Elastic Load Balance

Service Overview

Issue 01

Date 2022-09-30

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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.

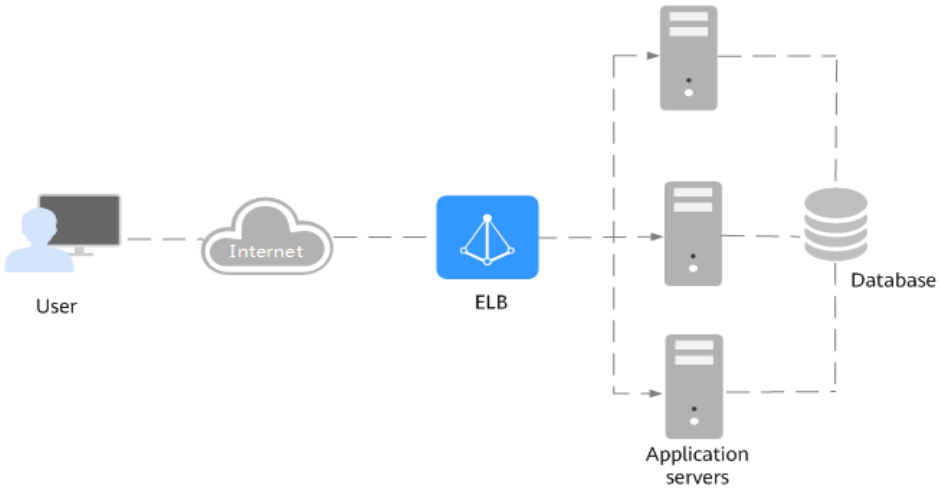
Contents

1 What Is ELB?

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

As shown in the example in the following figure, ELB distributes incoming traffic to three application servers, and each server processes one third of the requests. ELB also provides health checks, which can detect unhealthy servers. Traffic is distributed 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 following components:

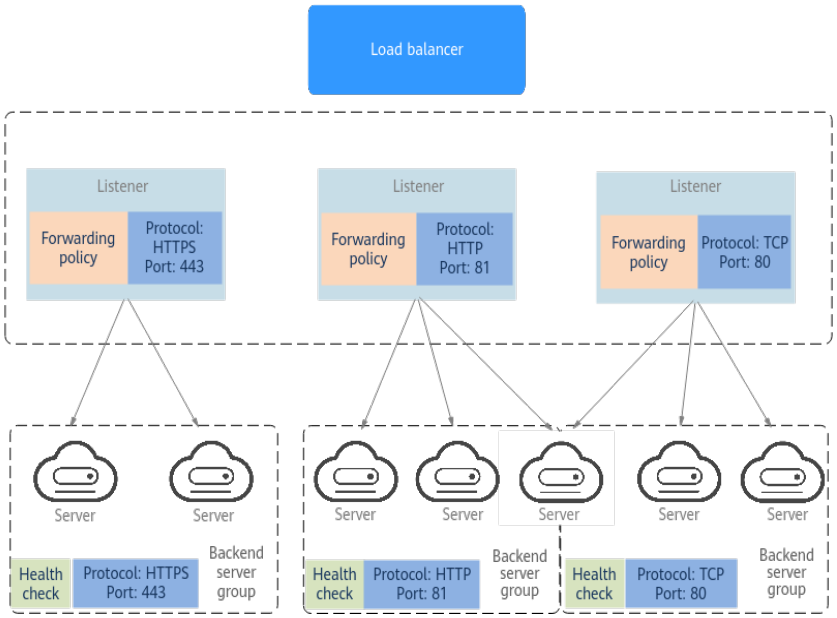
- Load balancer: distributes incoming traffic across backend servers in one or more availability zones (AZs).
- Listener: uses the protocol and port you specify to check for requests from clients and route the requests to associated backend servers based on the listening rules and forwarding policies 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. You need to add at least one backend server to a backend server group.

  You can set a weight for each backend server based on their performance.

  You can also configure health checks for a backend server group to check the health of each backend server. When a backend server is unhealthy, the load balancer stops routing new requests to this server.

Figure 1-2 ELB components



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 Product Advantages

Advantages of Dedicated Load Balancers

- Robust performance

  Each load balancer has exclusive use of isolated resources, meeting your requirements for handling a massive number of requests. A single load balancer deployed in one AZ can handle up to 20 million concurrent connections.

  If you deploy a load balancer in multiple AZs, its performance such as the number of new connections and the number of concurrent connections will multiply. For example, if you deploy a dedicated load balancer in two AZs, it can handle up to 40 million concurrent connections.

  NOTE

  - 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 two AZs, the requests the load balancers can handle will be doubled.
  - For requests from a private network:
    - If clients are in an AZ you select when you create the load balancer, requests are distributed by the load balancer in this AZ. If the load balancer is unhealthy, requests are distributed by the load balancer in another AZ you select.

      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 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 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.
- Ultra-high security

ELB supports TLS 1.3 and can route HTTPS requests to backend servers. You can select security policies or customize security policies that fit your security requirements.

- Multiple protocols

  ELB supports Quick UDP Internet Connection (QUIC), TCP, UDP, HTTP, and HTTPS, so that they can route requests to different types of applications.

- High flexibility

  ELB 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

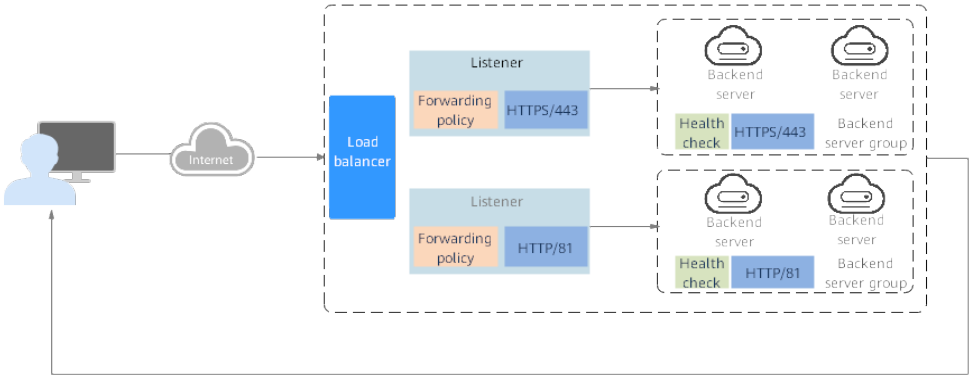
  ELB can route requests to both servers on the cloud and on premises, allowing you to leverage cloud resources to handle burst traffic.

- Ease-of-use

  ELB provides a diverse set of algorithms that allow you to configure different traffic routing policies to meet your requirements while keeping deployments simple.

3 How ELB Works

Figure 3-1 How ELB works



The following describes how ELB works:

1. A client sends a request to your application.
2. The listeners added to your load balancer use the protocols and ports you have configured to receive the request.
3. The listener forwards the request to the associated backend server group based on your configuration. If you have configured a forwarding policy for the listener, the listener evaluates the request based on the forwarding policy. If the request matches the forwarding policy, the listener forwards the request to the backend server group configured for the forwarding policy.
4. Healthy backend servers in the backend server group receive the request based on the load balancing algorithm and the routing rules you specify in the forwarding policy, handle the request, and return a result 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.

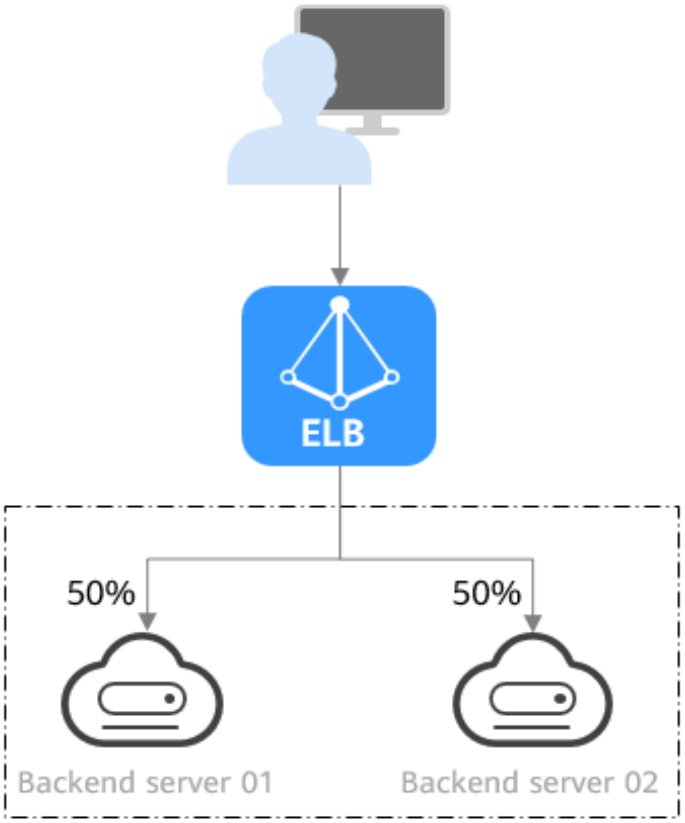
Load Balancing Algorithms

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

- Weighted round robin: Requests are routed 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. This algorithm is often used for short connections, such as HTTP connections.

  The following figure 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

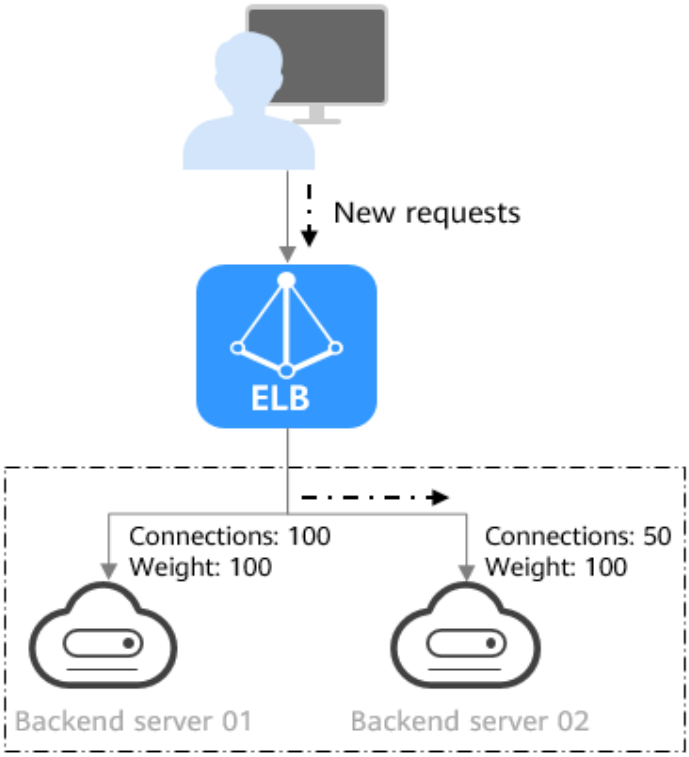


- Weighted least connections: In addition to the weight assigned to each server, the number of connections being processed by each backend server is also considered. Requests are routed to the server with the lowest connections-to-weight ratio. In addition to the number of connections, each server is assigned a weight based on its capacity. Requests are routed to the server with the lowest connections-to-weight ratio. This algorithm is often used for persistent connections, such as connections to a database.

  The following figure 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 connected with backend server 02. New requests are preferentially routed to backend server 02.

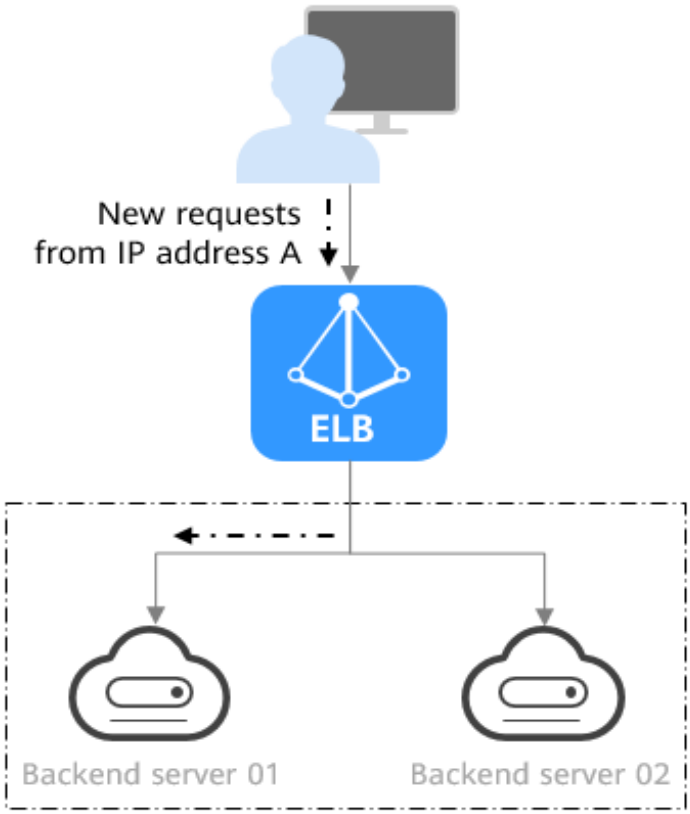
Figure 3-3 Traffic distribution using the weighted least connections algorithm



- Source IP hash: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hashing key, and all backend servers are numbered. The generated key is used to allocate the client to a particular server. This allows requests from different clients to be routed based on source IP addresses and ensures that a client is directed to the same server that it was using previously. This algorithm works well for TCP connections of load balancers that do not use cookies.

  The following figure 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



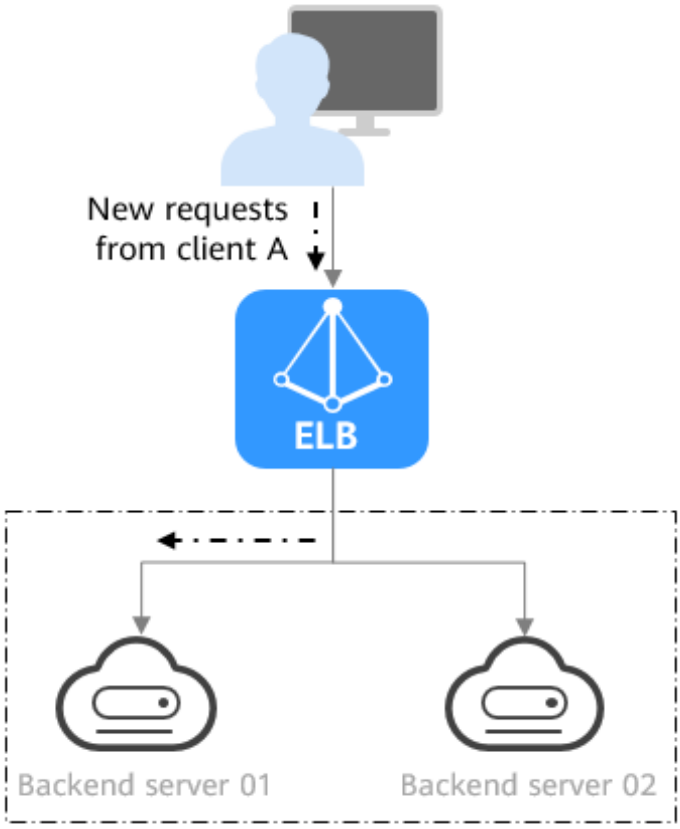
- Connection ID: The connection ID in the packet is calculated using the consistent hash algorithm to obtain a specific value, and backend servers are numbered. The generated value determines to which backend server the requests are routed. This allows requests with different connection IDs to be routed to different backend servers and ensures that requests with the same connection ID are routed to the same backend server. This algorithm applies to QUIC requests.

  NOTE

  Currently, only dedicated load balancers support the Connection ID algorithm.

  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 ECS 01 has processed a request from client A, the load balancer will route new requests from client A to ECS 01.

Figure 3-5 Traffic distribution using the connection ID algorithm



Factors Affecting Load Balancing

In addition to the load balancing algorithm, factors that affect load balancing generally include connection type, session stickiness, and server 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 weighted least connections algorithm, 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?**

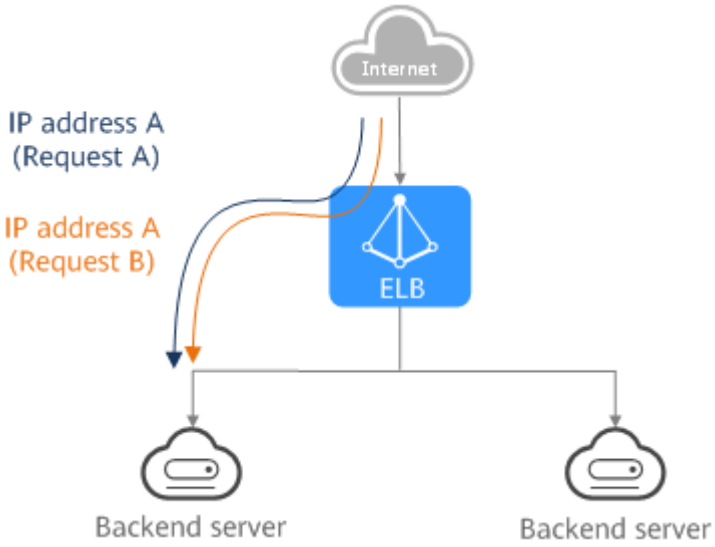
4 Application Scenarios

Heavy-Traffic Applications

For an application with heavy traffic, such as a large portal or mobile app store, ELB evenly distributes incoming traffic to multiple 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.

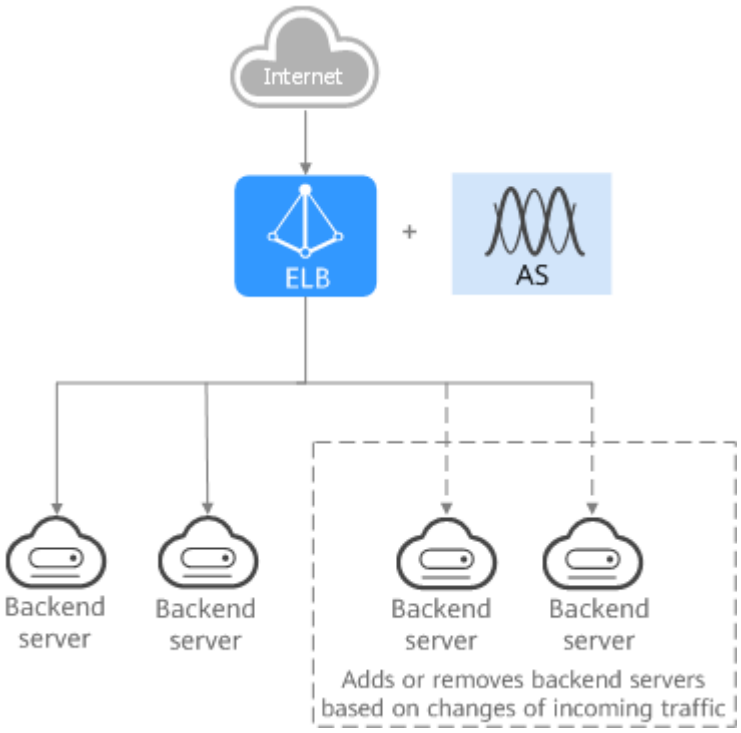
Figure 4-1 Session stickiness



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 add or remove backend servers to keep up with changing demands. An example is flash sales, during which application traffic spikes in a short period. ELB can work with Auto Scaling to run only the required number of backend servers to handle the load of your application.

Figure 4-2 Flexible scalability

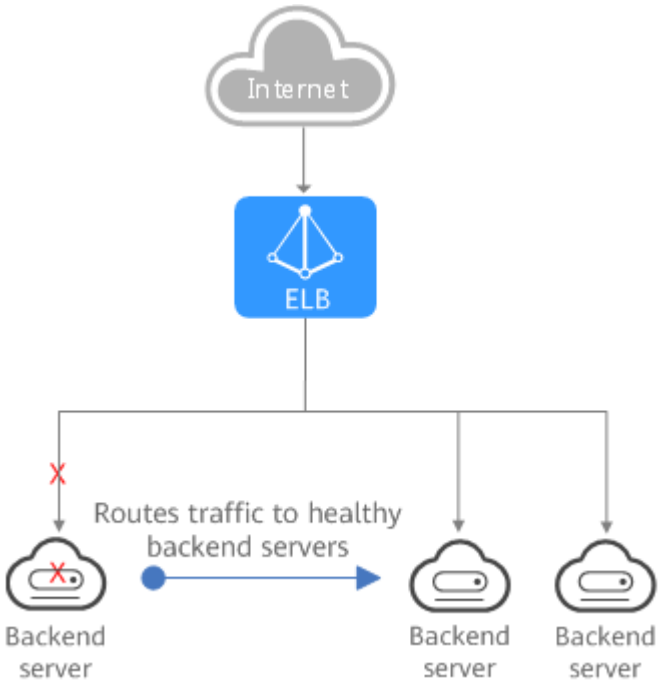


Zero SPOFs

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

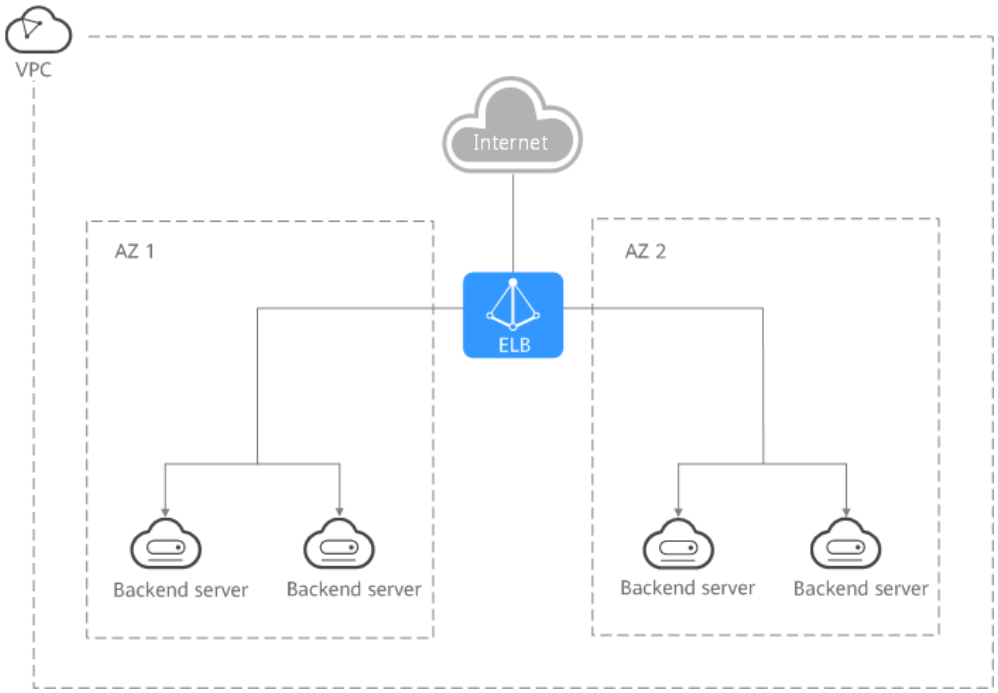


Cross-AZ Load Balancing

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

Each type of load balancer has their advantages.

Feature Comparison

Dedicated load balancers provide more powerful forwarding performance, while shared load balancers are less expensive. You can select the appropriate load balancer based on your application needs. The following tables compare the features supported by the two types of load balancers. (√ indicates that an item is supported, and x indicates that an item is not supported.)

Table 5-1 Performance

Item	Dedicated Load Balancers	Shared Load Balancers
Deployment mode	Their performance is not affected by other load balancers. You can select different specifications based on your requirements.	Shared load balancers are deployed in clusters, and all the load balancers share underlying resources, so that the performance of a load balancer is affected by other load balancers.

Item	Dedicated Load Balancers	Shared Load Balancers
Concurrent connections	A dedicated load balancer in an AZ can establish up to **20 million concurrent connections**. If you deploy a dedicated load balancer in two AZs, the number of concurrent connections will be doubled. For example, if you deploy a dedicated load balancer in two AZs, it can handle up to 40 million concurrent connections.	-

Item	Dedicated Load Balancers	Shared Load Balancers

	NOTE
	• 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 two AZs, the requests the load balancers can handle will be doubled.
	• For requests from a private network:
	  • If clients are in an AZ you select when you create the load balancer, requests are distributed by the load balancer in this AZ. If the load balancer is unhealthy, requests are distributed by the load balancer in another AZ you select.
	  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 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 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.	

Table 5-2 Supported protocols

Protocol	Description	Dedicated Load Balancers	Shared Load Balancers
QUIC	If you use UDP as the frontend protocol, you can select QUIC as the backend protocol, and select the connection ID algorithm to route requests with the same connection ID to the same backend server. QUIC has the advantages of low latency, high reliability, and no head-of-line blocking (HOL blocking), and is very suitable for the mobile Internet. No new connections need to be established when you switch between a Wi-Fi network and a mobile network.	√	x
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.	√	√
HTTP/HTTPS (Layer 7)	After receiving a request, the listener needs to identify the request and forward data based on the fields in the HTTP/HTTPS packet header. Though the routing efficiency is lower than that at Layer 4, load balancing at Layer 7 provides some advanced features such as encrypted transmission and cookie-based sticky sessions.	√	√
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.	√	√

Table 5-3 Supported backend types

Backend Type	Description	Dedicated Load Balancers	Shared Load Balancers
IP as backend servers	You can add servers in a VPC connected using a VPC peering connection, in a VPC 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. In this way, incoming traffic can be flexibly distributed to cloud servers and on-premises servers for hybrid load balancing.	√	x
Supplementary network interface	You can attach supplementary network interfaces to backend servers. Supplementary network interfaces are a supplement to elastic network interfaces and are attached to VLAN interfaces of elastic network interfaces used by backend servers when the number of elastic network interfaces attached to the backend servers exceeds the limit. Supplementary network interfaces allow you to attach more network interfaces to a single backend server for flexible and high-availability network configuration.	√	x
ECS	You can use load balancers to distribute incoming traffic across ECSs.	√	√
BMS	You can use load balancers to distribute incoming traffic across BMSs.	√	√

Table 5-4 Advanced features

Feature	Description	Dedicated Load Balancers	Shared Load Balancers
Multiple specifications	Load balancers allow you to select appropriate specifications based on your requirements.	√	x
HTTPS support	Load balancers can receive HTTPS requests from clients and route them to an HTTPS backend server group.	√	x
IPv6 addresses	Load balancers can route requests from IPv6 clients. You can change the IPv6 address bound to a load balancer and unbind the IPv6 address from the load balancer.	√	x
Changing the private IPv4 address bound to the load balancer	You can change the private IPv4 address bound to a load balancer.	√	x
Slow start	You can enable slow start for HTTP or HTTPS listeners. After you enable it, the load balancer linearly increases the proportion of requests to send to backend servers in this mode. Slow start gives applications time to warm up and respond to requests with optimal performance.	√	x
Mutual authentication	In this case, you need to deploy both the server certificate and client certificate. Mutual authentication is supported only by HTTPS listeners.	√	√

Feature	Description	Dedicated Load Balancers	Shared Load Balancers
Custom timeout durations	You can configure and modify timeout durations (idle timeout, request timeout, and response timeout) for your listeners to meet varied demands. For example, if the size of a request from an HTTP or HTTPS client is large, you can increase the request timeout duration to ensure that the request can be successfully routed. • Dedicated load balancers: You can change the timeout durations of TCP, UDP, HTTP, and HTTPS listeners. • Shared load balancers: You can only change the timeout durations of TCP, HTTP, and HTTPS listeners, but cannot change the timeout durations of UDP listeners.	√	√
Security policies	When you add HTTPS listeners, you can select appropriate security policies to improve service security. A security policy is a combination of TLS protocols and cipher suites.	√	√
Passing the listener's port number to backend servers	The listener's port number is stored in the X-Forwarded-Port header and passed to backend servers.	√	√
Passing the client's port number to backend servers	The client's port number is stored in the X-Forwarded-For-Port header and passed to backend servers.	√	√
Rewriting X-Forwarded-Host	• If you disable this option, the load balancer passes the X-Forwarded-Host field to backend servers. • If you enable this option, the load balancer rewrites the X-Forwarded-Host field based on the Host field in the request header sent from the client and sends the rewritten X-Forwarded-Host field to backend servers.	√	√

Table 5-5 Other features

Feature	Description	Dedicated Load Balancers	Shared Load Balancers
Customized cross-AZ deployment	You can create a load balancer in multiple AZs. Each AZ selects an optimal path to process requests. In addition, the AZs back up each other, improving service processing efficiency and reliability. If you deploy a load balancer in multiple AZs, its performance such as the number of new connections and the number of concurrent connections will multiply. For example, if you deploy a dedicated load balancer in two AZs, it can handle up to 40 million concurrent connections.	√	x

Feature	Description	Dedicated Load Balancers	Shared Load Balancers

	NOTE
	• 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 two AZs, the requests the load balancers can handle will be doubled.
	• For requests from a private network:
	  • If clients are in an AZ you select when you create the load balancer, requests are distributed by the load balancer in this AZ. If the load balancer is unhealthy, requests are distributed by the load balancer in another AZ you select.
	  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 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 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.		

Feature	Description	Dedicated Load Balancers	Shared Load Balancers
Connection ID	Load balancers can use the connection ID algorithm to route requests. The connection ID in the packet is calculated using the consistent hash algorithm to obtain a specific value, and backend servers are numbered. The generated value determines to which backend server the requests are routed.	√	x
Load balancing algorithms	Load balancers support weighted round robin, weighted least connections, and source IP hash.	√	√
Load balancing over public and private networks	• Each load balancer on a public network has a public IP address bound to it and routes requests from clients to backend servers over the Internet. • Load balancers on a private network work within a VPC and route requests from clients to backend servers in the same VPC.	√	√
Modifying the bandwidth	You can modify the bandwidth used by the EIP bound to the load balancer as required.	√	√
Binding/ Unbinding an IP address	You can bind an IP address to a load balancer or unbind the IP address from a load balancer based on service requirements.	√	√
Sticky session	If you enable sticky sessions, requests from the same client will be routed to the same backend server during the session.	√	√
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.	√	√
Health check	Load balancers periodically send requests to backend servers to check whether they can process requests.	√	√

Feature	Description	Dedicated Load Balancers	Shared Load Balancers
Certificate management	You can create two types of certificates: server certificate and CA certificate. If you need an HTTPS listener, you need to bind a server certificate to it. To enable mutual authentication, you also need to bind a CA certificate to the listener. You can also replace a certificate that is already used by a load balancer.	√	√
Tagging	If you have a large number of cloud resources, you can assign different tags to the resources to quickly identify them and use these tags to easily manage your resources.	√	√
Monitoring	You can use Cloud Eye to monitor load balancers and associated resources and view metrics on the management console.	√	√
Log auditing	You can use Cloud Trace Service (CTS) to record operations on load balancers and associated resources for query, auditing, and backtracking.	√	√

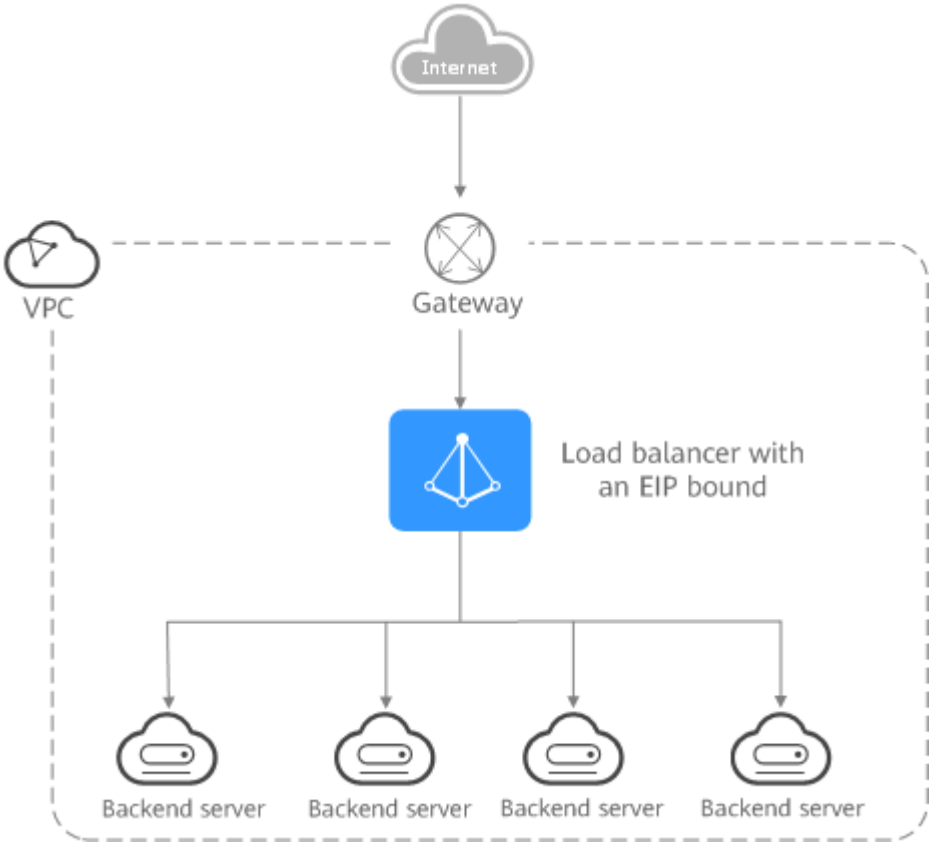
6 Load Balancing on a Public or Private Network

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.

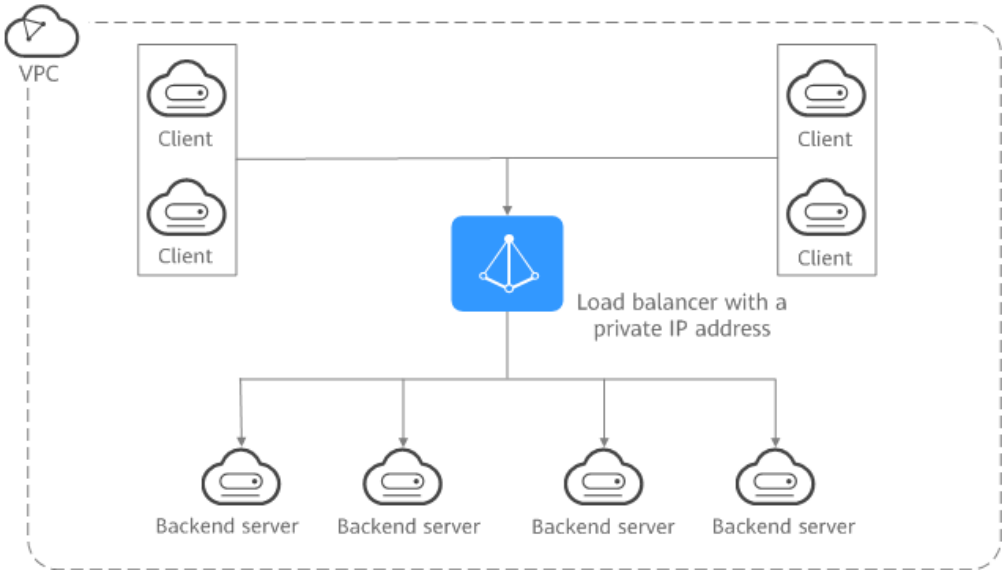
Figure 6-1 Load balancing on a public network



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.

7 Network Traffic Paths

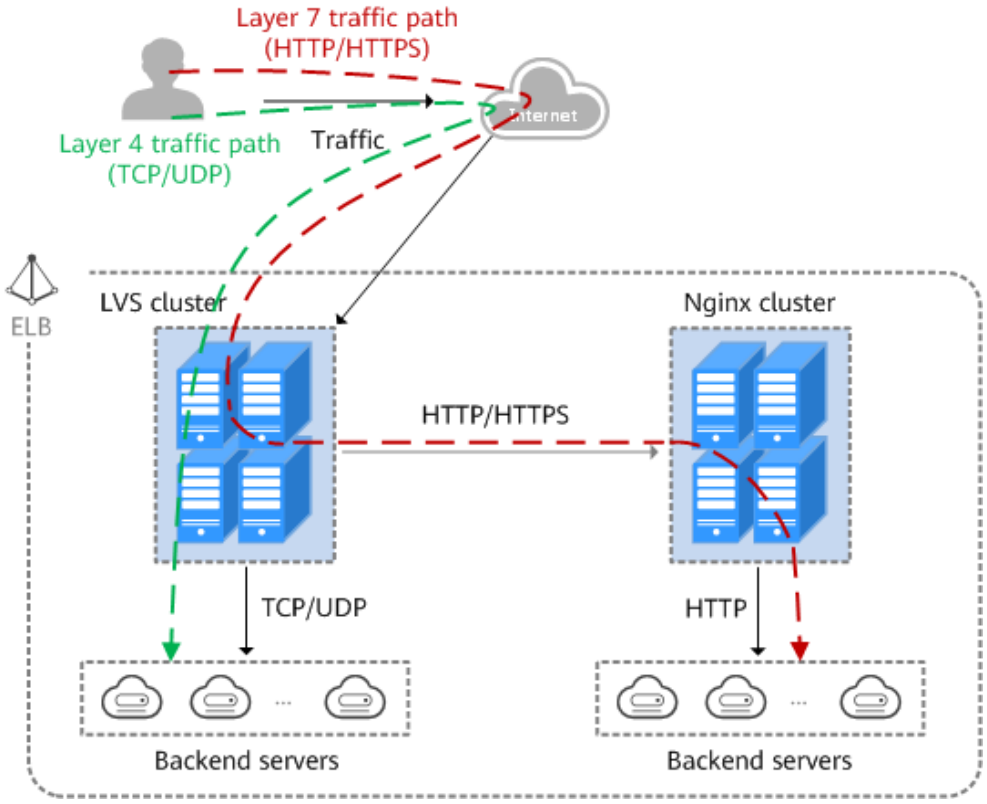
Load balancers communicate with backend servers over a private network.

- 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.

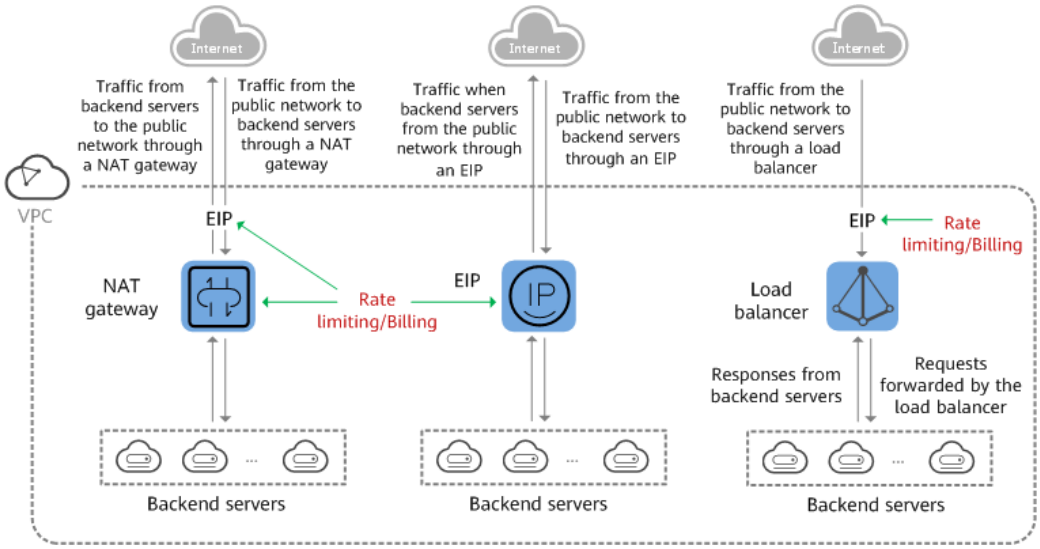
When a listener uses HTTP or HTTPS to receive incoming traffic:

- 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

Load balancers are available in different specifications. Choose the specifications that best meet your needs. If the traffic exceeds the selected specifications, new requests will be discarded.

- Maximum concurrent connections

  Indicates the maximum number of concurrent connections that a load balancer can handle. If the number reaches the maximum connections that defined in the specification, new requests will be discarded to ensure the performance of the 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 defined in the specification, 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 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 x Ratio of HTTP connections to HTTPS connections

    For a small I application load balancer, the ratio of HTTP connections to HTTPS connections is 10. For details, see **Table 8-1**.

Table 8-1 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 x 10 = 1,500	1,000 + 150 x 10 = 2,500
Description	The new connections do not reach the CPS (HTTP) defined in **Table 8-3**, and new requests can be properly routed.	The new connections exceed the CPS (HTTP) defined in **Table 8-3**, and new requests will be discarded.

NOTE

Details in the **Table 8-1** are for reference only.

- Queries per second (QPS)

  Indicates the number of HTTP or HTTPS requests sent to a backend server per second. If the QPS reaches that defined in the specification, 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 load balancer per second.

Table 8-2 and **Table 8-3** list the specifications of dedicated load balancers.

CAUTION

- **Available fixed specifications are displayed on the console and may vary depending on the resources in different regions.**
- The load balancing type cannot be changed after being selected.

  For example, after you select network load balancing, you cannot change it to application load balancing. You can add only TCP and UDP listeners and cannot add HTTP and HTTPS listeners.

Table 8-2 Specifications for network load balancing (TCP/UDP)

Type	Maximum Concurrent Connections	CPS	Bandwidth (Mbit/s)	LCUs in an AZ
Small I	500,000	10,000	50	10
Small II	1,000,000	20,000	100	20
Medium I	2,000,000	40,000	200	40
Medium II	4,000,000	80,000	400	80
Large I	10,000,000	200,000	1,000	200
Large II	20,000,000	400,000	2,000	400

Table 8-3 Specifications for application load balancing (HTTP/HTTPS)

Type	Maximum Concurrent Connections	CPS (HTTP)	CPS (HTTPS)	QPS (HTTP)	QPS (HTTPS)	Bandwidth (Mbit/s)	Number of 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
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 Quotas 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 number or amount 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 to adjust quotas 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

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 rules per forwarding policy	Forwarding rules that can be added to a forwarding policy	10
Backend servers per backend server group	Backend servers that can be added to a backend server group	500
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 size of files that a load balancer can forwarded:
  - Layer 4 listeners: any
  - Layer 7 listeners: 10 GB or smaller

Listener

- The listener of a dedicated load balancer can be associated with a maximum of 50 backend server groups.
- An HTTPS listener can have up to 50 SNI certificates.
- 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 URL	**Forward to another backend server group** and **Redirect to another listener**	**Forwarding Policy (Shared Load Balancers)**
Dedicated	Disabled	Domain name and URL	**Forward to another backend server group** and **Redirect to another listener**	**Forwarding Policy (Dedicated Load Balancers)**
	Enabled	Domain name, URL, HTTP request method, HTTP header, query string, and CIDR block	**Forward to a backend server group**, **Redirect to another listener**, **Redirect to another URL**, and **Return a specific response body**	**Advanced Forwarding (Dedicated Load Balancers)**

Backend Server Group

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

Table 9-4 The frontend and backend protocol

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	• UDP • QUIC
HTTP	HTTP
HTTPS	• HTTP • HTTPS

Backend Server

If **Transfer Client IP Address** is enabled, a server cannot serve as both a backend server and a client.

TLS Security Policy

You can create a maximum of 50 TLS security policies.

10 Billing (Dedicated Load Balancers)

Billing Items

The following table describes the billing items of dedicated load balancers.

Table 10-1 Billing items

Billing Mode	Load Balancer Price	LCU Price	Billing Formula
Pay-per-use	√	√	Load balancer price+ LCU price **NOTE** The load balancers are free of charge. You only need to pay for the LCUs.

Table 10-2 describes the billing items.

Table 10-2 Billing items

Billing Item	Description
Load balancer price	You will be charged for the duration that you use the dedicated load balancer. If the load balancer is used for less than 1 hour, you will be charged for the actual duration, accurate to seconds.
LCU price	You will be charged for the number of LCUs used by a dedicated load balancer.

NOTE

- √ indicates that the billing item is involved. × indicates that the billing item is not involved.
- An LCU measures the dimensions on which a dedicated load balancer routes the traffic. The four dimensions measured are as follows:
  - New connections: the number of new connections that a dedicated load balancer establishes per second
  - Maximum concurrent connections: the maximum number of concurrent connections that a dedicated load balancer can handle
  - Queries per second: the number of Layer-7 HTTP or HTTPS requests that a dedicated load balancer routes to a backend server per second
  - Processed traffic: each GB of data transferred through a dedicated load balancer
- If you bind an EIP to a dedicated load balancer, you will also be charged for the EIP and the bandwidth used by the EIP.

  For details about EIP pricing, see **Elastic IP Pricing Details**.

Billing Mode

Dedicated load balancers provide Layer-4 packages and Layer-7 packages. You can select a Layer-4 package, a Layer-7 package, or both based on your requirements.

NOTE

- The total bandwidth is the inbound or outbound bandwidth used for traffic to or from the backend servers.
- For details, see **Table 8-2** and **Table 8-3**.

- **Pay-per-use**

  Formula: Total price = Load balancer price + LCU price

  – Load balancer price = Unit price ($0 USD/hour) x Usage duration

  – LCU price = Unit price ($0.00695 USD/hour) x LCUs in a single AZ x Number of AZs x Usage duration

  The following table lists of the number of LCUs and their prices in different specifications in the pay-per-use mode.

Table 10-3 Dedicated load balancer for network load balancing (TCP/UDP)

Type	LCUs in an AZ	Load Balancer Unit Price (Hourly)	LCU Unit Price (Hourly)	Load Balancer Price (Hourly)	LCU Price (Unit Price x Number of AZs)
Small I	10	0	0.00695	0	0.0695
Small II	20	0	0.00695	0	0.139
Medium I	40	0	0.00695	0	0.278

Type	LCUs in an AZ	Load Balancer Unit Price (Hourly)	LCU Unit Price (Hourly)	Load Balancer Price (Hourly)	LCU Price (Unit Price x Number of AZs)
Medium II	80	0	0.00695	0	0.556
Large I	200	0	0.00695	0	1.39
Large II	400	0	0.00695	0	2.78

Table 10-4 Dedicated load balancers for application load balancing (HTTP/HTTPS)

Type	LCUs in an AZ	Load Balancer Unit Price (Hourly)	LCU Unit Price (Hourly)	Load Balancer Price (Hourly)	LCU Price (Unit Price x Number of AZs)
Small I	10	0	0.00695	0	0.0695
Small II	20	0	0.00695	0	0.139
Medium I	40	0	0.00695	0	0.278
Medium II	100	0	0.00695	0	0.695
Large I	200	0	0.00695	0	1.39
Large II	400	0	0.00695	0	2.78

NOTE

- LCU quantity refers to the number of LCUs corresponding to a specification in a single AZ.
- If you select multiple AZs for a load balancer, the number of LCUs is calculated as follows: Number of LCUs = Number of LCUs in the selected specification x Number of the selected AZs.
- You need to select at least an AZ when you create a dedicated load balancer. For more information about AZs, see **Region and AZ**.

Example:

If you select small I for network load balancing and small II for application load balancing and deploy the load balancer in two AZs, the total price for using the load balancer for 3 hours is calculated as follows:

Load balancer price + LCU price = $0/hour x 3 hours + ($0.0695/hour + $0.139/hour) x 2 AZs x 3 hours = $1.251

Changing the Billing Configuration

Ticket Management

- Load balancers support only pay-per-use billing. The billing mode cannot be changed.
- **Submit a service ticket**.

  New specifications take effect immediately upon change. You are then charged based on the new specifications.

Renewal

You can renew your resources on the Renewals page after logging in to the management console. For details, see **Renewal Management**.

Expiration and Overdue Payment

If your account is in arrears, you can view the arrears details in the Billing Center. To prevent your load balancers from being stopped or released, top up your account in a timely manner. For details, see **Repaying Outstanding Amount**.

If you do not renew your load balancers in time, your account will be frozen and your load balancers will be kept in retention.

During this period, the load balancers cannot be used. For details, see **What Functions Will Become Unavailable If a Load Balancer Is Frozen?**

If you still do not complete the renewal or payment after the retention period ends, your data stored in cloud services will be deleted and the resources will be released.

11 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 11-1 lists all the system-defined permissions for ELB.

Table 11-1 System-defined permissions for ELB

Role/Policy Name	Description	Type
ELB FullAccess	Permissions: all permissions on ELB resources Scope: project-level service	System-defined policy
ELB ReadOnlyAccess	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 11-2 describes common operations supported by each system policy of ELB.

Table 11-2 Common operations supported by system-defined policies

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Creating a load balancer	Supported	Not supported	Supported
Querying a load balancer	Supported	Supported	Supported

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Querying a load balancer and associated resources	Supported	Supported	Supported
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
Modifying a health check	Supported	Not supported	Supported

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
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 monitoring metrics, you also need to configure the **CES ReadOnlyAccess** permission. 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*.

12 Product Concepts

12.1 Basic Concepts

Table 12-1 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	A backend server is a cloud server added to a backend server group associated with a 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. If a backend server is detected as unhealthy, the load balancer stops routing 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.
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	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.

12.2 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.

13 How ELB Works with Other Services

Table 13-1 Related services

Service Name	Function	Reference
Elastic Cloud Server (ECS)	Provides cloud servers to run your applications in the cloud. Configure load balancers to route traffic to the servers or containers.	**Purchasing and Logging In to a Linux ECS**
Bare Metal Server (BMS)		**Creating a BMS**
Virtual Private Cloud (VPC)	Provides IP addresses and bandwidth for load balancers.	**Assigning an EIP**
Auto Scaling (AS)	Works with ELB to automatically scale the number of backend servers for faster traffic distribution.	**Creating an AS Group**
Identity and Access Management (IAM)	Provides authentication 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

14 Change History

Released On	Description
2022-09-30	This issue is the first official release.

