Elastic Load Balance

User Guide

Issue 01

Date 2024-04-15

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Contents

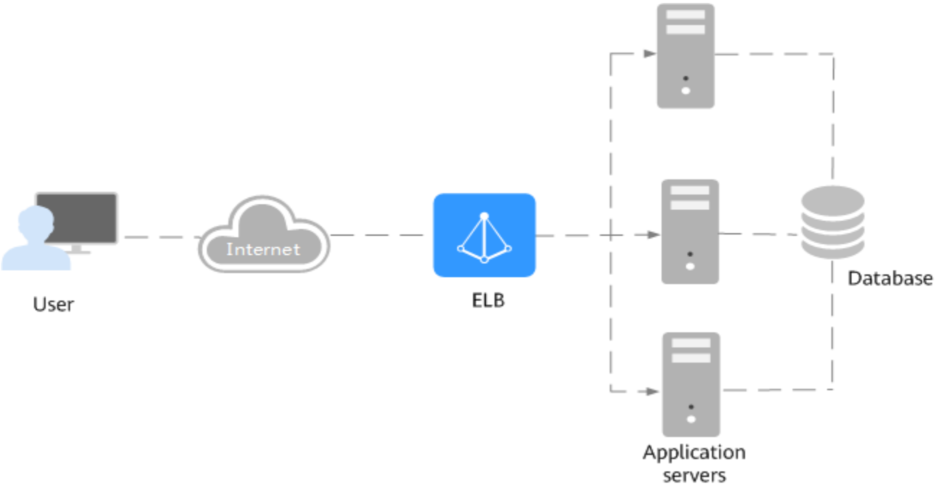
1 Service Overview

1.1 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:

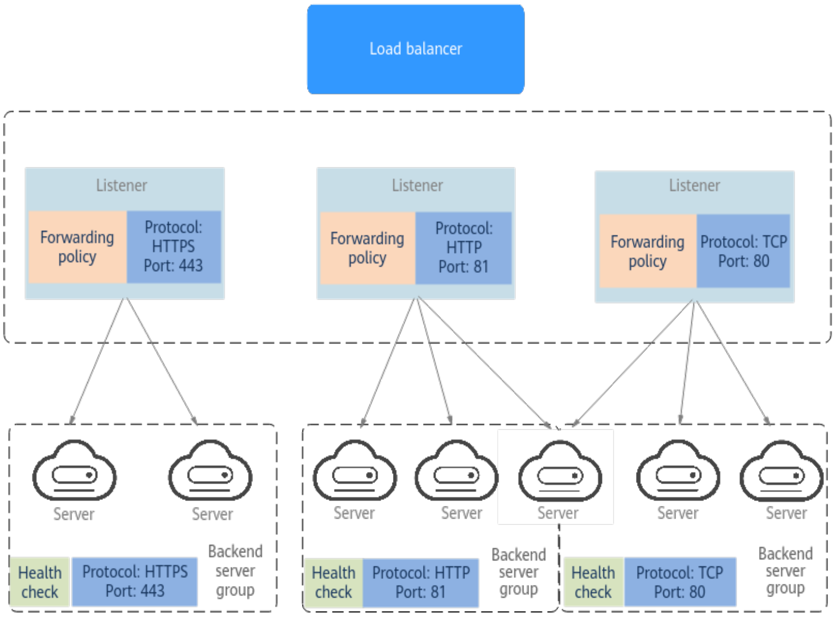
- **Load balancer**: distributes incoming traffic across backend servers in one or more availability zones (AZs).

- **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*.

1.2 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 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.

- 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

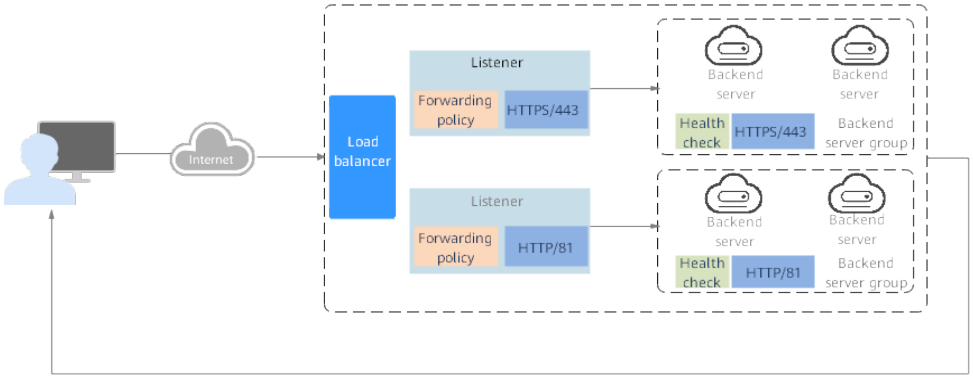
  ELB can route requests based on their content, such as the request method, header, URL, path, and source IP address. They can also redirect requests to another listener or URL, or return a fixed response to the clients.
- No limits

  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.

1.3 How ELB Works

Figure 1-3 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.

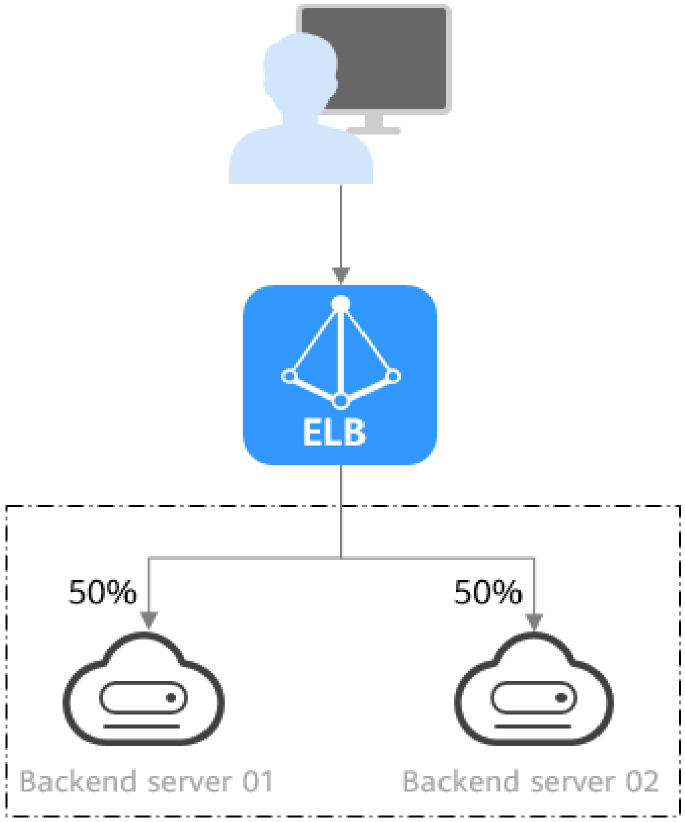
Load Balancing Algorithms

ELB supports the following load balancing algorithms:

- 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.

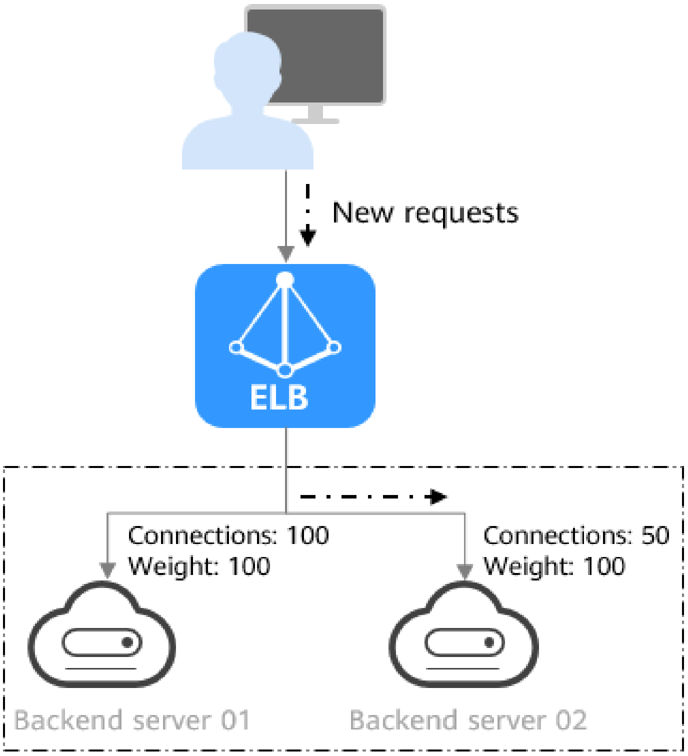
Figure 1-4 Traffic distribution using the weighted round robin algorithm



- 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.

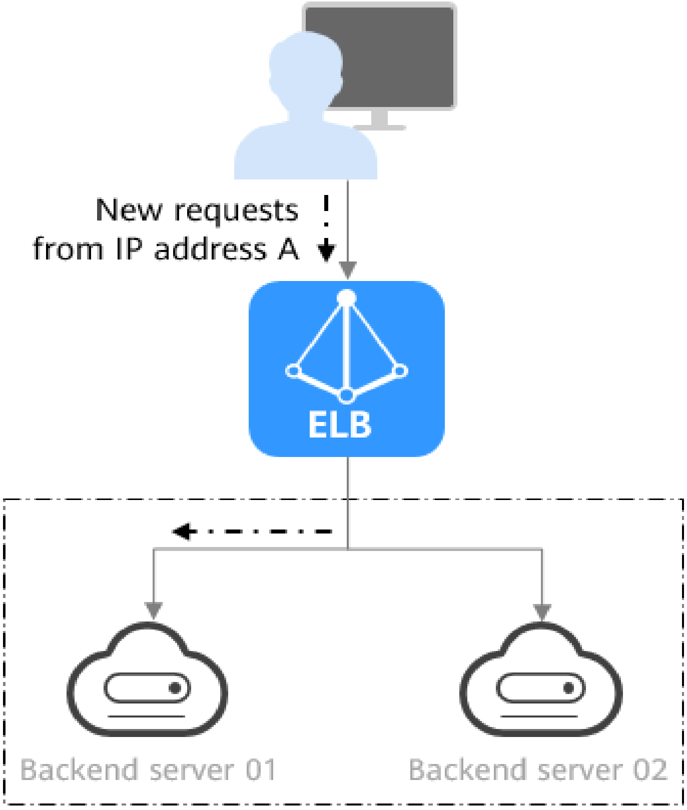
Figure 1-5 Traffic distribution using the weighted least connections algorithm



- 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 1-6 Traffic distribution using the source IP hash algorithm



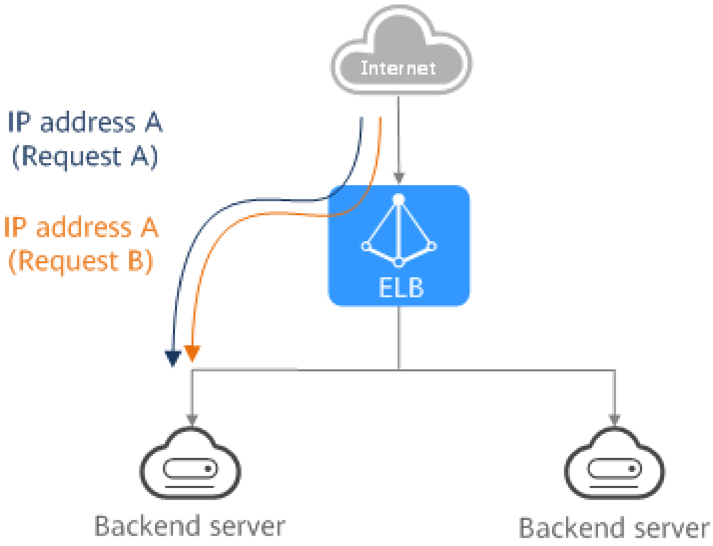
1.4 Application Scenarios

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.

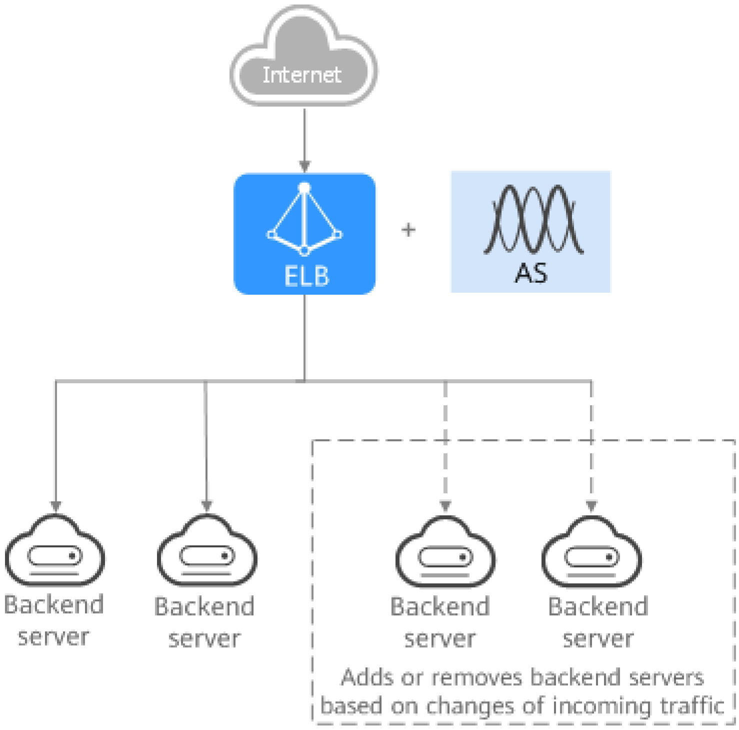
Figure 1-7 Session stickiness



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 1-8 Flexible scalability

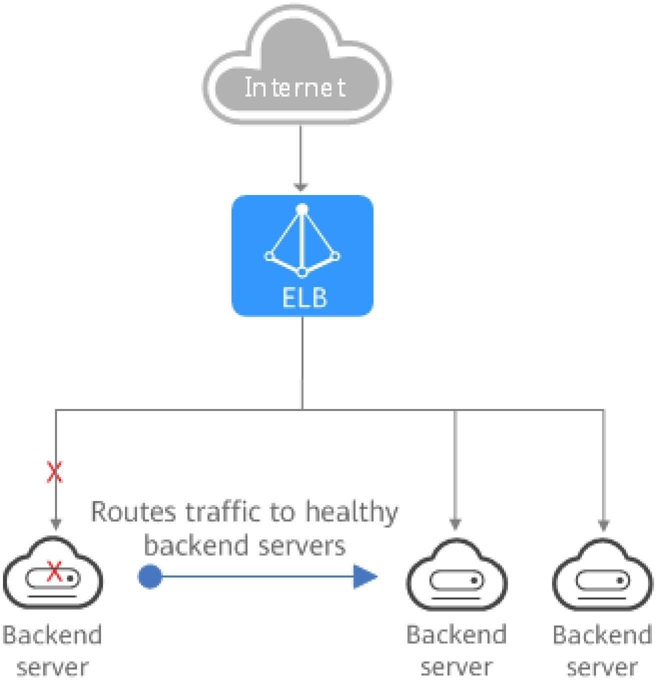


Zero SPOFs

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 1-9 Eliminating SPOFs

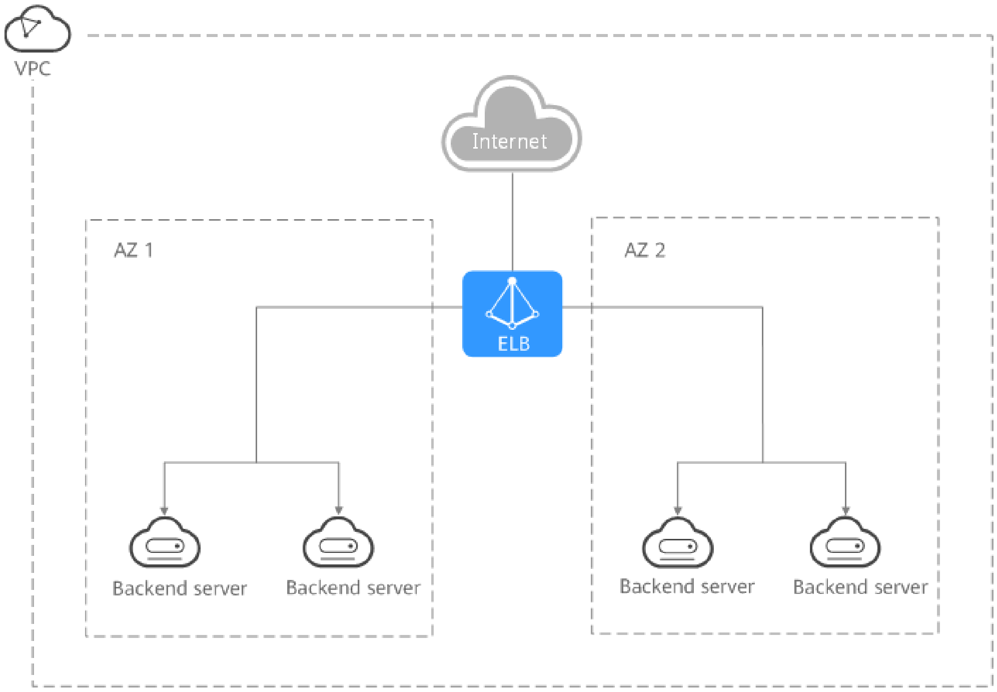


Cross-AZ Load Balancing

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 1-10 Traffic distribution to servers in one or more AZs



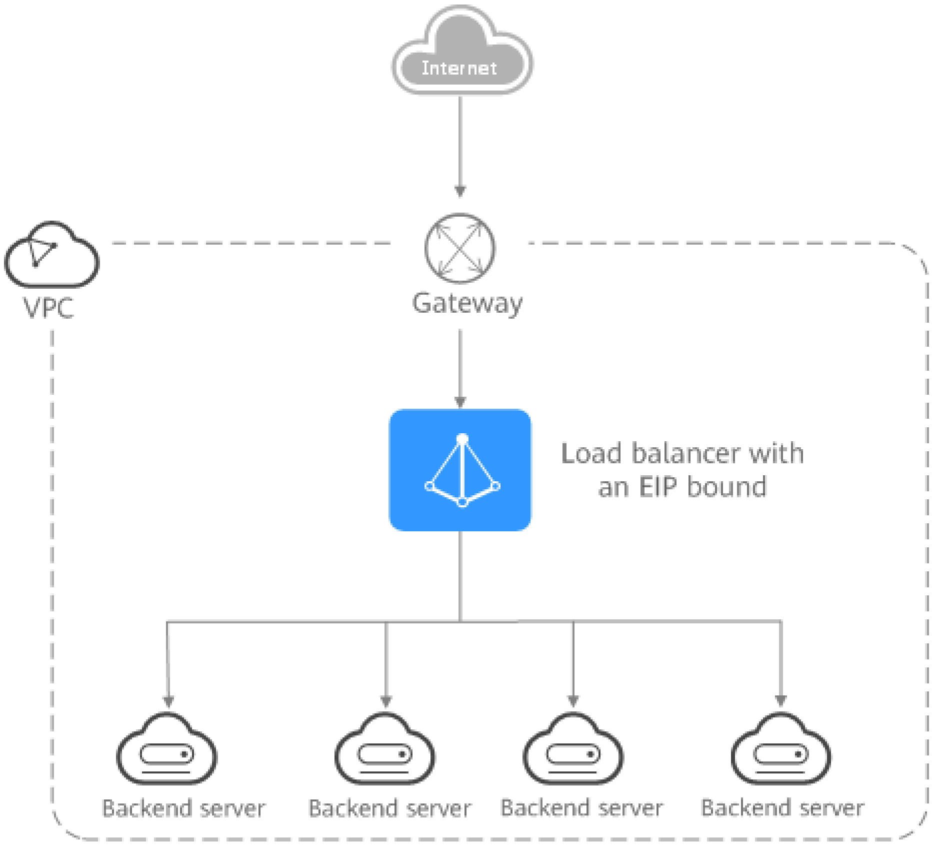
1.5 Load Balancing on a Public or Private Network

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.

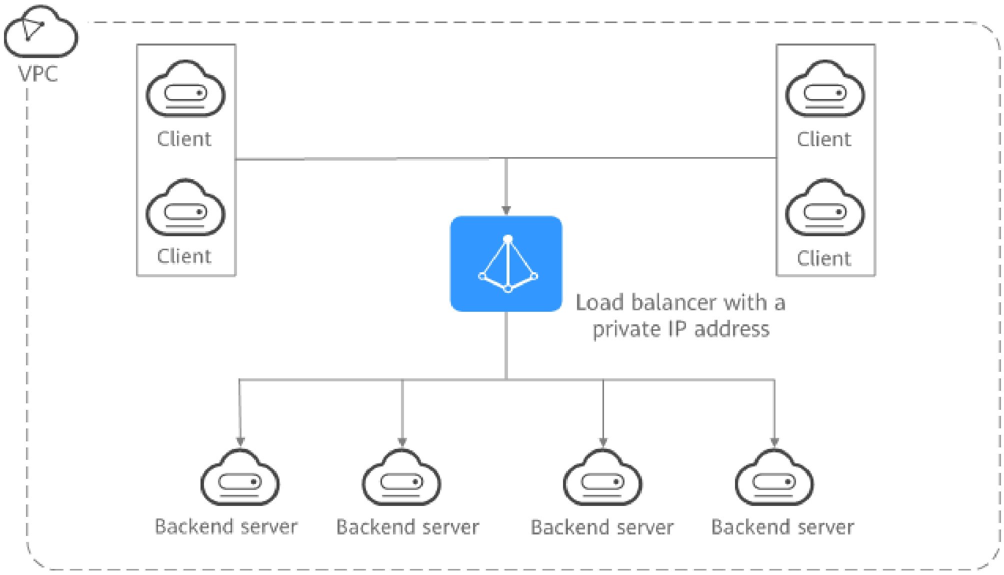
Figure 1-11 Load balancing on a public network



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 1-12 Load balancing on a private network



1.6 Network Traffic Paths

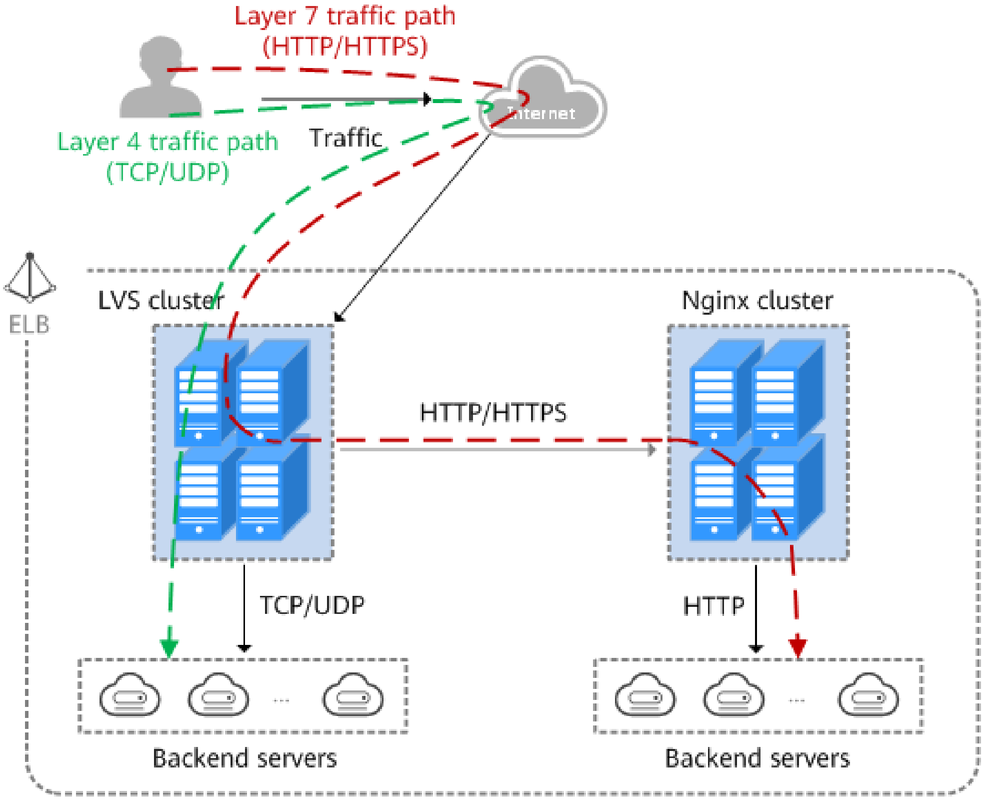
Load balancers communicate with backend servers over a private network.

- 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 1-13 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.

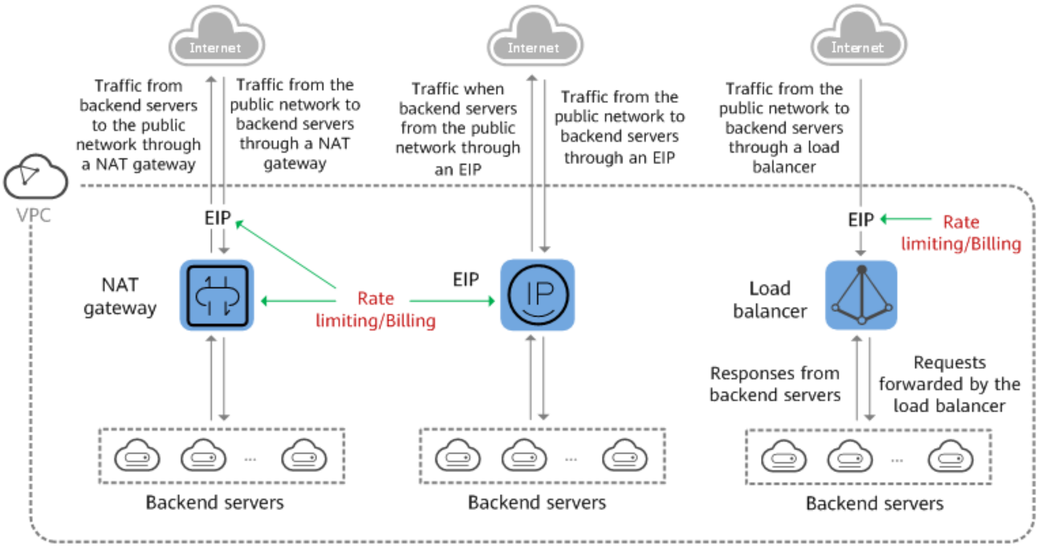
When a listener uses HTTP or HTTPS to receive incoming traffic:

- 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 1-14 Outbound network traffic


Internet
Internet
Internet
Traffic from backend servers to the public network through a NAT gateway
Traffic from the public network to backend servers through a NAT gateway
Traffic when backend servers from the public network through an EIP
Traffic from the public network to backend servers through an EIP
Traffic from the public network to backend servers through a load balancer
VPC
EIP
EIP
EIP
Rate limiting/Billing
Rate limiting/Billing
NAT gateway
Load balancer
Responses from backend servers
Requests forwarded by the load balancer
Backend servers
Backend servers
Backend servers

- 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.

1.7 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 1-1 lists the default resource quotas. Each user may have different resource quotas.

Table 1-1 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.

Table 1-2 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

- 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.
- 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 1-3 Restrictions on forwarding policies

Load Balancer Type	Advanced Forwarding	Forwarding Rule	Action
Dedicated	Disabled	Domain name and URL	**Forward to another backend server group** and **Redirect to another listener**
	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**

Backend Server Group

The backend protocol of the backend server group must match the frontend protocol of the listener as described in **Table 1-4**.

Table 1-4 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.

1.8 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 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 *Elastic Load Balance API Reference*.

Table 1-5 lists all the system-defined permissions for ELB.

Table 1-5 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 1-6 describes common operations supported by each system policy of ELB.

Table 1-6 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
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

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
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
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*.

1.9 Product Concepts

1.9.1 Basic Concepts

Table 1-7 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.
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.

Term	Definition
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.

1.9.2 Region and AZ

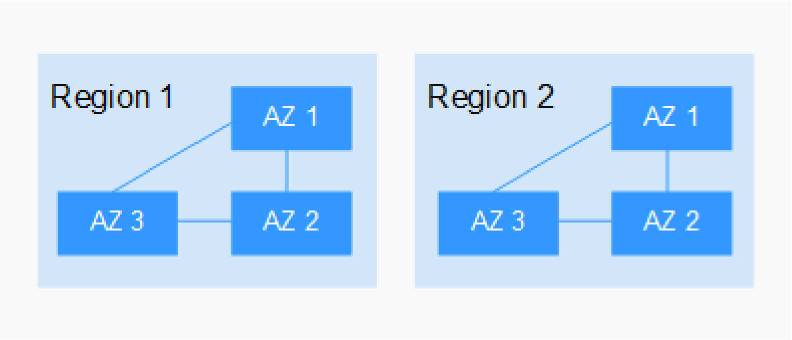
Concept

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

- A region is a physical data center, which is completely isolated to improve fault tolerance and stability. The region that is selected during resource creation cannot be changed after the resource is created.
- An AZ is a physical location where resources use independent power supplies and networks. A region contains one or more AZs that are physically isolated but interconnected through internal networks. Because AZs are isolated from each other, any fault that occurs in one AZ will not affect others.

Figure 1-15 shows the relationship between regions and AZs.

Figure 1-15 Regions and AZs



Selecting a Region

Select a region closest to your target users for lower network latency and quick access.

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**.

1.10 How ELB Works with Other Services

- Virtual Private Cloud (VPC)

  Provides IP addresses and bandwidth for load balancers.
- Auto Scaling (AS)

  Works with ELB to automatically scale the number of backend servers for faster traffic distribution.
- Identity and Access Management (IAM)

  Provides authentication for ELB.
- Elastic Cloud Server (ECS)

  Provides servers to run your applications in the cloud. Configure load balancers to route traffic to the servers or containers.
- Log Tank Service (LTS)

  Stores access logs of HTTP or HTTPS requests to your load balancer for query and analysis later if you have enabled access logging.
- Cloud Eye

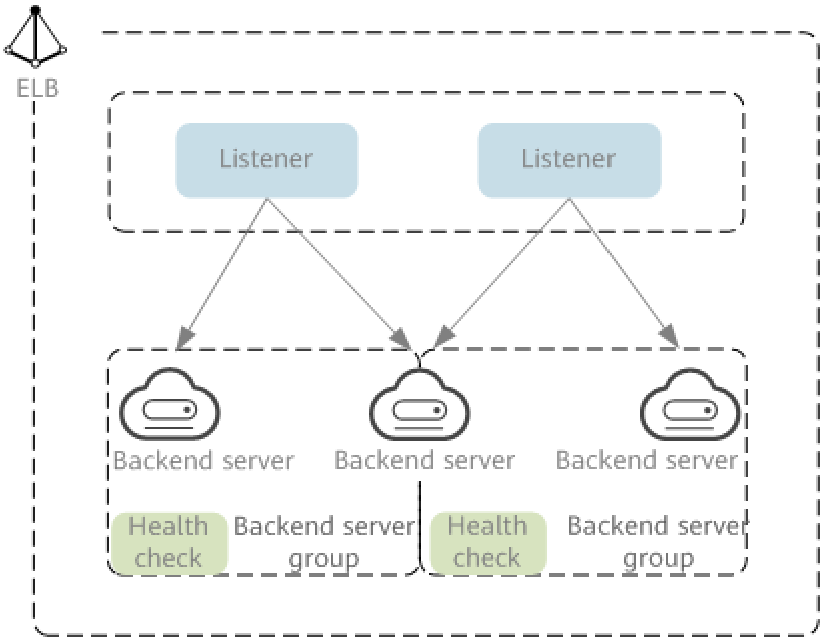
  Monitors the status of load balancers and listeners, without any additional plug-in.

2 Load Balancer

2.1 Overview

A load balancer distributes incoming traffic across multiple backend servers. Before using a load balancer, you need to add at least one listener to it and associate one backend server with it.

Figure 2-1 ELB components



Network Type

Load balancers can work on both public and private network.

- Load balancers on the public network route requests over the Internet.

  Each load balancer has an EIP bound so that it can receive requests from clients on the Internet and routes the requests across backend servers.

Application scenario

- A load balancer is used as a single point of contact for clients when a group of servers provide services over the Internet.
- Fault tolerance and fault recovery are necessary.

- Load balancers on a private network route requests within a VPC.

  This type of load balancers has only private IP addresses and can be accessed only in the VPC. They receive requests from clients in a VPC and route the requests across backend servers in the same VPC.

  Application scenario

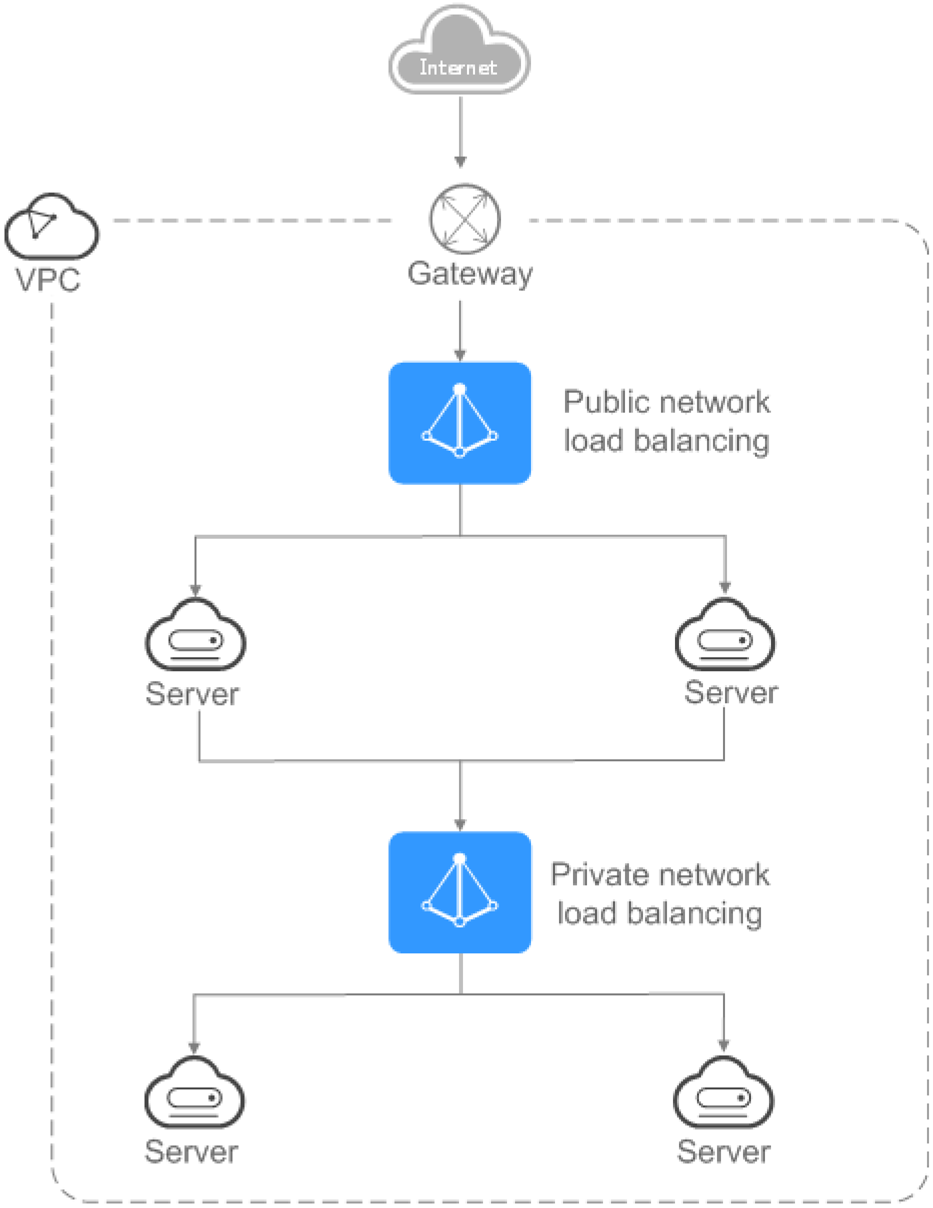
  Both clients and backend servers are in the same VPC as the load balancer.

  - There are multiple backend servers, and requests need to be evenly distributed across these servers.
  - Fault tolerance and fault recovery are necessary.
  - You do not want IP addresses of your physical devices to be exposed.

  Load balancing on both public and private networks

  Suppose that you have deployed both web servers and database servers. The web servers are accessible from users on the Internet, while the database servers can be accessed only on the private network. In this case, you can create two load balancers, one for the web servers and one for the database servers. The load balancer on the public network receives requests over the Internet and routes the requests to the web servers. Then, the load balancer on the private network forwards the requests to database servers.

Figure 2-2 Load balancing on both public and private networks



2.2 Preparations for Creating a Load Balancer

Before creating a load balancer, you must plan its region, network, protocol, and backend servers.

Region

When you select a region, note the following:

- The region must be close to your users to reduce network latency and improve the download speed.
- You can associate backend servers across regions or in a different VPC with a dedicated load balancer in either of the following ways:

  - To add backend servers in a different VPC or an on-premises data center, you need to enable **IP as a Backend** for the load balancer. For details, see Configuring Hybrid Load Balancing.

AZ

Dedicated load balancers can be deployed across AZs. If you select multiple AZs, a load balancer is created in each selected AZ.

Load balancers in these AZs work in active-active or multi-active mode and requests are distributed by the nearest load balancer in the same AZ.

To reduce network latency and improve access speed, you are suggested to deploy your load balancer in the AZ where backend servers are running.

If disaster recovery is required, create load balancers based on the scenario:

- **One load balancer in multiple AZs (disaster recovery at the AZ level)**

  If the number of requests does not exceed what the largest specifications (large II) can handle, you can create a load balancer and select multiple AZs. In this way, if the load balancer in a single AZ is abnormal, the load balancer in other AZs can route the traffic, and disaster recovery can be implemented among multiple AZs.
- **Multiple load balancers and each load balancer in multiple AZs (disaster recovery at both the load balancer and AZ level)**

  If the number of requests exceeds what the largest specifications (large II) can handle, you can create multiple load balancers and select multiple AZs for each load balancer. In this way, if a single load balancer is abnormal, other load balancers can distribute the traffic, and disaster recovery can be implemented among multiple load balancers and AZs.

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 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.

Network Type

Dedicated load balancers support IPv4 public network, IPv4 private network, and IPv6 network.

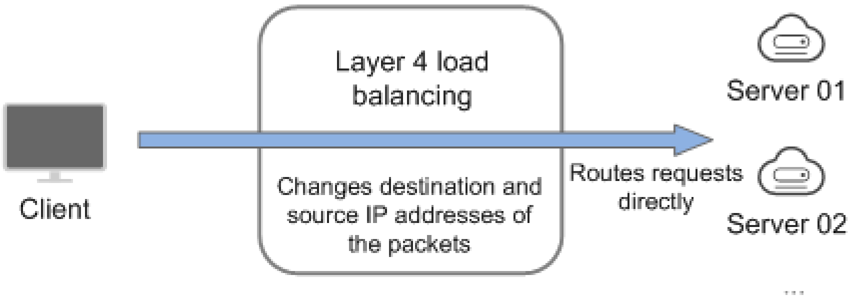
- If you select the public IPv4 network, the load balancer will have an IPv4 EIP bound to route requests over the Internet.
- If you select the private IPv4 network, a private IPv4 address will be assigned to the load balancer to route requests within a VPC.
- If you select the IPv6 network, the load balancer will have an IPv6 address, which allows the load balancer to route requests within a VPC. If you add the IPv6 address to a shared bandwidth, the load balancer can also process requests over the Internet.

Protocol

ELB provides load balancing at both Layer 4 and Layer 7.

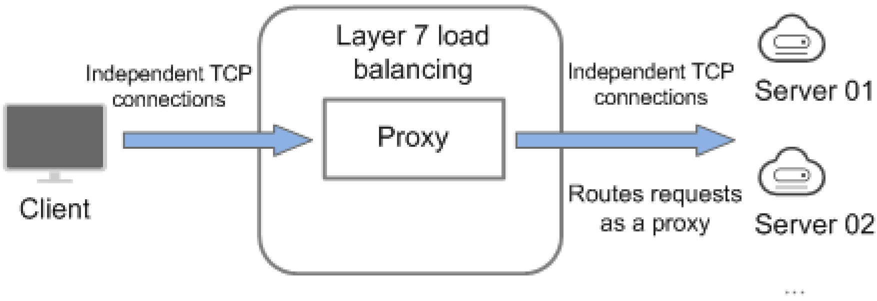
- If you choose TCP or UDP, the load balancer routes requests directly to backend servers. In this process, the destination IP address in the packets is changed to the IP address of the backend server, and the source IP address to the private IP address of the load balancer. A connection is established after a three-way handshake between the client and the backend server, and the load balancer only forwards the data.

Figure 2-3 Layer-4 load balancing



- Load balancing at Layer 7 is also called "content exchange". After the load balancer receives a request, it works as a proxy of backend servers to establish a connection (three-way handshake) with the client and then determines to which backend server the request is to be routed based on the fields in the HTTP/HTTPS request header and the load balancing algorithm you selected when you add the listener.

Figure 2-4 Layer-7 load balancing


Layer 7 load balancing
Independent TCP connections
Proxy
Independent TCP connections
Client
Routes requests as a proxy
Server 01
Server 02
...

Backend Servers

Before you use ELB, you need to create cloud servers, deploy required applications on them, and add the cloud servers to one or more backend server groups. When you create ECSs or BMSs, note the following:

- Cloud servers must be in the same region as the load balancer.
- Cloud servers that run the same OS are recommended so that you can manage them more easily.

2.3 Creating a Dedicated Load Balancer

Scenarios

You have prepared everything required for creating a load balancer. For details, see **Preparations for Creating a Load Balancer**.

Constraints

- After a load balancer is created, the VPC cannot be changed. If you want to change the VPC, create a load balancer and select a different VPC.
- To ping the IP address of a load balancer, you need to add a listener to it.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > .
4. On the **Load Balancers** page, click Create Elastic Load Balancer. Complete the basic configurations based on **Table 2-1**.

Table 2-1 Parameters for configuring the basic information

Parameter	Description	Example Value
Region	Specifies the desired region. Resources in different regions cannot communicate with each other over internal networks. For lower network latency and faster access to resources, select the nearest region.	-

Parameter	Description	Example Value
AZ	Specifies the AZ of the load balancer. You can deploy a load balancer in multiple AZs for high availability. If an AZ becomes faulty or unavailable, the load balancers in other AZs can route requests to backend servers to ensure service continuity and improve application reliability. **NOTE** If you change the AZs of an existing load balancer, the load balancer may fail to route requests for several seconds. It is recommended that you plan the AZs in advance, or change the AZs during off-peak hours when necessary.	-
Name	Specifies the load balancer name.	elb-test
Description	Provides supplementary information about the load balancer.	-

Table 2-2 Tag naming rules

Item	Requirement	Example Value
Tag key	● Cannot be empty. ● Must be unique for the same load balancer. ● Can contain a maximum of 36 characters.	elb_key1
Tag value	● Can contain a maximum of 43 characters.	elb-01

5. Configure the network parameters based on **Table 2-3**.

Table 2-3 Parameters for network configurations

Parameter	Description	Example Value
Network Type	Specifies the network where the load balancer works. You can select one or more network types. ● **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**: An IPv6 address will be assigned to the load balancer to route requests from IPv6 clients. **NOTE** If you do not select any of the options, the load balancer cannot communicate with the clients after it is created. When you are using ELB or testing network connectivity, ensure that the load balancer has a public or private IP address bound.	Public IPv4 network
VPC	Specifies the VPC where the load balancer works. Select an existing VPC or create a new one. For more information about VPC, see the *Virtual Private Cloud User Guide*.	vpc-test
Frontend Subnet	Specifies the subnet where the load balancer will work. The system assigns IP addresses to load balancers for receiving requests based on the configured network type. ● **IPv4 private network**: assigns IPv4 private addresses. ● **IPv6 network**: assigns IPv6 private or public addresses. **NOTE** If you select **IPv6 network** for **Network Type** and the selected VPC does not have any subnet that supports IPv6, enable IPv6 for the subnets or create a subnet that supports IPv6. For details, see the Virtual Private Cloud User Guide.	subnet-test

Parameter	Description	Example Value
Backend Subnet	The load balancer uses the IP addresses in the backend subnet to forward requests to the backend servers. ● Select **Subnet of the load balancer** by default. ● Select an existing subnet in the VPC where the load balancer works. ● Add a new subnet **NOTE** ● The number of IP addresses required depend on the specifications, number of AZs, and IP as a backend function you have configured when you create the load balancer. The actual number of occupied IP addresses depends on that displayed on the console. ● An application load balancer requires 8 to 30 additional IP addresses in the backend subnet for traffic forwarding. The actual number of required IP addresses depends on the ELB cluster size. If load balancers are deployed in the same cluster and work in the same backend subnet, they share the same IP addresses to save resources.	Subnet of the load balancer
Private IPv4 network configuration		
IPv4 Address	Specifies how you want the IPv4 address to be assigned. ● **Automatically assign IP address**: The system automatically assigns an IPv4 address to the load balancer. ● **Manually specify IP address**: Manually specify an IPv4 address to the load balancer. **NOTE** Network ACL rules configured for the backend subnet of the load balancer will not restrict the traffic from the clients to the load balancer. If these rules are configured, the clients can directly access the load balancer. To control access to the load balancer, configure access control for all listeners added to the load balancer. For details, see **Access Control**.	Automatically assign IP address
IPv6 network configuration		

Parameter	Description	Example Value
IPv6 Address	Specifies how you want the IPv6 address to be assigned. **NOTE** Network ACL rules configured for the backend subnet of the load balancer will not restrict the traffic from the clients to the load balancer. If network ACL rules are configured, the clients can directly access the load balancer. To control access to the load balancer, configure access control for all listeners added to the load balancer. For details, see **Access Control**.	Automatically assign IP address
Shared Bandwidth	Specifies the shared bandwidth that the IPv6 address will be added to. You can choose not to select a shared bandwidth, select an existing shared bandwidth, or assign a shared bandwidth.	Skip
Public IPv4 network configuration		
EIP	This parameter is mandatory when **Network Type** is set to **IPv4 public network**. ● **New EIP**: The system will assign a new EIP to the load balancer. ● **Use existing**: Select an existing IP address.	-
EIP Type	Specifies the link type (BGP) when a new EIP is used. **Dynamic BGP**: When changes occur on a network using dynamic BGP, routing protocols provide automatic, real-time optimization of network configurations, ensuring network stability and optimal user experience.	Dynamic BGP

Parameter	Description	Example Value
Billed By	Specifies how the bandwidth will be billed. You can select **Bandwidth**, **Traffic**, or **Shared Bandwidth**. ● **Bandwidth**: You specify the maximum bandwidth and pay for the amount of time you use the bandwidth. ● **Traffic**: You specify a maximum bandwidth and pay for the outbound traffic you use. ● **Shared Bandwidth**	Shared Bandwidth
Bandwidth	Specifies the maximum bandwidth.	100 Mbit/s

6. Confirm the configuration and submit your request.

2.4 Modifying the Bandwidth

Scenario

If you set the **Network Type** of a load balancer to **Public IPv4 network** or **IPv6 network**, the load balancer can route requests over the Internet and you can modify the bandwidth used by the EIP bound to the load balancer as required.

NOTE

- When changing bandwidth, you need to change the specifications of the dedicated load balancer to avoid speed limit due to insufficient bandwidth.
- The bandwidth of the EIP bound to the load balancer is the limit for traffic required by the clients to access the load balancer.

Modifying the Bandwidth

When you modify the bandwidth, traffic routing will not be interrupted.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click **More** in the **Operation** column.
5. Click **Modify IPv4 Bandwidth** or **Modify IPv6 Bandwidth**.
6. In the **New Configuration** area, modify the bandwidth and click **Next**.

   You can select the bandwidth defined by the system or customize the bandwidth. The bandwidth ranges from 1 Mbit/s to 2,000 Mbit/s.

7. Confirm the modified bandwidth and click **Submit**.

2.5 Changing an IP Address

Scenarios

You can change the private IPv4 address bound to a load balancer into another IPv4 IP address in the current subnet or other subnets.

NOTE

You can only change the IP address bound to a dedicated load balancer.

Changing a Private IPv4 Address

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer whose IP address you want to change, and click **More** > **Change Private IPv4 Address** in the **Operation** column.
5. In the **Change Private IPv4 Address** dialog box, select the subnet where the IP address resides and specify the IP address.
   - To use an IP address from another subnet, select **Automatically assign IPv4 address**. The system automatically assigns an IPv4 address for your load balancer.
   - To use another IP address from the current subnet, specify an IP address.
6. Click **OK**.

2.6 Binding an IP Address to or Unbinding an IP Address from a Load Balancer

Scenarios

You can bind an IP address to a load balancer or unbind the IP address from a load balancer based on service requirements.

NOTE

- Load balancers without IPv4 EIPs cannot route requests over the public IPv4 network.
- Load balancers without private IPv4 addresses cannot route requests over the private IPv4 network.
- After an IPv6 address is unbound, the load balancer cannot route requests over the IPv6 network.

Binding an IPv4 EIP

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer to which you want to bind an IPv4 EIP and click **More** > **Bind IPv4 EIP** in the **Operation** column.
5. In the **Bind IPv4 EIP** dialog box, select the EIP you want to bind to the load balancer.
6. Click **OK**.

Binding a Private IPv4 Address

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer to which you want to bind a private IPv4 address and click **More** > **Bind Private IPv4 Address** in the **Operation** column.
5. In the **Bind Private IPv4 Address** dialog box, select the subnet where the IP address resides and specify the IP address.
   - By default, the system automatically assigns an IP address. To manually specify an IP address, deselect **Automatically assign IP address** and enter the IP address.
   - Ensure that the entered IP address belongs to the selected subnet and is not in use.
6. Click **OK**.

Unbinding an IPv4 EIP

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer from which you want to unbind the IPv4 EIP and click **More** > **Unbind IPv4 EIP** in the **Operation** column.
5. In the displayed dialog box, confirm the IPv4 EIP that you want to unbind and click **Yes**.

NOTE

After the IPv4 EIP is unbound, the load balancer cannot route requests over the Internet.

Unbinding a Private IPv4 Address

Only dedicated load balancers support this function.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer from which you want to unbind the private IPv4 address and click **More** > **Unbind Private IPv4 Address** in the **Operation** column.
5. In the displayed dialog box, confirm the private IPv4 address that you want to unbind and click **Yes**.

NOTE

After the private IPv4 address is unbound, the load balancer cannot route requests over the private IPv4 network.

Unbinding an IPv6 Address

Only dedicated load balancers can have IPv6 addresses bound.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer from which you want to unbind the IPv6 address and click **More** > **Unbind IPv6 Address** in the **Operation** column.
5. In the displayed dialog box, confirm the IPv6 address that you want to unbind and click **Yes**.

NOTE

After an IPv6 address is unbound, the load balancer cannot route requests over the IPv6 network.

2.7 Exporting the Load Balancer List

Scenarios

You can export the load balancer list for backup.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the upper left corner of the load balancer list, click **Export**.

2.8 Deleting a Load Balancer

Scenarios

You can delete a load balancer if you do not need it any longer.

⚠ CAUTION

A deleted load balancer cannot be recovered.

After a public network load balancer is deleted, its EIP will not be released and can be used by other resources.

Prerequisites

Delete the resources configured for the load balancer in the following sequence:

1. Delete all the forwarding policies added to HTTP and HTTPS listeners of the load balancer.
2. Delete the redirect created for each HTTP listener of the load balancer.
3. Remove all the backend servers from the backend server groups associated with each listener of the load balancer.
4. Delete all the listeners added to the load balancer.
5. Delete all backend server groups associated with each listener of the load balancer.

Deleting a Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the target load balancer and choose **More** > **Delete** in the **Operation** column.

   A confirmation dialog box is displayed. Select **Release the EIP** as required.

5. Click **Yes**.

3 Listener

3.1 Overview

You need to add at least one listener after you have created a load balancer. This listener receives requests from clients and routes requests to backend servers using the protocol, port, and load balancing algorithm you select.

Supported Protocols

ELB provides load balancing at both Layer 4 and Layer 7.

Select TCP or UDP for load balancing at Layer 4 and HTTP or HTTPS at Layer 7.

Table 3-1 Protocols supported by ELB

Protocol		Description	Application Scenario
Layer 4	TCP	• Source IP address-based sticky sessions • Fast data transfer	• Scenarios that require high reliability and data accuracy, such as file transfer, email, and remote login • Web applications that receive a large number of concurrent requests and require high performance
Layer 4	UDP	• Low reliability • Fast data transfer	Scenarios that require quick response, such as video chat, gaming, and real-time financial quotations
Layer 7	HTTP	• Cookie-based sticky sessions • X-Forward-For request header	Web applications where data content needs to be identified, such as mobile games

Protocol		Description	Application Scenario
Layer 7	HTTPS	- An extension of HTTP for encrypted data transmission to prevent unauthorized access - Encryption and decryption performed on load balancers - Multiple versions of encryption protocols and cipher suites	Web applications that require encrypted transmission

3.2 Adding a TCP Listener

Scenarios

You can add a TCP listener, if high reliability and high accuracy are required but slow speed is acceptable, for example, during file transfer, email sending and receiving, and remote login.

Constraints

- If the listener protocol is TCP, the protocol of the backend server group is TCP by default and cannot be changed.
- If you only select application load balancing (HTTP/HTTPS) for your dedicated load balancer, you cannot add a TCP listener to this load balancer.

Adding a TCP Listener to a Dedicated Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 3-2**.

Table 3-2 Parameters for configuring a listener

Parameter	Description	Example Value
Name	Specifies the listener name.	listener-pnqy

Parameter	Description	Example Value
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients.	TCP
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.	80
Access Control	Specifies how access to the listener is controlled. For details, see **Access Control**. The following options are available: - **All IP addresses** - **Blacklist** - **Whitelist**	Blacklist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **Creating an IP Address Group**.	ipGroup-b2
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.	N/A
Advanced Settings		
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **10** to **4000.**	300
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.	N/A

6. Click **Next: Configure Request Routing Policy** to configure the backend server group. For details about how to configure a backend server group, see **Table 3-3**.

Table 3-3 Parameters for configuring a backend server group

Parameter	Description	Example Value
Backend Server Group	Specifies a group of servers with the same features to receive requests from the load balancer. Two options are available: ● **Create new** ● **Use existing** **NOTE** The backend protocol of the backend server group must match the frontend protocol. For example, if the frontend protocol is TCP, the backend protocol must be TCP.	Create new
Backend Server Group Name	Specifies the name of the backend server group.	server_group
Backend Protocol	Specifies the protocol used by backend servers to receive requests. The backend protocol is TCP by default and cannot be changed.	TCP

Parameter	Description	Example Value
Load Balancing Algorithm	Specifies the algorithm used by the load balancer to distribute traffic. The following options are available: ● **Weighted round robin**: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. ● **Weighted least connections**: In addition to the number of active connections established with each backend server, each server is assigned a weight based on their processing capability. Requests are routed to the server with the lowest connections-to-weight ratio. ● **Source IP hash**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash 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. **NOTE** ● Choose an appropriate algorithm based on your requirements for better traffic distribution. ● For **Weighted round robin** or **Weighted least connections**, no requests will be routed to a server with a weight of 0.	Weighted round robin

Parameter	Description	Example Value
Sticky Session	Specifies whether to enable sticky sessions. If you enable sticky sessions, all requests from a client during one session are sent to the same backend server. This parameter is optional and can be enabled if you have selected **Weighted round robin** for **Load Balancing Algorithm**.	N/A
Sticky Session Type	Specifies the type of sticky sessions. **Source IP address** is the only choice available when TCP or UDP is used as the frontend protocol. **Source IP address**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash key, and all the endpoints are numbered. The system allocates the client to a particular endpoint based on the generated key. Requests from the same IP address are forwarded to the same backend server for processing.	Source IP address
Description	Provides supplementary information about the backend server group. You can enter a maximum of 255 characters.	N/A

7. Click **Next: Add Backend Server**. Add backend servers and configure the health check for the backend server group. For details about how to add backend servers, see **Overview**. For the parameters required for configuring a health check, see **Table 3-4**.

Table 3-4 Parameters for configuring a health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks. If the health check is enabled, click ✎ next to **Advanced Settings** to set health check parameters.	N/A

Parameter	Description	Example Value
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. If the backend protocol is TCP, the health check protocol can be TCP, HTTP, or HTTPS.	HTTP
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. ● You can use the private IP address of the backend server as the domain name. ● You can also specify a domain name that consists of at least two labels separated by periods (.). Use only letters, digits, and hyphens (-). Do not start or end strings with a hyphen. Max total: 100 characters. Max label: 63 characters.	www.elb.com
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from 1 to 65535. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80
Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. The path can contain 1 to 80 characters and must start with a slash (/). The path can contain letters, digits, hyphens (-), slashes (/), periods (.), question marks (?), percent signs (%), ampersands (&), and underscores (_).	/index.html

Parameter	Description	Example Value
Interval (s)	Specifies the interval for sending health check requests, in seconds. The interval ranges from **1** to **50**.	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The timeout duration ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **Next: Confirm**.
9. Confirm the configuration and click **Submit**.

3.3 Adding a UDP Listener

Scenarios

UDP listeners are suitable for scenarios that focus more on timeliness than reliability, such as video chat, gaming, and real-time quotation in the financial market.

Constraints

- UDP listeners do not support fragmentation.
- The port of UDP listeners cannot be 4789.
- UDP packets can have any size less than 1,500 bytes. The packets will be discarded if they are too big. You need to modify the configuration files of the applications based on the maximum transmission unit (MTU) value.
- If you only select application load balancing (HTTP/HTTPS) for your dedicated load balancer, you cannot add a UDP listener to this load balancer.

Adding a UDP Listener to a Dedicated Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 3-5**.

Table 3-5 Parameters for configuring a listener

Parameter	Description	Example Value
Name	Specifies the listener name.	listener
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients.	UDP
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.	80
Access Control	Specifies how access to the listener is controlled. For details, see **Access Control**. The following options are available: ● **All IP addresses** ● **Blacklist** ● **Whitelist**	**Blacklist**
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **Creating an IP Address Group**.	ipGroup
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.	N/A
Advanced Settings		
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **10** to **4000**.	300

Parameter	Description	Example Value
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.	N/A

6. Click **Next: Configure Request Routing Policy** to configure the backend server group. **Table 3-6** describes the parameters for configuring a backend server group.

Table 3-6 Parameters for configuring a backend server group

Parameter	Description	Example Value
Backend Server Group	Specifies a group of servers with the same features to receive requests from the load balancer. Two options are available: ● **Create new** ● **Use existing** **NOTE** The backend protocol of the backend server group must match the frontend protocol. For example, if the frontend protocol is TCP, the backend protocol must be TCP.	Create new
Backend Server Group Name	Specifies the name of the backend server group.	server_group
Backend Protocol	Specifies the protocol that will be used by backend servers to receive requests. The backend protocol can be UDP or QUIC.	UDP

Parameter	Description	Example Value
Load Balancing Algorithm	Specifies the algorithm that will be used by the load balancer to distribute traffic. The following options are available: ● **Weighted round robin**: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. ● **Weighted least connections**: In addition to the number of active connections established with each backend server, each server is assigned a weight based on their processing capability. Requests are routed to the server with the lowest connections-to-weight ratio. ● **Source IP hash**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash 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. **NOTE** ● Choose an appropriate algorithm based on your requirements for better traffic distribution. ● For **Weighted round robin** or **Weighted least connections**, no requests will be routed to a server with a weight of 0.	Weighted round robin

Parameter	Description	Example Value
Sticky Session	Specifies whether to enable sticky sessions. If you enable sticky sessions, all requests from a client during one session are sent to the same backend server. This parameter is optional and can be enabled if you have selected **Weighted round robin** for **Load Balancing Algorithm**.	N/A
Sticky Session Type	Specifies the type of sticky sessions. **Source IP address** is the only choice available when TCP or UDP is used as the frontend protocol. **Source IP address**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash key, and all the endpoints are numbered. The system allocates the client to a particular endpoint based on the generated key. Requests from the same IP address are forwarded to the same backend server for processing.	Source IP address
Description	Provides supplementary information about the backend server group. You can enter a maximum of 255 characters.	N/A

7. Click **Next: Add Backend Server**. Add backend servers and configure the health check for the backend server group. For details about how to add backend servers, see **Overview**. For the parameters required for configuring a health check, see **Table 3-7**.

Table 3-7 Parameters for configuring a health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks. If the health check is enabled, click ✎ next to **Advanced Settings** to set health check parameters.	N/A

Parameter	Description	Example Value
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. If the backend protocol is UDP, the health check protocol is UDP and cannot be changed.	UDP
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from 1 to 65535. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80
Interval (s)	Specifies the interval for sending health check requests, in seconds. The interval ranges from **1** to **50**.	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The timeout duration ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **Next: Confirm**.
9. Confirm the configuration and click **Submit**.

3.4 Adding an HTTP Listener

Scenarios

HTTP listeners are suitable for applications that require identifying the data content, such as web applications and small mobile games.

Constraints

- If the listener protocol is HTTP, the protocol of the backend server group is HTTP by default and cannot be changed.
- If you only select network load balancing (TCP/UDP) for your dedicated load balancer, you cannot add an HTTP listener to this load balancer.

Adding an HTTP Listener to a Dedicated Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 3-8**.

Table 3-8 Parameters for configuring a listener

Parameter	Description	Example Value
Name	Specifies the listener name.	listener
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients.	HTTP
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.	80
Redirect	Specifies whether to enable redirection. If you have both HTTPS and HTTP listeners, you can use this function to redirect the requests from the HTTP listener to the HTTPS listener to ensure security.	N/A
Redirected To	Specifies the HTTPS listener to which requests are redirected if **Redirect** is enabled.	listener_HTTPS_443
Access Control	Specifies how access to the listener is controlled. For details, see **Access Control**. The following options are available: ● **All IP addresses** ● **Blacklist** ● **Whitelist**	**Blacklist**

Parameter	Description	Example Value
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **Creating an IP Address Group**.	ipGroup
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.	Enabled
Advanced Settings		
Transfer Load Balancer EIP	Specifies whether to store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header field and pass this field to backend servers.	N/A
Transfer Listener Port Number	Specifies whether to store the port number used by the listener in the X-Forwarded-Port header field and pass the field to backend servers.	N/A
Transfer Port Number in the Request	Specifies whether to store the port number used by the client in the X-Forwarded-For-Port header field and pass the field to backend servers.	N/A
Rewrite 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.	N/A

Parameter	Description	Example Value
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **0** to **4000**.	60
Request Timeout	Specifies the length of time (in seconds) after which the load balancer closes the connection if the load balancer does not receive a request from the client. The request timeout duration ranges from **1** to **300**.	60
Response Timeout	Specifies the length of time (in seconds) after which the load balancer sends a 504 Gateway Timeout error to the client if the load balancer receives no response from the backend server after routing a request to the backend server and receives no response after attempting to route the same request to other backend servers. The response timeout duration ranges from **1** to **300**. **NOTE** If you have enabled sticky sessions and the backend server does not respond within the response timeout duration, the load balancer returns 504 Gateway Timeout to the clients.	60
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.	N/A

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group and configure parameters as described in **Table 3-9**.

Table 3-9 Parameters for configuring a backend server group

Parameter	Description	Example Value
Backend Server Group	Specifies a group of servers with the same features to receive requests from the load balancer. Two options are available: ● **Create new** ● **Use existing** **NOTE** The backend protocol of the backend server group must match the frontend protocol. For example, if the frontend protocol is TCP, the backend protocol must be TCP.	Create new
Backend Server Group Name	Specifies the name of the backend server group.	server_group
Backend Protocol	Specifies the protocol that will be used by backend servers to receive requests. The backend protocol is HTTP by default and cannot be changed.	HTTP

Parameter	Description	Example Value
Load Balancing Algorithm	Specifies the algorithm that will be used by the load balancer to distribute traffic. The following options are available: ● **Weighted round robin**: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. ● **Weighted least connections**: In addition to the number of active connections established with each backend server, each server is assigned a weight based on their processing capability. Requests are routed to the server with the lowest connections-to-weight ratio. ● **Source IP hash**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash 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. **NOTE** ● Choose an appropriate algorithm based on your requirements for better traffic distribution. ● For **Weighted round robin** or **Weighted least connections**, no requests will be routed to a server with a weight of 0.	Weighted round robin

Parameter	Description	Example Value
Sticky Session	Specifies whether to enable sticky sessions. If you enable sticky sessions, all requests from a client during one session are sent to the same backend server. This parameter is optional and can be enabled if you have selected **Weighted round robin** for **Load Balancing Algorithm**.	N/A
Sticky Session Type	Specifies the type of sticky sessions for HTTP and HTTPS listeners. ● **Load balancer cookie**: The load balancer generates a cookie after receiving a request from the client. All subsequent requests with the same cookie are then routed to the same backend server. **NOTE**	Load balancer cookie
Slow Start	Specifies whether to enable slow start, which is disabled by default. After you enable slow start, the load balancer linearly increases the proportion of requests to send to backend servers in this mode. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode. For details, see **Slow Start (Dedicated Load Balancers)**.	N/A
Slow Start Duration	Specifies the slow start duration if **Slow Start** is enabled. The duration ranges from **30** to **1200**, in seconds, and the default value is **30**.	30
Description	Provides supplementary information about the backend server group. You can enter a maximum of 255 characters.	N/A

7. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group. For details about how to add backend servers, see **Overview**. For the parameters required for configuring a health check, see **Table 3-10**.

Table 3-10 Parameters for configuring a health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks. If the health check is enabled, click ✎ next to **Advanced Settings** to set health check parameters.	N/A
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. If the backend protocol is HTTP or HTTPS, the health check protocol can be TCP, HTTP, or HTTPS.	HTTP
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. ● You can use the private IP address of the backend server as the domain name. ● You can also specify a domain name that consists of at least two labels separated by periods (.). Use only letters, digits, and hyphens (-). Do not start or end strings with a hyphen. Max total: 100 characters. Max label: 63 characters.	www.elb.com
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from **1** to **65535**. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80

Parameter	Description	Example Value
Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. The path can contain 1 to 80 characters and must start with a slash (/).	/index.html
Interval (s)	Specifies the interval for sending health check requests, in seconds. The interval ranges from **1** to **50**.	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The timeout duration ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **Next: Confirm**.
9. Confirm the configuration and click **Submit**.

3.5 Adding an HTTPS Listener

Scenarios

HTTPS listeners are best suited for applications that require encrypted transmission. Load balancers decrypt HTTPS requests before routing them to backend servers, which then send the processed requests back to load balancers for encryption before they are sent to clients.

Constraints

- Dedicated load balancers: If the listener protocol is HTTPS, the protocol of the backend server group can be HTTP or HTTPS.
- If you only select network load balancing (TCP/UDP) for your dedicated load balancer, you cannot add an HTTPS listener to this load balancer.

Adding an HTTPS Listener to a Dedicated Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.

3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 3-11**.

Table 3-11 Parameters for configuring a listener

Parameter	Description	Example Value
Name	Specifies the listener name.	listener-pnqy
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients.	HTTPS
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.	80
SSL Authentication	Specifies whether how you want the clients and backend servers to be authenticated. There are two options: **One-way authentication** or **Mutual authentication**. ● If only server authentication is required, select **One-way authentication**. ● If you want the clients and the load balancer to authenticate each other, select **Mutual authentication**. Only authenticated clients will be allowed to access the load balancer.	One-way authentication
Server Certificate	Specifies the certificate that will be used by the backend server to authenticate the client when HTTPS is used as the frontend protocol. Both the certificate and private key are required.	N/A

Parameter	Description	Example Value
CA Certificate	Specifies the certificate that will be used by the backend server to authenticate the client when **SSL Authentication** is set to **Mutual authentication**. A CA certificate is issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.	N/A
Enable SNI	Specifies whether to enable SNI when HTTPS is used as the frontend protocol. SNI is an extension to TLS and is used when a server uses multiple domain names and certificates. This allows the client to submit the domain name information while sending an SSL handshake request. After the load balancer receives the request, the load balancer queries the corresponding certificate based on the domain name and returns it to the client. If no certificate is found, the load balancer will return the default certificate. For details, see **SNI Certificate**.	N/A
SNI Certificate	Specifies the certificate associated with the domain name when the frontend protocol is HTTPS and SNI is enabled. Select an existing certificate or create one.	N/A
Access Control	Specifies how access to the listener is controlled. For details, see **Access Control**. The following options are available: ● **All IP addresses** ● **Blacklist** ● **Whitelist**	Whitelist

Parameter	Description	Example Value
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **Creating an IP Address Group**.	ipGroup-b2
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.	Enabled
Advanced Settings		
HTTP/2	Specifies whether you want to use HTTP/2 if you select **HTTPS** for **Frontend Protocol**. For details, see **HTTP/2**.	N/A
Transfer Load Balancer EIP	Specifies whether to store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header field and pass this field to backend servers.	N/A
Transfer Listener Port Number	Specifies whether to store the port number used by the listener in the X-Forwarded-Port header field and pass the field to backend servers.	N/A
Transfer Port Number in the Request	Specifies whether to store the port number used by the client in the X-Forwarded-For-Port header field and pass the field to backend servers.	N/A
Rewrite 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.	N/A

Parameter	Description	Example Value
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **0** to **4000**.	60
Request Timeout	Specifies the length of time (in seconds) after which the load balancer closes the connection if the load balancer does not receive a request from the client. The request timeout duration ranges from **1** to **300**.	60
Response Timeout	Specifies the length of time (in seconds) after which the load balancer sends a 504 Gateway Timeout error to the client if the load balancer receives no response from the backend server after routing a request to the backend server and receives no response after attempting to route the same request to other backend servers. The request timeout duration ranges from **1** to **300**. **NOTE** If you have enabled sticky sessions and the backend server does not respond within the response timeout duration, the load balancer returns 504 Gateway Timeout to the clients.	60
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.	N/A

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group and configure parameters as described in **Table 3-12**.

Table 3-12 Parameters for configuring a backend server group

Parameter	Description	Example Value
Backend Server Group	Specifies a group of servers with the same features to receive requests from the load balancer. Two options are available: ● **Create new** ● **Use existing** **NOTE** The backend protocol of the backend server group must match the frontend protocol. For example, if the frontend protocol is TCP, the backend protocol must be TCP.	Create new
Backend Server Group Name	Specifies the name of the backend server group.	server_group-sq4v
Backend Protocol	Specifies the protocol that will be used by backend servers to receive requests. If the frontend protocol is HTTPS, the backend protocol can be HTTP, or HTTPS.	HTTP

Parameter	Description	Example Value
Load Balancing Algorithm	Specifies the algorithm that will be used by the load balancer to distribute traffic. The following options are available: ● **Weighted round robin**: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. ● **Weighted least connections**: In addition to the number of active connections established with each backend server, each server is assigned a weight based on their processing capability. Requests are routed to the server with the lowest connections-to-weight ratio. ● **Source IP hash**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hash 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. **NOTE** ● Choose an appropriate algorithm based on your requirements for better traffic distribution. ● For **Weighted round robin** or **Weighted least connections**, no requests will be routed to a server with a weight of 0.	Weighted round robin

Parameter	Description	Example Value
Sticky Session	Specifies whether to enable sticky sessions. If you enable sticky sessions, all requests from a client during one session are sent to the same backend server. This parameter is optional and can be enabled if you have selected **Weighted round robin** for **Load Balancing Algorithm**.	N/A
Sticky Session Type	Specifies the type of sticky sessions for HTTP and HTTPS listeners. ● **Load balancer cookie**: The load balancer generates a cookie after receiving a request from the client. All subsequent requests with the same cookie are then routed to the same backend server. **NOTE**	Load balancer cookie
Slow Start	Specifies whether to enable slow start, which is disabled by default. After you enable slow start, the load balancer linearly increases the proportion of requests to send to backend servers in this mode. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode. For details, see **Slow Start (Dedicated Load Balancers)**.	N/A
Slow Start Duration	Specifies how long the slow start will last. The duration ranges from **30** to **1200**, in seconds, and the default value is **30**.	30
Description	Provides supplementary information about the backend group. You can enter a maximum of 255 characters.	N/A

7. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group. For details about how to add backend servers, see **Overview**. For the parameters required for configuring a health check, see **Table 3-13**.

Table 3-13 Parameters for configuring a health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks. If the health check is enabled, click ✎ next to **Advanced Settings** to set health check parameters.	N/A
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. If the backend protocol is HTTP or HTTPS, the health check protocol can be TCP, HTTP, or HTTPS.	HTTP
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. ● You can use the private IP address of the backend server as the domain name. ● You can also specify a domain name that consists of at least two labels separated by periods (.). Use only letters, digits, and hyphens (-). Do not start or end strings with a hyphen. Max total: 100 characters. Max label: 63 characters.	www.elb.com
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from **1** to **65535**. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80

Parameter	Description	Example Value
Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. The path can contain 1 to 80 characters and must start with a slash (/).	/index.html
Interval (s)	Specifies the interval for sending health check requests, in seconds. The interval ranges from **1** to **50**.	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The timeout duration ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **Next: Confirm**.
9. Confirm the configuration and click **Submit**.

3.6 Configuring Timeout Durations

Scenarios

You can configure 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.

For dedicated load balancers, you can change the timeout durations of TCP, UDP, HTTP, and HTTPS listeners.

Figure 3-1 Timeout durations at Layer 7

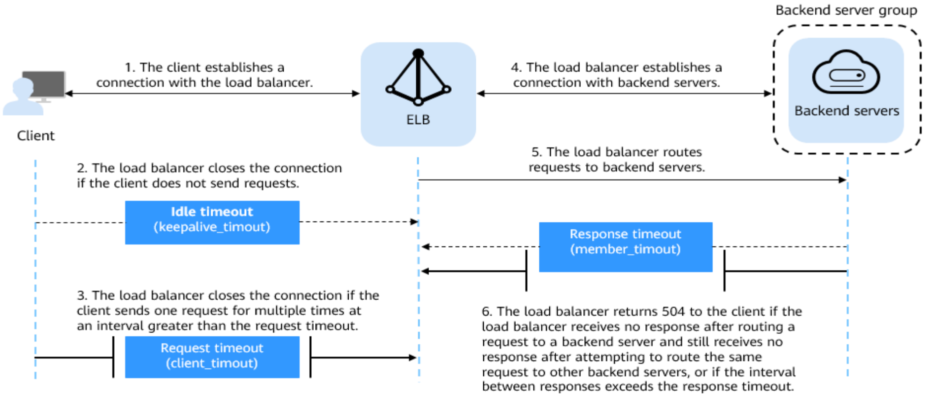

Backend server group
1. The client establishes a connection with the load balancer.
ELB
4. The load balancer establishes a connection with backend servers.
Backend servers
Client
5. The load balancer routes requests to backend servers.
2. The load balancer closes the connection if the client does not send requests.
Idle timeout (keepalive_timout)
Response timeout (member_timout)
3. The load balancer closes the connection if the client sends one request for multiple times at an interval greater than the request timeout.
Request timeout (client_timout)
6. The load balancer returns 504 to the client if the load balancer receives no response after routing a request to a backend server and still receives no response after attempting to route the same request to other backend servers, or if the interval between responses exceeds the response timeout.

Figure 3-2 Timeout durations at Layer 4

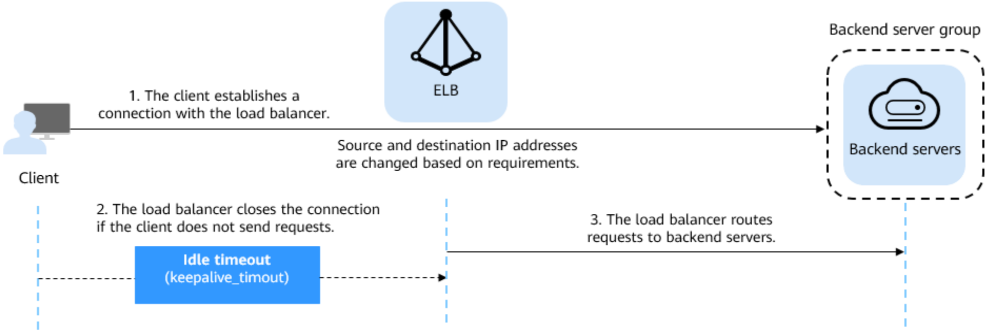

ELB
Backend server group
1. The client establishes a connection with the load balancer.
Backend servers
Client
Source and destination IP addresses are changed based on requirements.
2. The load balancer closes the connection if the client does not send requests.
3. The load balancer routes requests to backend servers.
Idle timeout (keepalive_timout)

Table 3-14 Timeout durations

Protocol	Type	Description	Value Range	Default Timeout Duration
TCP	Idle Timeout	Duration for a connection to be kept alive. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives.	10–4000s	300s
UDP	Idle Timeout		10–4000s	Dedicated load balancers: 300s
HTTP/ HTTPS	Idle Timeout		10–4000s	60s
	Request Timeout	Duration after which the load balancer closes the connection with the client if the load balancer does not receive a request from the client.	10–300s	60s

Protocol	Type	Description	Value Range	Default Timeout Duration
	Response Timeout	Duration after which the load balancer sends a 504 Gateway Timeout error to the client if the load balancer receives no response after routing a request to a backend server and receives no response after attempting to route the same request to other backend servers. **NOTE** If sticky sessions are enabled and the backend server does not respond within the response timeout duration, the load balancer returns the 504 error code without attempting to route the same request to other backend servers.	1–300s	60s

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click the name of the listener.
6. On the **Summary** tab page, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings**.
8. Configure **Idle Timeout (s)**, **Request Timeout (s)**, or **Response Timeout (s)** as you need.
9. Click **OK**.

3.7 Modifying or Deleting a Listener

Scenarios

You can modify a listener as needed or delete a listener if you no longer need it.

Deleted listeners cannot be recovered.

NOTE

Frontend Protocol/Port and **Backend Protocol** cannot be modified after you have configured them. If you want to modify the protocol or port of the listener, add another listener to the load balancer.

Modifying a listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Modify the listener in either of the following ways:
   - On the **Listeners** page, locate the listener, and click **Edit** in the **Operation** column.
   - Click the name of the target listener. On the **Summary** tab page, click **Edit** on the top right corner.
6. On the **Edit** dialog box, modify parameters, and click **OK**.

Deleting a listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click **Delete** in the **Operation** column.

NOTE

- If the listener has backend servers associated, disassociate the backend servers before deleting the listener.
- If HTTP requests are redirected to an HTTPS listener, delete the redirect before deleting the HTTPS listener.
- If the listener has a forwarding policy, delete the forwarding policy before deleting the listener.
- After a listener is deleted, the associated backend server group is also deleted.

6. In the displayed dialog box, enter **DELETE**.
7. Click **OK**.

3.8 Transfer Client IP Address

Transfer Client IP Address

If you enable **Transfer Client IP Address**, your load balancer will use the IP address of the client to access the backend server.

Table 3-15 lists whether you can enable or disable the transfer client IP address function.

Table 3-15 Transfer client IP address

Listener Type	Enabling Transfer Client IP Address	Disabling Transfer Client IP Address
TCP and UDP	Enabled by default	×
HTTP and HTTPS	Enabled by default	×

Constraints

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

  If the client and the backend server are using the same server and the **Transfer Client IP Address** option is enabled, the backend server will think the packet is sent by itself but not from the client and will not return a response packet to the load balancer. As a result, the return traffic will be interrupted.
- After this function is enabled, unidirectional download or push traffic may be interrupted when backend servers are being migrated. After the backend servers are migrated, retransmit the packets to restore the traffic.
- If you add IP addresses as backend servers, the source IP addresses of the clients cannot be passed to these servers.

Alternatives for Obtaining the IP Address of the Client

You can obtain the IP address of a client in one of the ways listed in **Table 3-16**.

Table 3-16 Alternatives

Listener Type	Alternatives
TCP and UDP	N/A
HTTP and HTTPS	**Layer 7 Load Balancing**

4 Advanced Features of HTTP/HTTPS Listeners

4.1 Forwarding Policy (Dedicated Load Balancers)

Overview

You can add forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups based on domain names or URLs.

A forwarding policy consists of one or more forwarding rules and an action. For details, see **Table 4-1**.

Table 4-1 Rules and actions supported by a forwarding policy

Policy Type	Forwarding Rules	Actions
Forwarding policy	Domain name and URL	**Forward to another backend server group** and **Redirect to another listener** (only for HTTP listeners)
Advanced forwarding policy	Domain name, URL, HTTP request method, HTTP header, query string, and CIDR block	The following actions are supported: forward to a backend server group, redirect to another listener, redirect to another URL, and return a specific response body.

How Requests Are Matched

- After you add a forwarding policy, the load balancer forwards requests based on the specified domain name or URL:
  - If the domain name or URL in a request matches that specified in the forwarding policy, the request is forwarded to the backend server group you create or select when you add the forwarding policy.

  - If the domain name or URL in a request does not match that specified in the forwarding policy, the request is forwarded to the default backend server group of the listener.
- Matching priority:
  - Forwarding policy priorities are independent of each other regardless of domain names. If a forwarding rule uses both domain names and URLs, requests are matched based on domain names first.
  - If the forwarding rule is a URL, the priorities follow the order of exact match, prefix match, and regular expression match. If the matching types are the same, the longer the URL length, the higher the priority.

Table 4-2 Example forwarding policies

Request	Forwarding policy	Forwarding Rule	Specified Value
www.elb.com/test	1	URL	/test
	2	Domain name	www.elb.com

NOTE

In this example, request **www.elb.com/test** matches both forwarding policies 1 and 2, but is routed based on forwarding policy 2.

Notes and Constraints

- You can add forwarding policies to 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.
- When you add a forwarding policy, note the following:
  - Each URL path must exist on the backend server. Otherwise, the backend server returns 404 when you access the backend server.
  - In the regular expression match, the rules are matched sequentially, and matching ends when any rule is successfully matched. Matching rules cannot overlap with each other.
  - A URL path cannot be configured for two forwarding policies.
  - A domain name cannot exceed 100 characters.

Adding a Forwarding Policy

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.

4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Listeners** tab page, add a forwarding policy in either of the following ways:
   - Click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click **Forwarding Policies**.
6. Click **Add Forwarding Policy**. Configure the parameters based on **Table 4-3**.

Table 4-3 Forwarding policy parameters

Parameter	Type	Description	Example Value
Forwarding Rule	Domain name	Specifies the domain name used for forwarding requests. The domain name in the request must exactly match that in the forwarding policy. You need to specify either a domain name or URL.	www.test.com
	URL	Specifies the URL used for forwarding requests. There are three URL matching rules: ● Exact match: The request URL must exactly match that specified in the forwarding policy. ● Prefix match: The requested URL starts with the specified URL string. ● Regular expression match: The URLs are matched using a regular expression.	/login.php
Action	Forward to a backend server group	If the request matches the configured forwarding rule, the request is forwarded to the specified backend server group.	-
	Redirect to another listener	If the request matches the configured forwarding rule, the request is redirected to the specified HTTPS listener. This action can be configured only for HTTP listeners. **NOTE** If you select **Redirect to another listener**, the HTTP listener will redirect requests to the specified HTTPS listener, but access control configured for the HTTP listener still takes effect.	-

7. Click **Save**.

4.2 Mutual Authentication

Scenarios

In common HTTPS service scenarios, only the server certificate is required for authentication. For some mission-critical services, such as financial transactions, you need to deploy both the server certificate and the client certificate for mutual authentication.

This section uses self-signed certificates as an example to describe how to configure mutual authentication. Self-signed certificates do not provide all the security properties provided by certificates signed by a CA. It is recommended that you purchase certificates from other CAs.

Creating a CA Certificate Using OpenSSL

1. Log in to a Linux server with OpenSSL installed.
2. Create the **server** directory and switch to the directory:

   mkdir ca

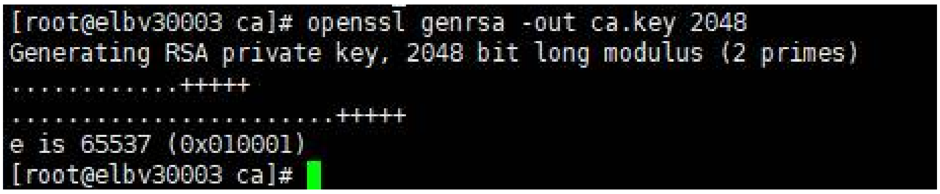
   cd ca
3. Create the certificate configuration file **ca_cert.conf**. The file content is as follows:

```
[ req ]
distinguished_name      = req_distinguished_name
prompt                  = no

[ req_distinguished_name ]
 O                      = ELB
```

4. Create the CA certificate private key **ca.key**.

   openssl genrsa -out ca.key 2048

Figure 4-1 Private key of the CA certificate



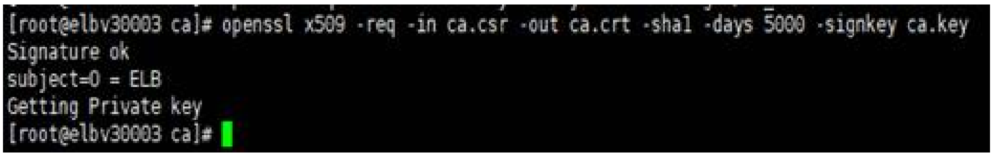
```
[root@elbv30003 ca]# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
............+++++
........................+++++
e is 65537 (0x010001)
[root@elbv30003 ca]#
```

5. Create the certificate signing request (CSR) file **ca.csr** for the CA certificate.

   openssl req -out ca.csr -key ca.key -new -config ./ca_cert.conf
6. Create the self-signed CA certificate **ca.crt**.

   openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key

Figure 4-2 Creating a self-signed CA certificate



```
[root@elbv30003 ca]# openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key
Signature ok
subject=O = ELB
Getting Private key
[root@elbv30003 ca]#
```

Issuing a Server Certificate Using the CA Certificate

The server certificate can be a CA signed certificate or a self-signed one. In the following steps, a self-signed certificate is used as an example to describe how to create a server certificate.

1. Log in to the server where the CA certificate is generated.
2. Create a directory at the same level as the directory of the CA certificate and switch to the directory.

   mkdir server

   cd server
3. Create the certificate configuration file **server_cert.conf**. The file content is as follows:
   ```
   [ req ]
   distinguished_name     = req_distinguished_name
   prompt                 = no

   [ req_distinguished_name ]
    O                     = ELB
    CN                    = www.test.com
   ```
   NOTE

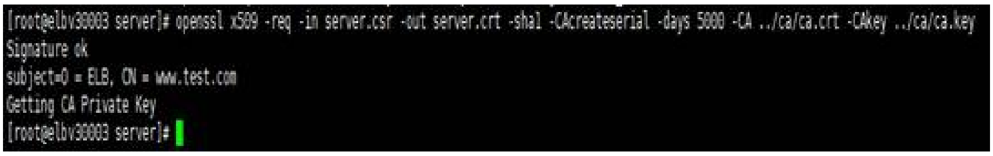
   Set the **CN** field to the domain name or IP address of the Linux server.
4. Create the server certificate private key **server.key**.

   openssl genrsa -out server.key 2048
5. Create the CSR file **server.csr** for the server certificate.

   openssl req -out server.csr -key server.key -new -config ./server_cert.conf
6. Use the CA certificate to issue the server certificate **server.crt**.

   openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key

   Figure 4-3 Issuing a server certificate

   ```
   [root@elbv30003 server]# openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key
   Signature ok
   subject=O = ELB, CN = www.test.com
   Getting CA Private Key
   [root@elbv30003 server]#
   ```

Issuing a Client Certificate Using the CA Certificate

1. Log in to the server where the CA certificate is generated.
2. Create a directory at the same level as the directory of the CA certificate and switch to the directory.

   mkdir client

   cd client
3. Create the certificate configuration file **client_cert.conf**. The file content is as follows:
   ```
   [ req ]
   distinguished_name     = req_distinguished_name
   prompt                 = no

   [ req_distinguished_name ]
   ```

```
O                = ELB
CN                = www.test.com
```

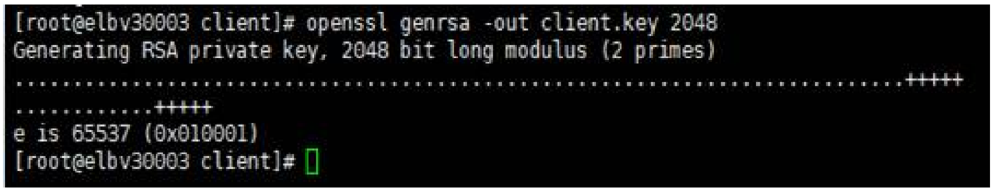
NOTE

Set the **CN** field to the domain name or IP address of the Linux server.

4. Create the client certificate private key **client.key**.

   openssl genrsa -out client.key 2048

   Figure 4-4 Creating a client certificate private key



```
[root@elbv30003 client]# openssl genrsa -out client.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..........................................................................+++++
...........+++++
e is 65537 (0x010001)
[root@elbv30003 client]#
```

5. Create the CSR file **client.csr** for the client certificate.

   openssl req -out client.csr -key client.key -new -config ./client_cert.conf

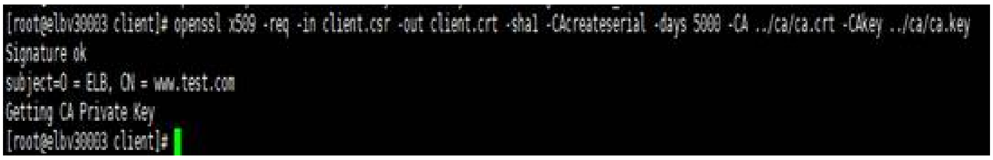
   Figure 4-5 Creating a client certificate CSR file

```
[root@elbv30003 client]# openssl req -out client.csr -key client.key -new -config ./client_cert.conf
```

6. Use the CA certificate to issue the client certificate **client.crt**.

   openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key

   Figure 4-6 Issuing a client certificate



```
[root@elbv30003 client]# openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key
Signature ok
subject=O = ELB, CN = www.test.com
Getting CA Private Key
[root@elbv30003 client]#
```

7. Convert the client certificate to a **.p12** file that can be identified by the browser.

   openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out client.p12

   NOTE

   A password is required during command execution. Save this password, which will be required when you import the certificate using the browser.

Configuring the Server Certificate and Private Key

1. Log in to the load balancer management console.
2. In the navigation pane on the left, choose **Certificates**.
3. In the navigation pane on the left, choose **Certificates**. On the displayed page, click **Add Certificate**. In the **Add Certificate** dialog box, select **Server certificate**, copy the content of server certificate **server.crt** to the **Certificate Content** area and the content of private key file **server.key** to the **Private Key** area, and click **OK**.

NOTE

Delete the last newline character before you copy the content.

NOTE

The certificate and private key must be PEM-encoded.

Configuring the CA Certificate

Step 1 Log in to the load balancer management console.

Step 2 In the navigation pane on the left, choose **Certificates**.

Step 3 Click **Add Certificate**. In the **Add Certificate** dialog box, select **CA certificate**, copy the content of CA certificate **ca.crt** created in **Creating a CA Certificate Using OpenSSL** to the **Certificate Content** area, and click **OK**.

NOTE

Delete the last newline character before you copy the content.

NOTE

The certificate must be PEM-encoded.

----End

Configuring Mutual Authentication

1. Log in to the load balancer management console.
2. Locate the load balancer and click its name. Under **Listeners**, click **Add Listener**. Select **HTTPS** for **Frontend Protocol** and **Mutual authentication** for **SSL Authentication**, and select a CA certificate and server certificate.

Add backend servers.

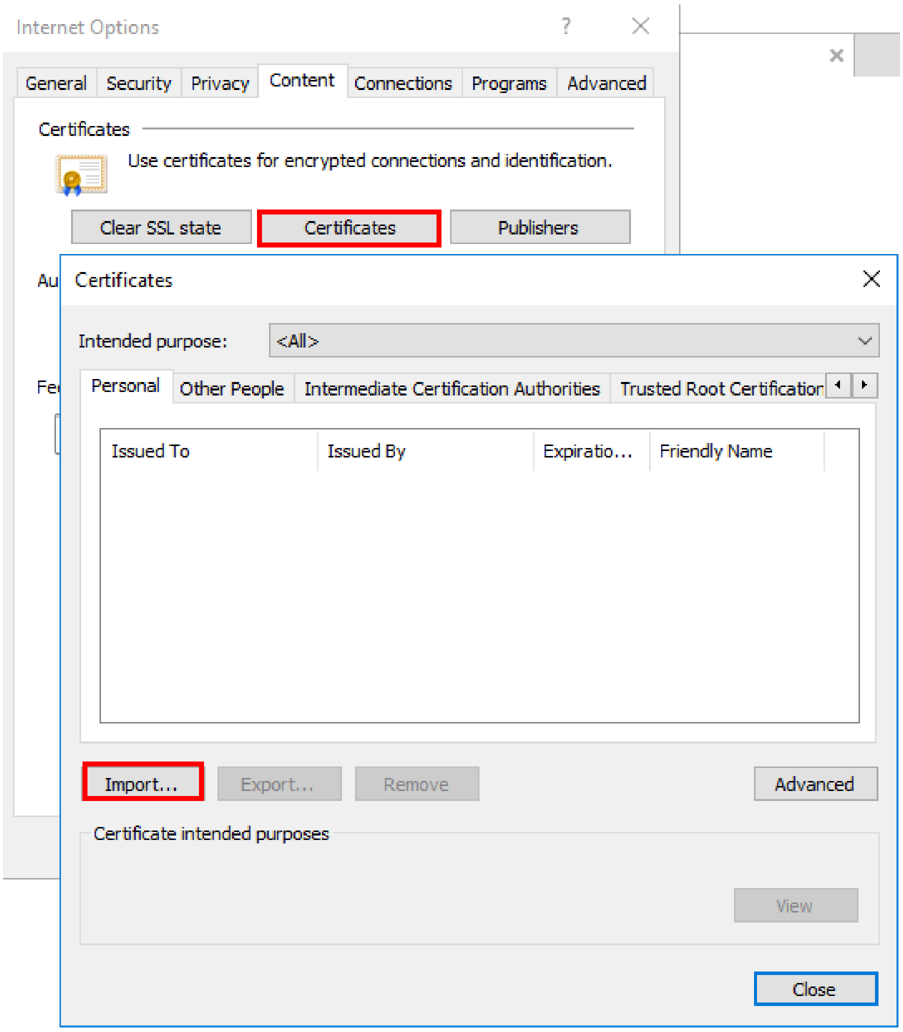
For detailed operations, see **Overview**.

Importing and Testing the Client Certificate

Method 1: Using a browser

1. Import the client certificate using a browser (Internet Explorer 11 is used as an example).
   a. Export **client.p12** from the Linux server.
   b. Open the browser, choose **Settings** > **Internet Options** and click **Content**.
   c. Click **Certificates** and then **Import** to import the **client.p12** certificate.

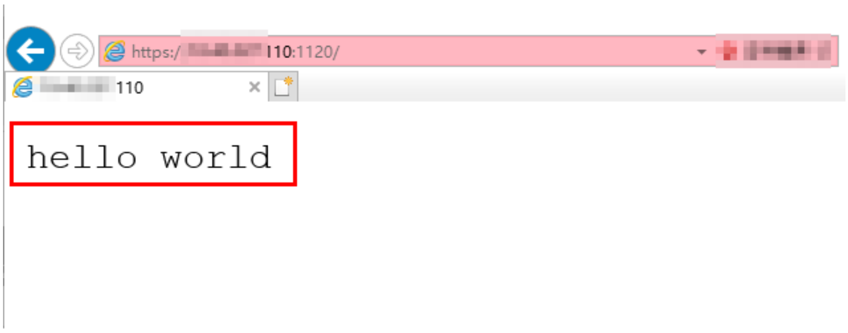
Figure 4-7 Importing the **client.p12** certificate



2. Verify the import.

   Enter the access address in the address box of your browser. A window is displayed asking you to select the certificate. Select the client certificate and click **OK**. If the website can be accessed, the certificate is successfully imported.

   Figure 4-8 Accessing the website

   ```
   hello world
   ```

Method 2: Using cURL

1. Import the client certificate.

   Copy client certificate **client.crt** and private key **client.key** to a new directory, for example, **/home/client_cert**.

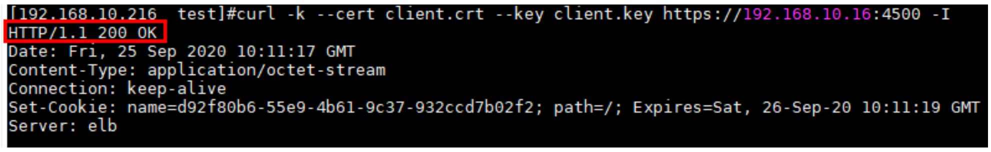
2. Verify the import.

   On the Shell screen, run the following command:

   ```
   curl -k --cert /home/client_cert/client.crt --key /home/client_cert/client.key https://
   XXX.XXX.XXX.XXX:XXX/ -I
   ```

   Ensure that the certificate address, private key address, IP address and listening port of the load balancer are correct. Replace **https://XXX.XXX.XXX.XXX:XXX** with the actual IP address and port number. If the expected response code is returned, the certificate is successfully imported.

   Figure 4-9 Example of a correct response code

   ```
   [192.168.10.216  test]#curl -k --cert client.crt --key client.key https://192.168.10.16:4500 -I
   HTTP/1.1 200 OK
   Date: Fri, 25 Sep 2020 10:11:17 GMT
   Content-Type: application/octet-stream
   Connection: keep-alive
   Set-Cookie: name=d92f80b6-55e9-4b61-9c37-932ccd7b02f2; path=/; Expires=Sat, 26-Sep-20 10:11:19 GMT
   Server: elb
   ```

4.3 HTTP/2

Scenarios

Hypertext Transfer Protocol 2.0 (HTTP/2) is the next-generation HTTP protocol. HTTP/2 is used to secure connections between the load balancer and clients. You can enable HTTP/2 when you add HTTPS listeners. If you have already added an HTTPS listener, you can also enable this function.

Constraints

You can enable HTTP/2 only for HTTPS listeners.

Enabling HTTP/2 When Adding a Listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.
7. Expand **Advanced Settings** and enable HTTP/2.
8. Confirm the configurations and click **Submit**.

Enabling or Disabling HTTP/2 When Modifying a Listener

1. Log in to the management console.

2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Summary** tab page, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings** and enable or disable HTTP/2.
8. Click **OK**.

4.4 HTTP Redirection to HTTPS

Scenarios

HTTPS is an extension of HTTP. HTTPS encrypts data between a web server and a browser.

If you enable redirection, all HTTP requests to your website are transmitted over HTTPS connections to improve security.

⚠ CAUTION

- If the listener protocol is HTTP, only the GET or HEAD method can be used for redirection. If you create a redirect for an HTTP listener, the client browser will change POST or other methods to GET. If you want to use other methods rather than GET and HEAD, add an HTTPS listener.
- HTTP requests are forwarded to the HTTPS listener as HTTPS requests, which are then routed to backend servers over HTTP.
- If an HTTP listener is redirected to an HTTPS listener, no certificate can be deployed on the backend servers associated with the HTTPS listener. If certificates are deployed, HTTPS requests will not take effect.

Prerequisites

- You have added an HTTPS listener.
- You have added an HTTP listener.

Creating Redirection to HTTPS

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.

4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the HTTP listener, and click its name.
6. On the **Forwarding Policies** tab page, click **Add Forwarding Policy**.

Table 4-4 Configuring parameters for redirection

Parameter	Setting
Action	Select **Redirect to another listener**.
Listener	Select the HTTPS listener to which requests are redirected.

7. After the forwarding policy is added, click **Save**.

NOTE

- If requests to an HTTP listener are redirected, the listener will become invalid, but access control to the listener will still take effect.
- If you create a redirect for an HTTP listener, the backend server will return HTTP 301 Move Permanently to the clients.

Modifying Redirection to HTTPS

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the HTTP listener, and click its name.
6. On the **Forwarding Policies** tab page, locate the target forwarding policy and click **Edit**.
7. You can change the HTTPS listener to which requests are redirected as required.
8. Click **Save**.

Deleting Redirection to HTTPS

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.

6. On the **Forwarding Policies** tab page, click **Delete** on the right of the target forwarding policy.
7. In the displayed dialog box, click **Yes**.

4.5 Transferring the Load Balancer EIP to Backend Servers

Scenarios

ELB allows the EIPs of the load balancers to be passed to backend servers. You can enable the function when you add HTTPS or HTTP listeners.

Load balancer EIPs are placed in the X-Forwarded-ELB-IP field in the HTTPS or HTTP header in the format of *XX.XXX.XX.XXX*, as shown below:

```
X-Forwarded-ELB-IP: XX.XXX.XX.XXX
```

Enabling the Function

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, expand **Advanced Settings** and enable the function.
7. Confirm the configurations and click **Submit**.

   NOTE

   This function can be enabled only for HTTPS or HTTP listeners.

Disabling the Function

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Summary** tab page, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings** and disable the function.
8. Click **OK**.

4.6 SNI Certificate

Scenarios

If you have an application that can be accessed through multiple domain names and each domain name uses a different certificate, you can enable Server Name Indication (SNI) when you add an HTTPS listener.

SNI, an extension to Transport Layer Security (TLS), enables a server to present multiple certificates on the same IP address and port number. SNI allows the client to indicate the domain name of the website while sending an SSL handshake request. Once receiving the request, the load balancer queries the right certificate based on the hostname or domain name and returns the certificate to the client. If no certificate is found, the load balancer will return the default certificate.

You can enable SNI only when you add HTTPS listeners. Load balancers can have multiple SNI certificates bound.

Constraints

An HTTPS listener can have up to 30 SNI certificates.

Prerequisites

- You have created an SNI certificate by performing the operations in **Adding, Modifying, or Deleting a Certificate**.
- You have added an HTTPS listener to the load balancer by performing the operations in **Adding an HTTPS Listener**.

NOTE

- You need to specify a domain name for an SNI certificate. The domain name must be the same as that in the certificate.
- A domain name can be used by both an ECC certificate and an RSA certificate. If there are two SNI certificates that use the same domain name, the ECC certificate is displayed preferentially.
- If a certificate has expired, you need to manually replace or delete it by following the instructions in **Adding, Modifying, or Deleting a Certificate**.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Summary** tab page, click **Edit** on the top right.

7. Enable SNI and select an SNI certificate.
8. Click **OK**.

5 Backend Server Group

5.1 Overview

Introduction

A backend server group is a logical collection of one or more backend servers to receive massive concurrent requests at the same time. A backend server can be an ECS, BMS, or IP address.

The following process describes how a backend server group forwards traffic:

1. A client sends a request to your application. The listeners added to your load balancer use the protocols and ports you have configured forward the request to the associated backend server group.
2. Healthy backend servers in the backend server group receive the request based on the load balancing algorithm, handle the request, and return a result to the client.
3. In this way, massive concurrent requests can be processed at the same time, improving the availability of your applications.

Advantages

Backend server groups can bring the following benefits:

- **Reduced costs and easier management**: You can add or remove backend servers as traffic changes over the time. This can help avoid low resource utilization and makes it easy to manage backend servers.
- **Higher reliability**: Traffic is routed only to healthy backend servers in the backend server group.

Key Functions

You can configure the key functions listed in **Table 5-1** for each backend server group to ensure service stability.

Table 5-1 Key functions

Key Function	Description	Detail
Health Check	Specifies whether to enable the health check option. Health checks determine whether backend servers are healthy. If a backend server is detected unhealthy, it will not receive requests from the associated load balancer, improving your service reliability.	**Health Check**
Load Balancing Algorithm	The load balancer distributes traffic based on the load balancing algorithm you have configured for the backend server group.	**Load Balancing Algorithms**
Sticky Session	Specifies whether to enable the sticky session option. If you enable this option, all requests from a client during one session are sent to the same backend server.	**Sticky Session**
Slow Start	Specifies whether to enable slow start. After you enable it, the load balancer linearly increases the proportion of requests to backend servers in this mode. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode. **NOTE** Slow start is only available for HTTP and HTTPS backend server groups of dedicated load balancers.	**Slow Start (Dedicated Load Balancers)**

Precautions for Creating a Backend Server Group

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 5-2**.

You can create a backend server group by referring to **Table 5-3**.

Table 5-2 The frontend and backend protocol

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	● UDP ● QUIC

Frontend Protocol	Backend Protocol
HTTP	HTTP
HTTPS	● HTTP ● HTTPS

Table 5-3 Creating a backend server group

Load Balancer Type	Reference
Dedicated	**Creating a Backend Server Group**

5.2 Key Features

5.2.1 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.

If backend servers have to handle large number of requests, frequent health checks may overload the backend servers and cause them to respond slowly. To address this problem, you can prolong the health check interval or use TCP or UDP instead of HTTP. You can also disable health check. If you choose to disable health check, requests may be routed to unhealthy servers, and service interruptions may occur.

Health Check Protocol

You can configure health checks when configuring backend server groups. Generally, you can use the default setting or select a different health check protocol as you need.

If you want to modify health check settings, see details in **Modifying Health Check Settings**.

Select a health check protocol that matches the backend protocol as described in **Table 5-4**.

Table 5-4 The backend protocol and health check protocols (dedicated load balancers)

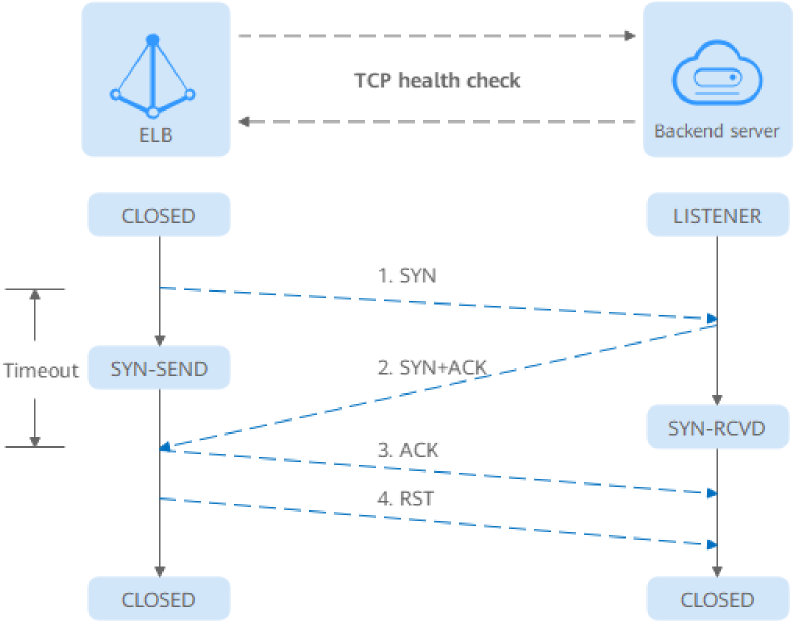
Backend Protocol	Health Check Protocol
TCP	TCP, HTTP, or HTTPS

Backend Protocol	Health Check Protocol
UDP	UDP
QUIC	UDP
HTTP	TCP, HTTP, or HTTPS
HTTPS	TCP, HTTP, or HTTPS

TCP Health Check

For TCP, HTTP, and HTTPS backend protocols, you can use TCP to initiate three-way handshakes to obtain the statuses of backend servers.

Figure 5-1 TCP health check



The TCP health check process is as follows:

1. The load balancer sends a TCP SYN packet to the backend server (in the format of {*Private IP address*}:{*Health check port*}).
2. The backend server returns an SYN-ACK packet.
   - If the load balancer does not receive the SYN-ACK packet within the timeout duration, it declares that the backend server is unhealthy and sends an RST packet to the backend server to terminate the TCP connection.
   - If the load balancer receives the SYN-ACK packet from the backend server within the timeout duration, it sends an ACK packet to the backend server and declares that the backend server is healthy. After that, the load balancer sends an RST packet to the backend server to terminate the TCP connection.

NOTICE

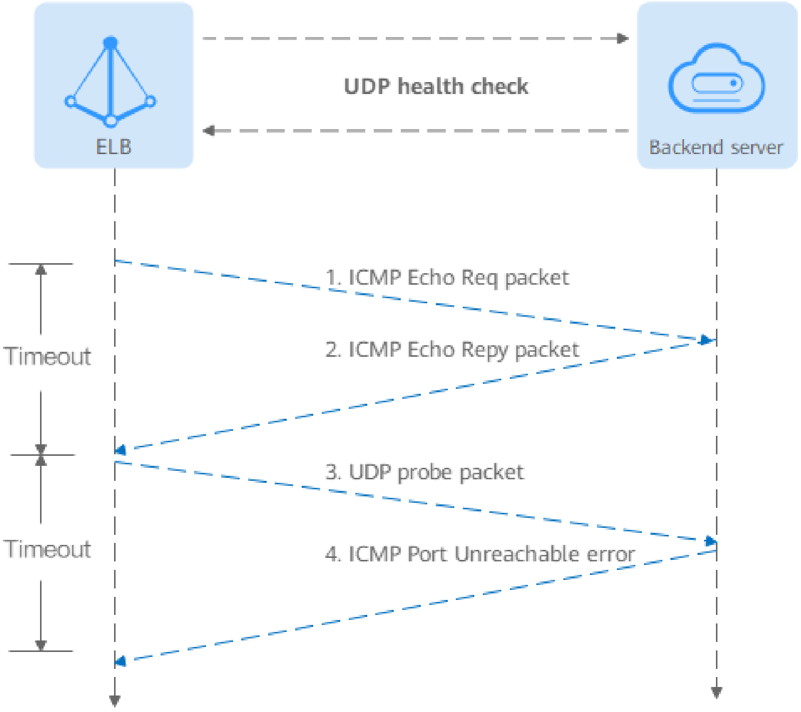
After a successful TCP three-way handshake, an RST packet will be sent to close the TCP connection. The application on the backend server may consider this packet a connection error and reply with a message, for example, "Connection reset by peer". To avoid this issue, take either of the following actions:

- Use **HTTP Health Check**.
- Have the backend server ignore the connection error.

UDP Health Check

For UDP backend protocol, ELB sends ICMP and UDP probe packets to backend servers to check their health.

Figure 5-2 UDP health check



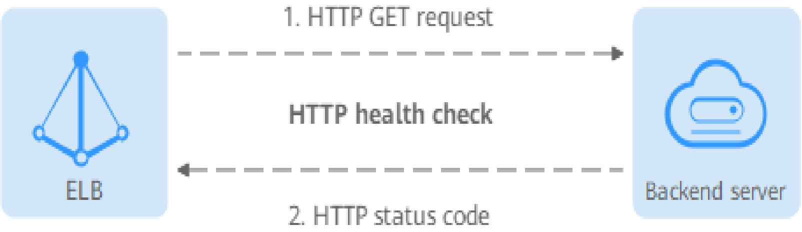
The UDP health check process is as follows:

1. The load balancer sends an ICMP Echo Request packet to the backend server.
   - If the load balancer does not receive an ICMP Echo Reply packet within the health check timeout duration, the backend server is declared unhealthy.
   - If the load balancer receives an ICMP Echo Reply packet within the timeout period, it sends a UDP probe packet to the backend server.
2. If the load balancer does not receive an ICMP Port Unreachable error within the health check timeout duration, it declares the backend server is healthy. If the load balancer receives an ICMP Port Unreachable error, the backend server is declared unhealthy.

HTTP Health Check

You can also configure HTTP health checks to obtain server statuses through HTTP GET requests if you select TCP, HTTP, or HTTPS as the backend protocol. **Figure 5-3** shows how an HTTP health check works.

Figure 5-3 HTTP health check



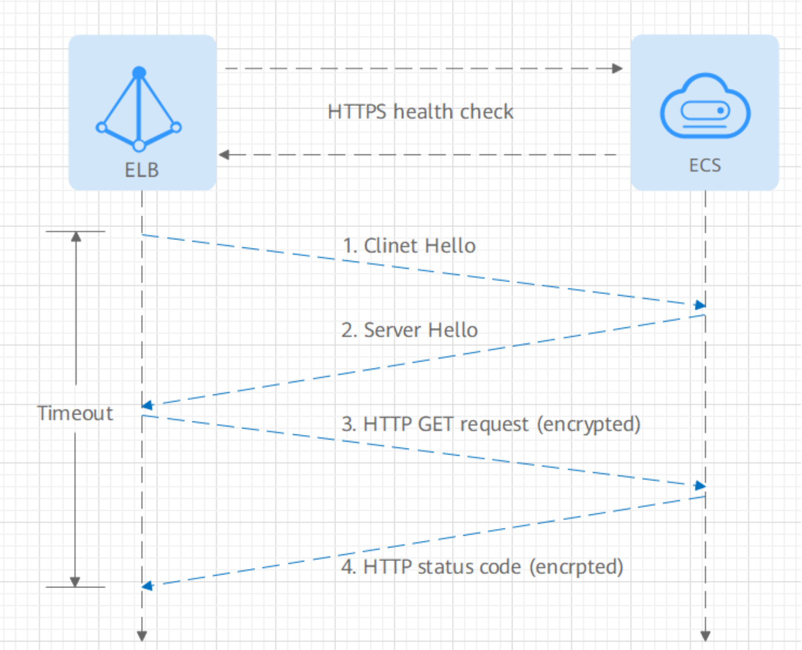
The HTTPS health check process is as follows:

1. The load balancer sends an HTTP GET request to the backend server (in format of *{Private IP address}*:*{Health check port}*/*{Health check path}*). (You can specify a domain name when configuring a health check.)
2. The backend server returns an HTTP status code to ELB.
   - If the load balancer receives the status code within the health check timeout duration, it compares the status code with the preset one. If the status codes are the same, the backend server is declared healthy.
   - If the load balancer does not receive any response from the backend server within the health check timeout duration, it declares the backend server is unhealthy.

HTTPS Health Check

For TCP, HTTP, and HTTPS backend protocols, you can use HTTPS to establish an SSL connection over TLS handshakes to obtain the statuses of backend servers. **Figure 5-4** shows how an HTTPS health check works.

Figure 5-4 HTTPS health check


ELB
HTTPS health check
ECS
1. Clinet Hello
2. Server Hello
Timeout
3. HTTP GET request (encrypted)
4. HTTP status code (encrpted)

The HTTPS health check process is as follows:

1. The load balancer sends a Client Hello packet to establish an SSL connection with the backend server.
2. After receiving the Server Hello packet from the backend server, the load balancer sends an encrypted HTTP GET request to the backend server (in the format of *{Private IP address}*:*{Health check port}*/*{Health check path}*). (You can specify a domain name when configuring a health check.)
3. The backend server returns an HTTP status code to the load balancer.
   - If the load balancer receives the status code within the health check timeout duration, it compares the status code with the preset one. If the status codes are the same, the backend server is declared healthy.
   - If the load balancer does not receive any response from the backend server within the health check timeout duration, it declares the backend server is unhealthy.

Health Check Time Window

Health checks greatly improve service availability. However, if health checks are too frequent, service availability will be compromised. To avoid the impact, ELB declares a backend server healthy or unhealthy after several consecutive health checks.

The health check time window is determined by the factors in **Table 5-5**:

Table 5-5 Factors affecting the health check time window

Factor	Description
Check Interval	How often health checks are performed.

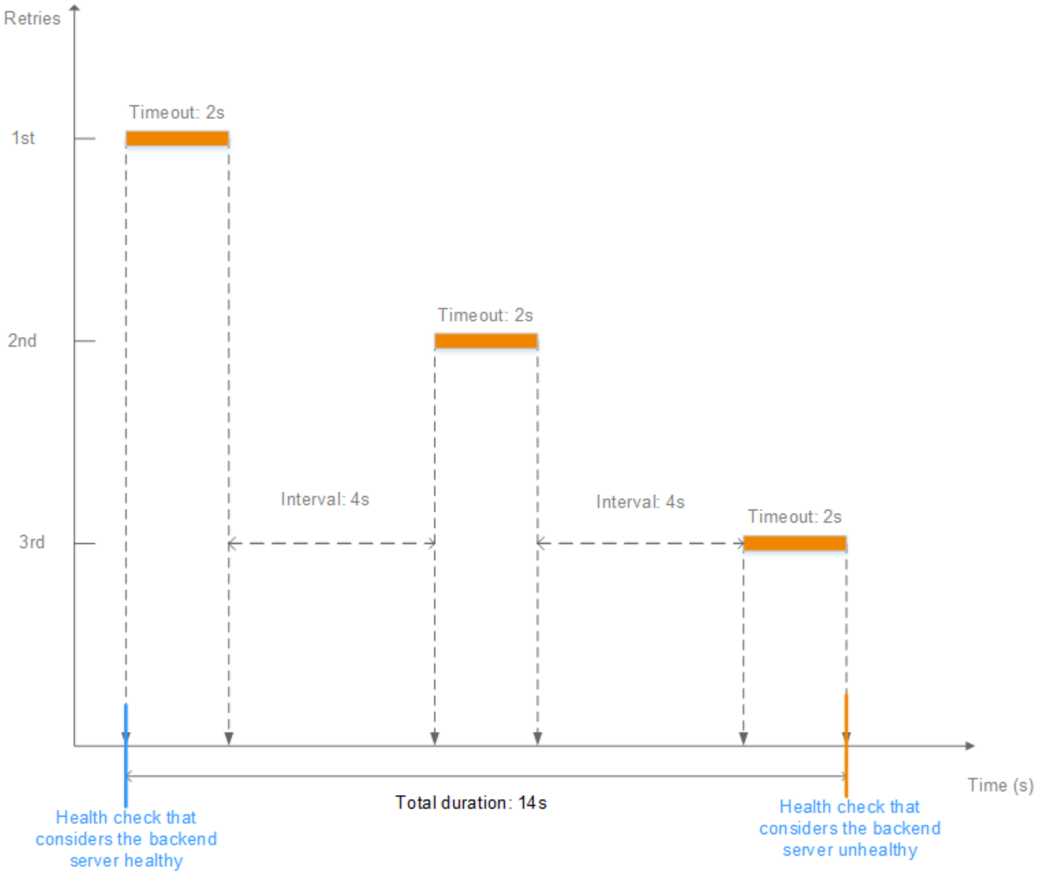
Factor	Description
Timeout Duration	How long the load balancer waits for the response from the backend server.
Health Check Threshold	The number of consecutive successful or failed health checks required for determining whether the backend server is healthy or unhealthy.

The following is a formula for you to calculate the health check time window:

- Time window for a backend server to be detected healthy = Timeout duration x Healthy threshold + Interval x (Healthy threshold – 1)
- Time window for a backend server to be detected unhealthy = Timeout duration x Unhealthy threshold + Interval x (Unhealthy threshold – 1)

As shown in **Figure 5-5**, if the health check interval is 4s, the health check timeout duration is 2s, and unhealthy threshold is 3, the time window for a backend server to be considered unhealthy is calculated as follows: 2 x 3 + 4 x (3 – 1) = 14s.

Figure 5-5 Health check timeout duration


Retries
1st
2nd
3rd
Timeout: 2s
Timeout: 2s
Interval: 4s
Interval: 4s
Timeout: 2s
Time (s)
Total duration: 14s
Health check that considers the backend server healthy
Health check that considers the backend server unhealthy

Rectifying an Unhealthy Backend Server

If a backend server is detected unhealthy, see **How Do I Troubleshoot an Unhealthy Backend Server?**

5.2.2 Load Balancing Algorithms

Overview

Load balancers receive requests from clients and forward them to backend servers in one or more AZs. Each load balancer has at least a listener and a backend server. The load balancing algorithm you select when you create the backend server group determines how requests are distributed.

ELB supports the following load balancing algorithms: weighted round robin, weighted least connections, source IP hash, and connection ID.

You can select the load balancing algorithm that best suits your needs.

Table 5-6 Load balancing algorithms

Load Balancing Algorithm	Description
Weighted round robin	Routes requests to backend servers in sequence based on their weights.
Weighted least connections	Routes requests to backend servers with the smallest connections-to-weight ratio.
Consistent hashing ● Source IP hash ● Connection ID	Calculates the request fields using the consistent hashing algorithm to obtain a hash value and routes requests with the same hash value to the same backend server, even if the number of backend servers in the backend server group changes. ● Source IP hash: Calculates the source IP address of each request and routes requests from the same source IP address to the same backend server. ● Connection ID: Calculates the QUIC connection ID and routes requests with the same ID to the same backend server.

Weighted Round Robin

Figure 5-6 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 5-6 Traffic distribution using the weighted round robin algorithm

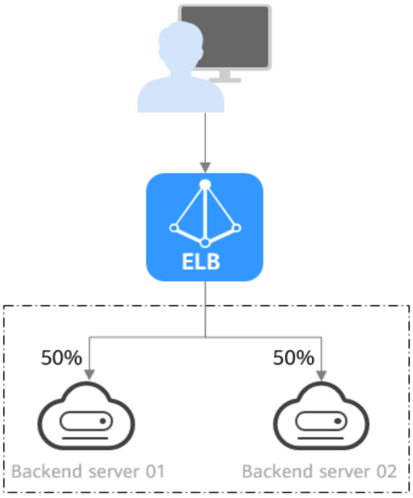


Table 5-7 Weighted round robin

Description	Requests are routed to backend servers based on their weights. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests.
When to Use	This algorithm is typically used for short connections, such as HTTP connections. ● Flexible load balancing: When you need more refined load balancing, you can set a weight for each backend server to specify the percentage of requests to each server. For example, you can set higher weights to backend servers with better performance so that they can process more requests. ● Dynamic load balancing: You can adjust the weight of each backend server in real time when the server performance or load fluctuates.
Disadvantages	● You need to set a weight for each backend server. If you have a large number of backend servers or your services require frequent adjustments, setting weights would be time-consuming. ● If the weights are inappropriate, the requests processed by each server may be imbalanced. As a result, you may need to frequently adjust server weights.

Weighted Least Connections

Figure 5-7 shows an example of how requests are distributed using the weighted least connections algorithm. Two backend servers are in the same AZ and have the same weight, 100 connections have been established with backend server 01, and 50 connections have been established with backend server 02. New requests are preferentially routed to backend server 02.

Figure 5-7 Traffic distribution using the weighted least connections algorithm

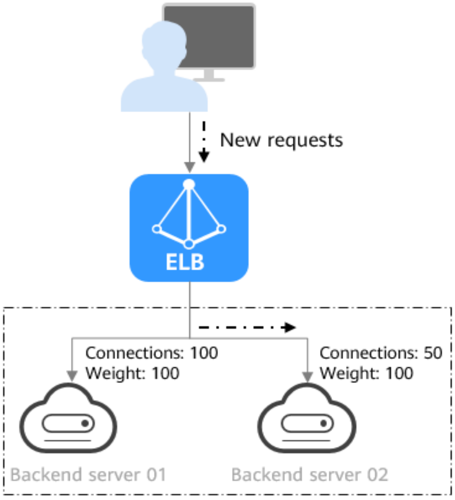


Table 5-8 Weighted least connections

Description	In addition to the number of active connections established with each backend server, each server is assigned a weight based on their processing capability. Requests are routed to the server with the lowest connections-to-weight ratio.
When to Use	This algorithm is often used for persistent connections, such as connections to a database. ● Flexible load balancing: Load balancers distribute requests based on the number of established connections and the weight of each backend server and route requests to the server with the lowest connections-to-weight ratio. This helps prevent servers from being underloaded or overloaded. ● Dynamic load balancing: When the number of connections to and loads on backend servers change, you can use the weighted least connection algorithm to dynamically adjust the requests distributed to each server in real time. ● Stable load balancing: You can use this algorithm to reduce the peak loads on each backend server and improve service stability and reliability.

Disadvantages	- Complex calculation: The weighted least connections algorithm needs to calculate and compare the number of connections established with each backend server in real time before selecting a server to route requests. - Dependency on connections to backend servers: The algorithm routes requests based on the number of connections established with each backend server. If monitoring data is inaccurate or outdated, requests may not be distributed evenly across backend servers. The algorithm can only collect statistics on the connections between a given load balancer and a backend server, but cannot obtain the total number of connections to the backend server if it is associated with multiple load balancers. - Too much loads on new servers: If existing backend servers have to handle a large number of requests, new requests will be routed to new backend servers. This may deteriorate new servers or even cause them to fail.

Source IP Hash

Figure 5-8 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 5-8 Traffic distribution using the source IP hash algorithm

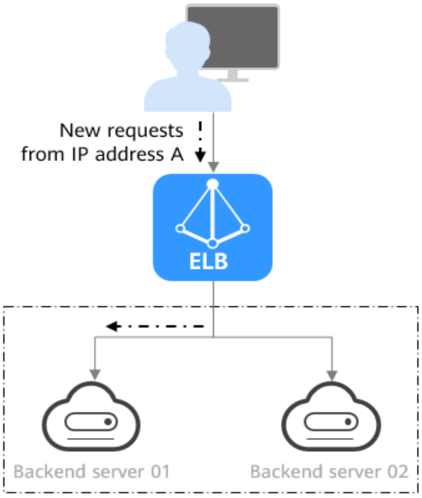


Table 5-9 Source IP hash

Description	The source IP hash algorithm calculates the source IP address of each request and routes requests from the same IP address to the same backend server.

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

Connection ID

Figure 5-9 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 5-9 Traffic distribution using the connection ID algorithm

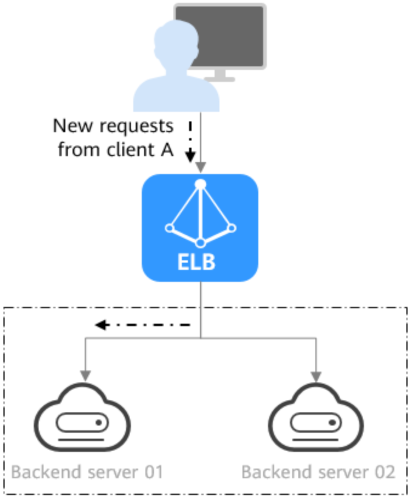


Table 5-10 Connection ID

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

5.2.3 Sticky Session

Sticky sessions ensure that requests from a client always get routed to the same backend server before a session elapses.

Here is an example that describes how sticky session works. Assume that you have logged in to a server. After a while, you send another request. If sticky sessions are not enabled, the request may be routed to another server, and you will be asked to log in again. If sticky sessions are enabled, all your requests are processed by the same server, and you do not need to repeatedly log in.

Differences Between Sticky Sessions at Layer 4 and Layer 7

The following table describes the differences of sticky sessions at Layer 4 at Layer 7.

Table 5-11 Sticky session comparison

OSI Layer	Listener Protocol	Sticky Session Type	Scenarios Where Sticky Sessions Become Invalid
Layer 4	TCP or UDP	**Source IP address**: 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 system allocates the client to a particular server based on the generated key. This allows requests from the same IP address are forwarded to the same backend server.	● Source IP addresses of the clients change. ● The session stickiness duration has been reached.
Layer 7	HTTP or HTTPS	● **Load balancer cookie**: The load balancer generates a cookie after receiving a request from the client. All subsequent requests with the cookie are routed to the same backend server. The load balancer itself does not generate cookies. ● **Application cookie**: The application deployed on the backend server generates a cookie after receiving the first request from the client. All subsequent requests with the same cookie are routed to the same backend server.	● If requests sent by the clients do not contain a cookie, sticky sessions will not take effect. ● Requests from the clients exceed the session stickiness duration.

NOTE

- If you set **Load Balancing Algorithm** to **Source IP hash**, you do not need to manually enable and configure **Sticky Session**. Source IP hash allows requests from the same client to be directed to the same server.
- If you set **Load Balancing Algorithm** to **Weighted round robin** or **Weighted least connections**, you need to manually enable and configure **Sticky Session**.

Constraints and Limitations

- If you use **Direct Connect** or **VPN** to access ELB, you must select **Source IP hash** as the load balancing algorithm and disable sticky sessions for ELB.
- Dedicated load balancers support **Source IP address** and **Load balancer cookie**.

NOTE

- For HTTP and HTTPS listeners, enabling or disabling sticky sessions may cause few seconds of service interruption.
- If you enable sticky sessions, traffic to backend servers may be unbalanced. If this happens, disable sticky sessions and check the requests received by each backend server.

5.2.4 Slow Start (Dedicated Load Balancers)

If you enable slow start, the load balancer linearly increases the proportion of requests to the new backend servers added to the backend server group. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode. For details about how to set weights for backend servers, see **Backend Server Weights**.

Slow start gives applications time to warm up and respond to requests with optimal performance.

NOTE

Slow start is only available for HTTP and HTTPS backend server groups of dedicated load balancers.

Backend servers will exit slow start in either of the following cases:

- The slow start duration elapses.
- Backend servers become unhealthy during the slow start duration.

Constraints

- Weighted round robin must be selected as the load balancing algorithm.
- Slow start takes effect only for new backend servers and does not take effect when the first backend server is added to a backend server group.
- After the slow start duration elapses, backend servers will not enter the slow start mode again.
- Slow start takes effect when health check is enabled and the backend servers are running normally.
- If health check is disabled, slow start takes effect immediately.

5.3 Creating a Backend Server Group

Scenario

To route requests, you need to associate at least one backend server group to each listener.

NOTE

This section describes how you can create a backend server group for a dedicated load balancer.

You can create a backend server group for a load balancer in any of the ways described in **Table 5-12**.

Table 5-12 Creating a backend server group

Scenario	Procedure
Creating a backend server group and associating it with a load balancer	**Procedure**
Creating a backend server group when adding a listener	You can add listeners using different protocols as required. For details, see **Overview**. References are as follows: ● **Adding a TCP Listener** ● **Adding a UDP Listener** ● **Adding an HTTP Listener** ● **Adding an HTTPS Listener**
Changing the backend server group associated with the listener	**Changing a Backend Server Group**

Constraints

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 5-13**.

Table 5-13 The frontend and backend protocol

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	● UDP ● QUIC
HTTP	HTTP

Frontend Protocol	Backend Protocol
HTTPS	• HTTP • HTTPS

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. Click **Create Backend Server Group** in the upper right corner.
6. Configure the routing policy based on **Table 5-14**.

Table 5-14 Parameters required for configuring a routing policy

Parameter	Description	Example Value
Load Balancer Type	Specifies the type of load balancers that can use the backend server group. Dedicated load balancers are recommended. The following parameters apply to exclusive load balancers.	-
Load Balancer	Specifies whether to associate a load balancer.	-
Backend Server Group Name	Specifies the name of the backend server group.	server_group
Backend Protocol	Specifies the protocol that backend servers in the backend server group use to receive requests from the listeners. The protocol varies depending on the forwarding mode:	HTTP

Parameter	Description	Example Value
Load Balancing Algorithm	Specifies the algorithm used by the load balancer to distribute traffic. The following options are available: ● **Weighted round robin**: Requests are routed to different servers based on their weights. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. ● **Weighted least connections**: 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. ● **Source IP hash**: Allows requests from different clients to be routed based on source IP addresses and ensures that requests from the same client are forwarded to the same server. For more information about load balancing algorithms, see **Load Balancing Algorithms**.	Weighted round robin
Sticky Session	Specifies whether to enable sticky sessions if you have selected **Weighted round robin** or **Weighted least connections** for **Load Balancing Algorithm**. If you enable sticky sessions, all requests from the same client during one session are sent to the same backend server. For more information about sticky sessions, see **Sticky Session**.	-

Parameter	Description	Example Value
Sticky Session Type	Specifies the sticky session type. This parameter is mandatory if **Sticky Session** is enabled. You can select one of the following type: ● **Source IP address**: 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 system allocates the client to a particular server based on the generated key. This allows requests from the same IP address are forwarded to the same backend server. ● **Load balancer cookie**: The load balancer generates a cookie after receiving a request from the client. All subsequent requests with the cookie are routed to the same backend server. **NOTE** ● **Source IP address** is available when you have selected **TCP** or **UDP** for **Backend Protocol**. ● **Load balancer cookie** is available when you have selected **HTTP** or **HTTPS** for **Backend Protocol**.	Source IP address
Slow Start	Specifies whether to enable slow start. This parameter is optional if you have selected **Weighted round robin** for **Load Balancing Algorithm**. After you enable this option, the load balancer linearly increases the proportion of requests to backend servers in this mode. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode. **NOTE** Slow start is only available for HTTP and HTTPS backend server groups of dedicated load balancers. For more information about the slow start, see **Slow Start (Dedicated Load Balancers)**.	-

Parameter	Description	Example Value
Slow Start Duration (s)	Specifies how long the slow start will last, in seconds. This parameter is mandatory if **Slow Start** is enabled.	30
Description	Provides supplementary information about the backend server group.	-

7. Click **Next** to add backend servers and configure health check.

   Add cloud servers, or IP addresses to this backend server group. For details, see **Overview**.

   Configure health check for the backend server group based on **Table 5-15**. For more information about health checks, see **Health Check**.

Table 5-15 Parameters required for configuring a health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks. If the health check is enabled, click ✎ next to **Advanced Settings** to set health check parameters.	-
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. ● The backend protocol can be TCP, HTTP, or HTTPS. ● If the protocol of the backend server group is UDP, the health check protocol is UDP by default.	HTTP

Parameter	Description	Example Value
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. ● You can use the private IP address of the backend server as the domain name. ● You can also specify a domain name that consists of at least two labels separated by periods (.). Use only letters, digits, and hyphens (-). Do not start or end strings with a hyphen. Max total: 100 characters. Max label: 63 characters.	www.elb.com
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from 1 to 65535. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80
Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. The path can contain 1 to 80 characters and must start with a slash (/). The path can contain letters, digits, hyphens (-), slashes (/), periods (.), question marks (?), percent signs (%), ampersands (&), and underscores (_).	/index.html
Interval (s)	Specifies the maximum time between two consecutive health checks, in seconds. The interval ranges from **1** to **50**.	5

Parameter	Description	Example Value
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The interval ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **Next**.
9. Confirm the specifications and click **Create Now**.

Related Operations

You can associate the backend server group with the listener of a dedicated load balancer in either ways listed in **Table 5-12**.

5.4 Modifying a Backend Server Group

5.4.1 Overview

After a backend server group is created, you can modify its health check settings and basic information.

Health Check

If backend servers have to handle large number of requests, frequent health checks may overload the backend servers and cause them to respond slowly. To address this problem, you can prolong the health check interval or use TCP or UDP instead of HTTP. You can also disable health check. If you choose to disable health check, requests may be routed to unhealthy servers, and service interruptions may occur.

For details about the health check, see **Health Check**.

For details about how to modify health check settings, see **Modifying Health Check Settings**.

Basic Information

You can modify the basic information of a backend server group listed in **Table 5-16**.

Table 5-16 Basic information that can be modified

Parameter	Description
Name	Change the name by performing the operations in **Changing the Load Balancing Algorithm**.
Load Balancing Algorithm	Change the load balancing algorithm by performing the operations in **Changing the Load Balancing Algorithm**. For details about load balancing algorithms, see **Load Balancing Algorithms**.
Sticky Session	Enable or disable sticky session by performing the operations in **Modifying Sticky Session Settings**. For details about the sticky session function, see **Sticky Session**.
Slow Start	Enable or disable slow start by performing the operations in **Modifying Slow Start Settings (Dedicated Load Balancers)**. For details about the slow start function, see **Slow Start (Dedicated Load Balancers)**.
Description	Change the description of the backend server group by performing the operations in **Changing the Load Balancing Algorithm**.

5.4.2 Modifying Health Check Settings

Scenario

This section describes how you can modify the health check settings.

After the protocol is changed, the load balancer uses the new protocol to check the health of backend servers. The load balancer continues to route traffic to the backend servers after they are detected healthy.

Before the new configurations take effect, the load balancer may return the HTTP 503 error code to the clients.

Constraints and Notes

- The health check protocol can be different from the backend protocol.
- To reduce the vCPU usage of the backend servers, it is recommended that you use TCP for health checks. If you want to use HTTP for health checks, you can use static files to return the health check results.
- If health check is enabled, security group rules must allow traffic from the health check port to the backend servers over the health check protocol.
  - Dedicated load balancers: For details, see **Security Group Rules**.

NOTE

After you enable health check, the load balancer immediately checks the health of backend servers.

- If a backend server is detected healthy, the load balancer will start routing requests to it over new connections based on the configured loading balancing algorithms and weights.
- If a backend server is detected unhealthy, the load balancer will stop routing traffic to it.

Enabling Health Check

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** tab page, locate the backend server group.
6. On the **Summary** page, click **Health Check** on the right.
7. In the **Configure Health Check** dialog box, configure the parameters based on **Table 5-17**.

Table 5-17 Parameters required for configuring health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable health checks.	-
Health Check Protocol	• The health check protocol can be TCP, HTTP, or HTTPS. • If the protocol of the backend server group is UDP, the health check protocol is UDP by default.	HTTP

Parameter	Description	Example Value
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. ● You can use the private IP address of the backend server as the domain name. ● You can also specify a domain name that consists of at least two labels separated by periods (.). Use only letters, digits, and hyphens (-). Do not start or end strings with a hyphen. Max total: 100 characters. Max label: 63 characters.	www.elb.com
Health Check Port	Specifies the port that will be used by the load balancer to check the health of backend servers. The port number ranges from **1** to **65535**. **NOTE** By default, the service port on each backend server is used. You can also specify a port for health checks.	80
Interval (s)	Specifies the maximum time between two consecutive health checks, in seconds. The interval ranges from **1** to **50**.	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The interval ranges from **1** to **50**.	3
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from **1** to **10**.	3

8. Click **OK**.

Disabling Health Check

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.

3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. On the **Summary** page, click **Health Check** on the right.
7. In the **Configure Health Check** dialog box, disable health check.
8. Click **OK**.

5.4.3 Changing the Load Balancing Algorithm

Scenario

This section describes how you can change the load balancing algorithm.

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

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the target backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, change the load balancing algorithm.
7. Click **OK**.

NOTE

The new load balancing algorithm takes effect immediately and will be used to route requests over new connections. However, the previous load balancing algorithm will still be used to route requests over established connections.

5.4.4 Modifying Sticky Session Settings

Scenario

This section describes how you can modify the sticky session settings.

NOTE

- This section applies to dedicated and shared load balancers.
- You can also configure sticky sessions when adding a listener or creating a backend server group.

Enabling Sticky Session

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, enable sticky session, select the sticky session type, and set the session stickiness duration.
7. Click **OK**.

Disabling Sticky Session

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, disable sticky session.
7. Click **OK**.

5.4.5 Modifying Slow Start Settings (Dedicated Load Balancers)

Scenario

This section describes how you can modify the slow start settings.

For details, see **Slow Start (Dedicated Load Balancers)**.

NOTE

- This section applies only to dedicated load balancers.
- You can also configure slow start when adding a listener or creating a backend server group.

Enabling Slow Start

1. Log in to the management console.

2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, enable slow start and set the slow start duration.

   The slow start duration ranges from 30 to 1200 in seconds. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits the slow start mode.
7. Click **OK**.

Disabling slow start

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, disable slow start.
7. Click **OK**.

5.5 Changing a Backend Server Group

Scenario

This section describes how you can change the default backend server group configured for a listener.

TCP or UDP listeners forward requests to the default backend server groups.

HTTP or HTTPS listeners forward requests based on the priorities of the forwarding policies. If you do not add a forwarding policy, the listener will route the requests to the default backend server group.

Constraints and Limitations

- The backend server group cannot be changed if redirection is enabled.
- The backend protocol of the backend server group must match the frontend protocol of the listener. For details, see **Table 5-2**.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the target load balancer and click its name.
5. On the **Listeners** tab, locate the target listener and click its name.
6. On the **Summary** page, click **Change Backend Server Group** on the right.
7. In the displayed dialog box, click the server group name box.

   Select a backend server group from the drop-down list or create a group.

   a. Click the name of the backend server group or enter the name in the search box to search for the target group.

   b. Click **Create Backend Server Group**. After the backend server group is created, click the refresh icon.

      NOTE

      The backend protocol of the new backend server group must match the frontend protocol of the listener.
8. Click **OK**.

5.6 Viewing a Backend Server Group

Scenario

This section describes how you can view the following information about a backend server group:

- Basic information: the name, ID, and backend protocol
- Health check: whether health check is enabled and health check configurations
- Backend servers: servers that have been added to the backend server group

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.

6. On the **Summary** tab page, view the basic information and health check settings.

5.7 Deleting a Backend Server Group

Scenario

This section describes how you can delete a backend server group.

Constraints and Limitations

- Before you delete a backend server group, you need to:
  - Disassociate it from the listener. For details, see **Changing a Backend Server Group**.
  - Ensure the backend server group is not used by a forwarding policy of an HTTP or HTTPS listener.
- Remove all backend servers from the backend server group.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click **Delete** in the **Operation** column.
6. In the displayed dialog box, click **Yes**.

6 Backend Server

6.1 Overview

Backend servers receive and process requests from the associated load balancer.

If the incoming traffic increases, you can add more backend servers to ensure the stability and reliability of applications and eliminate SPOFs. If the incoming traffic decreases, you can remove some backend servers to reduce the cost.

If the load balancer is associated with an AS group, instances are automatically added to or removed from the load balancer.

Precautions

- It is recommended that you select backend servers running the same OS for easier management and maintenance.
- The load balancer checks the health of each server added to the associated backend server group if you have configured health check for the backend server group. If the backend server responds normally, the load balancer will consider it healthy. If the backend server does not respond normally, the load balancer will periodically check its health until the backend server is considered healthy.
- If a backend server is stopped or restarted, connections established with the server will be disconnected, and data being transmitted over these connections will be lost. To avoid this from happening, configure the retry function on the clients to prevent data loss.
- If you enable sticky sessions, traffic to backend servers may be unbalanced. If this happens, disable sticky sessions and check the requests received by each backend server.

Constraints and Limitations

- A maximum of 500 backend servers can be added to a backend server group.
- Inbound security group rules must be configured to allow traffic over the port of each backend server and health check port. For details, see **Security Group Rules**.

- If you select only network load balancing, a server cannot serve as both a backend server and a client.

Backend Server Weights

You need to set a weight for each backend server in a backend server group to receive requests. The higher the weight you have configured for a backend server, the more requests the backend server receives.

You can set an integer from **0** to **100**. If you set the weight of a backend server to **0**, new requests will not be routed to this server.

Three load balancing algorithms allow you to set weights to backend servers, as shown in the following table. For more information about load balancing algorithms, see **Load Balancing Algorithms**.

Table 6-1 Server weights in different load balancing algorithms

Load Balancing Algorithm	Weight Setting
Weighted round robin	● If none of the backend servers have a weight of 0, the load balancer routes requests to backend servers based on their weights. Backend servers with higher weights receive proportionately more requests. ● If two backend servers have the same weights, they receive the same number of requests.
Weighted least connections	● If none of the backend servers have a weight of 0, the load balancer calculates the load of each backend server using the formula (Overhead = Number of current connections/Backend server weight). ● The load balancer routes requests to the backend server with the lowest overhead.
Source IP hash	● If none of the backend servers have a weight of 0, requests from the same client are routed to the same backend server within a period of time. ● If the weight of a backend server is 0, no requests are routed to this backend server.

6.2 Security Group Rules

Scenarios

To ensure normal communications between the load balancer and backend servers, you need to check the security group rules and network ACL rules configured for the backend servers.

- Security group rules must allow traffic from the backend subnet where the load balancer resides to the backend servers. (By default, the backend subnet of a load balancer is the same as the subnet where the load balancer resides.) For details about how to configure security group rules, see **Configuring Security Group Rules**.

- Network ACL rules are optional for subnets. If network ACL rules are configured for the backend subnet of the load balancer, the network ACL rules must allow traffic from the backend subnet of the load balancer to the backend servers. For details about how to configure rules, see **Configuring Network ACL Rules**.

NOTE

If the load balancer has a TCP or UDP listener and IP as a backend is disabled, security group rules and network ACL rules will not take effect.

You can use access control to limit which IP addresses are allowed to access the listener. Learn how to configure **Access Control**.

Constraints and Limitations

- If health check is enabled for a backend server group, security group rules must allow traffic from the health check port over the health check protocol.
- If UDP is used for health check, there must be a rule that allows ICMP traffic to check the health of the backend servers.

Configuring Security Group Rules

If you have no VPCs when creating a server, the system automatically creates one for you. Default security group rules allow only communications among the servers in the VPC. To ensure that the load balancer can communicate with these servers over both the frontend port and health check port, configure inbound rules for security groups containing these servers.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Under **Computing**, click **Elastic Cloud Server**.
4. On the **Elastic Cloud Server** page, click the name of the ECS that has been added to a backend server group.

   The page providing details about the ECS is displayed.
5. Click **Security Groups**, locate the security group, and view security group rules.
6. Click the ID of a security group rule or **Modify Security Group Rule**. The security group details page is displayed.
7. On the **Inbound Rules** tab page, click **Add Rule**. Configure an inbound rule based on **Table 6-2**.

Table 6-2 Security group rules

Backend Protocol	Protocol & Port	Source IP Address
HTTP or HTTPS	**Protocol**: TCP **Port**: the port used by the backend server and health check port	Backend subnet of the load balancer
TCP	**Protocol**: TCP **Port**: health check port	
UDP	**Protocol**: UDP and ICMP **Port**: health check port	

NOTE

- After a load balancer is created, do not change the subnet. If the subnet is changed, the IP addresses occupied by the load balancer will not be released, and traffic from the previous backend subnet is still need to be allowed to backend servers.
- Traffic from the new backend subnet is also need to be allowed to backend servers.

8. Click **OK**.

Configuring Network ACL Rules

To control traffic in and out of a subnet, you can associate a network ACL with the subnet. Network ACL rules control access to subnets and add an additional layer of defense to your subnets.

The default network ACL rule denies all inbound and outbound traffic. You can configure an inbound rule to allow traffic from the backend subnet of the load balancer through the port of the backend server.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Access Control** > **Network ACLs**.
5. In the network ACL list, click the name of the network ACL to switch to the page showing its details.
6. On the **Inbound Rules** or **Outbound Rules** tab page, click **Add Rule** to add an inbound or outbound rule.
   - **Type**: Select the same type as the backend subnet of the load balancer.
   - **Protocol**: The protocol must be the same as the backend protocol.
   - **Source**: Set it to the backend subnet of the load balancer.
   - **Source Port Range**: Select a port range.

  - **Destination**: Enter a destination address allowed in this direction. The default value is **0.0.0.0/0**, which indicates that traffic from all IP addresses is permitted.
  - **Destination Port Range**: Select a port range.
  - (Optional) **Description**: Describe the network ACL rule.

7. Click **OK**.

6.3 Managing Backend Servers

6.3.1 Adding Backend Servers

Scenario

When you use ELB to route traffic to backend servers, you need to ensure that at least one backend server is running properly and can receive requests from the associated load balancer.

If the incoming traffic increases, you can add more backend servers to ensure the stability and reliability of applications and eliminate SPOFs. If the incoming traffic decreases, you can remove some backend servers to reduce the cost.

Constraints and Limitations

- The cloud servers must be in the same VPC as the backend server group.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **Add** on the right.
7. You can search for backend servers using specified keywords.

   Select the backend servers you want to add and click **Next**.
8. Specify the weights and ports for the backend servers, and click **Finish**.

   Backend server ports can be set in batches.

6.3.2 Viewing Backend Servers

Scenario

You can view backend servers that have been added to a backend server group, including their status, private IP addresses, health check results, weights, and ports.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **Backend Servers**.
7. In the backend server list, view the backend servers.

6.3.3 Removing Backend Servers

Scenario

You can remove a backend server that is no longer needed from a backend server group.

Once a backend server is removed, it is disassociated from the load balancer and will no longer receive requests from the load balancer. The backend server still exists. You can add the backend server to the backend server group again when traffic increases or the reliability needs to be enhanced.

Notes

After the backend server is removed, requests are still routed to it. This is because a persistent connection is established between the load balancer and the backend server and requests are routed to this server until the TCP connection times out.

If no data is transmitted over this TCP connection after it times out, ELB disconnects the connection.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.

4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **Backend Servers**.
7. Select the backend servers you want to remove and click **Remove** above the backend server list.
8. In the displayed dialog box, click **Yes**.

6.3.4 Changing Backend Server Weights

Scenario

You can change the weights configured for backend servers based on their capability to process requests.

Constraints and Limitations

- You can set an integer from **0** to **100**. If you set the weight of a backend server to **0**, new requests will not be routed to this server.
- The weights can only be specified when you select weighted round robin, weighted least connections, or source IP hash as the load balancing algorithm. For more information about load balancing algorithms, see **Backend Server Weights**.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **Backend Servers**.
7. Select the backend servers and click **Modify Port/Weight** up above the backend server list.
8. In the displayed dialog box, modify weights as you need.
   - Modifying weights:
     - Changing the weight of a single backend server: Set the weight in the **New Weight** column.
     - Changing the weights of multiple backend servers: Set the weight next to **Batch Modify Weights** and click **OK**.

NOTE

You can change the weights of multiple backend servers to **0** so that they will not receive requests from the load balancer.

9. Click **OK**.

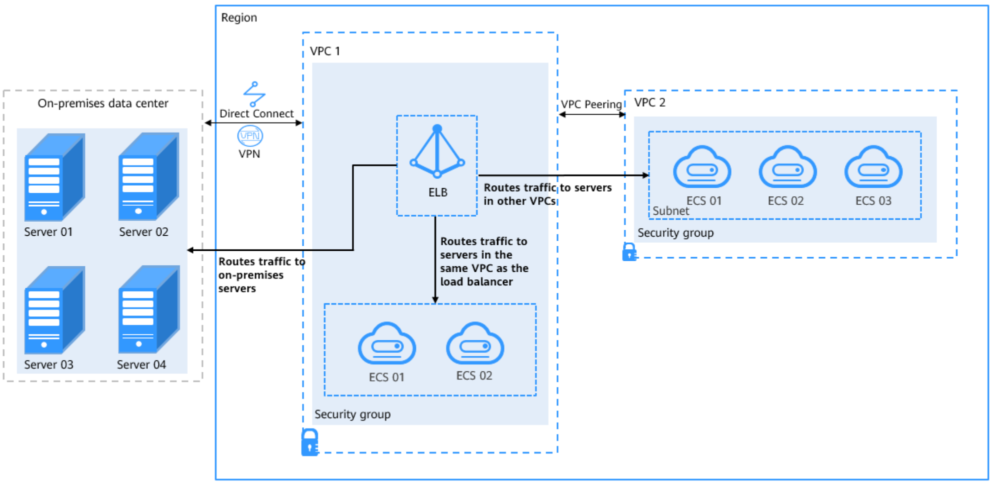
6.4 IP Addresses as Backend Servers

6.4.1 Overview

Dedicated load balancers support hybrid load balancing. You can add servers and supplementary network interfaces in the VPC where the load balancer is created, in a different VPC, or in an on-premises data center, by using private IP addresses of the servers to the backend server group of the load balancer.

In this way, incoming traffic can be flexibly distributed to cloud servers and on-premises servers.

Figure 6-1 Routing requests to cloud and on-premises servers



Constraints and Limitations

- IP as a backend cannot be disabled after it is enabled.
- Only private IPv4 addresses can be added as backend servers.
- A maximum of 50,000 concurrent connections can be established with a backend server that is added by using its IP address.
- If you add IP addresses as backend servers, the source IP addresses of the clients cannot be passed to these servers.

Scenario

After you enable IP as a backend, you can add backend servers by using their IP addresses. You need to get prepared for different scenarios as shown in **Table 6-3**.

Table 6-3 Adding IP addresses as backend servers

Where Servers Are Running	Preparations
In a different VPC from the load balancer	Set up a VPC peering connection between the VPC where the load balancer is running and the VPC where the servers are running. For details about how to set up a VPC peering connection, see the .
In on-premises data centers	Connect the on-premises data center to the VPC where the load balancer is running through Direct Connect or VPN. For details about how to connect on-premises data centers to the cloud, see the or .

6.4.2 Enabling IP as a Backend

Scenario

You can enable IP as a backend for an existing dedicated load balancer.

Constraints and Limitations

- IP as a backend cannot be disabled after it is enabled.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Summary** tab page, click **Enable** next to **IP as a Backend**.
6. Click **OK**.

6.4.3 Adding IP Addresses as Backend Servers

Scenario

If you enable IP as a backend, you can associate backend servers with the load balancer by using their IP addresses.

You need to get prepared for different scenarios as shown in **Table 6-4**.

Table 6-4 Adding IP addresses as backend servers

Where Servers Are Running	Preparations
In a different VPC from the load balancer	Set up a VPC peering connection between the VPC where the load balancer is running and the VPC where the servers are running. For details about how to set up a VPC peering connection, see the .
In on-premises data centers	Connect the on-premises data center to the VPC where the load balancer is running through Direct Connect or VPN. For details about how to connect on-premises data centers to the cloud, see the or .

Constraints and Limitations

- If IP as a backend is not enabled when you create a load balancer, you can enable it on the **Summary** page of the load balancer.
- Only private IPv4 addresses can be added as backend servers.
- The backend subnet of the load balancer must have sufficient IP addresses. Otherwise, backend servers cannot be added through IP addresses. If the IP addresses are insufficient, you can add more backend subnets on the **Summary** page of the load balancer.
- Security group rules of backend servers added through IP addresses must allow traffic from the backend subnet of the load balancer. If traffic is not allowed, health checks will fail.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **Add** on the **IP as Backend Servers** area.
7. Specify the IP addresses, ports, and weights for the backend servers.
8. Click **OK**.

6.4.4 Viewing Backend Servers

Scenario

You can view backend servers added to a backend server group, including their IP addresses, health check results, weights, and ports.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **IP as Backend Servers**.
7. In the backend server list, view the added backend servers.

6.4.5 Removing Backend Servers

Scenario

You can remove backend servers from a backend server group when you do not need them to process requests.

Notes

After the backend server is removed, requests are still routed to it. This is because a persistent connection is established between the load balancer and the backend server and requests are routed to this server until the TCP connection times out.

If no data is transmitted over this TCP connection after it times out, ELB disconnects the connection.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.

6. Switch to the **Backend Servers** tab page and click **IP as Backend Servers**.
7. Select the backend servers to be removed and click **Remove** above the backend server list.
8. In the displayed dialog box, click **Yes**.

6.4.6 Changing Backend Server Weights

Scenario

You can change the weights specified for backend servers based on their capability to process requests.

Constraints and Limitations

- You can set an integer from **0** to **100**. If you set the weight of a backend server to **0**, new requests will not be routed to this server.
- The weights can only be specified when you select weighted round robin, weighted least connections, or source IP hash as the load balancing algorithm. For more information about load balancing algorithms, see **Backend Server Weights**.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab page and click **IP as Backend Servers**.
7. Select the backend servers and click **Modify Port/Weight** up the backend server list.
8. In the displayed dialog box, modify weights as you need.
   - Modifying weights:
     - Changing the weight of a single backend server: Set the weight in the **New Weight** column.
     - Changing the weights of multiple backend servers: Set the weight next to **Batch Modify Weights** and click **OK**.

     NOTE

     You can change the weights of multiple backend servers to **0** so that they will not receive requests from the load balancer.
9. Click **OK**.

7 Certificate

7.1 Introduction to Certificates

ELB supports three types of certificates. 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.

- **Server SM certificates**: To support Chinese cryptographic algorithms, two certificates that must be used together are required, one signing certificate and one encryption certificate.
  - **Signing certificate**: This certificate is used only for identity authentication. The public and private keys are generated and kept by the server, rather than the CA.
  - **Encryption certificate**: This certificate is used for key negotiation. The public and private keys are generated and kept by the CA.

Precautions

- A certificate can be used by multiple load balancers but only needs to be uploaded to each load balancer once.
- You must specify a domain name for an SNI certificate. The domain name must be the same as that in the certificate. Only one domain name can be specified for each SNI certificate..
- For each certificate type, a listener can have only one certificate by default, but a certificate can be bound to more than one listener. If SNI is enabled for the listener, multiple server certificates can be bound.
- Only original certificates are supported. That is to say, you cannot encrypt your certificates.
- You can use self-signed certificates. However, note that self-signed certificates pose security risks. Therefore, it is recommended that you use certificates issued by third parties.
- ELB supports certificates only in PEM format. If you have a certificate in any other format, you must convert it to a PEM-encoded certificate.
- If a certificate has expired, you need to manually replace or delete it.

7.2 Certificate and Private Key Format

Certificate Format

You can copy and paste the certificate body to create a certificate or directly upload the certificate.

A certificate issued by the Root CA is unique, and no additional certificates are required. The configured site is considered trustable by access devices such as a browser.

The body of the server and CA certificates must meet the following requirements:

The body of SM signing and encryption certificates must meet the following requirements:

- The content starts with **-----BEGIN CERTIFICATE-----** and ends with **-----END CERTIFICATE-----**.
- Each row contains 64 characters except the last row.
- There are no empty rows.

The following is an example:

```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJALV96mEtVF4EMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAnh4MQswCQYDVQQIEwJ4eDELMAkGA1UEBxMCeHgxCzAJBgNVBAoTAnh4MQsw
CQYDVQQLEwJ4eDELMAkGA1UEAxMCeHgxGjAYBgkqhkiG9w0BCQEWC3h4eEAxNjMu
Y29tMB4XDTE3MTExMzAyMjYxM1oXDTIwMTExMjAyMjYxM1owajELMAkGA1UEBhMC
eHgxCzAJBgNVBAgTAnh4MQswCQYDVQQHEwJ4eDELMAkGA1UEChMCeHgxCzAJBgNV
BAsTAnh4MQswCQYDVQQDEwJ4eDEaMBgGCSqGSIb3DQEJARYLeHh4QDE2My5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMU832iM+d3FILgTWmpZBUoYcIWV
cAAYE7FsZ9LNerOyjJpyi256oypdBvGs9JAUBN5WaFk81UQx29wAyNixX+bKa0DB
WpUDqr84V1f9vdQc75v9WoujcnlKszzpV6qePPC7igJJpu4QOI362BrWzJCYQbg4
Uzo1KYBhLFxl0TovAgMBAAGjgc8wgcwwHQYDVR0OBBYEFMbTvDyvE2KsRy9zPq/J
WOjovG+WMIGcBgNVHSMEgZQwgZGAFMbTvDyvE2KsRy9zPq/JWOjovG+WoW6kbDBq
MQswCQYDVQQGEwJ4eDELMAkGA1UECBMCeHgxCzAJBgNVBAcTAnh4MQswCQYDVQQK
EwJ4eDELMAkGA1UECxMCeHgxCzAJBgNVBAMTAnh4MRowGAYJKoZIhvcNAQkBFgt4
eHhAMTYzLmNvbYIJALV96mEtVF4EMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAASkC/1iwiALa2RU3YCxqZFEEsZZvQxikrDkDbFeoa6Tk49Fnb1f7FCW6
PTtY3HPWl5ygsMsSy0Fi3xp3jmuIwzJhcQ3tcK5gC99HWp6Kw37RL8WoB8GWFU0Q
4tHLOjBIxkZROPRhH+zMIrqUexv6fsb3NWKhnlfh1Mj5wQE4Ldo=
-----END CERTIFICATE-----
```

Private Key Format

When creating a server certificate or server SM certificate, you also need to upload the private key of the certificate. You can copy and paste the private key content or directly upload the private key in the required format.

The private keys of server SM certificates must meet the requirements as described below.

Private keys must be unencrypted and meet the following requirements:

- The value must be in PEM format.
  - The content must start with **-----BEGIN RSA PRIVATE KEY-----** and end with **-----END RSA PRIVATE KEY-----**.

- The content must start with **-----BEGIN EC PRIVATE KEY-----** and end with **-----END EC PRIVATE KEY-----**.

- There are no empty rows. Each row must contain 64 characters except the last row.

The following is an example:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDFPN9ojPndxSC4E1pqWQVKGHCFlXAAGBOxbGfSzXqzsoyacotu
eqMqXQbxrPSQFATeVmhZPNVEMdvcAMjYsV/mymtAwVqVA6q/OFdX/b3UHO+b/VqL
o3J5SrM86Veqnjzwu4oCSabuEDiN+tga1syQmEG4OFM6NSmAYSxcZdE6LwIDAQAB
AoGBAJvLzJCyIsCJcKHWL6onbSUtDtyFwPViD1QrVAtQYabF14g8CGUZG/9fgheu
TXPtTDcvu7cZdUArvgYW3I9F9IBb2lmF3a44xfiAKdDhzr4DK/vQhvHPuuTeZA41
r2zp8Cu+Bp40pSxmoAOK3B0/peZAka01Ju7c7ZChDWrxleHZAkEA/6dcaWHotfGS
eW5YLbSms3f0m0GH38nRl7oxyCW6yMIDkFHURVMBKW1OhrcuGo8u0nTMi5IH9gRg
5bH8XcujlQJBAMWBQgzCHyoSeryD3TFieXIFzgDBw6Ve5hyMjUtjvgdVKoxRPvpO
kclc39QHP6Dm2wrXXHEej+9RILxBZCVQNbMCQQC42i+Ut0nHvPuXN/UkXzomDHde
h1ySsOAO4H+8Y6OSI87l3HUrByCQ7stX1z3L0HofjHqV9Koy9emGTFLZEzSdAkB7
Ei6cUKKmztkYe3rr+RcATEmwAw3tEJOHmrW5ErApVZKr2TzLMQZ7WZpIPzQRCYnY
2ZZLDuZWFFG3vW+wKKktAkAaQ5GNzbwkRLpXF1FZFuNF7erxypzstbUmU/31b7tS
i5LmxTGKL/xRYtZEHjya4Ikkkgt40q1MrUsgIYbFYMf2
-----END RSA PRIVATE KEY-----
```

7.3 Converting Certificate Formats

Scenarios

ELB supports certificates only in PEM format. If you have a certificate in any other format, you must convert it to a PEM-encoded certificate. There are some common methods for converting a certificate from any other format to PEM.

From DER to PEM

The DER format is usually used on a Java platform.

Run the following command to convert the certificate format:

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

Run the following command to convert the private key format:

```
openssl rsa -inform DER -outform PEM -in privatekey.der -out privatekey.pem
```

From P7B to PEM

The P7B format is usually used by Windows Server and Tomcat.

Run the following command to convert the certificate format:

```
openssl pkcs7 -print_certs -in incertificate.p7b -out outcertificate.cer
```

From PFX to PEM

The PFX format is usually used by Windows Server.

Run the following command to convert the certificate format:

```
openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
```

Run the following command to convert the private key format:

```
openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
```

7.4 Adding, Modifying, or Deleting a Certificate

Scenarios

To enable authentication for securing data transmission over HTTPS, you can add certificates to your load balancers. You can also modify and delete certificates.

NOTE

- A certificate can be bound to only one type of load balancer. Ensure that you have selected the correct type.

Adding a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Click **Add Certificate**. In the **Add Certificate** dialog box, configure the parameters.
   - **Certificate Name**
   - **Certificate Type**
     - **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
     - **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.
     - **Server SM certificates**: To support Chinese cryptographic algorithms, two certificates that must be used together are required, one signing certificate and one encryption certificate.
       - **Signing certificate**: This certificate is used only for identity authentication. The public and private keys are generated and kept by the server, rather than the CA.
       - **Encryption certificate**: This certificate is used for key negotiation. The public and private keys are generated and kept by the CA.
   - **Certificate Content**: The content must be in PEM format. This parameter is mandatory when **Certificate Type** is set to **Server certificate** or **CA certificate**.

     Click **Upload** and select the certificate to be uploaded. Ensure that your browser is of the latest version.

The format of the certificate body is as follows:

```
-----BEGIN CERTIFICATE-----
Base64–encoded certificate
-----END CERTIFICATE-----
```

- **Private Key**: This parameter is mandatory when **Certificate Type** is set to **Server certificate**.

  Click **Upload** and select the private key to be uploaded. Ensure that your browser is of the latest version.

  The value must be an unencrypted private key. The private key must be in PEM format. The format is as follows:

  ```
  -----BEGIN PRIVATE KEY-----
  [key]
  -----END PRIVATE KEY-----
  ```

- **Certificate**: The content must be in PEM format. This parameter is mandatory when **Certificate Type** is set to **Server SM certificates**.

  Click **Upload** and select the certificate to be uploaded. Ensure that your browser is of the latest version.

  The format of the certificate body is as follows:

  ```
  -----BEGIN CERTIFICATE-----
  Base64–encoded certificate
  -----END CERTIFICATE-----
  ```

- **Signing Private Key**: This parameter is mandatory when **Certificate Type** is set to **Server SM certificates**.

  Click **Upload** and select the private key to be uploaded. Ensure that your browser is of the latest version.

  Private Key: This must be an unencrypted private key. The format is as follows:

  ```
  -----BEGIN PRIVATE KEY-----
  [key]
  -----END PRIVATE KEY-----
  ```

- **Certificate**: The content must be in PEM format. This parameter is mandatory when **Certificate Type** is set to **Server SM certificates**.

  Click **Upload** and select the certificate to be uploaded. Ensure that your browser is of the latest version.

  The format of the certificate body is as follows:

  ```
  -----BEGIN CERTIFICATE-----
  Base64–encoded certificate
  -----END CERTIFICATE-----
  ```

- Encryption private key: This parameter is mandatory when **Certificate Type** is set to **Server SM certificates**.

  Click **Upload** and select the private key to be uploaded. Ensure that your browser is of the latest version.

  Private Key: This must be an unencrypted private key. The format is as follows:

  ```
  -----BEGIN PRIVATE KEY-----
  [key]
  -----END PRIVATE KEY-----
  ```

NOTE

If there is a certificate chain, you need to configure the certificates in the following sequence: sub-certificate (server certificate), intermediate certificate, and root certificate. If the root certificate has been preset on the server and is not contained in the issued certificates, first configure the sub-certificate (server certificate) and then the intermediate certificate.

For example, if a CA issued a private key **private.key** and two certificates: a sub-certificate (server certificate) **server.cer** and an intermediate certificate **mid.crt**, paste the content of **server.cer** in the **Certificate** text box, press **Enter**, then paste the content of **mid.crt** in the **Certificate** text box, and paste the content of **private.key** in the **Private Key** text box to make the entire certificate chain take effect. The format of the certificate body in a certificate chain is as follows:

Certificate body

```
-----BEGIN CERTIFICATE-----
Content of the server certificate server.cer
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Content of the intermediate certificate mid.crt
-----END CERTIFICATE-----
```

Private key

```
-----BEGIN PRIVATE KEY-----
Content of the private key private.key
-----END PRIVATE KEY-----
```

– **Domain Name**

  If the created certificate will be used for SNI, you need to specify a domain name for each certificate, and the domain name must be the same as that in the certificate.

– **Description**

6. Click **OK**.

Modifying a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the certificate and click **Modify** in the **Operation** column.
6. Modify the parameters as required.
7. Confirm the information and click **OK**.

Deleting a Certificate

Only certificates that are not in use can be deleted.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.

3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the certificate and click **Delete** in the **Operation** column.
6. Click **Yes**.

7.5 Replacing the Certificate Bound to a Listener

Scenarios

You need to bind a certificate when you add an HTTPS listener to a load balancer. If the certificate used by a listener has expired or needs to be replaced due to other reasons, you can replace the certificate on the **Listeners** tab page.

If the certificate is also used by other services such as WAF, replace the certificate on all these services to prevent service unavailability.

NOTE

Replacing certificates and private keys does not affect your applications.

Prerequisites

You have created a certificate by following the instructions in **Adding a Certificate**.

Binding a Certificate

You can bind certificates when you add an HTTPS listener. For details, see **Adding an HTTPS Listener**.

Replacing a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the listener, and click **Edit** in **Operation** column.
6. Select a server certificate.
7. Click **OK** in the **Edit** dialog box.

7.6 Replacing the Certificate Bound to Different Listeners

Scenario

If the certificate that is bound to different listeners has expired or needs to be replaced due to other reasons, you can replace the certificate by modifying it on the **Certificates** page.

NOTE

Replacing the certificate and private keys does not affect your applications.

Constraints

- Only HTTPS listeners require certificates.
- The new certificate takes effect immediately.

Modifying a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ≡ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the certificate and click **Modify** in the **Operation** column.
6. Modify the parameters as required.
7. Confirm the information and click **OK**.

7.7 Querying Listeners by Certificate

Scenarios

You need to quickly view details of the listeners to which a certificate is bound.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ≡ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.

5. In the certificate list, click the listener name in the **Listener (Frontend Protocol/Port)** column to view its details.

   If there are more than 5 listeners, no listener is displayed in the **Listener (Frontend Protocol/Port)** column. Click **View All**. On the displayed page, click **Listeners**, locate the listener, and click its name to view it details.

8 Access Control

8.1 Access Control

Access control allows you to add a whitelist or blacklist to specify IP addresses that are allowed or denied to access a listener. A whitelist allows specified IP addresses to access the listener, while a blacklist denies access from specified IP addresses.

NOTICE

- Adding the whitelist or blacklist may cause risks.
  - Once the whitelist is set, only the IP addresses specified in the whitelist can access the listener.
  - Once the blacklist is set, the IP addresses specified in the blacklist cannot access the listener.
- Whitelists and blacklists do not conflict with inbound security group rules. Whitelists define the IP addresses that are allowed to access the listeners, while blacklists specify IP addresses that are denied to access the listeners. Inbound security group rules control access to backend servers by specifying the protocol, ports, and IP addresses.
- Access control does not restrict the ping command. You can still ping backend servers from the restricted IP addresses.
- Access control policies only take effect for new connections, but not for connections that have been established. If a whitelist is configured for a listener but IP addresses that are not in the whitelist can access the backend server associated with the listener, one possible reason is that a persistent connection is established between the client and the backend server. To deny IP addresses that are not in the whitelist from accessing the listener, the persistent connection between the client and the backend server needs to be disconnected.

Configuring Access Control

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. You can configure access control for a listener in either of the following ways:
   - On the **Listeners** page, locate the listener and click **Configure** in the **Access Control** column.
   - Click the name of the target listener. On the **Summary** page, click **Configure** on the right of **Access Control**.
6. In the displayed **Configure Access Control** dialog box, configure parameters as shown in **Table 8-1**.

Table 8-1 Parameter description

Parameter	Description	Example Value
Access Control	Specifies how access to the listener is controlled. Three options are available: ● **All IP addresses**: All IP addresses can access the listener. ● **Whitelist**: Only IP addresses in the IP address group can access the listener. ● **Blacklist**: IP addresses in the IP address group are not allowed to access the listener.	Blacklist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group Overview**.	ipGroup-b2

Parameter	Description	Example Value
Access Control	If you have set **Access Control** to **Whitelist** or **Blacklist**, you can enable or disable access control. ● Only after you enable access control, the whitelist or blacklist takes effect. ● If you disable access control, the whitelist or blacklist does not take effect.	N/A

7. Click **OK**.

8.2 Managing IP Address Groups

8.2.1 Creating an IP Address Group

IP Address Group Overview

An IP address group is a collection of IP addresses that you can use to manage IP addresses with the same security requirements or whose security requirements change frequently.

ELB allows you to use a whitelist or blacklist for access control. If you want to configure an **access control** policy, you must select an IP address group.

- **Whitelist**: Only IP addresses in the IP address group can access the listener. If the IP address group does not contain any IP address and you have selected whitelist for access control, no IP addresses can access the listener.
- **Blacklist**: IP addresses in the IP address group are denied to access the listener. If the IP address group does not contain any IP address and you have selected blacklist for access control, all IP addresses can access the listener.

Constraints

- By default, you can create a maximum of 50 IP address groups.
- An IP address group can be associated with a maximum of 50 listeners.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.

3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the displayed page, click **Create IP Address Group**.
6. Configure the parameters based on **Table 8-2**.

Table 8-2 Parameters required for creating an IP address group

Parameter	Description	Example Value
Name	Specifies the name of the IP address group.	ipGroup-01
IP Addresses	Specifies IPv4 or IPv6 IP addresses or CIDR blocks that are added to the whitelist or blacklist for access control. ● Each line must contain an IP address or a CIDR block and end with a line break. ● Each IP address or CIDR block can include a description with a vertical bar (|) separated, for example, 192.168.10.10 | ECS01. The description is 0 to 255 characters long and cannot contain angle brackets (<>). ● You can add a maximum of 300 IP addresses or CIDR blocks in each IP address group.	10.168.2.24 10.168.16.0/24
Description	Provides supplementary information about the IP address group.	-

7. Click **OK**.

8.2.2 Viewing the Details of an IP Address Group

Scenarios

This section describes how you can view information about an IP address group, including:

- Name, ID, and creation time
- IP addresses and CIDR blocks
- Associated listeners

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, click the name of the target address group.
6. Viewing basic information about the IP address group.
   a. On the **IP Addresses** tab, view the IP addresses.
   b. On the **Associated Listeners** tab, view the listeners associated with the IP address group.

8.2.3 Managing IP Addresses in an IP Address Group

After an IP address group is created, you can manage the IP addresses in an IP address group as required:

- **Adding IP Addresses**
- **Changing IP Addresses**
- **Deleting an IP Address**

Constraints

The IP addresses can be in the following formats:

- Each line must contain an IP address or a CIDR block and end with a line break.
- Each IP address or CIDR block can include a description with a vertical bar (|) separated, for example, 192.168.10.10 | ECS01. The description is 0 to 255 characters long and cannot contain angle brackets (<>).
- You can add a maximum of 300 IP addresses or CIDR blocks in each IP address group.

Adding IP Addresses

After an IP address group is created, you can add IP addresses to an IP address group.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.

5. On the **IP Address Groups** page, click the name of the target address group.
6. In the lower part of the displayed page, choose **IP Addresses** tab and click **Add IP Addresses**.
7. On the **Add IP Addresses** page, add IP addresses.
8. Click **OK**.

Changing IP Addresses

You can perform the following steps to change all IP addresses in an IP address group:

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, you can:
   a. Modify the basic information and change IP addresses of an IP address group:
      i. Locate the target address group, click **Modify** in the **Operation** column.
      ii. You can modify the name and description of an IP address group, and change all its IP addresses.
      iii. Click **OK**.
   b. Only change IP addresses:
      i. Click the name of the target IP address group.
      ii. In the lower part of the displayed page, choose **IP Addresses** tab and click **Change IP Address**.
      iii. Change IP addresses as you needed.
      iv. Click **OK**.

Deleting an IP Address

If you want to delete IP addresses in batches from an IP address group, see **Changing IP Addresses**.

To delete an IP address from an IP address group, perform the following operations:

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.

4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, click the name of the target address group.
6. In the IP address list, locate the IP address you want to delete and click **Delete** in the **Operation** column.

   A confirmation dialog box is displayed.
7. Confirm the information and click **Yes**.

8.2.4 Deleting an IP Address Group

Scenarios

If you no longer need an IP address group, you can delete it. This section describes how you can delete an IP address group.

Constraints

An IP address group that has been used for controlling access to a listener cannot be deleted. You can view the listeners associated with an IP address group by referring to **Viewing the Details of an IP Address Group**. For details about how to disassociate an IP address group from a listener, see **Configuring Access Control**.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the IP address group, and click **Delete** in the **Operation** column.
6. Click **Yes**.

9 TLS Security Policy

Scenarios

When you add HTTPS listeners, you can select appropriate security policies to improve security. A security policy is a combination of TLS protocols of different versions and supported cipher suites.

Adding a Security Policy

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.
7. Expand **Advanced Settings** and select a security policy.

   Table 9-1 shows the default security policies.

Table 9-1 Default security policies

Security Policy	Description	TLS Versions	Cipher Suites
TLS-1-0	TLS 1.0, TLS 1.1, and TLS 1.2 and supported cipher suites (high compatibility and moderate security)	TLS 1.2 TLS 1.1 TLS 1.0	● ECDHE-RSA-AES256-GCM-SHA384 ● ECDHE-RSA-AES128-GCM-SHA256 ● ECDHE-ECDSA-AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-GCM-SHA256 ● AES128-GCM-SHA256 ● AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-SHA256 ● ECDHE-RSA-AES128-SHA256 ● AES128-SHA256 ● AES256-SHA256 ● ECDHE-ECDSA-AES256-SHA384 ● ECDHE-RSA-AES256-SHA384 ● ECDHE-ECDSA-AES128-SHA ● ECDHE-RSA-AES128-SHA ● ECDHE-RSA-AES256-SHA ● ECDHE-ECDSA-AES256-SHA ● AES128-SHA ● AES256-SHA
TLS-1-1	TLS 1.1 and TLS 1.2 and supported cipher suites (moderate compatibility and moderate security)	TLS 1.2 TLS 1.1	
TLS-1-2	TLS 1.2 and supported cipher suites (moderate compatibility and high security)	TLS 1.2	

Security Policy	Description	TLS Versions	Cipher Suites
TLS-1-2-Strict	Strict TLS 1.2 and supported cipher suites (low compatibility and ultra-high security)	TLS 1.2	● ECDHE-RSA-AES256-GCM-SHA384 ● ECDHE-RSA-AES128-GCM-SHA256 ● ECDHE-ECDSA-AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-GCM-SHA256 ● AES128-GCM-SHA256 ● AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-SHA256 ● ECDHE-RSA-AES128-SHA256 ● AES128-SHA256 ● AES256-SHA256 ● ECDHE-ECDSA-AES256-SHA384 ● ECDHE-RSA-AES256-SHA384

Security Policy	Description	TLS Versions	Cipher Suites
TLS-1-0-WITH-1-3	TLS 1.0 and later, and supported cipher suites (ultra-high compatibility and low security)	TLS 1.3 TLS 1.2 TLS 1.1 TLS 1.0	● ECDHE-RSA-AES256-GCM-SHA384 ● ECDHE-RSA-AES128-GCM-SHA256 ● ECDHE-ECDSA-AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-GCM-SHA256 ● AES128-GCM-SHA256 ● AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-SHA256 ● ECDHE-RSA-AES128-SHA256 ● AES128-SHA256 ● AES256-SHA256 ● ECDHE-ECDSA-AES256-SHA384 ● ECDHE-RSA-AES256-SHA384 ● ECDHE-ECDSA-AES128-SHA ● ECDHE-RSA-AES128-SHA ● ECDHE-RSA-AES256-SHA ● ECDHE-ECDSA-AES256-SHA ● AES128-SHA ● AES256-SHA ● TLS_AES_128_GCM_SHA256 ● TLS_AES_256_GCM_SHA384 ● TLS_CHACHA20_POLY1305_SHA256 ● TLS_AES_128_CCM_SHA256 ● TLS_AES_128_CCM_8_SHA256

Security Policy	Description	TLS Versions	Cipher Suites
TLS-1-2-FS-WITH-1-3	TLS 1.2 and later, and supported forward secrecy cipher suites (high compatibility and ultra-high security)	TLS 1.3 TLS 1.2	● ECDHE-RSA-AES256-GCM-SHA384 ● ECDHE-RSA-AES128-GCM-SHA256 ● ECDHE-ECDSA-AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-GCM-SHA256 ● ECDHE-ECDSA-AES128-SHA256 ● ECDHE-RSA-AES128-SHA256 ● ECDHE-ECDSA-AES256-SHA384 ● ECDHE-RSA-AES256-SHA384 ● TLS_AES_128_GCM_SHA256 ● TLS_AES_256_GCM_SHA384 ● TLS_CHACHA20_POLY1305_SHA256 ● TLS_AES_128_CCM_SHA256 ● TLS_AES_128_CCM_8_SHA256

Security Policy	Description	TLS Versions	Cipher Suites
hybrid-policy-1-0	TLS 1.1 and TLS 1.2 and supported cipher suites (moderate compatibility and moderate security)	TLS 1.2 TLS 1.1	• ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • AES128-GCM-SHA256 • AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • AES128-SHA256 • AES256-SHA256 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-SHA • AES256-SHA • ECC-SM4-SM3 • ECDHE-SM4-SM3

NOTE

- This table lists the cipher suites supported by ELB. Generally, clients also support multiple cipher suites. In actual use, the cipher suites supported by ELB and clients are used, and the cipher suites supported by ELB take precedence.

8. Click **OK**.

Differences Between Security Policies

Table 9-2 Differences between the security policies

Security Policy	TLS-1-0	TLS-1-1	TLS-1-2	TLS-1-2-Strict
TLS versions				

Security Policy	TLS-1-0	TLS-1-1	TLS-1-2	TLS-1-2-Strict
TLS 1.3	-	-	-	-
TLS 1.2	√	√	√	√
TLS 1.1	√	√	-	-
TLS 1.0	√	-	-	-
Cipher suites				
EDHE-RSA-AES128-GCM-SHA256	√	√	√	√
ECDHE-RSA-AES256-GCM-SHA384	√	√	√	√
ECDHE-RSA-AES128-SHA256	√	√	√	√
ECDHE-RSA-AES256-SHA384	√	√	√	√
AES128-GCM-SHA256	√	√	√	√
AES256-GCM-SHA384	√	√	√	√
AES128-SHA256	√	√	√	√
AES256-SHA256	√	√	√	√
ECDHE-RSA-AES128-SHA	√	√	√	-
ECDHE-RSA-AES256-SHA	√	√	√	-
AES128-SHA	√	√	√	-
AES256-SHA	√	√	√	-
ECDHE-ECDSA-AES128-GCM-SHA256	√	√	√	√
ECDHE-ECDSA-AES128-SHA256	√	√	√	√
ECDHE-ECDSA-AES128-SHA	√	√	√	-
ECDHE-ECDSA-AES256-GCM-SHA384	√	√	√	√
ECDHE-ECDSA-AES256-SHA384	√	√	√	√
ECDHE-ECDSA-AES256-SHA	√	√	√	-

Changing a Security Policy

When you change a security policy, ensure that the security group containing backend servers allows traffic from 100.125.0.0/16 to backend servers and allows ICMP packets for UDP health checks. Otherwise, backend servers will be considered unhealthy, and routing will be affected.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Summary** tab page, click **Edit** on the top right.
7. In the **Modify Listener** dialog box, expand **Advanced Settings** and change the security policy.
8. Click **OK**.

10 Access Logging

Scenarios

ELB logs HTTP and HTTPS requests received by load balancers, including the time when the request was sent, client IP address, request path, and server response. To enable access logging, you need to interconnect ELB with LTS and create a log group and a log stream on the LTS console.

NOTE

ELB displays operations data, such as access logs, on the LTS console. Do not transmit private or sensitive data through fields in access logs. Encrypt your sensitive data if necessary.

Configuring LTS

To view access logs, you first need to configure LTS by following the instructions in the *Log Tank Service User Guide*.

1. Create a log group.
   a. Log in to the management console.
   b. In the upper left corner of the page, click and select the desired region and project.
   c. Click ☰ in the upper left corner and **Management & Deployment** > **Log Tank Service**.
   d. In the navigation pane on the left, choose **Log Management**.
   e. Click **Create Log Group**. In the displayed dialog box, enter a name for the log group.

      Set **Log Retention Duration** as required.
   f. Click **OK**.
2. Create a log stream.
   a. On the LTS console, click ⌄ on the left of a log group name.
   b. Click **Create Log Stream**. In the displayed dialog box, enter a name for the log stream.

c. Click **OK**.

Configuring Access Logging

Configure access logging on the ELB console.

1. Hover on ☰ in the upper left corner to display **Service List** and choose **Network** > **Elastic Load Balance**.
2. Locate the load balancer and click its name.
3. Under **Access Logs**, click **Configure Access Logging**.
4. Enable access logging and select the log group and log stream you created.
5. Click **OK**.

Viewing Access Logs

After you enable access logging, you can obtain details about the requests sent to your load balancer.

There are two ways for you to view access logs.

- On the ELB console, click the name of the load balancer and click **Access Logs** to view logs.
- (Recommended) On the LTS console, click the name of the corresponding log topic. On the displayed page, click **Real-Time Logs**

The following is an example log. For details about the fields in the log, see **Table 10-1**. The log format cannot be modified.

```
$msec $access_log_topic_id [$time_iso8601] $log_ver $remote_addr:$remote_port $status
"$request_method $scheme://$host$router_request_uri $server_protocol" $request_length $bytes_sent
$body_bytes_sent $request_time "$upstream_status" "$upstream_connect_time" "$upstream_header_time"
"$upstream_response_time" "$upstream_addr" "$http_user_agent" "$http_referer" "$http_x_forwarded_for"
$lb_name $listener_name $listener_id
$pool_name "$member_name" $tenant_id $eip_address:$eip_port "$upstream_addr_priv" $certificate_id
$ssl_protocol $ssl_cipher $sni_domain_name $tcpinfo_rtt $self_defined_header
```

Table 10-1 Parameter description

Parameter	Description	Description	Example Value
msec	Time in seconds with a millisecond resolution	Floating-point data	1530153091.868
access_log_topic_id	Log stream ID	UUID	04465dfa-640f-4567-8b58-45c9f8bbc23f
time_iso8601	Local time in the ISO 8601 standard format	-	2018-06-28T10:31:31+08:00
log_ver	Log format version	Fixed value: **elb_01**	elb_01

Parameter	Description	Description	Example Value
remote_addr: remote_port	IP address and port number of the client	Records the IP address and port of the client.	10.184.30.170:59605
status	HTTP status code	Records the request status code.	200
request_method scheme:// host request_uri server_protocol	*Request method Protocol://Host name: Request URI Request protocol*	● **request_method**: request method ● **scheme**: HTTP or HTTPS ● **host**: host name, which can be a domain name or an IP address ● **request_uri**: indicates the native URI initiated by the browser without any modification does not include the protocol and host name.	POST https://setting1.hicloud.com/AccountServer/IUserInfoMng/stAuth?Version=26400&cVersion=ID_SDK_2.6.4.300
request_length	Length of the request received from the client, including the header and body	Integer	295
bytes_sent	Number of bytes sent to the client	Integer	58470080
body_bytes_sent	Number of bytes sent to the client (excluding the response header)	Integer	58469792

Parameter	Description	Description	Example Value
request_time	Request processing time in seconds from the time when the load balancer receives the first request packet from the client to the time when the load balancer sends the response packet	Floating-point data	499.769
upstream_status	Response status code returned by the backend server ● When the load balancer attempts to retry a request, there will be multiple response status codes. ● If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.	HTTP status code returned by the backend server to the load balancer	200 or "-, 200", or "502, 502: 200", or "502:"

Parameter	Description	Description	Example Value
upstream_connect_time	Time taken to establish a connection with the backend server, in seconds, with a millisecond resolution ● When the load balancer attempts to retry a request, there will be multiple connection times. ● If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.	Floating-point data	0.008, "-, 0.008", "0.008, 0.005: 0.004", or "0.008:"
upstream_header_time	Time taken to receive the response header from the backend server, in seconds, with a millisecond resolution ● When the load balancer attempts to retry a request, there will be multiple response times. ● If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.	Floating-point data	0.008, "-, 0.008", "0.008, 0.005: 0.004", or "0.008:"

Parameter	Description	Description	Example Value
upstream_response_time	Time taken to receive the response from the backend server, in seconds, with a millisecond resolution ● When the load balancer attempts to retry a request, there will be multiple response times. ● If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.	Floating-point data	0.008, "-, 0.008", "0.008, 0.005: 0.004", or "0.008:"
upstream_addr	IP address and port number of the backend server. There may be multiple values separated by commas and spaces, and each value is in the format of {*IP address*}:{*Port number*} or -. This parameter is only available for dedicated load balancers.	IP address and port number	-, or 192.168.1.2:8080
http_user_agent	**http_user_agent** in the request header received by the load balancer, indicating the system model and browser information of the client	Records the browser-related information.	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36

Parameter	Description	Description	Example Value
http_referer	**http_referer** in the request header received by the load balancer, indicating the page link of the request	Request for a page link	http://10.154.197.90/
http_x_forwarded_for	**http_x_forwarded_for** in the request header received by the load balancer, indicating the IP address of the proxy server that the request passes through	IP address	10.154.197.90
lb_name	Load balancer name in the format of **loadbalancer**_*Load balancer ID*	String	loadbalancer_789424af-3fd2-4292-8c62-2a2dd7005175
listener_name	Listener name in the format of **listener**_*Listener ID*	String	listener_fde03b66-f960-440e-954a-0be8b2b75093
listener_id	Listener ID (This field can be ignored.)	String	-
pool_name	Backend server group name in the format of **pool**_*backend server group ID*	String	pool_066a5dc5-a3e4-4ea1-99f1-2a5716b681f6
member_name	Backend server name in the format of **member**_*server ID* (this field is not supported yet). There may be multiple values separated by commas and spaces, and each value is a member ID (**member_id**) or **-**.	String	member_47b07465-075a-4d2f-8ce9-0b9f39bff160 (There may be multiple values separated by commas and spaces, and each value is a member ID (**member_id**) or -.)
tenant_id	Tenant ID	String	04dd36f921000fe20f95c00bba986340

Parameter	Description	Description	Example Value
eip_address:eip_port	EIP of the load balancer and frontend port that were set when the listener was added	EIP of the load balancer and frontend port that were set when the listener was added	4.17.12.248:443
upstream_addr_priv	IP address and port number of the backend server. There may be multiple values separated by commas and spaces, and each value is in the format of {*IP address*}:{*Port number*} or -. This parameter is only available for dedicated load balancers.	IP address and port number	-, 192.168.1.2:8080 (There may be multiple values by commas and spaces, and each value is in the format of {*IP address*}:{*Port number*} or -.)
certificate_id	[HTTPS listener] Certificate ID used for establishing an SSL connection This field is not supported yet.	String	17b03b19-b2cc-454e-921b-4d187cce31dc
ssl_protocol	[HTTPS listener] Protocol used for establishing an SSL connection For a non-HTTPS listener, a hyphen (-) is displayed as a null value for this field.	String	TLS 1.2
ssl_cipher	[HTTPS listener] Cipher suite used for establishing an SSL connection For a non-HTTPS listener, a hyphen (-) is displayed as a null value for this field.	String	ECDHE-RSA-AES256-GCM-SHA384

Parameter	Description	Description	Example Value
sni_domain_name	[HTTPS listener] SNI domain name provided by the client during SSL handshake For a non-HTTPS listener, a hyphen (-) is displayed as a null value for this field.	String	www.test.com
tcpinfo_rtt	TCP Round Trip Time (RTT) between the load balancer and client in microseconds	Integer	39032
self_defined_header	This field is reserved. The default value is -.	String	-

Example Log

```
1644819836.370 eb11c5a9-93a7-4c48-80fc-03f61f638595 [2022-02-14T14:23:56+08:00] elb_01
192.168.1.1:888 200 "POST https://www.test.com/example /HTTP/1.1" 1411 251 3 0.011 "200" "0.000"
"0.011" "0.011" "100.64.0.129:8080" "okhttp/3.13.1" "-" "-"
loadbalancer_295a7eee-9999-46ed-9fad-32a62ff0a687 listener_20679192-8888-4e62-a814-a2f870f62148
3333fd44fe3b42cbaa1dc2c641994d90 pool_89547549-6666-446e-9dbc-e3a551034c46 "-"
f2bc165ad9b4483a9b17762da851bbbb 121.64.212.1:443 "10.1.1.2:8080" - TLSv1.2 ECDHE-RSA-AES256-
GCM-SHA384 www.test.com 56704 -
```

The following table describes the fields in the log.

Table 10-2 Fields in the log

Field	Example Value
msec	1644819836.370
access_log_topic_id	eb11c5a9-93a7-4c48-80fc-03f61f638595
time_iso8601	[2022-02-14T14:23:56+08:00]
log_ver	elb_01
remote_addr: remote_port	192.168.1.1:888
status	200
request_method scheme://host request_uri server_protocol	"POST https://www.test.com/example/1 HTTP/1.1"
request_length	1411

Field	Example Value
bytes_sent	251
body_bytes_sent	3
request_time	0.011
upstream_status	"200"
upstream_connect_time	"0.000"
upstream_header_time	"0.011"
upstream_response_time	"0.011"
upstream_addr	"100.64.0.129:8080"
http_user_agent	"okhttp/3.13.1"
http_referer	"-"
http_x_forwarded_for	"-"
lb_name	loadbalancer_295a7eee-9999-46ed-9fad-32a62ff0a687
listener_name	listener_20679192-8888-4e62-a814-a2f870f62148
listener_id	3333fd44fe3b42cbaa1dc2c641994d90
pool_name	pool_89547549-6666-446e-9dbc-e3a551034c46
member_name	"-"
tenant_id	f2bc165ad9b4483a9b17762da851bbbb
eip_address:eip_port	121.64.212.1:443
upstream_addr_priv	"10.1.1.2:8080"
certificate_id	-
ssl_protocol	TLSv1.2
ssl_cipher	ECDHE-RSA-AES256-GCM-SHA384
sni_domain_name	www.test.com
tcpinfo_rtt	56704
self_defined_header	-

Log analysis:

At 14:23:56 GMT+08:00 on Feb 14, 2022, the load balancer receives an HTTP/1.1 POST request from a client whose IP address and port number are 192.168.1.1 and 888, then routes the request to a backend server whose IP address and port number are 100.64.0.129 and 8080, and finally returns 200 OK to the client after receiving the status code from the backend server.

Analysis results:

The backend server responds to the request normally.

Configuring Log Transfer

If you want to analyze access logs later, transfer the logs to OBS or Data Ingestion Service (DIS) for storage.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click ☰ in the upper left corner and **Management & Deployment** > **Log Tank Service**.
4. In the navigation pane on the left, choose **Log Transfer**.
5. On the **Log Transfer** page, click **Configure Log Transfer** in the upper right corner.

1. Configure the parameters. For details, see the *Log Tank Service User Guide*.

11 Monitoring

11.1 Monitoring Metrics

Overview

This section describes the namespace, the metrics that can be monitored by Cloud Eye, and dimensions of these metrics. You can view the metrics reported by ELB and the generated alarms on the Cloud Eye console. For details, see **Viewing Metrics**.

Namespace

SYS.ELB

Metrics

Table 11-1 Metrics supported by ELB

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object Unit: N/A	≥ 0	● Load balancer ● Listener	1 minute
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: N/A	≥ 0		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: N/A	≥ 0		
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second Unit: Count/s	≥ 0/second		
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second Unit: Packet/s	≥ 0/second		
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second Unit: Packet/s	≥ 0/second		
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet per second Unit: byte/s	≥ 0 bytes/s		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second Unit: byte/s	≥ 0 bytes/s		
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object Unit: N/A	≥ 0	● Load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object Unit: N/A	≥ 0		
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second Unit: Query/s	≥ 0 query/s	● Load balancer ● Listener	1 minute
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the monitored object Unit: Count/s	≥ 0/ second	● Load balancer ● Listener	1 minute
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the monitored object Unit: Count/s	≥ 0/ second	● Load balancer ● Listener	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the monitored object Unit: Count/s	≥ 0/ second		
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the monitored object Unit: Count/s	≥ 0/ second		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the monitored object except 2xx, 3xx, 4xx, and 5xx status codes Unit: Count/s	≥ 0/second		
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0 ms		

Dimensions

Key	Value
lbaas_instance_id	Load balancer ID
lbaas_listener_id	ID of a listener added to a load balancer
lbaas_pool_id	ID of the backend server group

11.2 Setting an Alarm Rule

You can add, modify, and delete alarm rules. For details, see the Cloud Eye User Guide.

11.2.1 Creating an Alarm Rule

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click ☰ in the upper left corner and choose **Management & Deployment** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.
5. On the displayed **Alarm Rules** page, click **Create Alarm Rule**.

   The following describes how to create an alarm rule for a load balancer.

   a. **Resource Type**: Select **Elastic Load Balance**.
   b. For **Dimension**, select **Load Balancers** or **Listeners**. In the following operations, a load balancer is used as an example.
   c. Select a load balancer that you want to monitor.
   d. Configure other parameters as required and then click **Create**.

   Once the alarm rule is created and the notification function has been enabled, the system automatically sends you a notification when an alarm is generated.

   NOTE

   For more information about alarm rules of load balancers and listeners, see the *Cloud Eye User Guide*.

11.2.2 Modifying an Alarm Rule

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.

3. Click ☰ in the upper left corner and choose **Management & Deployment** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.
5. On the **Alarm Rules** page, locate the alarm rule and click **Modify** in the **Operation** column.
   a. On the **Modify Alarm Rule** page, modify the parameters.
   b. Set other parameters as required and then click **Modify**.

      Once the alarm rule is set and you have enabled the notification function, the system automatically sends you a notification when an alarm is generated.

      NOTE

      For more information about alarm rules of load balancers and listeners, see the *Cloud Eye User Guide*.

11.3 Viewing Metrics

Scenarios

Cloud Eye provided by the cloud service platform monitors the running statuses of load balancers.

You can view the metrics of each load balancer on the Cloud Eye console.

The transmission of monitoring data takes a while, so the status of each load balancer displayed on the Cloud Eye dashboard is not its real-time status. For a newly created load balancer or a newly added listener, you need to wait for about 5 minutes to 10 minutes before you can view its metrics.

Prerequisites

- The load balancer is running properly.

  If backend servers are stopped, faulty, or deleted, no monitoring data is displayed.

  NOTE

  Cloud Eye stops monitoring a load balancer and removes it from the monitored object list if its backend servers have been deleted or are in stopped or faulty state for over 24 hours. However, the configured alarm rules will not be automatically deleted.
- You have interconnected ELB with Cloud Eye and configured an alarm rule for the load balancer on the Cloud Eye console.

  Without alarm rules, there is no monitoring data. For details, see **Setting an Alarm Rule**.
- If an IAM user wants to view the ELB monitoring data on the Cloud Eye console, the IAM user must be granted the **ELB Administrator** permission. Otherwise, the IAM user cannot view all monitoring data.

Viewing Monitoring Metrics on the Cloud Eye Console

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click ☰ in the upper left corner and choose **Management & Deployment** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Elastic Load Balance**.
5. On the **Cloud Service Monitoring** page, click the name of the load balancer. Alternatively, locate the load balancer and click **View Metric** in the **Operation** column.
6. Select the time period during which you want to view metrics. You can select a system-defined time period (for example, last 1 hour) or specify a time period.
7. Click **Select Metric** in the upper right corner and select the metrics to be viewed.

NOTE

For more details, see the *Cloud Eye User Guide*.

12 Quotas

What Is Quota?

Quotas can limit the number or amount of resources available to users, such as the maximum number of ECSs or EVS disks that can be created.

If the existing resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. In the upper right corner of the page, click 📊 .

   The **Service Quota** page is displayed.
3. View the used and total quota of each type of resources on the displayed page.

   If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. If you need to adjust a quota, contact the operations administrator.

Before contacting the operations administrator, make sure that the following information has been obtained:

- Account name, which can be obtained by performing the following operations:

  Log in to the management console using the cloud account, click the username in the upper right corner, select **My Credentials** from the drop-down list, and obtain the account name on the **My Credentials** page.
- Quota information, which includes service name, quota type, and required quota

13 FAQ

13.1 Popular Questions

- How Can I Transfer the IP Address of a Client?
- How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?
- What Types of Sticky Sessions Does ELB Support?
- How Is WebSocket Used?
- How Do I Check If Sticky Sessions Failed to Take Effect?
- What Are the Relationships Between Load Balancing Algorithms and Sticky Session Types?
- How Does ELB Distribute Traffic?

13.2 ELB Functionality

13.2.1 Can ELB Be Used Separately?

ELB cannot be used alone.

ELB distributes incoming traffic to multiple backend servers based on the forwarding policy to balance workloads. So, it can expand external service capabilities of your applications and eliminate single points of failure (SPOFs) to improve service availability. To use a load balancer, you must associated backend servers (such as ECSs) with it.

13.2.2 Does ELB Support Persistent Connections?

Yes.

The connections between the client and load balancer are persistent connections. After a TCP persistent connection is established, the client continuously sends HTTP requests to the load balancer until the connection times out. The reuse of TCP connections reduces the costs for a large number of short connections.

13.2.3 Does ELB Support FTP on Backend Servers?

ELB does not support File Transfer Protocol (FTP), but supports Secure File Transfer Protocol (SFTP) on backend servers.

13.2.4 Is an EIP Assigned Exclusively to a Load Balancer?

During the lifecycle of a load balancer, the EIP can be unbound from the load balancer. If the EIP is unbound, the load balancer becomes a private network load balancer, and the EIP can be bound to other resources.

13.2.5 How Many Load Balancers and Listeners Can I Have?

By default, each account can have up to 50 load balancers and 100 listeners. If you need more load balancers or listeners, apply to increase your quotas.

All load balancers in your account share the same quota for listeners.

13.2.6 What Types of APIs Does ELB Provide? What Are Permissions of ELB?

ELB supports the following policies:

Table 13-1 ELB policies

Policy Type	Policy Name	Description
RBAC policy	ELB Administrator	Has all permissions on ELB Before assigning the RBAC policy to a user group, check whether the user group has a dependent policy. If yes, set the dependent permission to make the RBAC policy take effect.
Fine-grained policy	ELB FullAccess	Has all permissions on ELB. If this function is not enabled, you cannot assign a fine-grained policy to a user group.
	ELB ReadOnlyAccess	Has the read-only permission on 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

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

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Configuring a health check	Supported	Not supported	Supported
Querying a health check	Supported	Supported	Supported
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

For details about fine-grained permissions, see the *Elastic Load Balance API Reference*.

13.2.7 Can I Adjust the Number of Backend Servers When a Load Balancer is Running?

You can adjust the number of backend servers associated with a load balancer at any time. You can also change the type of backend servers according to your service needs. To ensure service stability, ensure that health checks are normal and that at least one healthy backend server is associated with the load balancer.

13.2.8 Can Backend Servers Run Different OSs?

Yes.

ELB does not restrict OSs of backend servers as long as applications on these servers are the same and the data is consistent. However, it is recommended that you install the same OS on backend servers to simplify management.

13.2.9 Can I Configure Different Backend Ports for a Load Balancer?

Yes. You can configure different backend ports for backend servers associated with a load balancer.

13.2.10 Can ELB Be Used Across Accounts or VPCs?

- For dedicated load balancers, you can add servers in a VPC connected using a VPC peering connection, in a VPC 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 their IP addresses. For details, see .

13.2.11 Can Backend Servers Access the Ports of a Load Balancer?

No. Backend servers cannot access the ports of the load balancer they are associated with.

13.2.12 Can Both the Listener and Backend Server Group Use HTTPS?

Dedicated load balancers support this function.

You can select HTTPS as the listener's protocol and the backend server group's protocol. For details about how to add a listener, see section "Adding an HTTPS Listener" in the *Elastic Load Balance User Guide*.

13.2.13 Can I Change the VPC and Subnet for My Load Balancer?

You can change the subnet but not the VPC for your dedicated load balancers.

13.3 Load Balancers

13.3.1 How Does ELB Distribute Traffic?

ELB uses FullNAT to forward the incoming traffic. For load balancing at Layer 4, LVS forwards the incoming traffic to backend servers directly. For load balancing at Layer 7, LVS forwards the incoming traffic to Nginx, which then forwards the traffic to backend servers.

NOTE

In FullNAT, LVS translates source IP addresses and destination IP addresses of the clients.

Figure 13-1 Load balancing at Layer 4

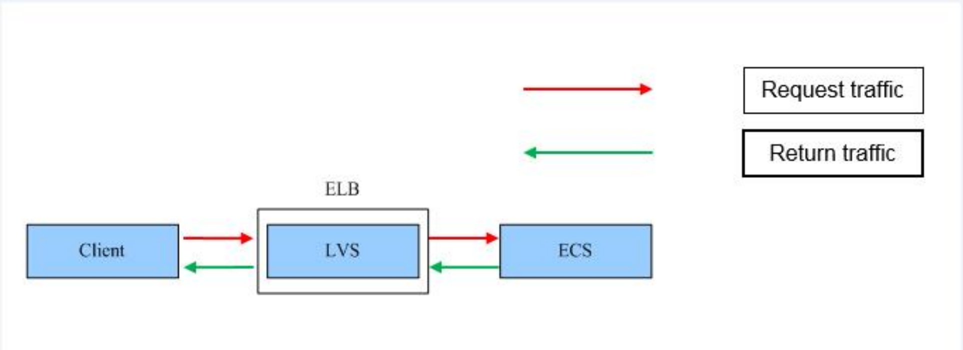
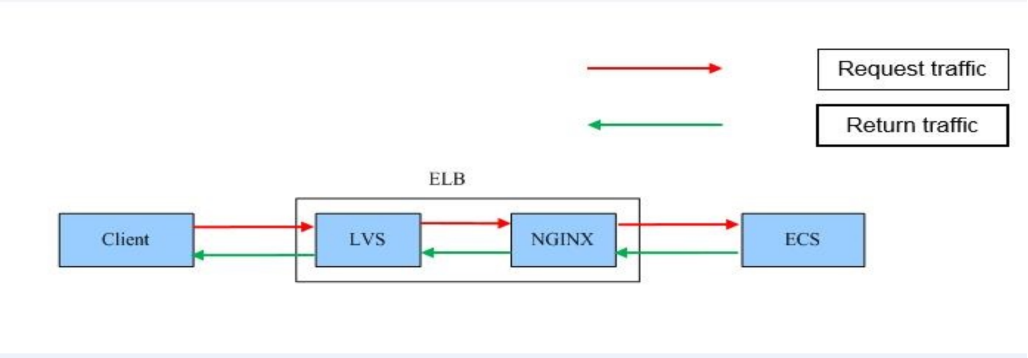


Figure 13-2 Load balancing at Layer 7



13.3.2 How Can I Access a Load Balancer Across VPCs?

VPC Peering can help you achieve this. For example, if another user has created load balancer ELB01 in VPC01, and you are in VPC02 and want to access ELB01, you just need to set up a VPC peering connection between VPC01 and VPC02 and add a route for the connection.

13.3.3 Do I Need to Configure EIP Bandwidth for My Load Balancers?

If you use a load balancer on a private network, you do not need to configure EIP bandwidth. You only need to bind an EIP and configure bandwidth if you are using a load balancer on a public network.

13.3.4 Can I Bind Multiple EIPs to a Load Balancer?

No.

- If you want to use the load balancer on a public network, you can only bind one EIP to the load balancer to receive requests from the Internet.
- If you want to use the load balancer in a VPC, bind a private IP address. To route requests from a different VPC, you need to create a VPC peering connection between the VPC where the load balancer works and the other VPC. For details, see section "Creating a VPC Peering Connection with Another VPC in Your Account" in the *Virtual Private Cloud User Guide*.

13.3.5 Why Multiple IP Addresses Are Required When I Create or Enable a Dedicated Load Balancer?

These IP addresses are used by underlying resources.

Generally, 2 IP addresses are required for creating a load balancer in a single AZ, and 6 IP addresses are required for creating a load balancer with IP as a backend enabled. If you create a load balancer in multiple AZs, more IP addresses will be required. There is an algorithm to determine how many IP addresses are required.

13.3.6 Why Are Requests from the Same IP Address Routed to Different Backend Servers When the Load Balancing Algorithm Is Source IP Hash?

One possible cause is that the backend server receiving requests from the client has become unhealthy. The source IP hash algorithm uses the source IP address of each request as a hashing key to route traffic from a particular client to the same backend server, as long as it is available. This allows requests from different clients to be routed based on their source IP addresses and ensures that a given client is always directed to the same backend server.

However, if a backend server become unhealthy and then recovers, ELB will generate a new hash key based on the source IP address of the request and numbers the backend server. As a result, requests from the same IP address are routed to different backend servers.

13.3.7 Can Backend Servers Access the Internet Using the EIP of the Load Balancer?

No.

The load balancer uses the EIP to receive requests from the Internet and routes the requests to backend servers over a private network.

If you want the backend servers to access the Internet or provide Internet-accessible services directly, you can bind an EIP to each backend server. You can also configure a NAT gateway for the backend servers so that they can share an EIP to access the Internet.

13.3.8 Will Traffic Routing Be Interrupted If the Load Balancing Algorithm Is Changed?

No. If the load balancing algorithm is changed, established connections will not be affected. Therefore, traffic routing will not be interrupted.

13.3.9 What Is the Difference Between the Bandwidth Included in Each Specification of a Dedicated Load Balancer and the Bandwidth of an EIP?

The bandwidth included in the specifications of dedicated load balancers is the upper limit of the inbound or the outbound traffic. The bandwidth of the EIP bound to the load balancer is the limit for traffic required by the clients to access the load balancer.

13.4 Listeners

13.4.1 What Are the Relationships Between Load Balancing Algorithms and Sticky Session Types?

Sticky sessions ensure that requests from the same client are routed to the same backend server. **Table 13-3** lists the types of sticky sessions.

Table 13-3 Sticky sessions supported by dedicated load balancers

Load Balancing Algorithm	Sticky Session Type	Layer 4 (TCP/UDP)	Layer 7 (HTTP/HTTPS)
Weighted round robin	Source IP address	Supported	Not supported
	Load balancer cookie	N/A	Supported
	Application cookie	N/A	Not supported
Weighted least connections	Source IP address	Not supportedSupported	Not supported
	Load balancer cookie	N/A	Not supportedSupported
	Application cookie	N/A	Not supportedSupported
Source IP hash	Source IP address	N/A	Not supported
	Load balancer cookie	N/A	Not supported
	Application cookie	N/A	Not supported

Generally, the weighted round robin algorithm is recommended. Sticky sessions at Layer 4 use source IP addresses to main sessions, and sticky sessions at Layer 7 use load balancer cookies.

13.4.2 Can I Bind Multiple Certificates to a Listener?

You can configure multiple certificates for an HTTPS listener by enabling SNI so that different certificates can be used for authentication based on the domain names of the requests.

For details, see **SNI Certificate**.

13.4.3 Will ELB Stop Distributing Traffic Immediately After a Listener Is Deleted?

- If a TCP or UDP listener is deleted, the load balancer immediately stops routing traffic because the client uses short connections to communicate with the load balancer.
- If an HTTP or HTTPS listener is deleted, persistent connections that have been established between the client and the load balancer will be kept alive until they time out, and therefore request routing is not affected. After the connections time out, the client stops sending requests over these connections. The default timeout duration is 300s.

  NOTE

  The duration for which persistent connections are kept alive is called idle timeout, and this takes effect only for persistent connections established between the client and load balancer.

13.4.4 Does ELB Have Restrictions on the File Upload Speed and Size?

- ELB has no restrictions on the file upload speed on the clients. However, the bandwidth may limit the upload speed.
- For HTTP or HTTPS listeners, the maximum file size is 10 GB. However, TCP or UDP listeners have no limit on the file size.

13.4.5 Can Multiple Load Balancers Route Requests to One Backend Server?

Yes. This is supported as long as the load balancers are in the same subnet as the backend server.

13.4.6 How Is WebSocket Used?

For HTTP listeners, unencrypted WebSocket (ws://) is supported by default. For HTTPS listeners, encrypted WebSocket (wss://) is supported by default.

13.4.7 Why Can't I Select the Target Backend Server Group When Adding or Modifying a Listener?

The backend server group's protocol (backend protocol) you want to select is not supported by the listener protocol (frontend protocol). There are some constraints on the backend protocol when you associate a backend server group with a listener.

Table 13-4 Frontend and backend protocols of dedicated load balancers

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	UDP/QUIC

Frontend Protocol	Backend Protocol
HTTP	HTTP
HTTPS	HTTP/HTTPS

Table 13-5 Frontend and backend protocols of shared load balancers

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	UDP
HTTP	HTTP
HTTPS	HTTP

13.4.8 Why Cannot I Add a Listener to a Dedicated Load Balancer?

If you select either network load balancing (TCP/UDP) or application load balancing (HTTP/HTTPS) when creating the load balancer, you can only add listeners of the matched protocol.

The load balancing type cannot be changed after being selected. For example, if you have selected network load balancing during load balancer creation, you cannot change it to application load balancing and you cannot add HTTP or HTTPS listeners.

Table 13-6 Protocols and load balancing types

Load Balancing Type	Protocol	Listener Types
Network load balancing	TCP/UDP	TCP and UDP listeners
Application load balancing	HTTP/HTTPS	HTTP and HTTPS listeners

13.5 Backend Servers

13.5.1 Why Is the Interval at Which Backend Servers Receive Health Check Packets Different from What I Have Configured?

Each LVS node and Nginx node in the ELB system send detection packets to backend servers at the health check interval that you have specified for the backend server group.

During this period, backend servers receive multiple detection packets from LVS and Nginx nodes. This makes it seem like backend servers are receiving packets at intervals shorter than the specified health check interval.

13.5.2 Can Backend Servers Access the Internet After They Are Associated with a Load Balancer?

Yes. Backend servers can access the Internet whether they are associated with a load balancer.

13.5.3 Can ELB Route Traffic Across Regions?

- Dedicated load balancers can distribute traffic across VPCs. For details about how to add backend servers in a different VPC or an on-premises data center, see **Overview**.

13.5.4 Does Each Backend Server Need an EIP to Receive Requests from a Public Network Load Balancer?

No. There is no need to bind an EIP to each backend server because the load balancer routes requests through the private network.

13.5.5 How Do I Check the Network Conditions of a Backend Server?

1. Verify that an IP address has been assigned to the server's primary NIC.
   a. Log in to the server. (An ECS is used as an example here.)
   b. Use **ifconfig** or **ip address** to view the IP address.

      NOTE

      For Windows ECSs, use **ipconfig** on the CLI.
2. Ping the gateway of the subnet where the ECS resides to check for network connectivity.
   a. On the VPC details page, locate the subnet and view the gateway address in the **Gateway** column. Generally, the gateway address ends with **.1**.
   b. Ping the gateway from the ECS. If the gateway cannot be pinged, check the networks at Layer 2 and Layer 3.

13.5.6 How Can I Check the Network Configuration of a Backend Server?

1. Check whether the security group of the server is correctly configured.
   a. On the server details page, view the security group.
   b. Check whether the security group rules allow access from the corresponding IP address range.
      - Dedicated load balancers: Check whether the security group of the backend server has inbound rules to allow traffic from the VPC where the load balancer works. If traffic is not allowed, add an inbound rule to allow traffic from the VPC to the backend server.

      CAUTION

2. Ensure that the network ACLs of the subnet where the server resides does not intercept the traffic.

   In the navigation pane of the VPC console, choose **Access Control** > **Network ACLs** and check whether the subnet allows traffic.

13.5.7 How Do I Check the Status of a Backend Server?

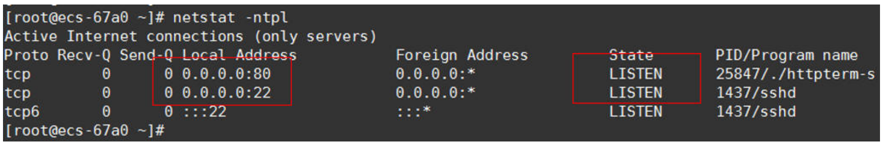
1. Verify that the applications on the backend server are enabled.
   a. Log in to the backend server. (An ECS is used as an example here.)
   b. Check the port status.

      netstat -ntpl

      NOTE

      For Windows ECSs, use **netstat -ano** on the CLI to view the port status or server software status.

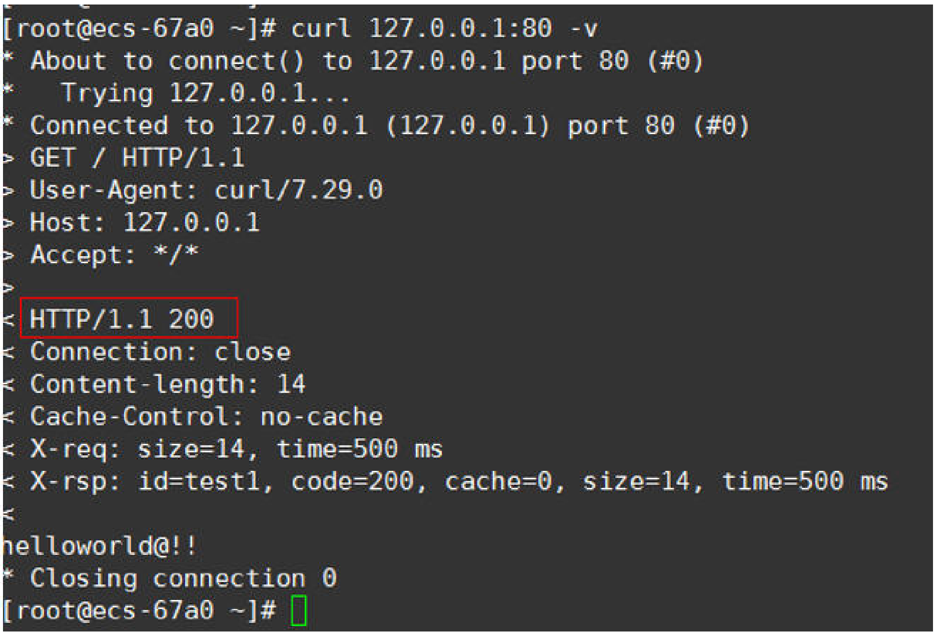
      Figure 13-3 Port status



```
[root@ecs-67a0 ~]# netstat -ntpl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      25847/./httpterm-s
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1437/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1437/sshd
[root@ecs-67a0 ~]#
```

2. Check the network communication of the ECS.

   For example, if the ECS uses port 80, use **curl** to check whether network connectivity is normal.



```
[root@ecs-67a0 ~]# curl 127.0.0.1:80 -v
* About to connect() to 127.0.0.1 port 80 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1
> Accept: */*
>
< HTTP/1.1 200
< Connection: close
< Content-length: 14
< Cache-Control: no-cache
< X-req: size=14, time=500 ms
< X-rsp: id=test1, code=200, cache=0, size=14, time=500 ms
<
helloworld@!!
* Closing connection 0
[root@ecs-67a0 ~]# 
```

13.5.8 When Is a Backend Server Considered Healthy?

When a backend server is associated with a load balancer for the first time, the backend server is considered healthy after one health check. After this, the server is considered healthy only after the maximum number of health checks has been attempted.

13.5.9 Why Can I Access Backend Servers After a Whitelist Is Configured?

The whitelist controls only access to a listener. Only IP addresses in the whitelist can access the listener. To control access to backend servers, you can configure Network ACL or security group rules.

13.5.10 When Will Modified Weights Take Effect?

The new weights for backend servers take effect 5 seconds after the weights are configured.

- TCP and UDP listeners forward requests over new connections based on the new weights. However, connections that have been established with backend servers will not be affected.
- HTTP and HTTPS listeners forward requests based on the new weights. However, requests that have been forwarded to backend servers will not be affected.

NOTE

If the weight of a backend server is changed to 0, the new weight does not take effect immediately, and requests are still routed to this backend server. This is because a persistent connection is established between the load balancer and the backend server and requests are routed to this server until the connection times out.

- TCP and UDP listeners: Persistent connections are disconnected after the idle timeout duration expires.
- HTTP and HTTPS listeners: Persistent connections are disconnected after the response timeout duration expires.

13.5.11 Why Must the Subnet Where the Load Balancer Resides Have at Least 16 Available IP Addresses for Enabling IP as a Backend?

These IP addresses are used by the ELB system. Generally, two IP addresses are required for creating a dedicated load balancer in a single AZ, and six IP addresses are required for creating a dedicated load balancer with IP as a backend enabled. If you create a dedicated load balancer in multiple AZs, more IP addresses will be required. There is an algorithm to calculate how many IP addresses are required.

13.6 Health Checks

13.6.1 How Do I Troubleshoot an Unhealthy Backend Server?

Symptom

If a client cannot access a backend server through a load balancer, the backend server is declared unhealthy. You can view the health check results for a backend server on the ELB console.

On the **Load Balancers** page, click the name of the load balancer to view its details. Click **Backend Server Groups** and locate the server group. You can find the health check results for backend servers in the **Basic Information** area.

Background

Load balancers use the IP addresses from the backend subnet where the load balancers work to send heartbeat requests to backend servers. To ensure that health checks can be performed normally, you need to ensure that the IP addresses from the backend subnet where the load balancers work are allowed to access the backend servers.

CAUTION

- Ensure that security group rules allow access from IP addresses in the backend subnet where the backend server resides.
- If **IP as a Backend** is not enabled for a load balancer and a TCP or UDP listener is added to the load balancer, there is no need to configure security group rules to allow traffic from the VPC where the load balancer backend subnet works to the backend servers associated with TCP or UDP listener.

If a backend server is considered unhealthy, ELB will not route traffic to it until it is declared healthy again.

Troubleshooting

Possible causes are described here in order of how likely they are to occur.

Check these causes one by one until you find the cause of this issue.

NOTE

It takes a while for the modification to take effect after you change the health check configuration. The required time depends on health check interval and timeout duration. You can view the health check result in the backend server list of target load balancer.

Figure 13-4 Troubleshooting process

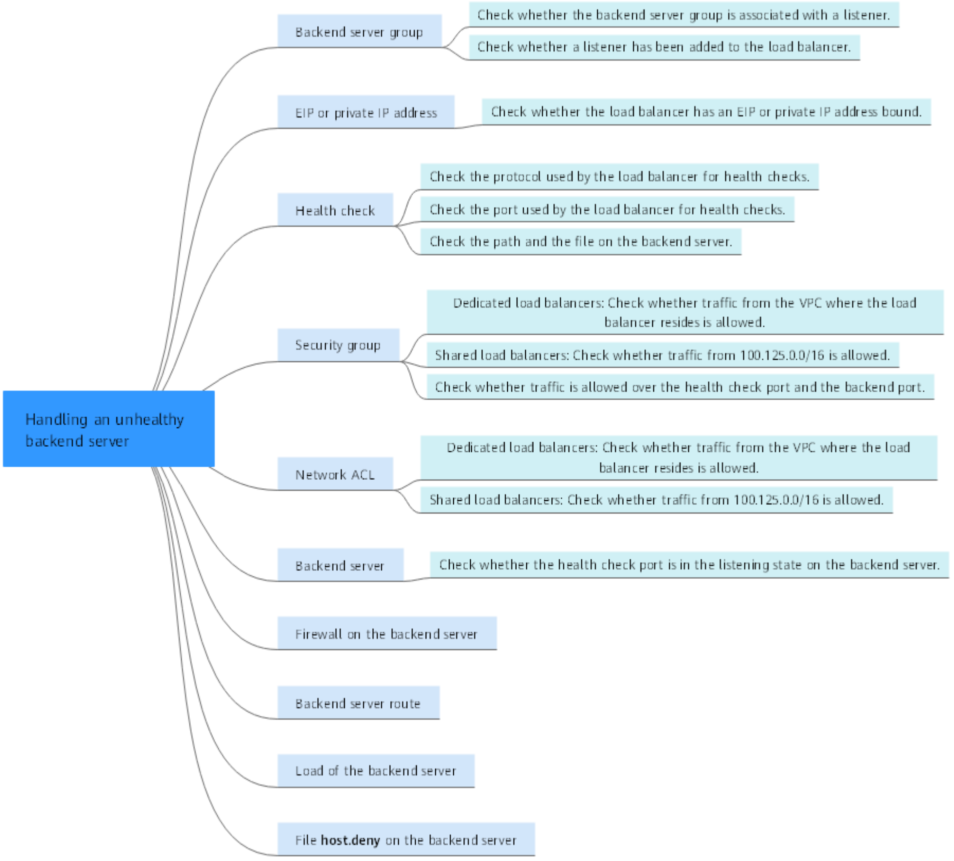

Handling an unhealthy backend server
Backend server group
Check whether the backend server group is associated with a listener.
Check whether a listener has been added to the load balancer.
EIP or private IP address
Check whether the load balancer has an EIP or private IP address bound.
Health check
Check the protocol used by the load balancer for health checks.
Check the port used by the load balancer for health checks.
Check the path and the file on the backend server.
Security group
Dedicated load balancers: Check whether traffic from the VPC where the load balancer resides is allowed.
Shared load balancers: Check whether traffic from 100.125.0.0/16 is allowed.
Check whether traffic is allowed over the health check port and the backend port.
Network ACL
Dedicated load balancers: Check whether traffic from the VPC where the load balancer resides is allowed.
Shared load balancers: Check whether traffic from 100.125.0.0/16 is allowed.
Backend server
Check whether the health check port is in the listening state on the backend server.
Firewall on the backend server
Backend server route
Load of the backend server
File **host.deny** on the backend server

Figure 13-5 Troubleshooting process

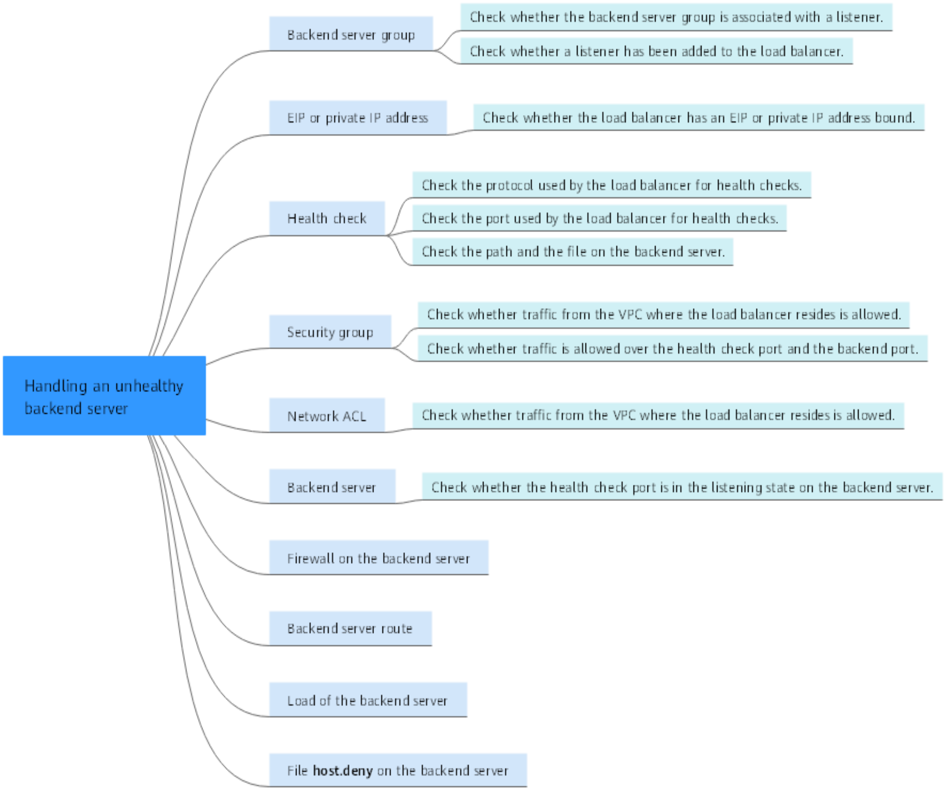

Handling an unhealthy backend server
Backend server group
Check whether the backend server group is associated with a listener.
Check whether a listener has been added to the load balancer.
EIP or private IP address
Check whether the load balancer has an EIP or private IP address bound.
Health check
Check the protocol used by the load balancer for health checks.
Check the port used by the load balancer for health checks.
Check the path and the file on the backend server.
Security group
Check whether traffic from the VPC where the load balancer resides is allowed.
Check whether traffic is allowed over the health check port and the backend port.
Network ACL
Check whether traffic from the VPC where the load balancer resides is allowed.
Backend server
Check whether the health check port is in the listening state on the backend server.
Firewall on the backend server
Backend server route
Load of the backend server
File host.deny on the backend server

Table 13-7 Troubleshooting process

Possible Cause	Solution
Backend server group	**Checking Whether the Backend Server Group Is Associated with a Listener**
EIP or private IP address	**Checking Whether an EIP or a Private IP Address Is Bound to the Load Balancer**
Health check configuration	**Checking the Health Check Configuration**
Security group rules	**Checking Security Group Rules**
Network ACL rules	**Checking Network ACL Rules**
Backend server listening configuration	**Checking the Backend Server**
Network ACL rules	**Checking the Firewall on the Backend Server**
Backend server route	**Checking the Backend Server Route**
Backend server load	**Checking the Backend Server Load**
Backend server **host.deny** file	**Checking the hosts.deny File**

Checking Whether the Backend Server Group Is Associated with a Listener

Check whether the backend server group that the unhealthy backend server belongs to is associated with a listener.

- If the backend server group is not associated with a listener, check whether a listener has been added to the load balancer.
  - If there is a listener, associate the backend server group with the listener.
  - If there are no listeners, add a listener. Select **Use existing** and then select the backend server group when you add the listener.
- If the backend server group has been associated with a listener, proceed with the following operations.

Checking Whether an EIP or a Private IP Address Is Bound to the Load Balancer

NOTE

- Check this only when you add a TCP or UDP listener to the load balancer.
- If you add an HTTP or HTTPS listener to the load balancer, health checks will not be affected no matter whether an EIP or private IP address is bound to the load balancer.

If you add a TCP or UDP listener to the load balancer, check whether the load balancer has an EIP or private IP address bound.

If the load balancer has no EIP or private IP address bound, bind one.

NOTE

When you create a load balancer for the first time, if no EIP or private IP address is bound to the load balancer, the health check result of backend servers associated with a TCP or UDP listener is **Unhealthy**. After you bind an EIP or private IP address to the load balancer, the health check result becomes **Healthy**. If you unbind the EIP or private IP address from the load balancer, the health check result is still **Healthy**.

Checking the Health Check Configuration

Click the name of the load balancer to view its details. Navigate to **Backend Server Groups** and then click the name of the server group. In the **Basic Information** area, to the right of **Health Check**, click **Configure**. Check the following parameters:

- **Domain Name**: If you use HTTP for health checks and the backend server is configured to verify the Host header, enter the domain name configured for the backend server.
- **Protocol**: The protocol used for health checks.
- **Port**: The port must be the one used on the backend server, and it cannot be changed. Check whether the health check port is in the listening state on the backend server. If it is not, the backend server will be identified as unhealthy.
- **Check Path**: If HTTP is used for health checks, you must check this parameter. A simple static HTML file is recommended.

NOTE

- If the health check protocol is HTTP, the port and the path are used for health checks.
- If the health check protocol is TCP, only the port is used for health checks.
- If health check protocol is HTTP and the health check port is normal, change the path or change the health check protocol to TCP.
- Enter an absolute path.

  For example:

  If the URL is **http://www.example.com** or **http://192.168.63.187:9096**, enter **/** as the health check path.

  If the URL is **http://www.example.com/chat/try/**, enter **/chat/try/** as the health check path.

  If the URL is **http://192.168.63.187:9096/chat/index.html**, enter **/chat/index.html** as the health check path.

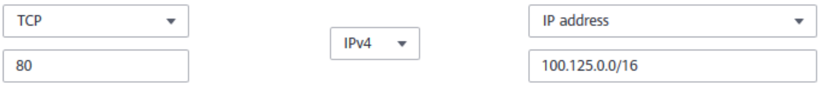
Checking Security Group Rules

- **TCP, HTTP, or HTTPS listeners**: Verify that the inbound security group rule allows TCP traffic from the VPC where the load balancer works to the backend server over the health check port.
  - **If the health check port is the same as the backend port**, the inbound rule must allow traffic over the backend port, for example, port 80.
  - **If the port (port 80 as an example) for health check is different from that used by the backend server (port 443 as an example)**, inbound security group rules must allow traffic over both ports.

    NOTE

    You can check the protocol and port in the **Basic Information** area of the backend server group.

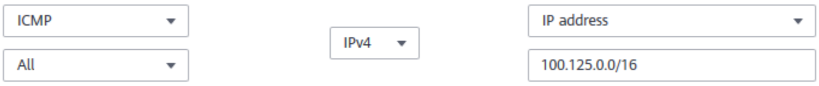
  Figure 13-6 Example inbound rule

  TCP
  80
  IPv4
  IP address
  100.125.0.0/16

- **UDP listeners**: Verify that the inbound security group rule allows traffic from the VPC where the load balancer works to the backend server using the health check protocol and over the health check port. In addition, the rule must allow inbound ICMP traffic.

  Figure 13-7 Example inbound rule that allows ICMP traffic

  ICMP
  All
  IPv4
  IP address
  100.125.0.0/16

NOTE

- If you are not sure about the security group rules, change the **Protocol & Port** to **All** for testing.
- For UDP listeners, see **How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?**

Checking Network ACL Rules

To control traffic in and out of a subnet, you can associate a network ACL with the subnet. Network ACL rules control access to subnets and add an additional layer of defense to your subnets. Default network ACL rules reject all inbound and outbound traffic. If the subnet of a load balancer or associated backend servers has a network ACL associated, the load balancer cannot receive traffic from the Internet or route traffic to backend servers, and backend servers cannot receive traffic from and respond to the load balancer.

Configure an inbound network ACL rule to allow traffic from the VPC where the load balancer resides to backend servers.

Checking the Backend Server

NOTE

If the backend server runs on Windows, use a browser to access **https://***{Backend server IP address}*:*{Health check port}*. If a 2xx or 3xx code is returned, the backend server is running normally.

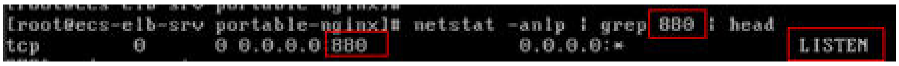
- Run the following command on the backend server to check whether the health check port is listened on:

  ```
  netstat -anlp | grep port
  ```

  If the health check port and **LISTEN** are displayed, the health check port is in the listening state. As shown in **Figure 13-8**, TCP port 880 is listened on.

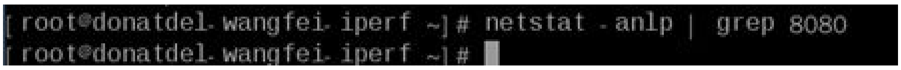
  If you do not specify a health check port, backend ports are used by default.

  Figure 13-8 Backend server port listened on

  ```
  [root@ecs-elb-srv portable-nginx]# netstat -anlp | grep 880 | head
  tcp        0      0 0.0.0.0:880             0.0.0.0:*               LISTEN
  ```

  Figure 13-9 Backend server port not listened on

  ```
  [root@donatdel.wangfei.iperf ~]# netstat -anlp | grep 8080
  [root@donatdel.wangfei.iperf ~]#
  ```

  If the health check port is not in the listening state, the backend server is not listened on. You need to start the application on the backend server and check whether the health check port is listened on.

- For HTTP health checks, run the following command on the backend server to check the status code:

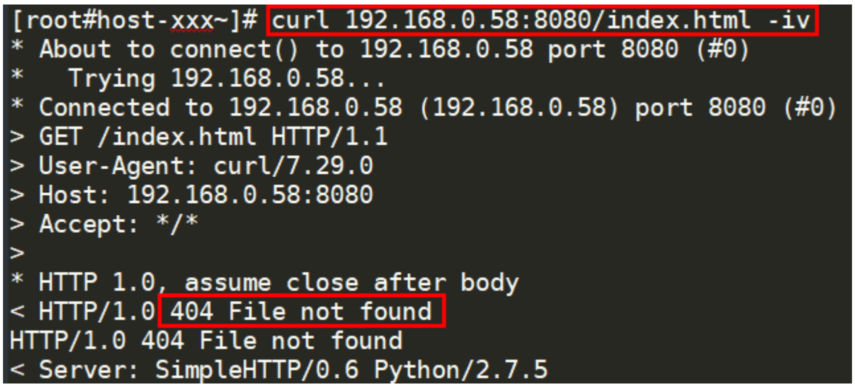
  ```
  curl Private IP address of the backend server:Health check port/Health check path -iv
  ```

  To perform an HTTP health check, the load balancer initiates a GET request to the backend server. If the following response status codes are displayed, the backend server is considered healthy:

  TCP listeners: 200

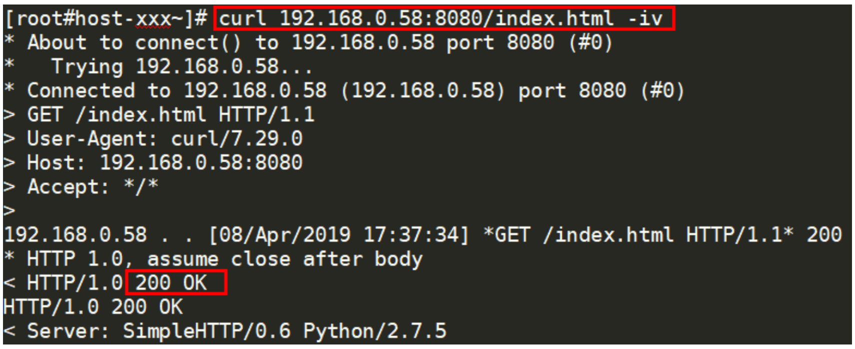
  The status code is 200, 202, or 401 if the backend server is healthy.

Figure 13-10 Unhealthy backend server



```
[root#host-xxx~]# curl 192.168.0.58:8080/index.html -iv
* About to connect() to 192.168.0.58 port 8080 (#0)
*   Trying 192.168.0.58...
* Connected to 192.168.0.58 (192.168.0.58) port 8080 (#0)
> GET /index.html HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.58:8080
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 404 File not found
HTTP/1.0 404 File not found
< Server: SimpleHTTP/0.6 Python/2.7.5
```

Figure 13-11 Healthy backend server



```
[root#host-xxx~]# curl 192.168.0.58:8080/index.html -iv
* About to connect() to 192.168.0.58 port 8080 (#0)
*   Trying 192.168.0.58...
* Connected to 192.168.0.58 (192.168.0.58) port 8080 (#0)
> GET /index.html HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 192.168.0.58:8080
> Accept: */*
>
192.168.0.58 . . [08/Apr/2019 17:37:34] *GET /index.html HTTP/1.1* 200
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/2.7.5
```

- If HTTP is used for health checks and the backend server is detected unhealthy, perform the following steps to configure a TCP health check:

  On the **Listeners** tab page, modify the target listener, select the backend server group for which TCP health check has been configured, or add a backend server group and select TCP as the health check protocol. After you complete the configuration, wait for a while and check the health check result.

Checking the Firewall on the Backend Server

If the firewall or other security software is enabled on the backend server, the software may block the IP addresses in the backend subnet of the load balancer, or 100.125.0.0/16.

Checking the Backend Server Route

Check whether the default route configured for the primary NIC (for example, eth0) has been manually modified. If the default route is changed, health check packets may fail to reach the backend server.

Run the following command on the backend server to check whether the default route points to the gateway (For Layer 3 communications, the default route must be configured to point to the gateway of the VPC subnet where the backend server resides):

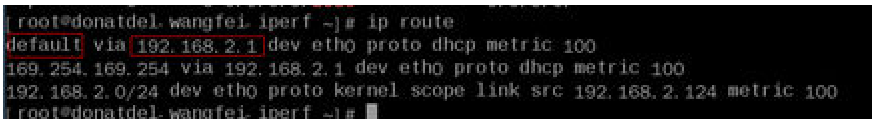
```
ip route
```

Alternatively, run the following command:

```
route -n
```

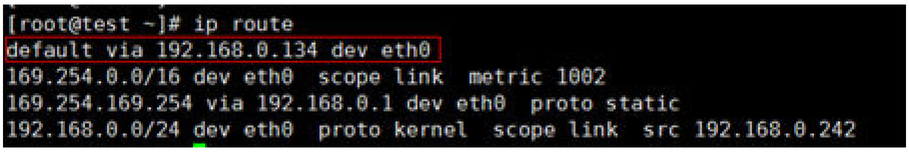
Figure 13-12 shows the command output when the backend server route is normal.

Figure 13-12 Example default route pointing to the gateway



```
[root@donatdel.wangfei.iperf ~]# ip route
default via 192.168.2.1 dev etho proto dhcp metric 100
169.254.169.254 via 192.168.2.1 dev etho proto dhcp metric 100
192.168.2.0/24 dev etho proto kernel scope link src 192.168.2.124 metric 100
[root@donatdel.wangfei.iperf ~]#
```

Figure 13-13 Example default route not pointing to the gateway



```
[root@test ~]# ip route
default via 192.168.0.134 dev eth0
169.254.0.0/16 dev eth0  scope link  metric 1002
169.254.169.254 via 192.168.0.1 dev eth0  proto static
192.168.0.0/24 dev eth0  proto kernel  scope link  src 192.168.0.242
```

If the command output does not contain the first route, or the route does not point to the gateway, configure or modify the default route to point to the gateway.

Checking the Backend Server Load

View the vCPU usage, memory usage, network connections of the backend server on the Cloud Eye console to check whether the backend server is overloaded.

If the load is high, connections or requests for health checks may time out.

Checking the hosts.deny File

Verify that IP addresses from the VPC where the load balancers work are not written into the **/etc/hosts.deny** file.

13.6.2 Why Is the Interval at Which Backend Servers Receive Health Check Packets Different from the Configured Interval?

Each LVS node and Nginx node in the ELB system detect backend servers at the health check interval that you have specified for the backend server group.

During this period, backend servers receive detection packets from multiple nodes. This makes it seem that backend servers receive these packets at intervals shorter than the specified health check interval.

13.6.3 How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?

How UDP Health Checks Work

UDP is a connectionless protocol. A UDP health check is implemented as follows:

- The health check node sends an ICMP request to the backend server based on the health check configuration.
  - If the health check node receives an ICMP reply from the backend server, it considers the backend server healthy and continues the health check.

  - If the health check node does not receive an ICMP reply from the backend server, it considers the backend server unhealthy.
- After receiving the ICMP reply, the health check node sends a UDP probe packet to the backend server.
  - If the health check node receives an ICMP Port Unreachable message from the backend server within the timeout duration, the backend server is considered unhealthy.
  - If the health check node does not receive an ICMP Port Unreachable message from the backend server within the timeout duration, the backend server is considered healthy.

When you use UDP for health checks, retain default parameter settings.

Troubleshooting

If the backend server is unhealthy, use either of the following methods to locate the fault:

- Check whether the timeout duration is too short.

  One possible cause is that the ICMP Echo Reply or ICMP Port Unreachable message returned by the backend server does not reach the health check node within the timeout duration. As a result, the health check result is inaccurate.

  It is recommended that you change the timeout duration to a larger value.

  UDP health checks are different from other health checks. If the health check timeout duration is too short, the health check result of the backend server frequently toggles back and forth between **Healthy** and **Unhealthy**.
- Check whether the backend server restricts the rate at which ICMP messages are generated.

For Linux servers, run the following commands to query the rate limit and rate mask:

```
sysctl -q net.ipv4.icmp_ratelimit
```

The default rate limit is **1000**.

```
sysctl -q net.ipv4.icmp_ratemask
```

The default rate mask is **6168**.

If the returned value of the first command is the default value or **0**, run the following command to remove the rate limit of Port Unreachable messages:

```
sysctl -w net.ipv4.icmp_ratemask=6160
```

For more information, see the *Linux Programmer's Manual*. On the Linux CLI, run the following command to display the manual:

```
man 7 icmp
```

Alternatively, visit **http://man7.org/linux/man-pages/man7/icmp.7.html**.

NOTE

Once the rate limit is lifted, the number of ICMP Port Unreachable messages on the backend server will not be limited.

Precautions

Note the following when you configure UDP health checks:

- UDP health checks use ping packets to check the health of the backend server. To ensure smooth transmission of these packets, ensure that ICMP is enabled on the backend server by performing the following:

  Log in to the server and run the following command as user **root**:

  cat /proc/sys/net/ipv4/icmp_echo_ignore_all

  - If the returned value is **1**, ICMP is disabled.
  - If the returned value is **0**, ICMP is enabled.
- The health check result may be different from the actual health of the backend server.

  If the backend server runs Linux, the rate of ICMP packets may be limited due to Linux's defense against ping flood attacks when there is a large number of concurrent requests. In this case, if a service exception occurs, the load balancer will not receive error message **port XX unreachable** and will consider the health check to be successful. As a result, there is an inconsistency between the health check result and the actual server health.

13.6.4 Why Does ELB Frequently Send Requests to Backend Servers During Health Checks?

ELB is deployed in clusters, and all nodes for request forwarding in the cluster send requests to backend servers at the same time. If the health check interval is too short, health checks are performed once every few seconds, and a large number of packets are sent to backend servers. To control the frequency of access to backend servers, change the health check interval by referring to **Modifying Health Check Settings**.

13.6.5 When Does a Health Check Start?

After a backend server is added to a backend server group, the health check is performed at a random time during the first interval and then at the specified interval.

13.6.6 Do Maximum Retries Include Health Checks That Consider Backend Servers Unhealthy?

Yes. Maximum retries are the maximum number of health checks after which a backend server is detected healthy or the maximum number of health checks after which the same backend server is detected unhealthy.

13.6.7 What Do I Do If a Lot of Access Logs Are Generated During Health Checks?

1. You can increase the health check interval by referring to **Modifying Health Check Settings**.

   Risk: After the health check interval is prolonged, the time for the load balancer to detect unhealthy servers will increase.

2. You can disable the health check by referring to **Modifying Health Check Settings**.

   Risk: After health checks are disabled, the load balancer will not check the backend servers. If a backend server becomes faulty, the load balancer will still route requests to this server.

13.6.8 What Status Codes Will Be Returned If Backend Servers Are Identified as Healthy?

Table 13-8 Status Code

Load Balancer Type	Health Check Protocol	Status Code
Load balancers	HTTP	200
	HTTPS	200

13.7 Obtaining Source IP Addresses

13.7.1 How Can I Transfer the IP Address of a Client?

When you use ELB to route requests to backend servers, IP addresses of the clients will be translated by the ELB. This FAQ guides you to obtain the IP addresses of the clients.

- Load balancing at Layer 7 (HTTP or HTTPS listeners): Configure the application server and obtain the IP address of a client from the HTTP header.

  For details, see **Layer 7 Load Balancing**.

Constraints and Limitations

- If Network Address Translation (NAT) is used, you cannot obtain the IP addresses of the clients.
- If the client is a container, you can obtain only the IP address of the node where the container is located, but cannot obtain the IP address of the container.
- If **Transfer Client IP Address** is enabled for TCP or UDP listeners, a cloud server cannot be used as a backend server and a client at the same time.
- By default, the **Transfer Client IP Address** function is enabled for TCP and UDP listeners of dedicated load balancers and cannot be disabled.

Layer 7 Load Balancing

Configure the application server and obtain the IP address of a client from the HTTP header.

The real IP address is placed in the X-Forwarded-For header field by the load balancer in the following format:

```
X-Forwarded-For: IP address of the client,Proxy server 1-IP address,Proxy server 2-IP address,...
```

If you use this method, the first IP address obtained is the IP address of the client.

Apache Server

1. Install Apache 2.4.

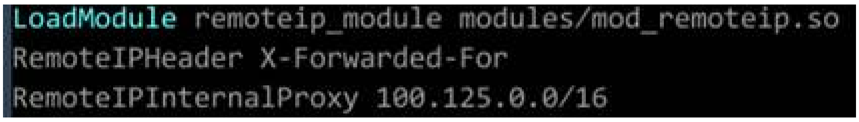
   For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

   ```
   yum install httpd
   ```

2. Add the following content to the end of Apache configuration file **/etc/httpd/conf/httpd.conf**:

   ```
   LoadModule remoteip_module modules/mod_remoteip.so
   RemoteIPHeader X-Forwarded-For
   RemoteIPInternalProxy 100.125.0.0/16
   ```

   Figure 13-14 Content to be added

   ```
   LoadModule remoteip_module modules/mod_remoteip.so
   RemoteIPHeader X-Forwarded-For
   RemoteIPInternalProxy 100.125.0.0/16
   ```

   NOTE

   Add the IP address range of the proxy server after **RemoteIPInternalProxy**.

   CIDR block of the subnet where the load balancer resides

3. Change the log output format in the Apache configuration file to the following (**%a** indicates the source IP address):

   ```
   LogFormat "%a %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
   ```

4. Restart Apache.

   ```
   systemctl restart httpd
   ```

5. Obtain the actual IP address of the client from the httpd access logs.

Nginx Server

For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

1. Run the following commands to install http_realip_module:

   ```
   yum -y install gcc pcre pcre-devel zlib zlib-devel openssl openssl-devel
   wget http://nginx.org/download/nginx-1.17.0.tar.gz
   tar zxvf nginx-1.17.0.tar.gz
   cd nginx-1.17.0
   ./configure --prefix=/path/server/nginx --with-http_stub_status_module --without-http-cache --with-http_ssl_module --with-http_realip_module
   make
   make install
   ```

2. Run the following command to open the **nginx.conf** file:

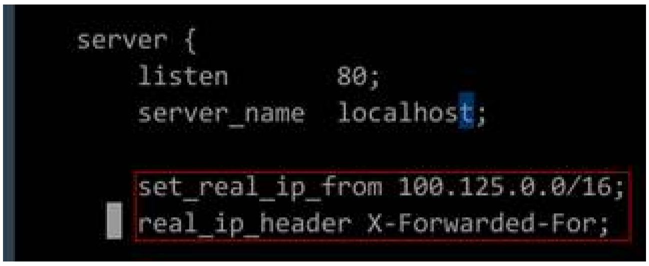
   ```
   vi /path/server/nginx/conf/nginx.conf
   ```

3. Add new fields and information to the end of the following configuration information:

   Add the following information under **http** or **server**:

   ```
   set_real_ip_from 100.125.0.0/16;
   real_ip_header X-Forwarded-For;
   ```

Figure 13-15 Adding information



```
server {
    listen       80;
    server_name  localhost;

    set_real_ip_from 100.125.0.0/16;
    real_ip_header X-Forwarded-For;
```

NOTE

Add the IP address range of the proxy server after **RemoteIPInternalProxy**.

CIDR block of the subnet where the load balancer resides

4. Start Nginx.

```
/path/server/nginx/sbin/nginx
```

5. Obtain the actual IP address of the client from the Nginx access logs.

```
cat /path/server/nginx/logs/access.log
```

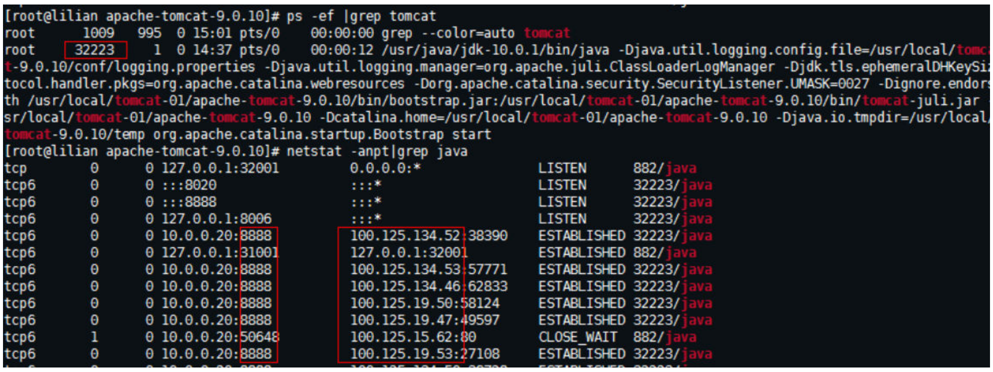
Tomcat Servers

In the following operations, the Tomcat installation path is **/usr/tomcat/tomcat8/**.

1. Log in to a server on which Tomcat is installed.
2. Check whether Tomcat is running properly.

```
ps -ef|grep tomcat
netstat -anpt|grep java
```

Figure 13-16 Tomcat running properly

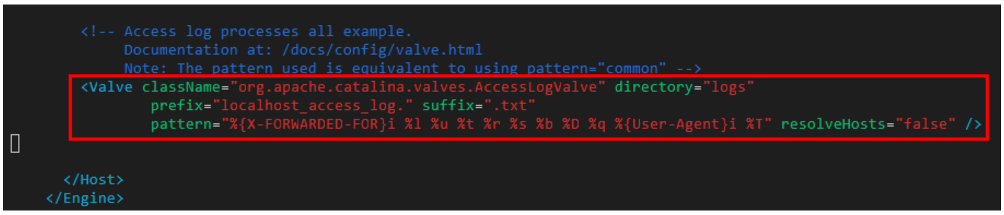


```
[root@lilian apache-tomcat-9.0.10]# ps -ef |grep tomcat
root      1009   995  0 15:01 pts/0    00:00:00 grep --color=auto tomcat
root     32223     1  0 14:37 pts/0    00:00:12 /usr/java/jdk-10.0.1/bin/java -Djava.util.logging.config.file=/usr/local/tomc
t-9.0.10/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySi
tocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endors
th /usr/local/tomcat-01/apache-tomcat-9.0.10/bin/bootstrap.jar:/usr/local/tomcat-01/apache-tomcat-9.0.10/bin/tomcat-juli.jar
sr/local/tomcat-01/apache-tomcat-9.0.10 -Dcatalina.home=/usr/local/tomcat-01/apache-tomcat-9.0.10 -Djava.io.tmpdir=/usr/local/
tomcat-9.0.10/temp org.apache.catalina.startup.Bootstrap start
[root@lilian apache-tomcat-9.0.10]# netstat -anpt|grep java
tcp        0      0 127.0.0.1:32001         0.0.0.0:*               LISTEN      882/java
tcp6       0      0 :::8020                 :::*                    LISTEN      32223/java
tcp6       0      0 :::8888                 :::*                    LISTEN      32223/java
tcp6       0      0 127.0.0.1:8006          :::*                    LISTEN      32223/java
tcp6       0      0 10.0.0.20:8888          100.125.134.52:38390    ESTABLISHED 32223/java
tcp6       0      0 127.0.0.1:31001         127.0.0.1:32001         ESTABLISHED 882/java
tcp6       0      0 10.0.0.20:8888          100.125.134.53:57771    ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888          100.125.134.46:62833    ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888          100.125.19.50:58124     ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888          100.125.19.47:49597     ESTABLISHED 32223/java
tcp6       1      0 10.0.0.20:50648         100.125.15.62:80        CLOSE_WAIT  882/java
tcp6       0      0 10.0.0.20:8888          100.125.19.53:27108     ESTABLISHED 32223/java
```

3. Modify **className="org.apache.catalina.valves.AccessLogValve"** in the **server.xml** file as follows:

```
vim /usr/tomcat/tomcat8/conf/server.xml
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt"
pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T"
resolveHosts="false" />
```

Figure 13-17 Example configuration



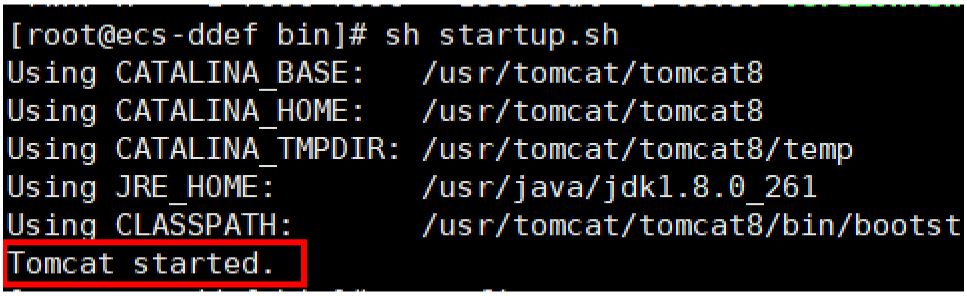
```
        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log." suffix=".txt"
               pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T" resolveHosts="false" />
▯

      </Host>
    </Engine>
```

4. Restart the Tomcat service.

   ```
   cd /usr/tomcat/tomcat8/bin && sh shutdown.sh && sh startup.sh
   ```

   /usr/tomcat/tomcat8/ is where Tomcat is installed. Change it based on site requirements.

   Figure 13-18 Restarting the Tomcat service

   ```
   [root@ecs-ddef bin]# sh startup.sh
   Using CATALINA_BASE:   /usr/tomcat/tomcat8
   Using CATALINA_HOME:   /usr/tomcat/tomcat8
   Using CATALINA_TMPDIR: /usr/tomcat/tomcat8/temp
   Using JRE_HOME:        /usr/java/jdk1.8.0_261
   Using CLASSPATH:       /usr/tomcat/tomcat8/bin/bootst
   Tomcat started.
   ```

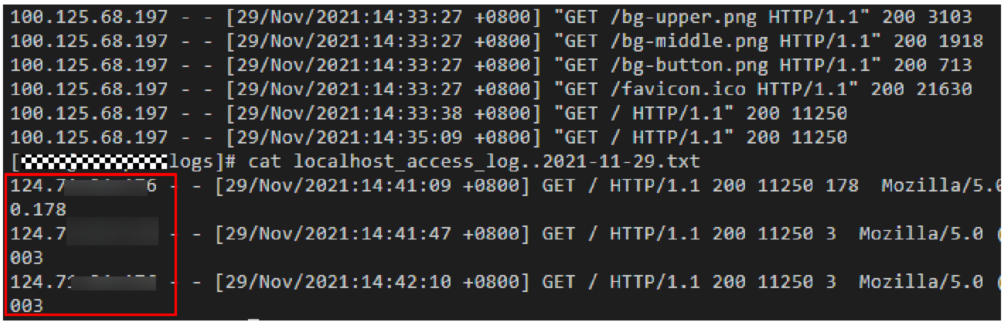
5. View the latest logs.

   As highlighted in the following figure, IP addresses that are not in the IP address range starting with 100.125 are the source IP addresses.

   ```
   cd /usr/tomcat/tomcat8/logs/
   cat localhost_access_log..2021-11-29.txt
   ```

   In this command, **localhost_access_log..2021-11-29.txt** indicates the log path of the current day. Change it based on site requirements.

   Figure 13-19 Querying the source IP address

   ```
   100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-upper.png HTTP/1.1" 200 3103
   100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-middle.png HTTP/1.1" 200 1918
   100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-button.png HTTP/1.1" 200 713
   100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /favicon.ico HTTP/1.1" 200 21630
   100.125.68.197 - - [29/Nov/2021:14:33:38 +0800] "GET / HTTP/1.1" 200 11250
   100.125.68.197 - - [29/Nov/2021:14:35:09 +0800] "GET / HTTP/1.1" 200 11250
   [[redacted]logs]# cat localhost_access_log..2021-11-29.txt
   124.7[redacted]6 - - [29/Nov/2021:14:41:09 +0800] GET / HTTP/1.1 200 11250 178  Mozilla/5.0
   0.178
   124.7[redacted] - - [29/Nov/2021:14:41:47 +0800] GET / HTTP/1.1 200 11250 3  Mozilla/5.0
   003
   124.7[redacted] - - [29/Nov/2021:14:42:10 +0800] GET / HTTP/1.1 200 11250 3  Mozilla/5.0
   003
   ```

Windows Server with IIS Deployed

The following uses Windows Server 2012 with IIS7 as an example to describe how to obtain the source IP address.

1. Download and install IIS.
2. Download the **F5XForwardedFor.dll** plug-in and copy the plug-ins in the **x86** and **x64** directories to a directory for which IIS has the access permission, for example, **C:\F5XForwardedFor2008**.

3. Open the Server Manager and choose **Modules** > **Configure Native Modules**.

Figure 13-20 Selecting modules

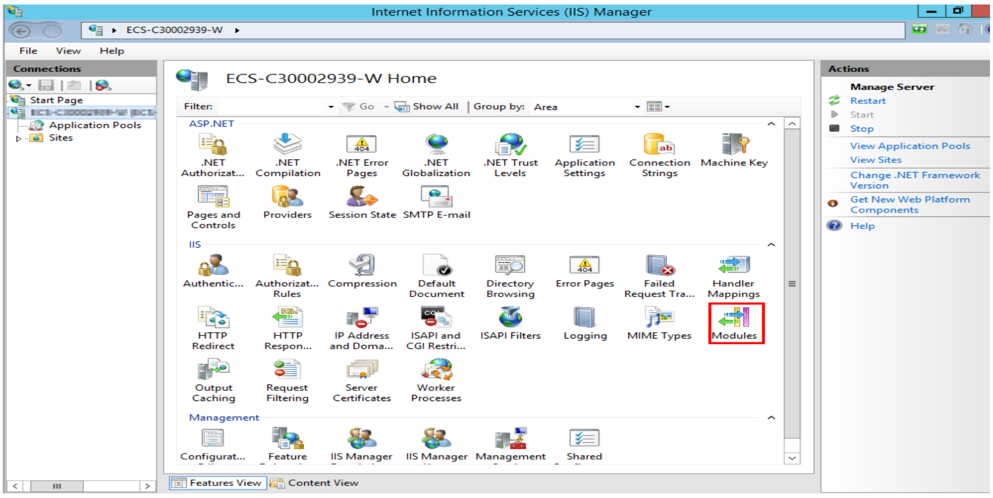
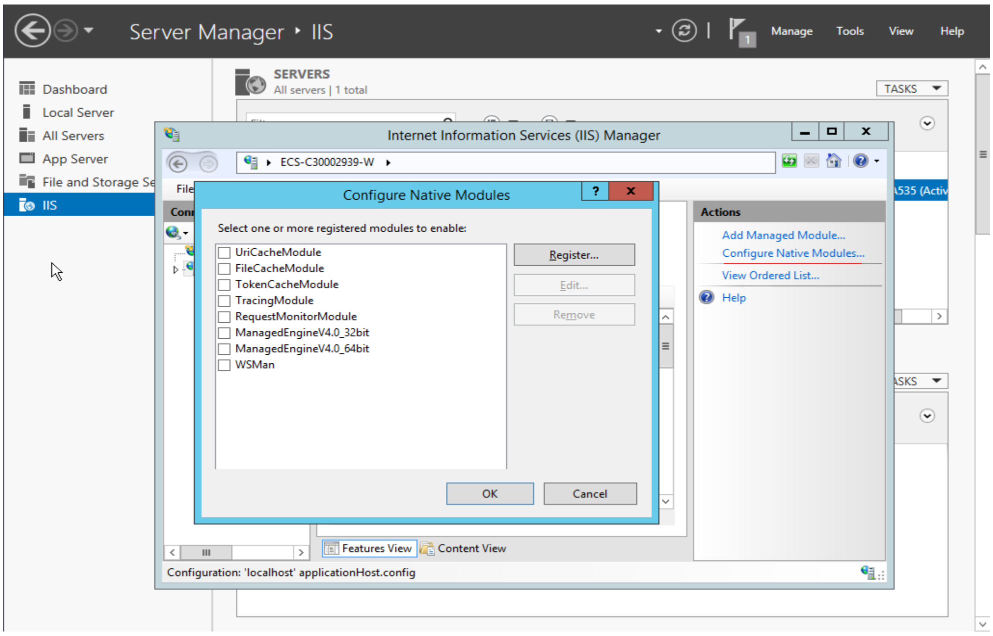


Figure 13-21 Configure Native Modules



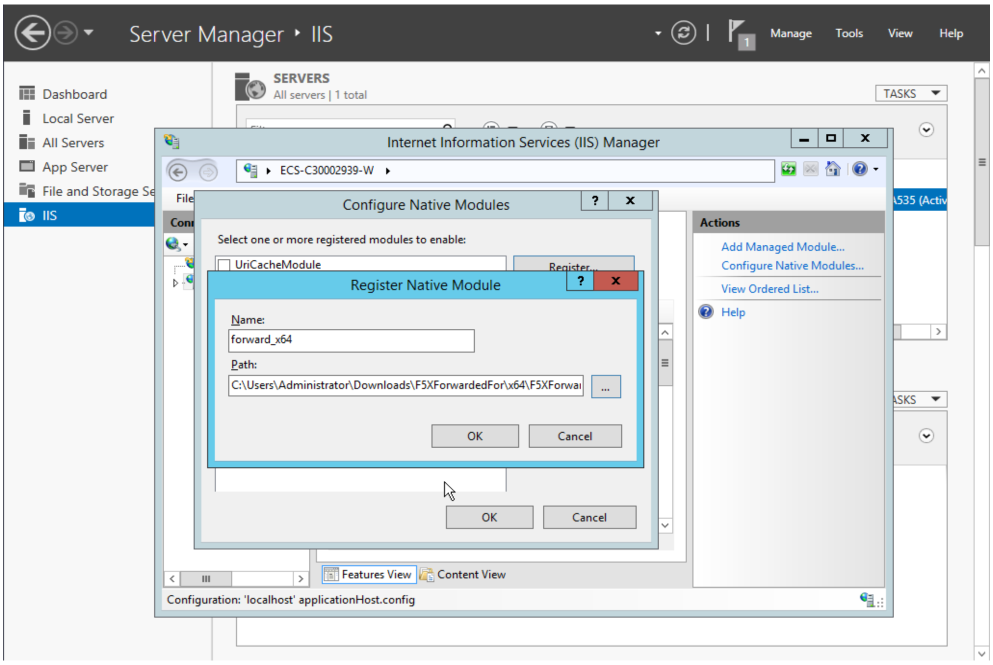
Select one or more registered modules to enable:

- □ UriCacheModule
- □ FileCacheModule
- □ TokenCacheModule
- □ TracingModule
- □ RequestMonitorModule
- □ ManagedEngineV4.0_32bit
- □ ManagedEngineV4.0_64bit
- □ WSMan

Configuration: 'localhost' applicationHost.config

4. Click **Register** to register the x86 and x64 plug-ins.

Figure 13-22 Registering plug-ins

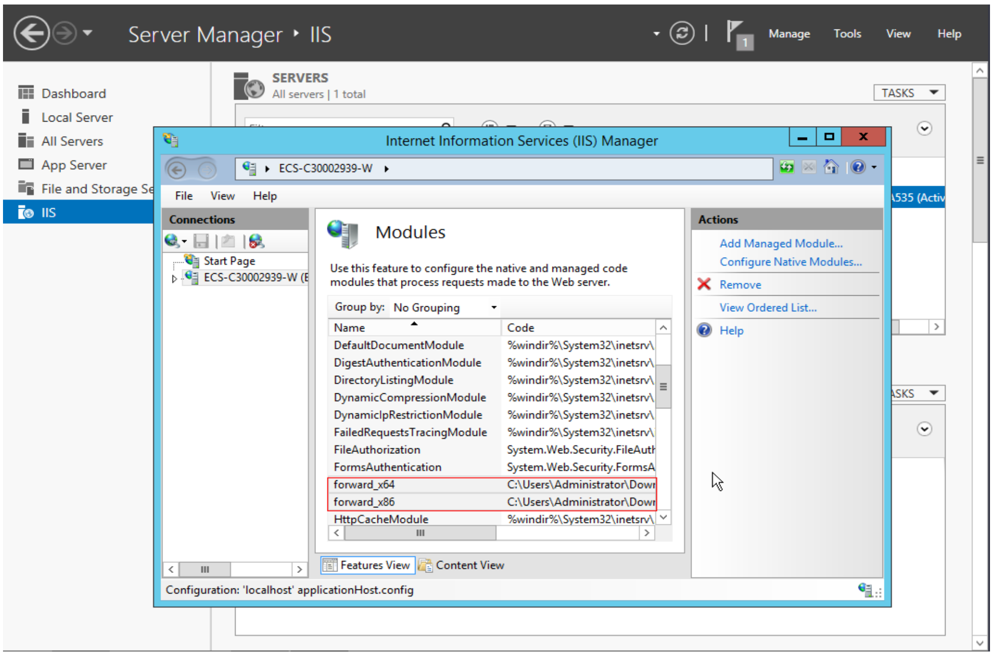


Name: forward_x64

Path: C:\Users\Administrator\Downloads\F5XForwardedFor\x64\F5XForwa

5. In the **Modules** dialog box, verify that the registered plug-ins are displayed in the list.

Figure 13-23 Confirming the registration



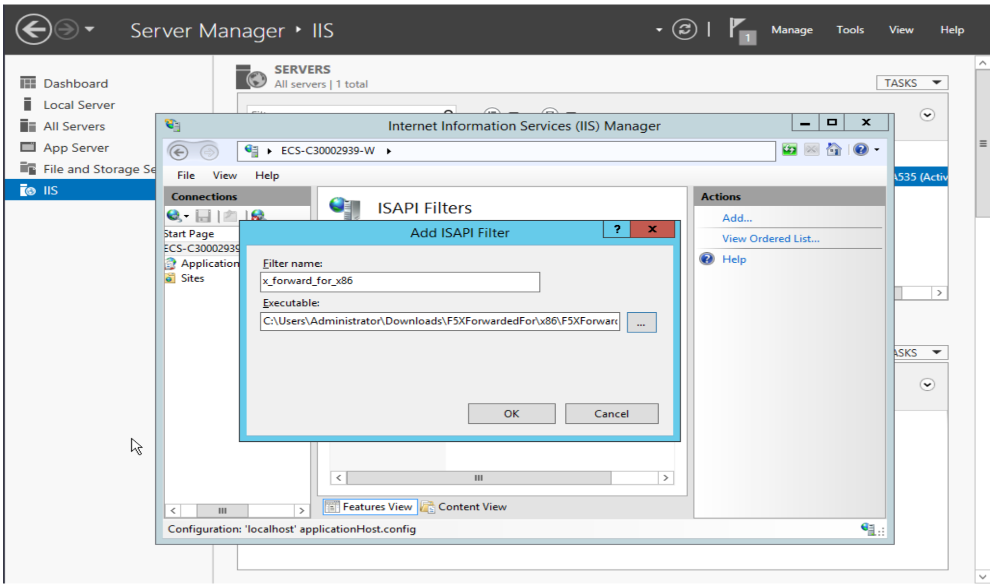
Use this feature to configure the native and managed code modules that process requests made to the Web server.

Group by: No Grouping

Name	Code
DefaultDocumentModule	%windir%\System32\inetsrv\
DigestAuthenticationModule	%windir%\System32\inetsrv\
DirectoryListingModule	%windir%\System32\inetsrv\
DynamicCompressionModule	%windir%\System32\inetsrv\
DynamicIpRestrictionModule	%windir%\System32\inetsrv\
FailedRequestsTracingModule	%windir%\System32\inetsrv\
FileAuthorization	System.Web.Security.FileAuth
FormsAuthentication	System.Web.Security.FormsA
forward_x64	C:\Users\Administrator\Dow
forward_x86	C:\Users\Administrator\Dow
HttpCacheModule	%windir%\System32\inetsrv\

6. Select **ISAPI Filters** on the Server Manager homepage and authorize two plug-ins to run ISAPI and CGI extensions.

Figure 13-24 Adding authorization



Add ISAPI Filter

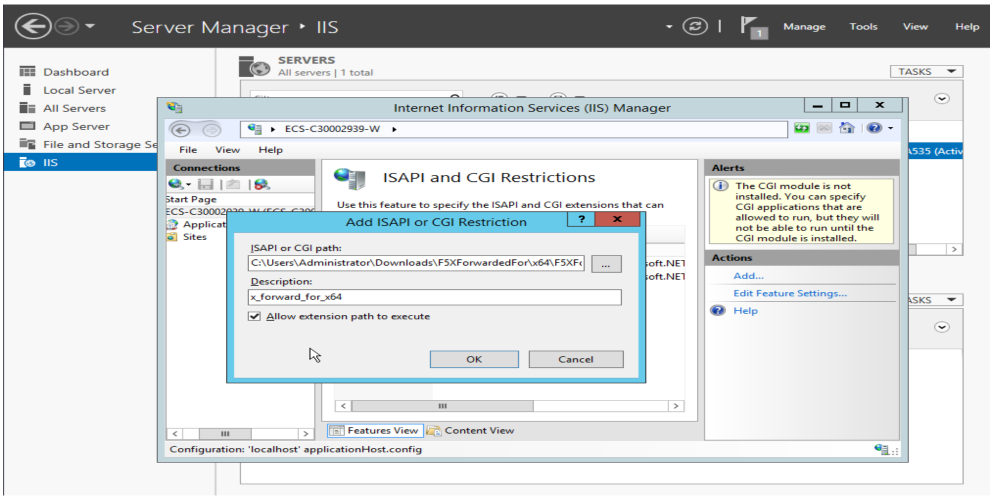
Filter name:
x_forward_for_x86

Executable:
C:\Users\Administrator\Downloads\F5XForwardedFor\x86\F5XForwar

OK Cancel

7. Select **ISAPI and CGI Restriction** to set the execution permission for the two plug-ins.

Figure 13-25 Allowing the plug-ins to execute


Server Manager ▸ IIS
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
App Server
File and Storage Se
IIS
SERVERS
All servers | 1 total
TASKS
Internet Information Services (IIS) Manager
ECS-C30002939-W
File
View
Help
Connections
Start Page
Applicat
Sites
ISAPI and CGI Restrictions
Use this feature to specify the ISAPI and CGI extensions that can
Alerts
The CGI module is not installed. You can specify CGI applications that are allowed to run, but they will not be able to run until the CGI module is installed.
Actions
Add...
Edit Feature Settings...
Help
Features View
Content View
Configuration: 'localhost' applicationHost.config

Add ISAPI or CGI Restriction

ISAPI or CGI path:
C:\Users\Administrator\Downloads\F5XForwardedFor\x64\F5XF

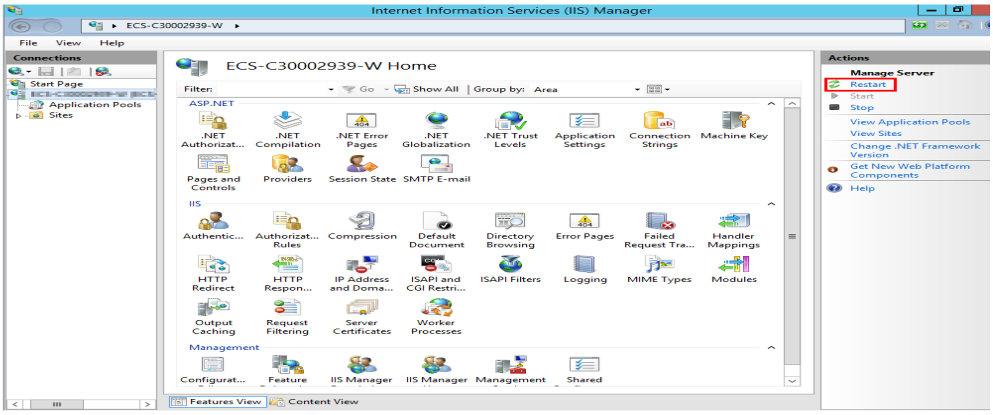
Description:
x_forward_for_x64

☑ Allow extension path to execute

OK Cancel

8. Click **Restart** on the homepage to restart IIS. The configuration will take effect after the restart.

Figure 13-26 Restarting IIS



13.8 HTTP/HTTPS Listeners

13.8.1 Which Protocol Should I Select for the Backend Server Group When Adding an HTTPS Listener?

To use HTTPS at both the frontend and backend, you can create a dedicated load balancer, add an HTTPS listener to the load balancer, and set the backend protocol to HTTPS.

To use HTTPS at the frontend only, you can create a dedicated load balancer, add an HTTPS listener to the load balancer, and set the backend protocol to HTTP.

NOTE

Using HTTPS at both the frontend and backend only allows you to enable mutual authentication on the load balancer and backend servers.

13.8.2 Why Is There a Security Warning After a Certificate Is Configured?

The following may cause the Not Secure warning even after a certificate is configured:

- The domain name used by the certificate is different from the domain name accessed by users. (If this is the case, check the domain name used the certificate to ensure that the domain names are the same or create a self-signed certificate.)
- SNI is configured, but the specified domain name is different from the one used by the certificate.
- The domain name level is inconsistent with the certificate level.

If the problem persists, run the **curl** *{Domain name}* command to locate the fault based on the error information returned by the system.

13.8.3 Why Is a Forwarding Policy in the Faulty State?

A possible cause is that you added a forwarding policy that is the same as an existing one. Even if you delete the existing forwarding policy, the newly-added forwarding policy is still faulty.

To resolve this issue, delete the newly-added forwarding policy and add a different one.

13.8.4 Why Can't I Add a Forwarding Policy to a Listener?

Check the listener protocol.

Forwarding policies can only be added to HTTP and HTTPS listeners.

13.8.5 Why Cannot I Select an Existing Backend Server Group When Adding a Forwarding Policy?

This is because the backend server group has been used by another forwarding policy. A backend server group can be used by only one forwarding policy.

13.9 Sticky Sessions

13.9.1 What Are the Differences Between Persistent Connections and Sticky Sessions?

Persistent connections are not necessarily related to sticky sessions.

A persistent connection allows multiple data packets to be sent continuously over a TCP connection. If no data packets are sent over the connection, the client and the server need to send link detection packets to each other. Sticky sessions enable all requests from the same client during one session to be sent to the same backend server.

13.9.2 How Do I Check If Sticky Sessions Failed to Take Effect?

1. Check whether sticky sessions are enabled for the backend server group. If sticky sessions are enabled, go to the next step.
2. Check the health check result of the backend server. If the health check result is **Unhealthy**, traffic is routed to other backend servers and sticky sessions become invalid.
3. If you select the source IP hash algorithm, check whether the IP address of the request changes before the load balancer receives the request.
4. If sticky sessions are enabled for an HTTP or HTTPS listener, check whether the request carries a cookie. If they are, check whether the cookie value changed (because load balancing at Layer 7 uses cookies to maintain sessions).

13.9.3 How Do I Test Sticky Sessions Using Linux Curl Commands?

1. Prepare required resources.
   1. Buy three ECSs, one as the client and the other two as backend servers.
   2. Create a load balancer and add an HTTP listener to the load balancer. Enable sticky sessions when you add the listener.
2. Start the HTTP service of the two backend servers.

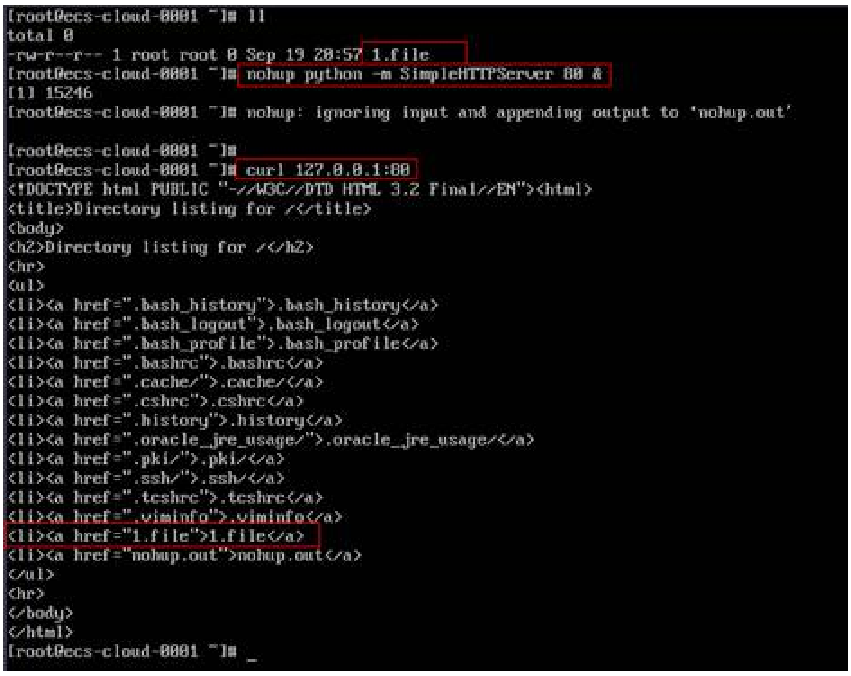
   Log in to a backend server and create a file named **1.file** in the current directory to mark this server.

   Run the following command in the current directory to start the HTTP service:

   nohup python –m SimpleHTTPServer 80 &

   Run the following command to check whether the HTTP service is normal:

   curl http://127.0.0.1:80



```
[root@ecs-cloud-0001 ~]# ll
total 0
-rw-r--r-- 1 root root 0 Sep 19 20:57 1.file
[root@ecs-cloud-0001 ~]# nohup python -m SimpleHTTPServer 80 &
[1] 15246
[root@ecs-cloud-0001 ~]# nohup: ignoring input and appending output to 'nohup.out'

[root@ecs-cloud-0001 ~]#
[root@ecs-cloud-0001 ~]# curl 127.0.0.1:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="1.file">1.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-cloud-0001 ~]# _
```

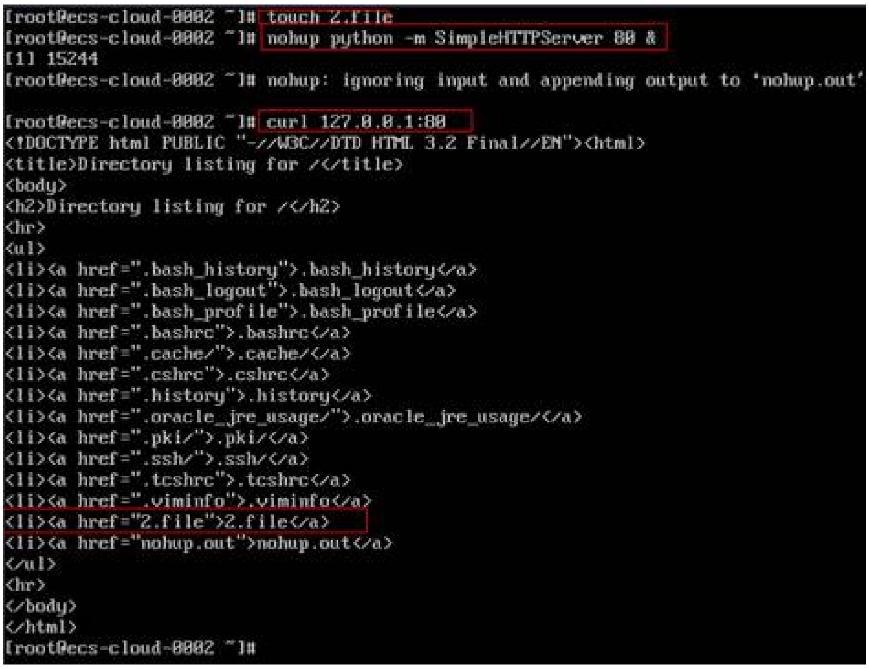
   Log in to the other backend server and create a file named **2.file** in the current directory.

   Run the following command in the current directory to start the HTTP service:

   nohup python –m SimpleHTTPServer 80 &

   Run the following command to check whether the HTTP service is normal:

   curl http://127.0.0.1:80

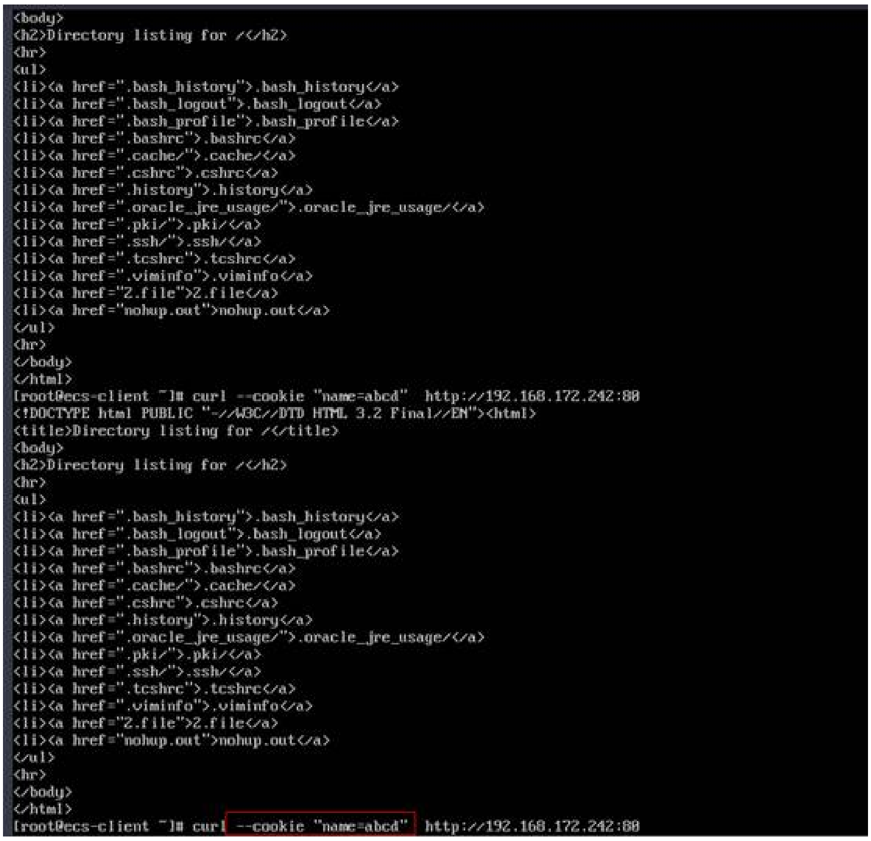


```
[root@ecs-cloud-0002 ~]# touch 2.file
[root@ecs-cloud-0002 ~]# nohup python -m SimpleHTTPServer 80 &
[1] 15244
[root@ecs-cloud-0002 ~]# nohup: ignoring input and appending output to 'nohup.out'

[root@ecs-cloud-0002 ~]# curl 127.0.0.1:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-cloud-0002 ~]#
```

3. Access the load balancer from the client and specify the cookie value.

   The following is an example command. Change the parameters as needed. Ensure that the returned file names of each request are the same.

   curl --cookie "name=abcd" http://ELB_IP:Port



```
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-client ~]# curl --cookie "name=abcd"  http://192.168.172.242:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-client ~]# curl --cookie "name=abcd"  http://192.168.172.242:80
```

13.9.4 What Types of Sticky Sessions Does ELB Support?

ELB supports sticky sessions source IP address, load balancer cookie, and application cookie.

13.10 Certificates

13.10.1 How Can I Create Server Certificates and CA Certificates?

Refer to to create server certificates and CA certificates. Generally, only backend servers need to be authenticated. You only need to configure server certificates.

13.10.2 Does ELB Support Wildcard Certificates?

Yes.

Dedicated load balancers using a SNI certificate support wildcard match by default. Only the subdomain names of the same level can be matched. You can change wildcard match to longest suffix match by changing the value of **sni_match_algo**. For details, see *Elastic Load Balance API Reference*.

Table 13-9 Examples of wildcard-domain matching rules

Domain Name	Wildcard Match	Longest Suffix Match
*.example.com	Domain names, such as abc.example.com, sport.example.com, and good.example.com	Domain names, such as abc.example.com and mycalc.good.example.com

13.10.3 Why Is Access to Backend Servers Still Abnormal Even If I Have Created a Certificate?

The following are possible causes:

- You have created a certificate on the ELB console, but you do not have an HTTPS listener.

  To solve this problem, perform the following steps:

  - Continue using the current listener and install the certificate on the backend server.
  - Delete the current listener, add an HTTPS listener, and bind a certificate to the HTTPS listener.

- You have created a certificate on the **Certificates** page and are using an HTTPS listener, but you have not bound the certificate to the listener.
- Your certificate has expired.

- The domain name is different from the one specified when you create the certificate.
- A certificate chain is used, but its format is incorrect.

13.10.4 Will the Network or Load Balancing Be Interrupted When a Certificate Is Being Replaced?

No.

The new certificate takes effect immediately after the replacement. The old certificate is used for established connections, and the new one is used for new connections.

NOTE

When the certificate expires, the system displays a message indicating that the connection is insecure. However, you can ignore the warning and continue accessing the website.

13.11 Monitoring

13.11.1 Why Is the Outgoing Rate on the ELB Console Inconsistent with the Bandwidth Usage Statistics on the Cloud Eye Console?

In the following scenarios, outgoing rate monitored by ELB is inconsistent with EIP bandwidth usage statistics on Cloud Eye:

- If the traffic does not exceed the bandwidth set for the EIP, the bandwidth is not limited and Cloud Eye collects statistics on the public network while ELB collects data on the private network.
- If the traffic exceeds the bandwidth set for the EIP, the bandwidth is limited. Traffic to the ELB system passes through a path that is different from the path in which traffic passes to the EIP.

13.11.2 What Are the Differences Between Layer-7 Status Codes and Backend Status Codes in ELB Metrics?

HTTP or HTTPS listeners terminate TCP connections. In other words, there are two TCP connections between the client and a backend server, one between the client and load balancer, and the other between the load balancer and backend server. The communication between the client and the backend server is divided into two parts. After receiving an HTTP request, the load balancer parses the request and routes the parsed request to the backend server for processing. The backend server returns a response to the load balancer after receiving the request. The load balancer then parses the response and returns the parsed response to the client. Therefore, there are two types of status codes: backend status codes returned by the backend server to the load balancer and Layer-7 status codes returned by the load balancer to the client.

You may encounter the following situations:

- The backend server returns a status code, and the load balancer directly transmits the status code to the client. In this case, the Layer-7 status code is the same as the backend status code.
- If the connection between the load balancer and backend server is abnormal or times out, the load balancer returns HTTP 502 or 504 to the client.
- If the listener configuration or the request format or content is incorrect, the load balancer directly returns an HTTP 4xx status code or 502 to the client, and does not route the request to the backend server. In this case, there will be only a Layer-7 status code, but no backend status code.

13.11.3 Why Is There a Large Number of HTTP 499 Errors?

When you are seeing the HTTP 499 status code, the client has closed the connection while the server is still processing the request.

The possible causes are as follows:

- The request timeout may not be long enough for the client to send HTTP requests before a connection is closed. Check the **request_time** field in the access log to view the total time for processing requests and set an appropriate request timeout.
- Your load balancer may be overloaded with traffic, causing packet loss due to bandwidth limit. Check the outbound bandwidth usage of your load balancer on the Cloud Eye console. For more information, see **Monitoring Metrics**.
- The network that connects the client and your load balancer may be unstable, causing long round-trip delay or packet loss. Check the **request_time** and **tcpinfo_rtt** fields in the access log or capture packets to check whether the network is normal.
- The backend server may take a longer time than the request timeout interval to process requests. Check whether the CPU, memory, and network of the backend server have performance bottlenecks.
- The client closes the connection before receiving a response from the server due to some unknown reasons. Check whether the client closes the connection before an HTTP request is complete.

14 Change History

Released On	Description
2024-04-15	This issue is the first official release.