Elastic Load Balance

Service Overview

Issue 9

Date 2023-08-21

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.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

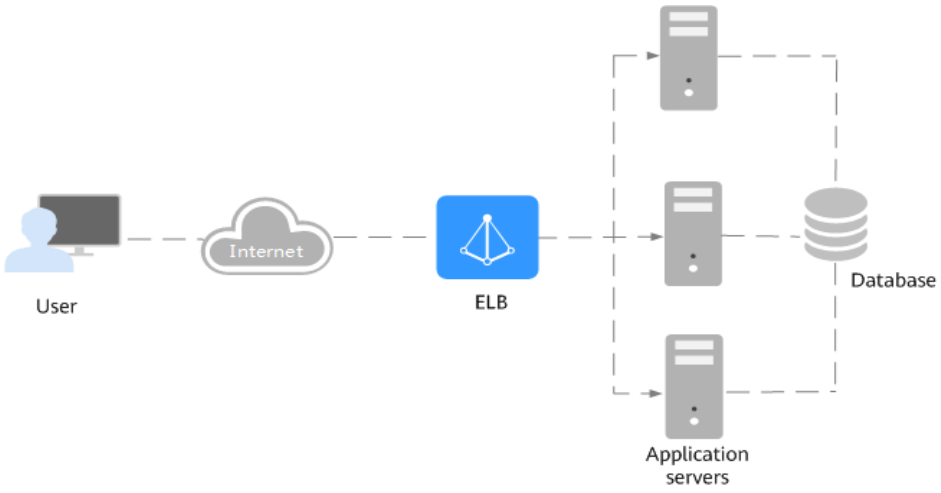
Contents

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:

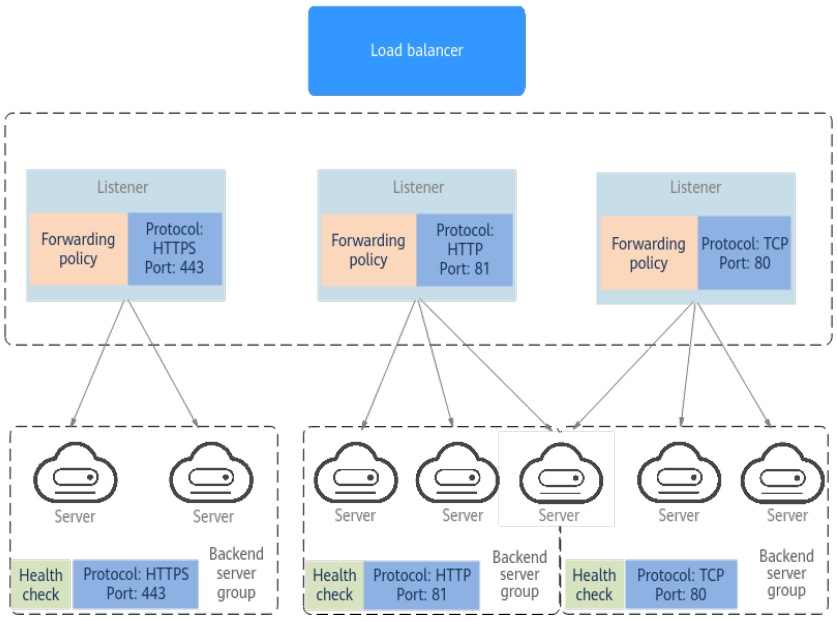
- **Load balancer**: distributes incoming traffic across backend servers in one or more availability zones (AZs).
- **Listener**: uses the protocol and port you specify to check for requests from clients and route the requests to associated backend servers based on the listening rules and forwarding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

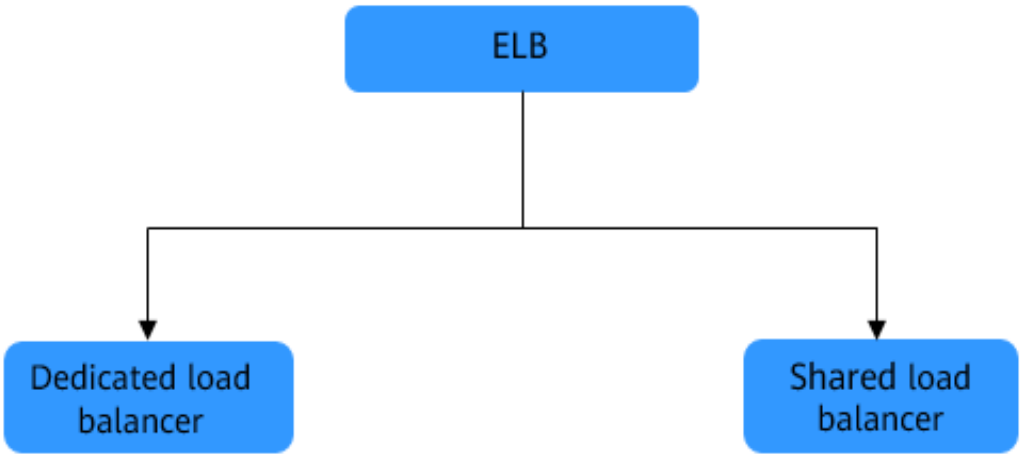
Figure 1-2 ELB components



Load Balancer Types

ELB provides shared load balancers and dedicated load balancers.

Figure 1-3 Load balancer types



- Dedicated load balancers have exclusive use of underlying resources, so that the performance of a dedicated load balancer is not affected by other load

balancers. In addition, there are a wide range of specifications available for selection.

NOTE

Dedicated load balancers are not available in AF-Johannesburg and LA-Mexico City1.

- Shared load balancers are deployed in clusters and share underlying resources, so their performance may be affected by other load balancers.

For details about the differences between shared and dedicated load balancers, see **ELB Product Types**.

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 access to isolated resources, allowing your services to handle 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 the AZ you selected when you created the load balancer, requests are distributed by the load balancer in this AZ. If the load balancer is unavailable, requests are distributed by the load balancer in another AZ you select.

      If the load balancer is available 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 in this AZ 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 in this AZ is unavailable, requests are distributed by the load balancer in another AZ.

- High availability

  ELB can route traffic uninterruptedly. If your servers in one AZ are unhealthy, it automatically routes traffic to healthy servers in other AZs. ELB provides a comprehensive health check system to ensure that incoming traffic is routed

only to healthy backend servers, improving the availability of your applications.

- 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.

Advantages of Shared Load Balancers

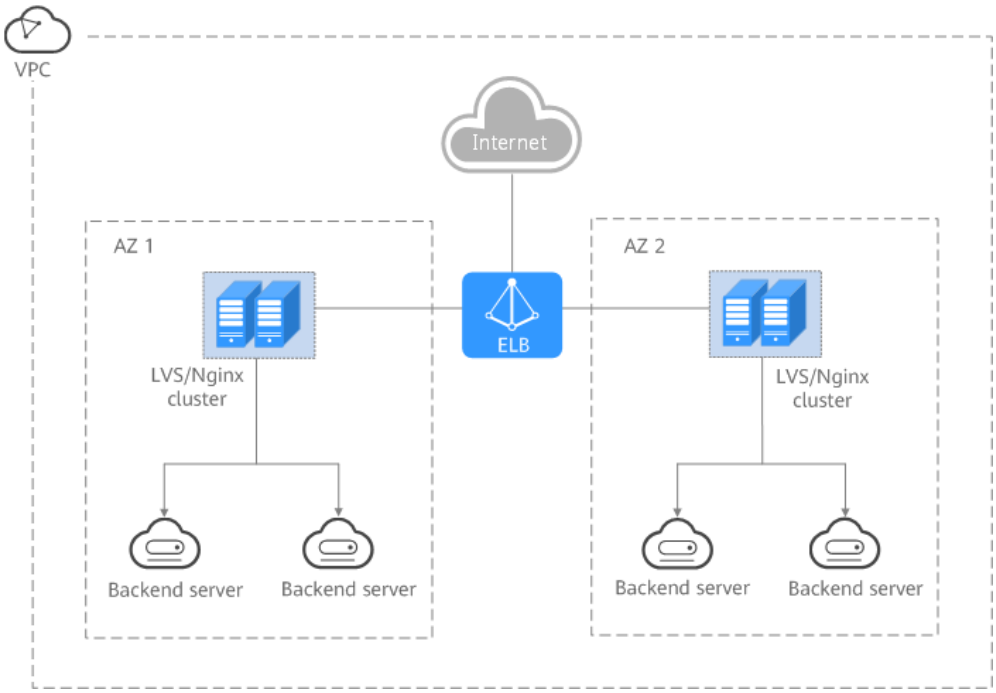
- High performance

  Shared load balancers provide guaranteed performance, which can handle up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.

- 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 mechanism to ensure that incoming traffic is routed to only healthy backend servers, improving the availability of your applications.

Figure 2-1 High availability



- Multiple protocols

  ELB supports TCP, UDP, HTTP, and HTTPS protocols to route requests to different types of applications.

- Ease-of-use

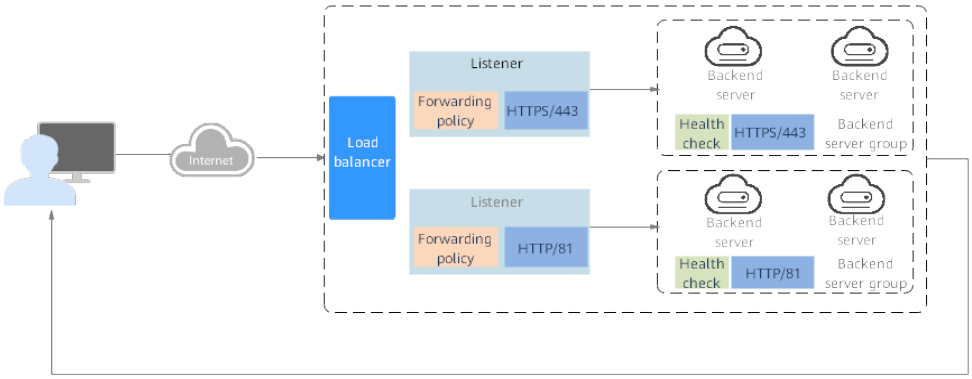
  ELB provides a diverse set of algorithms that allow you to configure different traffic routing policies to meet your requirements while keeping deployments simple.

- High reliability

  Load balancers are deployed in two AZs and can distribute traffic more evenly.

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

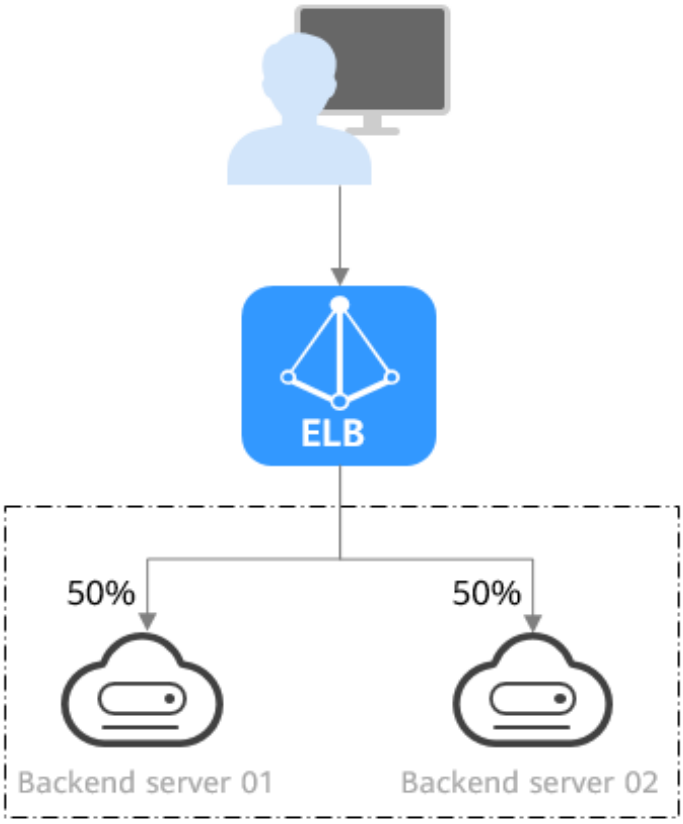
Dedicated load balancers support four load balancing algorithms: weighted round robin, weighted least connections, source IP hash, and connection ID. Shared load

balancers support three load balancing algorithms: weighted round robin, weighted least connections, and source IP hash.

- 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

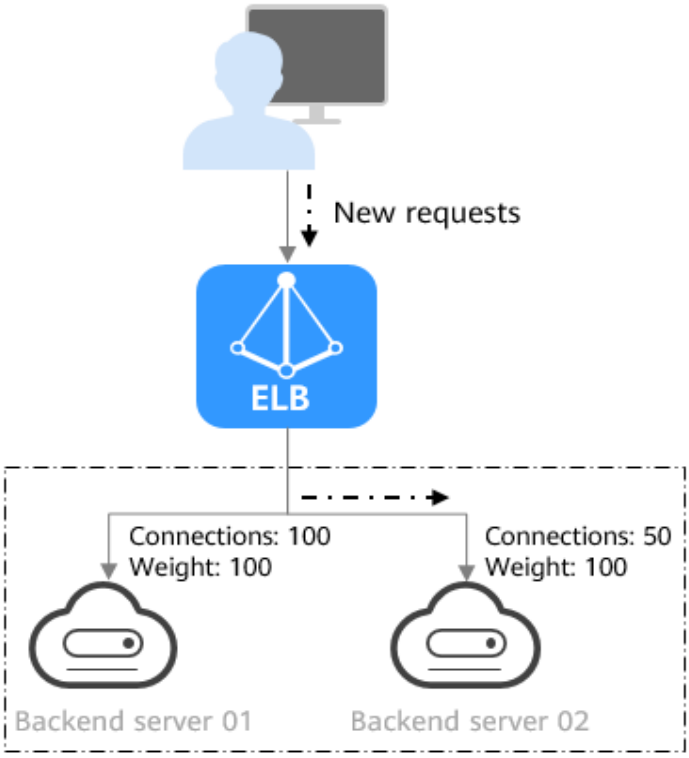
  

- Weighted least connections: In addition to the weight assigned to each server, the number of connections being processed by each backend server is also considered. Requests are routed to the server with the lowest connections-to-weight ratio. In addition to the number of connections, each server is 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.

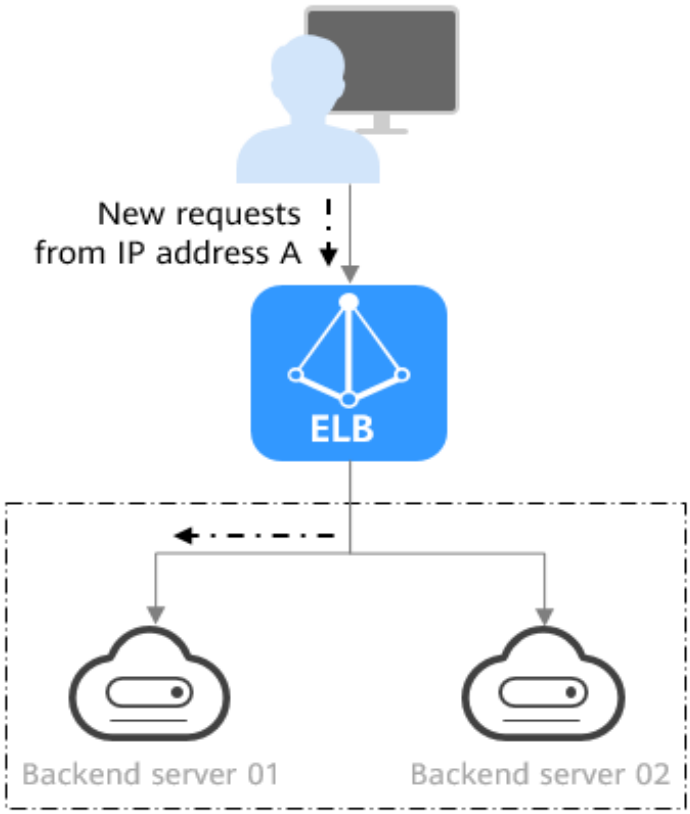
Figure 3-3 Traffic distribution using the weighted least connections algorithm



- 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



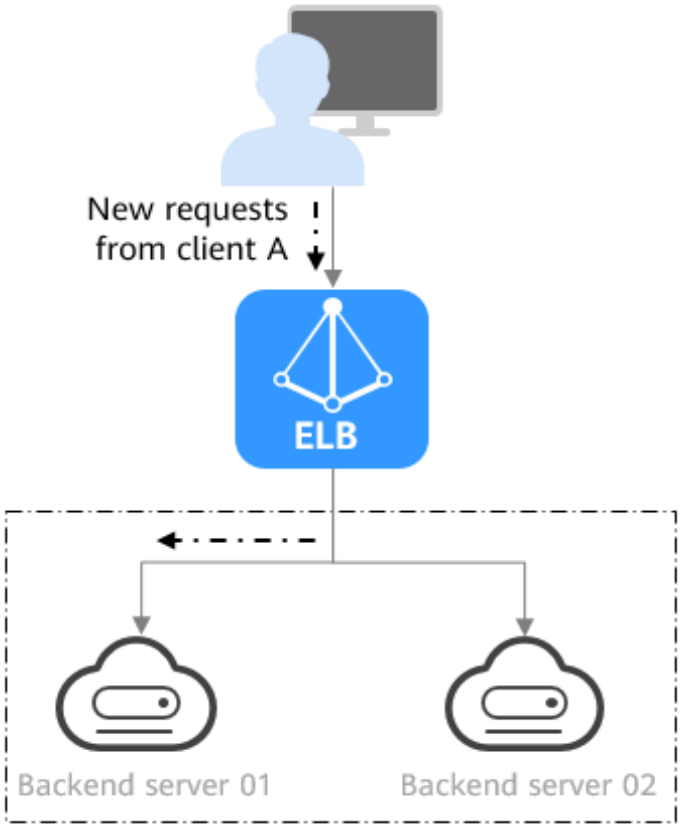
- Connection ID: The connection ID in the packet is calculated using the consistent hash algorithm to obtain a specific value, and backend servers are numbered. The generated value determines to which backend server the requests are routed. This allows requests with different connection IDs to be routed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



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?**

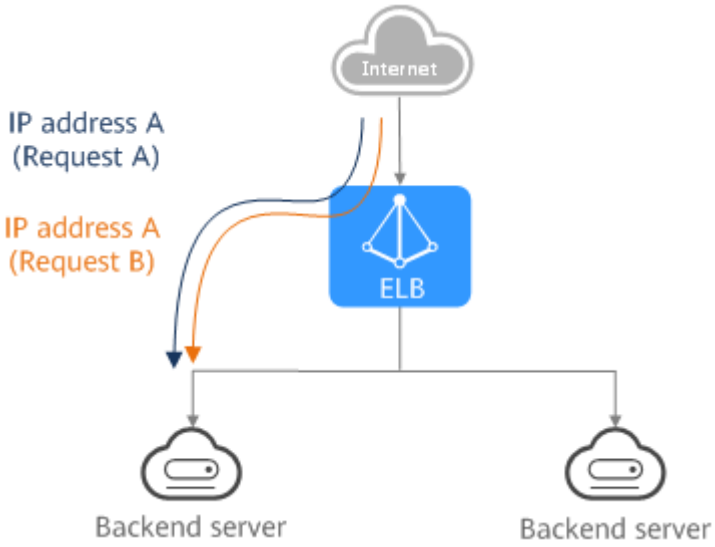
4 Application Scenarios

Heavy-Traffic Applications

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

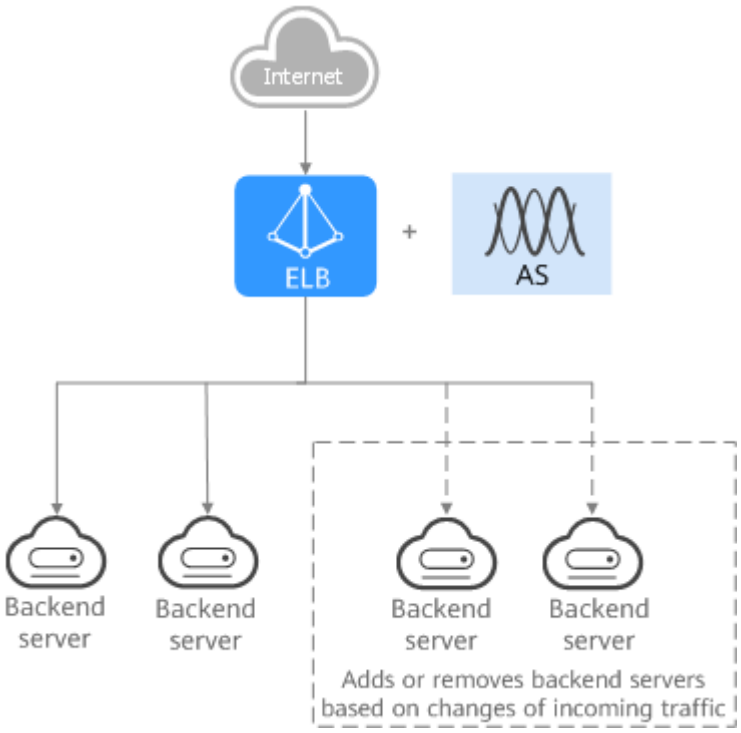
Figure 4-1 Session stickiness



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

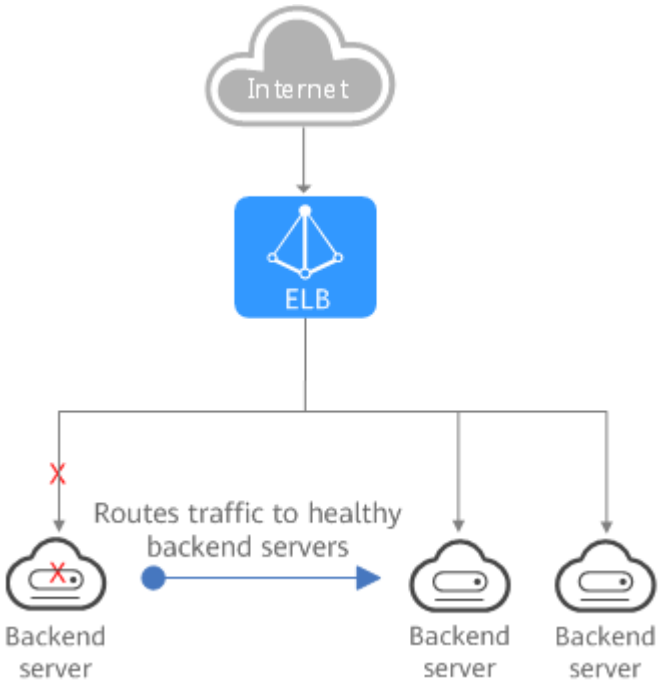


Zero SPOFs

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

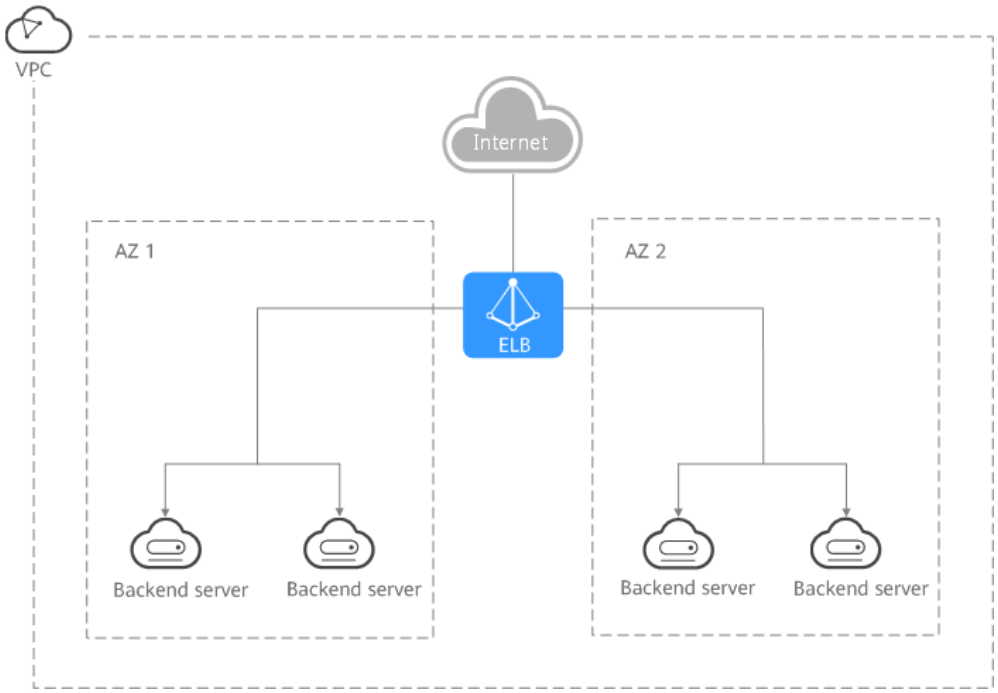


Cross-AZ Load Balancing

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 ELB Product 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. ELB expands the service capabilities 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	You get exclusive access to load balancer resources. The performance of a dedicated load balancer 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. They support guaranteed performance.

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**.	-
Performance	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.	If guaranteed performance is enabled, shared load balancers can handle up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.

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 two AZs, the requests the load balancers can handle will be doubled. ● For requests from a private network: – If clients are in the 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	-

Item	Dedicated Load Balancer	Shared Load Balancer
	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 you use your load balancers	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
Backend protocols	TCP, UDP, HTTP, HTTPS, and QUIC	TCP, UDP, and HTTP

Item	Dedicated Load Balancer	Shared Load Balancer
Forwarding capabilities	Provide powerful Layer 4 and Layer 7 processing capabilities to forward requests based on the following: • Forwarding rules: domain name, URL, 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 forward requests based on the following: • Forwarding rules: domain name and URL • Actions: forward to a backend server group and redirect to another listener
Key functions of backend server groups	• Health check • Sticky session • Slow start	• Health check • Sticky session
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	Load balancing
Backend type	• ECS • IP as backend server • Supplementary network interface • BMS • CCE Turbo cluster	• ECS • BMS • 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.	√	√
HTTP/HTTPS (Layer 7)	After receiving an access request, the listener needs to identify the request and forward data based on the fields in the HTTP/HTTPS packet header. Load balancing at Layer 7 provides some advanced features such as encrypted transmission and cookie-based sticky sessions.	√	√
HTTPS support	HTTPS can be used as both the frontend and backend protocol.	√	x
QUIC	If you use UDP and QUIC 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

Protocol	Description	Dedicated Load Balancer	Shared Load Balancer
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.	√	√
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.	√	√

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.	√	√
Private IPv4 network	The load balancer routes requests from the clients to backend servers in a VPC.	√	√
IPv6 network	Load balancers can route requests from IPv6 clients.	√	x
Changing a private IPv4 address	You can change the private IPv4 address into another one in the current subnet or other subnets.	√	x
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.	√	√
Modifying the bandwidth	You can change the bandwidth of public network load balancers as required.	√	√

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.	√	√
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.	√	√
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.	√	√
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.	√	√
Advanced features of HTTP/HTTPS listeners			
Default Security Policy	Allows you to 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.	√	√
Custom Security Policy	Allows you to select a TLS protocol and cipher suite to custom a security policy when you add HTTPS listeners.	√	x

Feature	Description	Dedicated Load Balancer	Shared Load Balancer
Transfer Load Balancer EIP	Allows you to store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header and pass it to backend servers.	√	√

Forwarding Capabilities

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

You can set forwarding rules and actions for a forwarding policy. For details, see **Table 5-6** and **Table 5-7**.

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.	√	√
URL	Route requests based on the URLs. There are three URL matching rules: exact match, prefix match, and regular expression match.	√	√
HTTP request method	Route requests based on the HTTP method. The options include GET, POST, PUT, DELETE, PATCH, HEAD, and OPTIONS.	√	x
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.	√	x
Query string	Route requests based on the query string.	√	x

Forwarding Rule	Description	Dedicated Load Balancer	Shared Load Balancer
CIDR block	Route requests based on source IP addresses from where the requests originate.	√	x

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.	√	√
Redirect to another listener	Redirect requests to an HTTPS listener, which then routes the requests to its associated backend server group.	√	x
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.	√	x
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.	√	x

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
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.	√	√
Sticky session	Requests from the same client will be routed to the same backend server during the session.	√	√
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.	√	x
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.	√	x

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.	√	√
Weighted least connections	Route requests to backend servers with the smallest ratio (current connections divided by weight).	√	√
Source IP hash	Route requests from the same client to the same backend server within a period of time.	√	√
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.	√	x

Backend Server Type

Table 5-10 Supported backend server types

Backend Server Type	Description	Dedicated Load Balancer	Shared Load Balancer
IP as backend server	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.	√	x
Supplementary network interface	You can attach supplementary network interfaces to backend servers.	√	x

Backend Server Type	Description	Dedicated Load Balancer	Shared Load Balancer
ECS	You can use load balancers to distribute incoming traffic across ECSs.	√	√
BMS	You can use load balancers to distribute incoming traffic across BMSs.	√	√
CCE Turbo cluster	You can use load balancers to distribute incoming traffic across CCE Turbo clusters. For details, see the *Cloud Container Engine User Guide*.	√	√

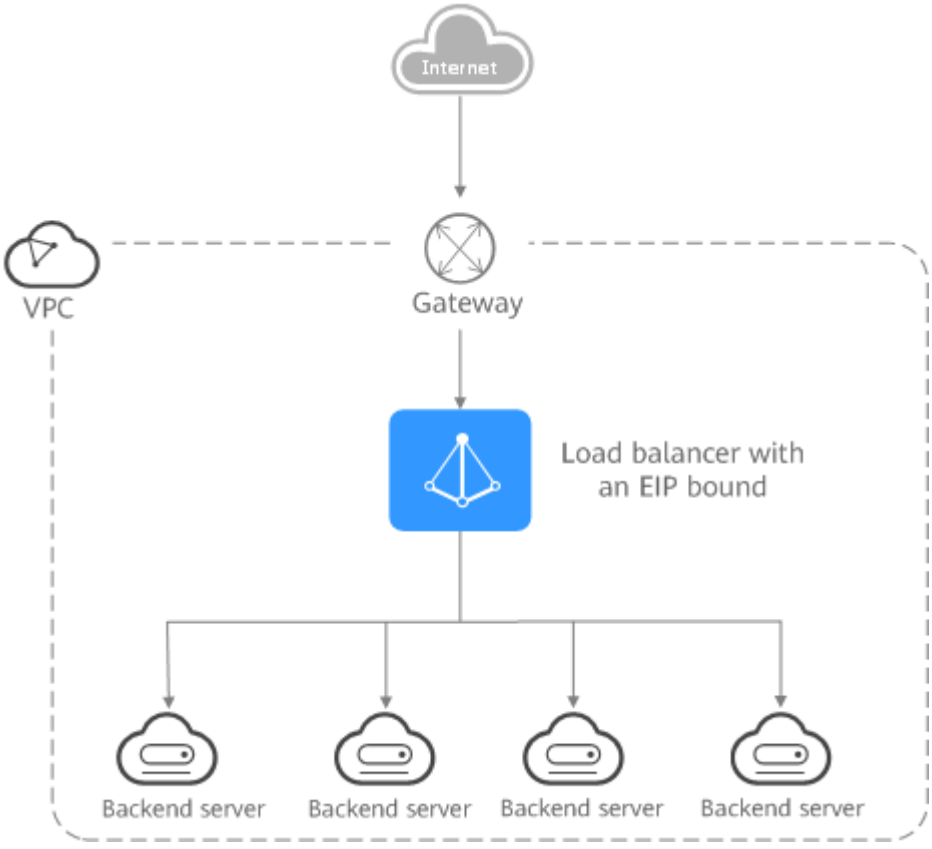
6 Load Balancing on a Public or Private Network

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.

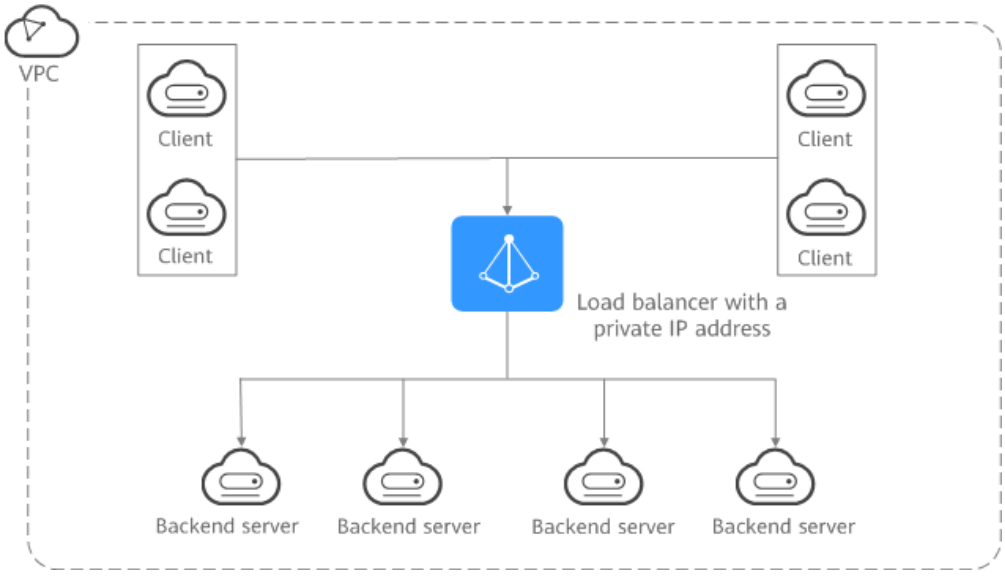
Figure 6-1 Load balancing on a public network



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

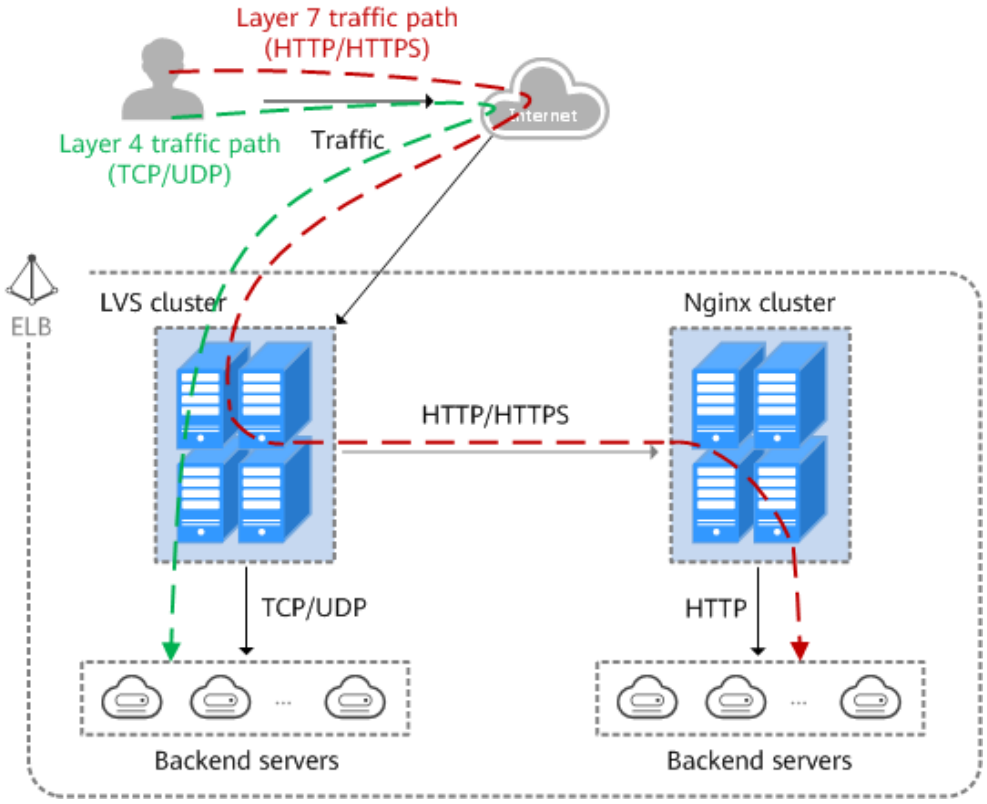
Load balancers communicate with backend servers over a private network.

- 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.

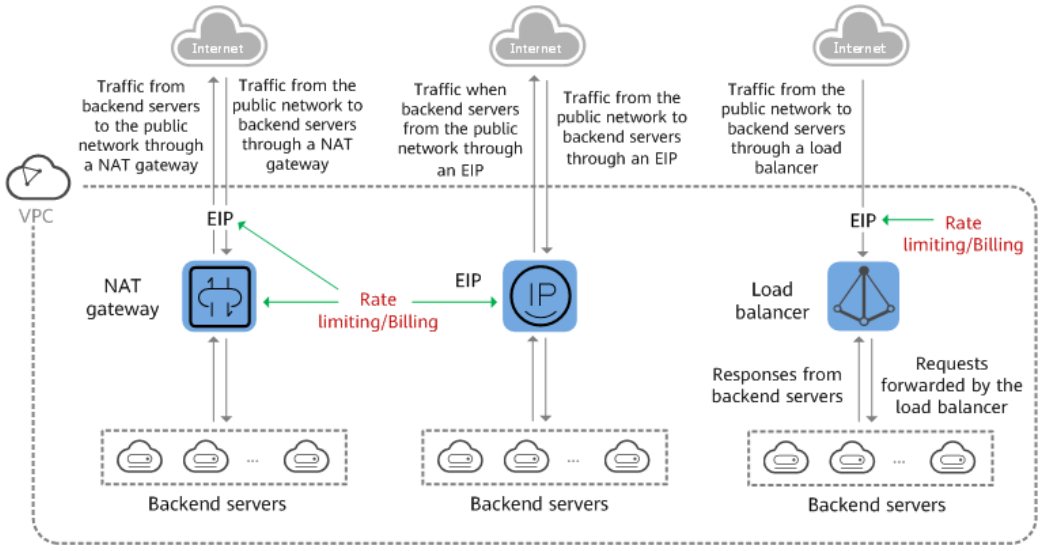
When a listener uses HTTP or HTTPS to receive incoming traffic:

- 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 specifications.

Table 8-1 Specifications comparison

Item	Elastic	Fixed
Application scenarios	• For fluctuating traffic • When you need to use resources temporarily and for urgent purposes	• For stable traffic • When you need to use resources for a long term
Network (TCP/UDP/TLS) load balancer performance	The performance multiplies as the number of AZs increases. **Table 8-2** shows the maximum performance in an AZ.	The performance multiplies as the number of AZs increases. **Table 8-4** shows the maximum performance in an AZ.
Application (HTTP/HTTPS) load balancer performance	The performance multiplies as the number of AZs increases. **Table 8-2** shows the maximum performance in an AZ.	The performance multiplies as the number of AZs increases. **Table 8-5** shows the maximum performance in an AZ.
Billing mode	Pay-per-use	• Pay-per-use • Yearly/Monthly
Billing items	• LCU • Load balancer	LCU
Capabilities	Same	

Elastic Specifications

If your service traffic fluctuates greatly, you can choose elastic specifications and select network or application load balancing that best meets your service needs.

> **NOTE**
>
> The listener protocol must match the load balancing type. For example, if you create an application load balancer, you can only add an HTTP or HTTPS listener to this load balancer.

Load balancers are available in different elastic 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.

- Maximum concurrent 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 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 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 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 8-2 Maximum elastic specifications

Protocol	Maximum Concurrent Connections	CPS	QPS	Bandwidth (Mbit/s)
Network load balancing (TCP/UDP)	20,000,000	400,000	-	10,000
Network load balancing (TLS)	20,000,000	400,000	-	10,000
Application load balancing (HTTP)	8,000,000	80,000	160,000	10,000

Protocol	Maximum Concurrent Connections	CPS	QPS	Bandwidth (Mbit/s)
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

- Maximum concurrent 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 **Table 8-4** and **Table 8-5**, new requests will be discarded to ensure the performance of existing 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 **Table 8-4** and **Table 8-5**, 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 8-3**.

Table 8-3 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 8-5**, and new requests will be properly routed.	• The new connections reach the CPS (HTTP) defined in **Table 8-5**, and new requests will be properly routed.

NOTE

Details in the **Table 8-3** 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 is defined in **Table 8-5**, 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 8-4 and **Table 8-5** list the fixed 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 listener protocol must match the load balancing type. For example, if you create an application load balancer, you can only add an HTTP or HTTPS listener to this load balancer.

Table 8-4 Fixed specifications for a network load balancer

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 8-5 Fixed specifications for an application load balancer

Type	Maximum Concurrent 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
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 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

- 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 30 SNI certificates. All the certificates can have up to 30 domain names.

  NOTE

  Listeners of a dedicated load balancer can have up to 50 SNI certificates. You can **submit a service ticket** to increase the quota.

- 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 (Shared Load Balancers)

Billing Items

- If your shared load balancers were created after February 10, 2023, guaranteed performance were enabled for them by default. To enable guaranteed performance for shared load balancers created before that time, refer to **Enabling Guaranteed Performance for a Shared Load Balancer**.
- Shared load balancers are billed on a pay-per-use or yearly/monthly basis as described in **Table 10-1**.

For details about load balancer pricing, see **ELB Pricing Details**. You can use the **price calculator** to quickly estimate the price for the load balancers that you select.

Table 10-1 Billing items

Billing Mode	Billing Item	Description
Pay-per-use	Load balancer	You are charged for how long you use each load balancer.

NOTE

- Shared load balancers will be billed based on the billing mode you have selected on the console.
- If you bind an EIP to a shared 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**.

Changing the Billing Mode

If you expect to use a load balancer for a long period of time, change its billing mode to yearly/monthly to save money. For details, see **Table 10-2**.

Table 10-2 Changing the billing mode

Billing Mode	Description
Pay-per-use	**Changing the Bandwidth Billing Option**

Renewal

You can renew your load balancers on the **Renewals** page of 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 the load balancers will be retained. During this period, certain functions of the load balancers cannot be used. For details, see **What Functions Will Become Unavailable If a Load Balancer Is Frozen?**

If you still fail to 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 Billing (Dedicated Load Balancers)

This section describes how dedicated load balancers will be billed.

Billing Item

You will be charged for how many LCUs you use and how long you use your load balancers as described in **Table 11-1**.

For details about the pricing, see **ELB Price Calculator**. Resources vary in different regions. Resources may vary depend on regions, see actual prices shown on the console.

Table 11-1 Billing items

Billing Item	Description
LCU	You are charged based on **the number of load balancer capacity units (LCUs) used by a dedicated load balancer per hour**.
Load balancer	You are charged for **how long you use each load balancer**. If the load balancer is used for less than 1 hour, you will be charged for the actual duration, accurate to seconds.

NOTE

- An LCU measures the dimensions on which a dedicated load balancer routes the traffic. See LCU price in **LCU Pricing**.
- If you deploy a dedicated load balancer in multiple AZs, its performance will multiply by the number of AZs. The number of LCUs is calculated as follows: Number of LCUs = LCUs of the selected specifications x Number of the selected AZs.
- For details about AZs, see **Region and AZ**.

Billing Mode

The billing items of dedicated load balancers vary by billing mode. For details, see **Table 11-2**.

Table 11-2 Varied billing items by billing mode

Billing Mode	Description	Specifications	LCU Price	Load Balancer Price
Pay-per-use	You are charged for how long you use each load balancer.	Elastic	√	√
		Fixed	√	×

NOTE

- √ indicates that the billing item is involved. × indicates that the billing item is not involved.
- 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**.

Constraints

The elastic specifications are available in CN Southwest-Guiyang1, CN East-Shanghai1, CN-Hong Kong, and AP-Singapore. They will soon be available in other regions.

LCU Pricing

An LCU measures the dimensions on which a dedicated load balancer routes the traffic. See LCU price in **Table 11-3**.

The unit price of LCU varies depending on the billing mode and specifications. See the actual price of LCU on the console. **LCU price (USD) = Unit price x Number of LCUs x Usage duration**.

Table 11-3 LCU pricing

Billing Mode	Specifications	Application Scenario	Description
Pay-per-use	Elastic	For fluctuating traffic	You are charged for how many LCUs you use.
	Fixed	For stable traffic	You are charged for the LCUs based on each fixed specification you select.

LCU Billing for Elastic Specifications

An LCU has four dimensions: **new connections**, **maximum concurrent connections**, **processed traffic**, and **rule evaluations**.

You can calculate the number of LCUs by taking the maximum LCUs consumed across the four dimensions.

NOTE

The number of LCUs is rounded up to the nearest integer.

Table 11-4 LCU dimensions

Dimension	Description
New connections	Number of new connections per second.
Maximum concurrent connections	The maximum number of concurrent connections that a load balancer can handle per minute.
Processed traffic	The amount of data transferred through a load balancer in GBs.
Rule evaluations (application load balancing)	The product of the number of rules processed by a load balancer and the number of queries per second (QPS). The first 10 processed rules are free. • When there are more than 10 processed rules, the number of rule evaluations is calculated as follows: Rule evaluations = QPS x (Number of processed rules – 10). • When there are 10 or less processed rules, the number of rule evaluations is equal to the QPS.

Table 11-5 lists the LCU performance supported by different protocols.

Table 11-5 LCU performance supported by different protocols

Protocol	New connections per second	Maximum concurrent connections per minute	Processed traffic	Rule evaluations per second
TCP	800	100,000	1 GB	-
UDP	400	50,000	1 GB	-
HTTP/HTTPS	25	3,000	1 GB	1,000

A pricing example for a network load balancer

Assume your network load balancer establishes 1,000 new TCP connections per second, each lasting for three minutes, and the traffic processed by your load balancer is 1,000 KB per second.

The unit price of LCU in the current region is $0.00833 USD/hour. The LCU price is calculated as the table shown below.

Table 11-6 LCU calculation

Dimension	Example	LCUs	Rounded Up LCUs
New connections per second	1,000 new TCP connections	1000 ÷ 800 = 1.25	2
Maximum concurrent connections per minute	The maximum established concurrent connections are calculated as: 1,000 new HTTP/HTTPS connections per second x 60s x 3 minutes = 180,000 connections	180000 ÷ 100000 = 1.8	2
Processed traffic per hour	1,000 KB/s x 60s x 60 minutes = 3.6 GB	3.6 ÷ 1 = 3.6	4

In this example, the traffic dimension consumes the most LCUs (**4** LCUs). Therefore, the LCU price is calculated based on the number of LCUs converted from the traffic.

The total LCU price for using this network load balancer for 2 hours is calculated as follows:

LCU price = Unit price x Number of LCUs x Usage duration = $0.00833 USD/hour x 4 LCUs x 2 hours =$0.06664 USD

A pricing example for an application load balancer

Assume your application load balancer establishes 1,000 new HTTP/HTTPS connections per second, each lasting for three minutes. A client sends an average of 400 requests per second and the traffic processed by this load balancer is 1,000 KB per second. You have configured 20 forwarding rules for your load balancer to route your client requests.

The unit price of LCU in the current region is $0.00833 USD/hour. The LCU price is calculated as the table shown below.

Table 11-7 LCU calculation

Dimension	Example	LCUs	Rounded Up LCUs
New connections per second	1,000 new connections	1000 ÷ 25 = 40	40
Maximum concurrent connections per minute	The maximum established concurrent connections are calculated as: 1,000 new HTTP/HTTPS connections per second x 60s x 3 minutes = 180,000 connections	180000 ÷ 3000 = 60	60
Processed traffic per hour	1,000 KB/s x 60s x 60 minutes = 3.6 GB	3.6 ÷ 1= 3.6	4
Rule evaluations per second	Rule evaluations are calculated as: Rule evaluations = QPS x (Number of processed rules – 10) = 400 x (20 – 10) = 4,000	4000 ÷ 1000 = 4	4

In this example, the maximum concurrent dimension consumes the most LCUs (**60** LCUs). Therefore, the LCU price is calculated based on the number of LCUs converted from the maximum concurrent connections.

The total LCU price for using this application load balancer for 2 hours is calculated as follows:

LCU price = Unit price x Number of LCUs x Usage duration = $0.00833 USD/hour x 60 LCUs x 2 hours = $0.9996 USD

LCU Billing for Fixed Specifications

You are charged for the LCUs based on each fixed specification you select. You can select either application load balancing (HTTP/HTTPS) or network load balancing (TCP/UDP), or both.

You can refer to **Specifications of Dedicated Load Balancers** for each fixed specification and select a fixed specification that best meets your service requirements.

Pay-per-use

The following table lists the converted number of LCUs of each fixed specification.

Table 11-8 Converted number of LCUs of each fixed specification

Type	LCUs in an AZ (TCP/UDP)	LCUs in an AZ (HTTP/HTTPS)
Small I	10	10
Small II	20	20
Medium I	40	40
Medium II	80	100
Large I	200	200
Large II	400	400

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 = LCUs of the selected specification x Number of the selected AZs.

Load Balancer Price

You are charged for how long you use each load balancer. If the load balancer is used for less than 1 hour, you will be charged for the actual duration, accurate to seconds. The billing cycle is from the time when the dedicated load balancer is created to the time when it is deleted.

Only load balancers with elastic specifications in pay-per-use billing mode are charged.

Changing Specifications or Billing Modes

You can change the specifications or billing mode of a dedicated load balancer.

Table 11-9 lists the specifications that you can change. For details about how to change the specifications of a dedicated load balancer, see **Changing the Specifications of a Dedicated Load Balancer**.

Table 11-9 Supported change options for a pay-per-use load balancer

Billing Mode	Specifications	Change to Elastic	Change to Fixed	Adding Load Balancing Type	Removing Load Balancing Type	Upgrading Specifications	Downgrading Specifications
Pay-per-use	Elastic	-	√	√	√	x	x

Billing Mode	Specifications	Change to Elastic	Change to Fixed	Adding Load Balancing Type	Removing Load Balancing Type	Upgrading Specifications	Downgrading Specifications
	Fixed	√	-	√	√	√	√

Table 11-10 describes whether you can change the billing mode of a load balancer.

Table 11-10 Changing the billing mode

Billing Mode	Specifications	Description
Pay-per-use	Elastic	Cannot be changed.
	Fixed	For details, see **Changing the Billing Mode or Bandwidth Billing Option**

Renewal

You can renew a dedicated load balancer in either of the following ways:

- On the ELB console, locate the load balancer and click **More** > **Renew** in the **Operation** column.
- On the **Renewals** page of the Billing Center, renew the subscription. 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.

12 Security

12.1 Shared Responsibilities

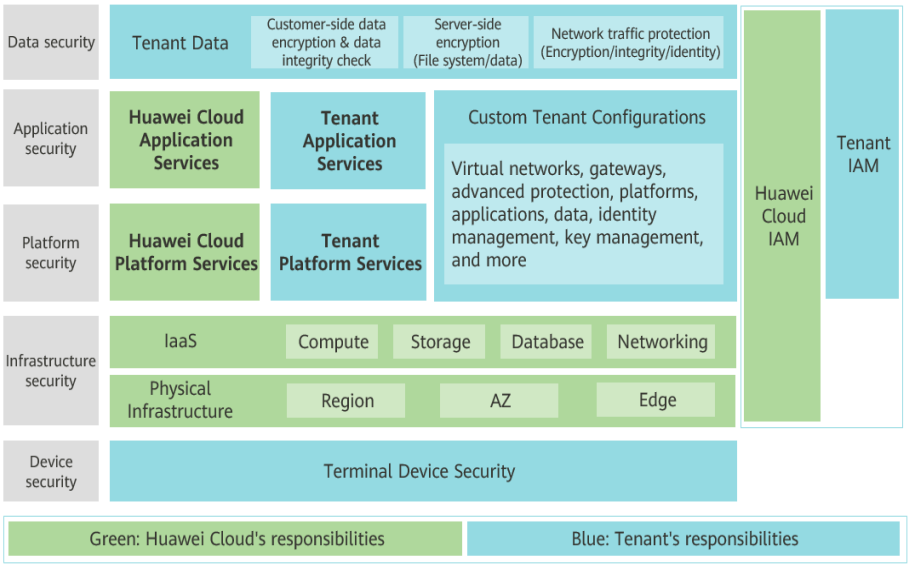
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Figure 12-1 illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud**: Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in the broader sense, the security and compliance of our infrastructure and services.
- **Tenant**: Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OS of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

Huawei Cloud Security White Paper elaborates on the ideas and measures for building Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 12-1 Huawei Cloud shared security responsibility model



12.2 Identity and Access Management

Identity Authentication

You can use Identity and Access Management (IAM) to control access to your ELB resources. IAM permissions define which actions on your cloud resources are allowed or denied. After creating an IAM user, the administrator needs to add it to a user group and grant the permissions required by ELB to the user group. Then, all users in this group automatically inherit the granted permissions.

For details, see **Permissions**.

Access Control

Access control allows you to add a whitelist or blacklist to specify IP addresses that can or cannot access a listener. A whitelist allows specified IP addresses to access the listener, while a blacklist denies access from specified IP addresses. For details, see **Access Control**.

12.3 Data Protection

When you add HTTPS listeners, you can select appropriate security policies to improve service security. A security policy is a combination of TLS protocols of different versions and supported cipher suites. You can select the default security policy or create a custom security policy. For details, see **TLS Security Policy**.

12.4 Auditing and Logging

Cloud Trace Service (CTS) is a log audit service for Huawei Cloud security. It allows you to collect, store, and query cloud resource operation records. You can use

these records to perform security analysis, audit compliance, track resource changes, and locate faults.

After CTS is enabled, it can record ELB operations.

- For details about how to enable and configure CTS, see **Enabling CTS**.
- For details about supported operations on ELB, refer to **Key Operations Recorded by CTS**.
- For details about how to view traces, see **Viewing Traces**.

12.5 Resilience

Huawei Cloud ELB provides multi-AZ, multi-cluster disaster recovery in more than 20 countries and regions around the world. Even if some nodes, clusters, or regions are faulty, your services will not be interrupted, greatly improving service reliability.

12.6 Risk Control

With Cloud Eye, you can dynamically analyze potential risks by viewing the network traffic and error logs of ELB during selected period of time.

You can also configure Cloud Eye to view updated logs on ELB to alert you of any potential issues in real time.

13 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 13-1 lists all the system-defined permissions for ELB.

Table 13-1 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 ReadOnly Access	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 13-2 describes common operations supported by each system policy of ELB.

Table 13-2 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
Querying a load balancer and associated resources	Supported	Supported	Supported

Operation	ELB FullAccess	ELB ReadOnlyAccess	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	ELB FullAccess	ELB ReadOnlyAccess	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 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*.

14 Product Concepts

14.1 Basic Concepts

Table 14-1 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.
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.

14.2 Region and AZ

Concept

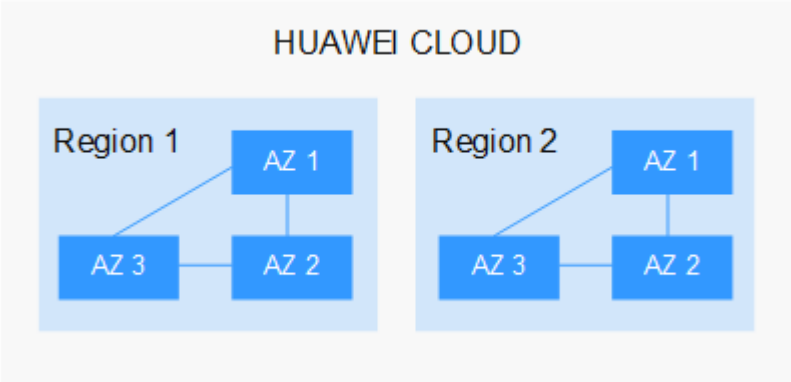
A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- 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. AZs within a region are interconnected using high-speed optical fibers, to support cross-AZ high-availability systems.

Figure 14-1 shows the relationship between regions and AZs.

Figure 14-1 Regions and AZs



Huawei Cloud provides services in many regions around the world. You can select a region and an AZ based on requirements. For more information, see **Huawei Cloud Global Regions**.

Selecting a Region

When selecting a region, consider the following factors:

- Location

  It is recommended that you select the closest region for lower network latency and quick access.

  - If your target users are in Asia Pacific (excluding the Chinese mainland), select the **CN-Hong Kong**, **AP-Bangkok**, or **AP-Singapore** region.
  - If your target users are in Africa, select the **AF-Johannesburg** region.
  - If your target users are in Latin America, select the **LA-Santiago** region.

    NOTE

    The **LA-Santiago** region is located in Chile.

- Resource price

  Resource prices may vary in different regions. For details, see **Product Pricing Details**.

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.

Regions and Endpoints

Before you use an API to call resources, specify its region and endpoint. For more details, see **Regions and Endpoints**.

15 How ELB Works with Other Services

Table 15-1 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 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**

16 Change History

Released On	Description
2023-08-21	This issue is the ninth official release. Added **Quotas and Constraints**
2022-02-15	This issue is the eighth official release, which incorporates the following changes: Added ELB infographics.
2021-12-14	This issue is the seventh official release, which incorporates the following changes: Added the following section: Billing (Dedicated Load Balancers).
2021-12-09	This issue is the sixth official release, which incorporates the following changes: Added HTTPS queries per second in the specifications of application load balancing in section *Specifications of Dedicated Load Balancers*.
2021-10-28	This issue is the fifth official release, which incorporates the following changes: Added **Permissions**.
2021-06-18	This issue is the fourth official release, which incorporates the following changes: Deleted all descriptions and operations related to classic load balancers.

Released On	Description
2020-05-30	This issue is the third official release, which incorporates the following changes: Changed the name of enhanced load balancers to shared load balancers.
2018-12-30	This issue is the second official release, which incorporates the following changes: ● Adjusted the table of contents. ● Optimized the content of each section.
2018-11-21	This issue is the first official release.

