Supported
Adding a backend server	Supported	Not supported	Supported
Querying a backend server	Supported	Supported	Supported
Modifying a backend server	Supported	Not supported	Supported
Deleting a backend server	Supported	Not supported	Supported
Configuring a health check	Supported	Not supported	Supported
Querying a health check	Supported	Supported	Supported

Operation	ELBFullAccess	ELBReadOnlyAccess	ELB Administrator
Modifying a health check	Supported	Not supported	Supported
Disabling a health check	Supported	Not supported	Supported
Assigning an EIP	Not supported	Not supported	Supported
Binding an EIP to a load balancer	Not supported	Not supported	Supported
Querying an EIP	Supported	Supported	Supported
Unbinding an EIP from a load balancer	Not supported	Not supported	Supported
Viewing metrics	Not supported	Not supported	Supported
Viewing access logs	Not supported	Not supported	Supported

NOTE

- To unbind an EIP, you also need to configure the **vpc:bandwidths:update** and **vpc:publicIps:update** permission of the VPC service. For details, see the *Virtual Private Cloud API Reference*.
- To view metrics, obtain the **CES ReadOnlyAccess** permissions. For details, see the *Cloud Eye API Reference*.
- To view access logs, you also need to configure the **LTS ReadOnlyAccess** permission. For details, see the *Log Tank Service API Reference*.

11 Product Concepts

11.1 Basic Concepts

Table 11-1 Some concepts about ELB

Term	Definition
Load balancer	A load balancer distributes incoming traffic across backend servers.
Listener	A listener listens on requests from clients and routes the requests to backend servers based on the settings that you configure when you add the listener.
Backend server	Backend servers receive and process requests from the associated load balancer. When you add a listener to a load balancer, you can create or select a backend server group to receive requests from the load balancer by using the port and protocol you specify for the backend server group and the load balancing algorithm you select.
Backend server group	A backend server group is a collection of cloud servers that have same features. When you add a listener, you select a load balancing algorithm and create or select a backend server group. Incoming traffic is routed to the corresponding backend server group based on the listener's configuration.
Health check	ELB periodically sends requests to backend servers to check whether they can process requests. This process is called health check. If a backend server is detected unhealthy, the load balancer will stop route requests to it. After the backend server recovers, the load balancer will resume routing requests to it.
Redirect	HTTPS is an extension of HTTP. HTTPS encrypts data between a web server and a browser. You can use ELB to redirect HTTP requests to an HTTPS listener to improve your service security.
Sticky session	Sticky sessions ensure that requests from a client always get routed to the same backend server before a session elapses.

Term	Definition
WebSocket	WebSocket is a new HTML5 protocol that provides full-duplex communication between the browser and the server. WebSocket saves server resources and bandwidth, and enables real-time communication. Both WebSocket and HTTP depend on TCP to transmit data. A handshake connection is required between the browser and server, so that they can communicate with each other only after the connection is established. However, as a bidirectional communication protocol, WebSocket is different from HTTP. After the handshake succeeds, both the server and browser (or client agent) can actively send data to or receive data from each other.
SNI	Server Name Indication (SNI) can be enabled for an HTTPS listener. SNI, an extension to Transport Layer Security (TLS), enables a server to present multiple certificates on the same IP address and port number. SNI allows the client to indicate the domain name of the website while sending an SSL handshake request. Once receiving the request, the load balancer queries the right certificate based on the hostname or domain name and returns the certificate to the client. If no certificate is found, the load balancer will return the default certificate.
Persistent connection	A persistent connection allows multiple data packets to be sent continuously over a TCP connection. If no data packet is sent during the connection, the client and server send link detection packets to each other to maintain the connection.
Short connection	A short connection is a connection established when data is exchanged between the client and server and immediately closed after the data is sent.
Concurrent connection	Concurrent connections are total number of TCP connections initiated by clients and routed to backend servers by a load balancer per second.

11.2 Region and AZ

Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. to support high-availability systems.

Selecting a Region

If your target users are in Europe, select the **EU-Dublin** region.

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

12 ELB and Other Services

Table 12-1 Related services

Service Name	Function	Reference
Elastic Cloud Server (ECS)	Provides servers to run your applications in the cloud. Configure load balancers to route traffic to the servers or containers.	**Purchasing an ECS**
Bare Metal Server (BMS)		**Creating a BMS**
Elastic IP (EIP)	Allows you to bind an EIP to a load balancer so that the load balancer can process Internet traffic.	**Assigning an EIP**
Auto Scaling (AS)	Works with ELB to automatically scale the number of backend servers for faster traffic distribution.	**Adding a Load Balancer to an AS Group**
Identity and Access Management (IAM)	Provides fine-grained permissions control for ELB.	**Creating a User Group and Assigning Permissions**
Cloud Trace Service (CTS)	Records the operations performed on ELB resources.	**Viewing Traces**
Cloud Eye	Monitors the status of load balancers and listeners, without any additional plug-in.	**Viewing Metrics**
Anti-DDoS	Defends public network load balancers against DDoS attacks, keeping your business stable and reliable.	Configuring an Anti-DDoS Protection Policy

