Elastic Load Balance User Guide

User Guide

Issue 01

Date 2025-09-25

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

Notice

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

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

Contents

1 User Guide for Dedicated Load Balancers

1.1 Using a Dedicated Load Balancer

If you are using a dedicated load balancer for the first time, you can start from this section.

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

ELB Architecture

Figure 1-1 ELB architecture

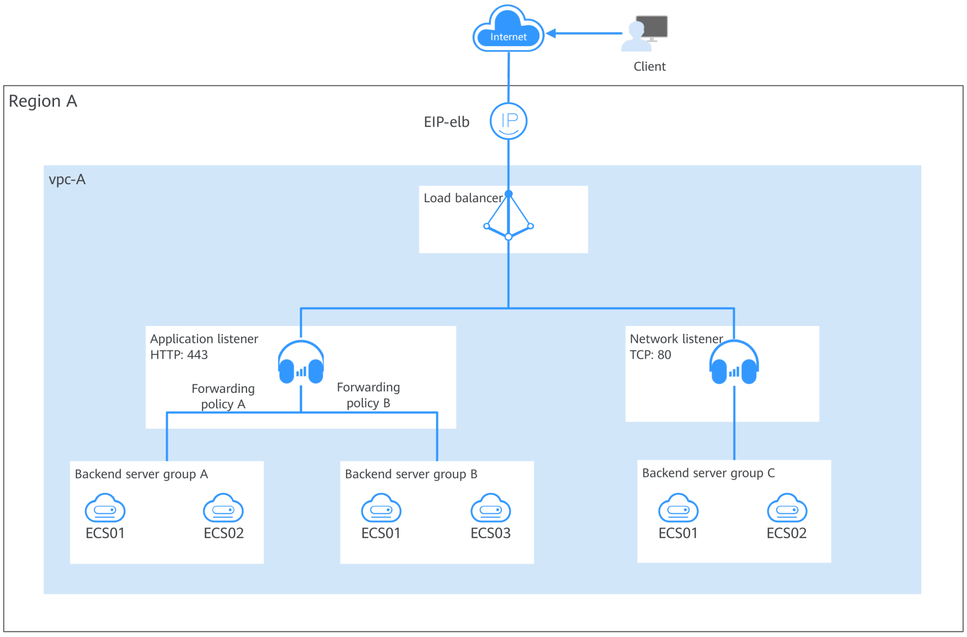


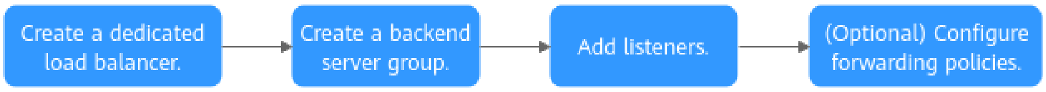
Table 1-1 ELB components

Component	Description	Helpful Links
Load balancer	Distributes incoming traffic across backend servers in one or more AZs. Before using a load balancer, you need to add at least one listener to it.	**Dedicated Load Balancer Overview**
Listener	Works as the minimum service unit. It uses a protocol and port (for example, TCP port 80) you have specified to check requests from clients and route the requests to associated backend servers. Each load balancer must have at least one listener to check and distribute traffic. You can add different types of listeners to distribute traffic using different protocols and ports. Network listeners forward traffic to the default backend server group, while application listeners forward traffic based on the forwarding policies you configure.	**Listener Overview**
Forwarding policy	Determines how application load balancers distribute traffic across one or more backend server groups. Forwarding policies can be only configured for application listeners. Application load balancers distribute Layer 7 requests more efficiently. They support various protocols and forwarding policies to suit your service needs.	**Advanced Forwarding**
Backend server group	Contains one or more backend servers to process requests distributed by load balancers. A backend server group can be created independently. A backend server group can be associated with one or more load balancers.	**Backend Server Group Overview**
Backend server	Processes client requests. A backend server can be an ECS, BMS, supplementary network interface, or IP address. If a supplementary network interface or an IP address is added as a backend server, the server with the supplementary network interface attached or using the IP address processes client requests. 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 identified as unhealthy, the load balancer will stop routing requests to it.	**Backend Server Overview**

Procedure for Using a Dedicated Load Balancer

The following describes how to quickly create and use a dedicated load balancer.

Figure 1-2 Procedure for using a dedicated load balancer


Create a dedicated load balancer.
Create a backend server group.
Add listeners.
(Optional) Configure forwarding policies.

Procedure	What to Do
Creating a Dedicated Load Balancer	Create a dedicated load balancer and be careful with the following configurations: ● Basic information: type, billing mode, region, and AZ. ● Specifications: elastic or fixed specifications; network or application load balancing, or both. ● Network configuration: network type (private IPv4 or IPv6), VPC, and subnet planning.
Creating a Backend Server Group	Create a backend server group and add backend servers to the group for easier management and scheduling. You can create a backend server group first and select it when creating a dedicated load balancer. Plan the backend protocol appropriately because the backend protocol of each backend server group must match the frontend protocol of the associated listeners.
● **Network Listeners** ● **Application Listeners**	Add listeners and choose the protocols and ports based on service requirements. ● Application listeners (HTTP/HTTPS): work well for workloads that require high performance at Layer 7, such as real-time audio and video, interactive livestreaming, and game applications. ● Networking listeners (TCP/UDP): are good for heavy-traffic and high-concurrency workloads at Layer 4, such as file transfer, instant messaging, and online video applications.
Advanced Forwarding	Configure advanced forwarding policies for application listeners to forward traffic to specified backend server groups based on the domain name, path, HTTP request method, HTTP header, query string, and CIDR block.

Backend Server Group and Listener Protocols

You can associate a backend server group with different listeners or different dedicated load balancers under the same enterprise project.

The backend protocol of each backend server group must match the frontend protocol of the associated listeners as described in **Table 1-2**.

Table 1-2 The frontend and backend protocol

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

1.2 Permissions Management

1.2.1 Creating a User and Granting Permissions

Use **IAM** to implement fine-grained permissions control over your ELB resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing ELB resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust another account or cloud service to perform efficient O&M on your ELB resources.

Skip this section if your account does not need individual IAM users.

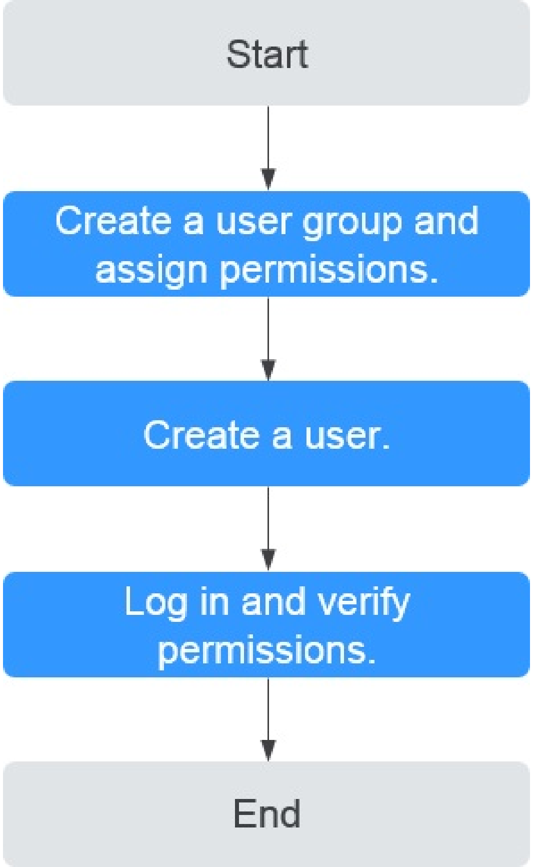
This following describes the procedure for granting permissions.

Prerequisites

You have learned about ELB policies and can select the appropriate policies based on service requirements. Learn about **permissions** supported by ELB. For the permissions of other services, see **System Permissions**.

Process Flow

Figure 1-3 Process for granting ELB permissions


Start
Create a user group and assign permissions.
Create a user.
Log in and verify permissions.
End

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the **ELB ReadOnlyAccess** policy to the group.

2. **Create a user and add it to a user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the ELB console by using the created user, and verify that the user only has read permissions for ELB.

   - Choose **Service List** > **Elastic Load Balance**. Then click **Buy Elastic Load Balancer** on the ELB console. If you cannot create a load balancer, the **ELB ReadOnlyAccess** policy has taken effect.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **ELB ReadOnlyAccess** policy has already taken effect.

1.2.2 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of ELB. For the actions supported for custom policies, see "Permissions Policies and Supported Actions" in the **Elastic Load Balance API Reference**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details, see **Creating a Custom Policy**. The following section contains examples of common ELB custom policies.

Example Custom Policies

- Example 1: Allowing users to update a load balancer

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elb:loadbalancers:put"
            ]
        }
    ]
 }
```

- Example 2: Denying load balancer deletion

  A deny policy must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

  If you grant the system policy **ELB FullAccess** to a user but do not want the user to have the permission to delete load balancers defined in the policy, you can create a custom policy that rejects the deletion of load balancers and grant the **ELB FullAccess** and deny policies to the user, so that the user can perform all operations on ELB except deleting load balancers. The following is an example deny policy:

```
{
      "Version": "1.1",
      "Statement": [
            {
          "Effect": "Deny",
                  "Action": [
                        "elb:loadbalancers:delete"
                  ]
            }
      ]
 }
```

- Example 3: Defining permissions for multiple services in a policy

  A custom policy can contain the actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elb:loadbalancers:get",
                "elb:loadbalancers:list",
                "elb:loadbalancers:delete",
                "ecs:cloudServers:delete"
            ]
    }
  ]
}
```

1.3 Load Balancer

1.3.1 Dedicated Load Balancer Overview

A load balancer automatically distributes incoming traffic across multiple backend servers based on the routing policies you configure. It expands the service availability and scalability of your applications. You can plan the load balancer configurations by referring to this section.

Region

- Resources in different regions cannot communicate with each other over internal networks. For lower network latency and faster access to resources, select the region nearest to where your services will be accessed.
- You can add servers in a different VPC from where the load balancer is created, or in an on-premises data center, by using private IP addresses of the servers. For details, see **Adding Backend Servers in a Different VPC from a Load Balancer**.

AZ

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

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

Table 1-3 Disaster recovery planning

DR Solution	Application Scenario	Advantage
Select multiple AZs for a load balancer.	If the number of requests does not exceed what the largest specifications can handle, you can create a load balancer and select multiple AZs.	If the load balancer in an AZ goes down, the load balancer in other AZs takes over to route traffic.
Create multiple load balancers and select multiple AZs for each load balancer.	If the number of requests exceeds what the largest specifications can handle, you can create multiple load balancers and select multiple AZs for each load balancer.	If a load balancer in an AZ goes down, another load balancer in the same AZ or other AZs takes over to distribute traffic.

Table 1-4 Traffic distribution

Source	Traffic Distribution
Internet	If requests are from the Internet, the load balancer in each AZ you select routes the requests based on source IP addresses. If you select two AZs for a load balancer, the requests the load balancers can handle will be doubled.
Private network	● If clients are **in the same AZ as the load balancer**, requests are distributed by the load balancer in this AZ. If the load balancer goes down, requests are distributed by the load balancer in another AZ. If the load balancer is healthy but the connections that the load balancer needs to handle exceed the amount defined in the specifications, service may be interrupted. To address this issue, you need to upgrade specifications. You can monitor traffic usage on private networks by AZ. ● If clients are **in an AZ that is different from the load balancer**, requests are distributed by the load balancer in each AZ you select based on source IP addresses.
Direct Connect connection	If requests are from a Direct Connect connection, the load balancer **in the same AZ as the Direct Connect connection** routes the requests. If the load balancer in this AZ goes down, requests are distributed by the load balancer in another AZ.
A VPC that is different from where the load balancer works	If the client is in a VPC that is different from where the load balancer works, the load balancer in the AZ where the client subnet works routes the requests. If the load balancer in this AZ goes down, requests are distributed by the load balancer in another AZ.

Specifications

Network load balancers can route TCP or UDP requests, while application load balancers route HTTP or HTTPS requests.

Select appropriate specifications based on your traffic volume and service requirements.

For details, see **Table 1-5**.

Table 1-5 Guide for selecting a specification

Specifications	Description
Network load balancing (TCP/UDP)	Pay attention to the maximum number of concurrent connections and consider maximum concurrent connections as a key metric. Estimate the maximum number of concurrent connections that a load balancer needs to handle and select the corresponding specification.

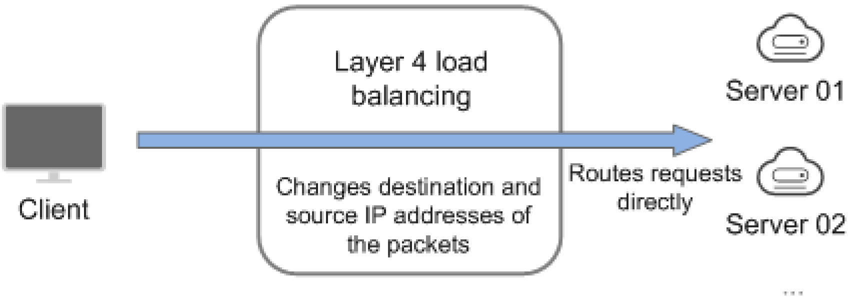
Specifications	Description
Application load balancing (HTTP/HTTPS)	Consider QPS as a key metric, which determines the service throughput of an application system. Estimate the QPS that a load balancer needs to handle and select the corresponding specification.

Protocols

ELB provides load balancing at both Layer 4 and Layer 7. Choose an appropriate protocol when you add a listener to a load balancer.

- Network load balancers work well for heavy-traffic workloads that need to handle massively concurrent requests at Layer 4, such as file transfer, instant messaging, and online video services.

Figure 1-4 Layer 4 load balancing



- Application load balancers handle Layer 7 requests and support advanced forwarding policies.

Figure 1-5 Layer 7 load balancing

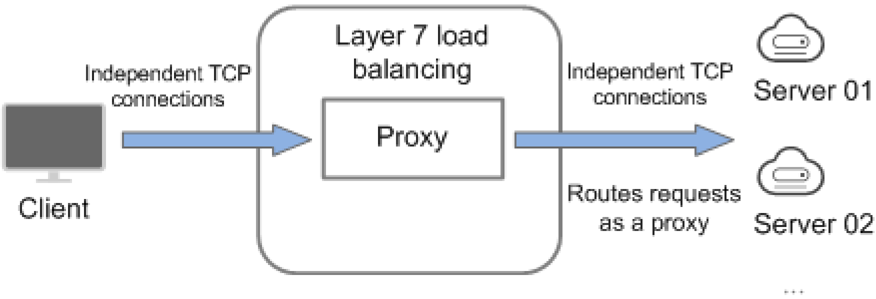

Layer 7 load balancing
Independent TCP connections
Client
Proxy
Independent TCP connections
Routes requests as a proxy
Server 01
Server 02
...

Table 1-6 Protocols

Protocol	Description
TCP/UDP	After receiving a request, the listener routes it directly to backend servers. In this process, the destination IP address in a packet 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.
HTTP/HTTPS	Once the load balancer receives a request, it works as a proxy for backend servers and initiates a connection (three-way handshake) with the client. It then determines which backend server to route the request to based on the fields in the HTTP/HTTPS request header and the load balancing algorithm you select when you add the listener.

NOTE

ELB establishes persistent connections between the clients and the load balancers to reduce the costs of a large number of short connections. After a persistent connection is established, the client can keep sending HTTP or HTTPS requests to the load balancer until the connection times out.

Network Type

Dedicated load balancers can work on both public and private network.

Table 1-7 ELB network types

Network Type	Note	Application Scenarios
Load balancing on a public network	You need to bind an EIP or global EIP to this type of load balancers. They can receive requests from the Internet and route the requests to backend servers.	● 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.

Network Type	Note	Application Scenarios
Load balancing on a private network	This type of load balancers has only private IP addresses and can be only accessed within a VPC. They receive requests from clients in a VPC and route the requests across backend servers in the same VPC.	● 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.

Backend Server

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 cloud servers, note the following:

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

1.3.2 Creating a Dedicated Load Balancer

Scenarios

ELB distributes heavy incoming traffic across backend servers, maintaining high service availability at both network and application layers. It provides multiple load balancing algorithms and health checks to keep your services running smoothly.

This section describes how to create a dedicated load balancer. Before that, ensure you have gotten everything ready. For details, see **Dedicated Load Balancer Overview**.

Procedure

1. Go to the **Buy Elastic Load Balancer** page.
2. Complete the basic configurations based on **Table 1-8**.

Table 1-8 Parameters for configuring the basic information

Parameter	Description
Type	Specifies the type of the load balancer. **The type cannot be changed after the load balancer is created**. Dedicated load balancers work well for heavy-traffic and high-concurrency workloads, such as large websites, cloud native applications, IoV, and multi-AZ disaster recovery applications.
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.
AZ	Specifies the AZs where the dedicated load balancer works. An AZ is a part of a region and has its own independent power supplies and networks. AZs are physically isolated but interconnected through internal networks. You can select multiple AZs for a load balancer to ensure high availability. If the load balancer in an AZ goes down, the load balancer in another AZ routes requests to backend servers to ensure service continuity and improve application reliability. For details about AZ planning, see **AZ**. If you select multiple AZs for a load balancer, its performance, such as the number of new connections and concurrent connections, will multiply by the number of AZs. For example, a dedicated load balancer in an AZ can handle 20 million concurrent connections. If you select two AZs for a dedicated load balancer, it can handle up to 40 million concurrent connections. To reduce network latency and improve access speed, you are recommended to deploy your load balancer in the AZ where backend servers are running.
Name	Specifies the load balancer name. The name can contain: ● 1 to 64 characters. ● Letters, digits, underscores (_), hyphens (-), and periods (.).
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

3. Select specifications for the dedicated load balancer based on **Table 1-9**.

Table 1-9 Load balancer specifications

Parameter	Description
Specifications	Select **Elastic** or **Fixed** if pay-per-use is chosen as the billing mode. ● Specification type – **Elastic specifications** work well for fluctuating traffic, and you will be charged for how many LCUs you use. – **Fixed specifications** are suitable for stable traffic, and you will be charged for the specifications you select. ● Load balancing type – **Application load balancing (HTTP/HTTPS)**: supports HTTP and HTTPS. This option is a great fit for workloads that require high performance at Layer 7, such as real-time audio and video, interactive livestreaming, and game applications. – **Networking load balancing (TCP/UDP)**: supports TCP and UDP. This option works well for heavy-traffic and high-concurrency workloads at Layer 4, such as file transfer, instant messaging, and online video applications. Select either application load balancing or network load balancing, or both, and then select the desired specification. You can select only one specification for each load balancing type. Select the desired specifications based on your service size.

4. Complete the network configurations based on **Table 1-10**.

Table 1-10 Configuring network parameters

Parameter	Description
Network Type	Specifies the network where the load balancer works. You can select one or more network types. If you do not select any option, no IP address will be assigned to the load balancer and 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. ● **Private IPv4 network**: The load balancer routes IPv4 requests from the clients to backend servers in a VPC. If you want the load balancer to route requests from the Internet, bind an EIP to the load balancer. ● **IPv6 network**: An IPv6 address will be assigned to the load balancer to route requests from IPv6 clients.
VPC	Specifies the VPC where the dedicated load balancer works. **You cannot change the VPC after the load balancer is created**. Plan the VPC as required. Select an existing VPC, or click **View VPCs** to create a desired one. For more information about VPC, see the **Virtual Private Cloud User Guide**.
Frontend Subnet	Specifies the frontend subnet from which an IP address will be assigned to the dedicated load balancer to **communicate with resources over the private network**. After the load balancer is created, you can unbind the existing IP address and bind IPv4 and IPv6 addresses in a different subnet to the load balancer. Unbinding an IP address may affect service running. IP addresses will be assigned to the load balancer based on the network type you configure. ● **Private IPv4 network**: IPv4 private addresses will be assigned. ● **IPv6 network**: IPv6 addresses will be assigned. If you select **IPv6 network** for **Network Type** and the selected VPC does not have any subnet that supports IPv6, enable IPv6 for at least one subnet or create a subnet that supports IPv6. For details, see the **Virtual Private Cloud User Guide**.

Parameter	Description
IPv4 Address	Specifies how you want the IPv4 address to be assigned if **Network Type** is set to **Private IPv4 network**. ● **Automatically assign IP address**: The system assigns an IPv4 address to the load balancer. ● **Manually specify IP address**: You need to manually specify an IPv4 address for the load balancer. NOTE Network ACL rules configured for the frontend subnet of a load balancer do not restrict traffic from clients to the load balancer. Use access control to limit which IP addresses can access the load balancer. For details, see **What Is Access Control?**
Backend Subnet	Specifies the backend subnet from which IP addresses will be assigned to the dedicated load balancer to **forward requests to and perform health checks on backend servers**. ● **Subnet of the load balancer** is selected by default. ● You can select other subnets in the VPC of the load balancer or click **Create Subnet** to create a subnet. NOTE ● If IPv6 is not enabled for the backend subnet you select when creating a dedicated load balancer, the load balancer cannot use IPv6 addresses to route requests. ● The number of IP addresses required by a load balancer to communicate with the backend servers depends on how many AZs you have selected, how you configure the specifications, and whether you enable the IP as a backend option. See how many IP addresses are actually required on the console. ● An application load balancer requires 8 to 30 additional IP addresses in the backend subnet for forwarding traffic. 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.
IPv6 Address	Specifies how you want the IPv6 address to be assigned if **Network Type** is set to **IPv6 network**. ● **Assign automatically**: The system automatically assigns an IPv6 address to the load balancer. ● **Manually specify**: You need to manually specify an IPv6 address for the load balancer. NOTE Network ACL rules configured for the frontend subnet of a load balancer do not restrict traffic from clients to the load balancer. Use access control to limit which IP addresses can access the load balancer. For details, see **What Is Access Control?**

Parameter	Description
Shared Bandwidth	Specifies the shared bandwidth that the IPv6 address will be added to. A shared bandwidth allows multiple EIPs in the same region to share the same bandwidth. You can choose not to select a shared bandwidth, select an existing shared bandwidth, or buy a shared bandwidth.
IP as a Backend	Specifies whether to associate backend servers with the load balancer by using their IP addresses. After this option is enabled, you can associate backend servers that are not in the VPC of the load balancer by referring to **Adding Backend Servers in a Different VPC from a Load Balancer**. If you enable this option, more IP addresses in the backend subnet need to be reserved for the load balancer to communicate with backend servers. Ensure that the selected subnet has sufficient IP addresses.

5. Configure an EIP for the load balancer to enable it to route IPv4 requests over the Internet based on **Table 1-11**.

Table 1-11 Selecting an EIP for the load balancer

Parameter	Description
EIP	Specifies the EIP that will be bound to the load balancer for receiving and forwarding requests over the Internet. ● **Auto assign**: A new EIP will be assigned to the load balancer. ● **Use existing**: Select an existing EIP. ● **Not required**: You can bind an EIP to the load balancer later.
EIP Type	Specifies the link type (BGP) when a new EIP is used.
Billed By	Specifies how the bandwidth will be billed. You can select one from the following options: ● **Bandwidth**: You specify the maximum bandwidth and pay for the amount of time you use the bandwidth. ● **Traffic**: You specify the maximum bandwidth and pay for the outbound traffic you use. ● **Shared Bandwidth**: Load balancers that have EIPs bound in the same region can share the selected bandwidth, helping you reduce public network bandwidth costs.

Parameter	Description
Bandwidth (Mbit/s)	Specifies the maximum bandwidth.

6. Configure other parameters for the load balancer as described in **Table 1-12**.

Table 1-12 Configuring other parameters

Parameter	Description
Advanced Settings (Optional) > Description	Click ⌄ to expand the configuration area and set this parameter. Enter a description about the load balancer in the text box as required. Enter up to 255 characters. Angle brackets (<>) are not allowed.
Advanced Settings (Optional) > Tag	Click ⌄ to expand the configuration area and set this parameter. Add tags to the load balancer so that they can be easily found. A tag consists of a tag key and a tag value. The tag key marks a tag, and the tag value specifies specific tag content. For details about the naming rules, see **Table 1-13**. You can add a maximum of 20 tags.

Table 1-13 Tag key and value requirements

Parameter	Requirement
Tag key	• Cannot be empty. • Must be unique for the same load balancer. • Can contain a maximum of 36 characters. • Can contain only letters, digits, underscores (_), hyphens (-), at signs (@).
Tag value	• Can contain a maximum of 43 characters. • Can contain only letters, digits, underscores (_), hyphens (-), at signs (@).

7. Click **Buy Now**.
8. Return to the load balancer list page to check the new load balancer.

   To ping the IP address of this load balancer, you need to add a listener to it.

1.3.3 Configuring Modification Protection or Deletion Protection for Dedicated Load Balancers

You can enable modification protection or deletion protection for load balancers to prevent them from being modified or deleted by accident.

- **Deletion protection**: prevents your load balancers from being deleted by accident. Disable **Deletion Protection** if you want to delete a load balancer.
- **Modification protection**: prevents your load balancers from being modified by accident. Disable **Modification Protection** if you want to modify or delete a load balancer.

Enabling or Disabling Deletion Protection

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Switch to the **Summary** tab of the load balancer and enable or disable **Deletion Protection**.

   ⚠ CAUTION

   If your load balancer is managed by CCE, modifying the load balancer will affect the running of the CCE cluster.

4. After deletion protection is enabled, the load balancer cannot be deleted. Other operations are not affected.

Enabling or Disabling Modification Protection

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Switch to the **Summary** tab and click **Configure** next to **Modification Protection**.
4. In the **Configure Modification Protection** dialog box, enable or disable **Modification Protection**.

   Fill in the reason if needed.

   ⚠ CAUTION

   If your load balancer is managed by CCE, modifying the load balancer will affect the running of the CCE cluster.

5. Click **OK**.
6. After modification protection is enabled, you cannot modify or delete the load balancer. Other operations are not affected.

1.3.4 Modifying the Basic Configurations of a Dedicated Load Balancer

As your service develops, the service traffic might surge, service types might change, and backend services might have to be migrated. If these happen, you can change the basic information of a dedicated load balancer, such as its specifications and AZ.

Modifying Specifications

A load balancer with elastic specifications has obvious advantages over the one with fixed specifications in scaling scenarios. If your service fluctuates greatly, you can use load balancers with elastic specifications to simplify management and reduce O&M complexity.

If you use a load balancer with fixed specifications and the specifications do not match your service needs, you can upgrade the specifications to ensure stable service running or downgrade the specifications to save costs. If your service type changes, you can also change the load balancing type of the load balancer.

On the console, you can:

- **Change elastic specifications to fixed specifications**, or the other way round.
- **Change application load balancing to network load balancing**, or the other way round.

  You must keep at least one load balancing type. Before removing a load balancing type, you must delete the:

  – HTTP or HTTPS listeners added to an application load balancer.

  – TCP or UDP listeners added to a network load balancer.
- **Upgrade or downgrade the fixed specifications**, for example, upgrade small I to medium I, or downgrade large I to medium I.

⚠ WARNING

- Upgrading specifications does not interrupt your services.
- Downgrading specifications will temporarily disconnect services.
  - New TCP/UDP connections may fail to be established.
  - New HTTP/HTTPS connections may fail to be established and some persistent connections may be interrupted.

Pay-per-Use

Table 1-14 Supported change options for a pay-per-use load balancer

Billing Mode	Specification Type	Change to Elastic	Change to Fixed	Add Load Balancing Type	Remove Load Balancing Type	Upgrade Specifications	Downgrade Specifications
Pay-per-use	Elastic	N/A	Supported	Supported	Supported	N/A	N/A
	Fixed	Supported	N/A	Supported	Supported	Supported	Supported

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer, click **More** in the **Operation** column, and select **Change Specifications**.
3. Select the new specifications and click **Next**.
4. Confirm the information and click **Submit**.
5. On the load balancer list page, check the new specifications in the **Specifications** column of the target dedicated load balancer.

Can I Change the Load Balancing Type of a Load Balancer?

Yes, you can change an application load balancer to a network load balancer, or the other way around.

Does Changing Specifications Interrupt Services?

Upgrading specifications does not interrupt your services, but downgrading specifications temporarily does.

1.3.5 Modifying the Network Configurations of a Dedicated Load Balancer

A load balancer communicates with external networks through its private or public IP addresses, which are used to forward traffic. You can follow this section to change the IP addresses of your load balancer for upgrade, security, or compliance purposes.

Network Type

A load balancer can work on **public and private networks**.

- **Public network load balancer**: You need to bind EIPs or global EIPs to this type of load balancers, so that they can receive requests from the Internet and route the requests to backend servers.

- **Private network load balancer**: This type of load balancers receives requests from clients in a VPC and route the requests across backend servers in the same VPC.

Binding or Unbinding an IP Address

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

An **IPv4 EIP**, a **private IPv4 address**, or an **IPv6 address** can be bound to or unbound from a load balancer.

⚠ CAUTION

After an IP address is unbound, the load balancer cannot use this IP address to forward traffic.

Binding or Unbinding an IPv4 EIP

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
   a. Binding an IPv4 EIP
      i. Click **Bind IPv4 EIP**.
      ii. In the **Bind IPv4 EIP** dialog box, select the EIP you want to bind to the load balancer and click **OK**.
   b. Unbinding an IPv4 EIP
      i. Click **Unbind IPv4 EIP**.
      ii. In the displayed dialog box, confirm the IPv4 EIP that you want to unbind and click **OK**.

Binding or Unbinding a Private IPv4 Address

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
   a. Binding a private IPv4 address
      i. Click **Bind Private IPv4 Address**.
      ii. In the **Bind Private IPv4 Address** dialog box, select the subnet where the IP address resides, specify an IP address, and click **OK**.

         NOTE

         - By default, an IP address is automatically assigned. To manually specify an IP address, deselect **Automatically assign IP address** and enter an IP address.
         - Ensure that the specified IP address is in the selected subnet and is not in use.

   b. Unbinding a private IPv4 address

      i. Click **Unbind IPv4 Private IPv4 Address**.

      ii. In the displayed dialog box, confirm the private IPv4 address that you want to unbind and click **OK**.

Binding or Unbinding an IPv6 Address

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.

   a. Binding an IPv6 address

      i. Click **Bind IPv6 Address**.

      ii. In the **Bind IPv6 Address** dialog box, select the subnet where the IP address resides and click **OK**.

   b. Unbinding an IPv6 address

      i. Click **Unbind IPv6 Address**.

      ii. In the displayed dialog box, confirm the IPv6 address that you want to unbind and click **OK**.

Changing an IP Address

Before changing the private IPv4 address or IPv6 address bound to a dedicated load balancer, note the following:

- The new IPv4 IP address can be in the current subnet or a different subnet.
- The new IPv6 IP address must be in a different subnet with IPv6 enabled.

Changing a Private IPv4 Address

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and choose **More** > **Change Private IPv4 Address** in the **Operation** column.
3. In the **Change Private IPv4 Address** dialog box, select the subnet where the IP address resides and specify an IP address.
   - To use an IP address in another subnet, if you select **Automatically assign IPv4 address**, an IPv4 address will be assigned to your load balancer.
   - To use another IP address from the current subnet, specify an IP address.
4. Click **OK**.

Changing an IPv6 Address

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and choose **More** > **Change IPv6 Address** in the **Operation** column.
3. In the **Change IPv6 Address** dialog box, select a different subnet where the IP address resides and specify an IP address.

The system will automatically assign an IPv6 address to the load balancer from the subnet you select.

4. Click **OK**.

Modifying the Bandwidth

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

⚠ CAUTION

- When modifying bandwidth, you need to change the specifications of the dedicated load balancer to avoid speed limit due to insufficient bandwidth.
- The EIP bandwidth defines the limit for clients to access the load balancer.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
3. Click **Modify IPv4 Bandwidth** or **Modify IPv6 Bandwidth**.
4. In the **New Configuration** area, modify the bandwidth size and click **Next**.

   You can select the bandwidth defined by the system or customize a bandwidth. The bandwidth ranges from 1 Mbit/s to 2,000 Mbit/s.
5. Confirm the new bandwidth and click **Submit**.

   NOTE

   After you change the billing option and bandwidth, the price will be recalculated accordingly.

Adding or Removing an IPv6 Address to or from a Shared Bandwidth

If the IPv6 address of a load balancer is added to a shared bandwidth, the load balancer can route IPv6 requests over the Internet.

You can add or remove an IPv6 address to or from a shared bandwidth.

NOTE

If the IPv6 address of a load balancer is removed from a shared bandwidth, the load balancer can only route IPv6 requests within a VPC.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
   a. Adding an IPv6 address to a shared bandwidth
      i. Click **Add to IPv6 Shared Bandwidth**.
      ii. In the **Add to IPv6 Shared Bandwidth** dialog box, select the shared bandwidth to which you want to add.

If no shared bandwidth is available, assign one as prompted.

b. Removing an IPv6 address from a shared bandwidth

i. Click **Remove from IPv6 Shared Bandwidth**.

ii. In the displayed dialog box, confirm the shared bandwidth you want to remove.

3. Click **OK**.

1.3.6 Exporting Dedicated Load Balancers

Scenarios

You can export the information of all or part of the load balancers in your account as an Excel file to a local directory.

You can export:

- The basic information of all or selected load balancers.
- The details of the selected load balancers.

Basic information includes the name, ID, status, type, and specifications of the load balancers.

Details include the basic information of load balancers and listeners by default. In addition, the forwarding policies, backend server groups, backend servers, and certificate names/IDs can also be exported.

Exporting the Basic Information of Load Balancers

1. Go to the **load balancer list page**.
2. In the upper left corner of the load balancer list, click **Export**.

   a. **Basic information of all resources**: The system automatically exports the basic information of all the load balancers in the current region as an Excel file to a local directory.

   b. **Basic information of selected resources**: The system automatically exports the basic information of the selected load balancers in the current region as an Excel file to a local directory.

1.3.7 Deleting or Unsubscribing from Dedicated Load Balancers

Scenarios

You can delete or unsubscribe load balancers if you on longer need them.

Deleting One or More Pay-per-Use Load Balancers

When deleting load balancers, you can select the following options based on your service requirements:

- Release the EIPs together to avoid unnecessary charges.
- Delete the associated backend server groups. (If a backend server group is associated with other load balancers, it cannot be deleted.)

Deleting a Pay-per-Use Load Balancer

1. Go to the **load balancer list page**.
2. On the displayed page, locate the target load balancer and choose **More** > **Delete** in the **Operation** column.

   A confirmation dialog box is displayed.
3. In the displayed dialog box, enter **DELETE**.
4. Click **OK**.

1.4 Listener

1.4.1 Listener Overview

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

Supported Protocols and Application Scenarios

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

You can select TCP or UDP for network load balancing and HTTP or HTTPS for application load balancing.

Table 1-15 Protocols supported by ELB

Type	Protocol	Description	Application Scenario
Network listeners	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 do not need to handle a large number of concurrent requests and do not require high performance
Network listeners	UDP	● Relatively low reliability ● Fast data transfer	Scenarios that require quick response, such as video chat, gaming, and real-time financial news
Application listeners	HTTP	● Cookie-based sticky sessions ● X-Forward-For request header	Applications that require content identification, for example, web applications and mobile games

Type	Protocol	Description	Application Scenario
Application listeners	HTTPS	● An extension of HTTP for encrypted data transmission that can prevent unauthorized access ● Encryption and decryption performed on load balancers ● Multiple versions of encryption protocols and cipher suites	Workloads that require encrypted transmission, such as e-commerce and financial services

Frontend Protocols and Ports

Frontend protocols and ports are used by load balancers to receive requests from clients.

Load balancers use TCP or UDP for network load balancing, and HTTP or HTTPS for application load balancing. Select protocols and ports that best suit your requirements.

⚠ CAUTION

The frontend protocols and ports cannot be changed once a listener is added. If you want to use a different protocol and port, add another listener.

Table 1-16 Frontend protocols and ports

Frontend Protocol	TCP, UDP, HTTP, or HTTPS
Frontend Port	Listeners using different protocols of a load balancer cannot use the same port. However, UDP listeners can use the same port as those using other protocols. For example, if there is a UDP listener that uses port 88, you can add a TCP listener that also uses port 88. The port ranges from 1 to 65535. Common ports: TCP/80 and HTTPS/443

Backend Protocols and Ports

Backend protocols and ports are used by backend servers to receive requests from load balancers. If Windows servers have Internet Information Services (IIS) installed, the default backend protocol and port are HTTP and 80.

Table 1-17 Backend protocols and ports

Backend Protocol	TCP, UDP, HTTP, HTTPS, QUIC
Backend Port	Backend servers of a load balancer can use the same port. The port ranges from 1 to 65535. Common ports: TCP/80, HTTP/80, and HTTPS/443

Forwarding by Port Ranges

Forwarding by Port Ranges is available only when you select TCP or UDP as the frontend protocol.

If this option is enabled, the listener checks requests from all ports in the port ranges you specify and routes them to the corresponding ports on the backend servers.

Timeout Durations

You can configure and modify timeout durations 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 prolong the request timeout duration to ensure that the request can be successfully routed.

Figure 1-6 Timeout durations at Layer 4

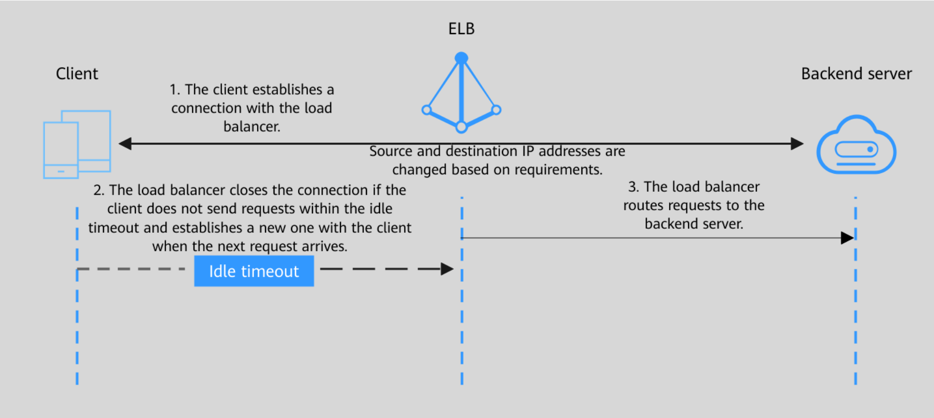

ELB
Client
Backend server
1. The client establishes a connection with the load balancer.
Source and destination IP addresses are changed based on requirements.
2. The load balancer closes the connection if the client does not send requests within the idle timeout and establishes a new one with the client when the next request arrives.
3. The load balancer routes requests to the backend server.
Idle timeout

Figure 1-7 Timeout durations at Layer 7

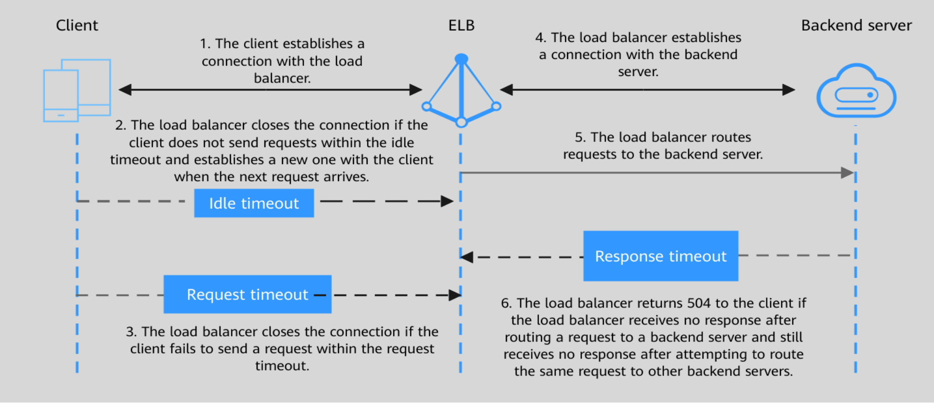

Client
ELB
Backend server
1. The client establishes a connection with the load balancer.
4. The load balancer establishes a connection with the backend server.
2. The load balancer closes the connection if the client does not send requests within the idle timeout and establishes a new one with the client when the next request arrives.
5. The load balancer routes requests to the backend server.
Idle timeout
Response timeout
Request timeout
3. The load balancer closes the connection if the client fails to send a request within the request timeout.
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.

Table 1-18 Timeout durations for Layer 4 listeners

Protocol	Type	Description	Value Range	Default Timeout Duration
• TCP • UDP	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.	10–4000s	300s

Table 1-19 Timeout durations for Layer 7 listeners

Protocol	Type	Description	Value Range	Default Timeout Duration
• HTTP • HTTPS	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.	0–4000s	60s

Protocol	Type	Description	Value Range	Default Timeout Duration
	Request Timeout	Specifies the length of time (in seconds) that a load balancer is willing to wait for a client request to finish. The load balancer terminates the connection if a request takes too long to complete.	1–300s	60s
	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. If sticky session is enabled and the load balancer receives no response from the backend server within the response timeout duration, the load balancer returns a 504 Gateway Timeout error to the client directly.	1–300s	60s

1.4.2 Network Listeners

1.4.2.1 Adding a TCP Listener

Scenarios

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

Constraints

- If the front protocol is TCP, the backend protocol defaults to TCP and cannot be changed.
- If you only select the application load balancing type for your dedicated load balancer, you cannot add TCP listeners to this load balancer.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on **Table 1-20**.

Table 1-20 Parameters for configuring a TCP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **TCP**.
Listening Port	Specifies a port or port ranges that will be used by the load balancer to receive requests from clients. ● **Single port**: The listener listens only on the specified port. ● **Port ranges**: The listener listens on all ports in the specified port ranges and routes the received packets to the corresponding ports on the backend servers, if the frontend protocol is TCP, UDP, or TLS. **NOTE** Listening on port ranges is available in certain regions. You can see which regions support this option on the console.
Name (Optional)	Specifies the listener name.

Parameter	Description
Transfer Client IP Address	This option is enabled for dedicated load balancers by default. When a TCP listener is used to forward requests, its load balancer communicates with backend servers using client IP addresses. In this case, you can check the backend server logs to obtain client IP addresses. Note that client IP addresses cannot be passed to IP as backend servers.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	
Idle Timeout (s)	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. Value range: 10–4000
Tag	Adds tags to the listener. Each tag is a key-value pair, and the tag key is unique.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-40**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-41**.

5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

1.4.2.2 Adding a UDP Listener

Scenarios

You can add a UDP listener, if quick response is required but low reliability is acceptable. UDP listeners are suitable for scenarios such as video chat, gaming, and real-time financial news.

Constraints

- UDP listeners do not support fragmentation.
- Any UDP packet larger than 1,500 bytes will be discarded. To avoid this, ensure that the MTU value of the network interface is not greater than 1,500 bytes and modify the configuration files of applications based on the MTU value.
- The backend protocol can be UDP or QUIC if the frontend protocol is UDP.
- If you only select the application load balancing type for your dedicated load balancer, you cannot add UDP listeners to this load balancer.
- When a UDP listener routes traffic to IP as backend servers in a UDP backend server group over a Direct Connect or VPN connection, the health check result may be unhealthy. In this case, **submit a service ticket**.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on **Table 1-21**.

Table 1-21 Parameters for configuring a UDP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **UDP**.

Parameter	Description
Listening Port	Specifies a port or port ranges that will be used by the load balancer to receive requests from clients. ● **Single port**: The listener listens only on the specified port. ● **Port ranges**: The listener listens on all ports in the specified port ranges and routes them to the corresponding ports on the backend servers, if the frontend protocol is TCP, UDP, or TLS.
Name (Optional)	Specifies the listener name.
Transfer Client IP Address	This option is enabled for dedicated load balancers by default. When a UDP listener is used to forward requests, its load balancer communicates with backend servers using client IP addresses. In this case, you can check the backend server logs to obtain client IP addresses. Note that client IP addresses cannot be passed to IP as backend servers.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	
Idle Timeout (s)	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**.
Tag	Adds tags to the listener. Each tag is a key-value pair, and the tag key is unique.

Parameter	Description
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-40**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-41**.
5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

1.4.2.3 Adding a UDP Listener (with a QUIC Backend Server Group Associated)

Scenarios

If you use UDP as the frontend protocol, you can select QUIC as the backend protocol, and select the connection ID algorithm to route requests with the same connection ID to the same backend server. QUIC is a great fit for the mobile Internet because it offers low latency, high reliability, and no head-of-line blocking (HOL blocking). Additionally, no new connections need to be established when you switch between a Wi-Fi network and mobile network.

Constraints

- You can add only UDP listeners if you want to use QUIC as the backend protocol.
- QUIC versions include Q043, Q046, and Q050.
- UDP listeners using QUIC as backend protocol do not support fragmentation.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.

   Select **Network load balancing (TCP/UDP/TLS)** for the load balancer.
3. On the **Listeners** tab, click **Add Listener**.
4. In the **Configure Listener** step, set **Frontend Protocol** to **UDP**, configure other parameters as required, and click **Next: Configure Request Routing Policy**.

5. In the **Configure Routing Policy** step, set **Backend Protocol** to **QUIC** and configure other parameters as required.
6. Configure the parameters and click **Submit.**

Related Operations

After you add a listener, associate backend servers with the listener by performing the operations in **Backend Server Overview**.

1.4.3 Application Listeners

1.4.3.1 Adding an HTTP Listener

Scenarios

You can add an HTTP listener if content identification is required. HTTP is a great fit for workloads such as web applications and mobile mini-games.

Constraints

- HTTP listeners can only be associated with HTTP backend server groups.
- If you only select the network load balancing type for your dedicated load balancer, you cannot add HTTP listeners to this load balancer.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on **Table 1-22**.

Table 1-22 Parameters for configuring an HTTP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTP**.
Listening 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.
Name (Optional)	Specifies the listener name.

Parameter	Description
Redirect to another listener	Specifies the HTTPS listener to which HTTP requests are redirected to encrypt the communication and improve service security. For example, if you configure an HTTP redirection, HTTP requests to access a web page will be redirected to the HTTPS listener you select and handled by the backend servers associated with the HTTPS listener. As a result, the clients access the web page over HTTPS. Note that the configurations for the HTTP listener will not be applied. Requests will be forwarded to backend servers by the HTTPS listener. After the redirection is configured for an HTTP listener, the backend server will return 301 Moved Permanently to the clients.
Transfer Client IP Address	This option is enabled for dedicated load balancers by default. When you use an HTTP listener to forward requests, you can use the X-Forwarded-For header to transfer client IP addresses. The first IP address recorded in the X-Forwarded-For header is the client IP address. For details, see **Using Dedicated Load Balancers to Transfer Client IP Address**.
Advanced Forwarding	Specifies whether to enable advanced forwarding. This option allows you to configure advanced forwarding policies to forward requests to different backend server groups. For more information, see **Advanced Forwarding**.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	

Parameter	Description
Idle Timeout (s)	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**.
Request Timeout (s)	Specifies the length of time (in seconds) that a load balancer is willing to wait for a client request to finish. The load balancer terminates the connection if a request takes too long to complete. The request timeout duration ranges from **1** to **300**.
Response Timeout (s)	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. If sticky session is enabled and the load balancer receives no response from the backend server within the response timeout duration, the load balancer returns a 504 Gateway Timeout error to the client directly. The response timeout duration ranges from **1** to **300**.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.
HTTP Headers	Select HTTP headers as needed. ● Transferring client information – Rewrite X-Forwarded-For-Port to transfer the client port. – Rewrite X-Forwarded-Host to transfer the client domain name. ● Transferring load balancer information – Rewrite X-Forwarded-ELB-IP to transfer the load balancer EIP. – Rewrite X-Forwarded-Port to transfer the listener port. For details, see **HTTP Headers**.

4. Click **Next: Configure Request Routing Policy**.

   a. You are advised to select an existing backend server group.

   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-40**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-41**.

5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

1.4.3.2 Adding an HTTPS Listener

Scenarios

You can add an HTTPS listener if you require encrypted transmission. Load balancers decrypt HTTPS requests before routing them to backend servers. Once the servers process the requests, they send them back to the load balancers for encryption. Finally, the load balancers send the encrypted requests to the clients.

Constraints

- HTTPS listeners can only be associated with HTTP and HTTPS backend server groups.
- If you only select the network load balancing type for your dedicated load balancer, you cannot add HTTPS listeners to this load balancer.
- When you add an HTTPS listener, ensure that the subnet of the load balancer has sufficient IP addresses. If the IP addresses are insufficient, add more subnets on the summary page of the load balancer. After you select a subnet, do not configure network ACL rules for this subnet. If rules are configured, access to the load balancer may be denied.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on **Table 1-23**.

Table 1-23 Parameters for configuring an HTTPS listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTPS**.
Listening 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.

Parameter	Description
Name (Optional)	Specifies the listener name.
QUIC	Specifies whether to upgrade HTTPS to QUIC. When adding an HTTPS listener, you can select a QUIC listener to route requests. Upgrading HTTPS to QUIC lowers latency and speeds up communication, particularly in poor networks with high latency.
Transfer Client IP Address	This option is enabled for dedicated load balancers by default. When you use an HTTPS listener to forward requests, you can use the X-Forwarded-For header to transfer client IP addresses. The first IP address recorded in the X-Forwarded-For header is the client IP address. For details, see **Using Dedicated Load Balancers to Transfer Client IP Address**.
Advanced Forwarding	Specifies whether to enable advanced forwarding. This option allows you to configure advanced forwarding policies to forward requests to different backend server groups. For more information, see **Advanced Forwarding**.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
Configure Certificate	
SSL Authentication	Specifies whether how you want the clients and backend servers to be authenticated. ● **One-way authentication**: Backend servers will be authenticated by clients. ● **Mutual authentication**: The clients and backend servers will authenticate each other.

Parameter	Description
CA Certificate	Specifies the certificate that will be used to authenticate the client when **SSL Authentication** is set to **Mutual authentication** and the frontend protocol is HTTPS. CA certificates are also called client CA public key certificate. They are used to verify the issuer of a client certificate. HTTPS connections can only be established when the client provides a certificate issued by a specific CA.
Server Certificate	Specifies a server certificate that will be used to authenticate the server when HTTPS is used as the frontend protocol. Both the certificate and private key are required.
SNI	Specifies whether to enable SNI. Server Name Indication (SNI) is an extension to TLS. It allows clients to specify which domain name of a listener they are trying to connect in the first request. Once receiving the request, the load balancer searches for the certificate based on the domain name. The client includes the domain name in the initial SSL handshake. Once receiving the request, the load balancer searches for the certificate based on the domain name. If an SNI certificate is found, this certificate will be used for authentication. If no SNI certificates are found, the server certificate is used for authentication. For details, see **Using SNI Certificates for Access Through Multiple Domain Names**.
SNI Certificate	Specifies one or more certificates associated with the domain name when the frontend protocol is HTTPS and SNI is enabled. You can only select the server certificate with SNI domain names. For details, see **Using SNI Certificates for Access Through Multiple Domain Names**.
More (Optional)	
Security Policy	Specifies the security policy you can use if you select HTTPS as the frontend protocol. For more information, see **Configuring TLS Security Policies for Encrypted Communication**.

Parameter	Description
HTTP/2	Specifies whether you want to use HTTP/2 if you select **HTTPS** for **Frontend Protocol**. For details, see **Enabling HTTP/2 for Faster Communication**.
Idle Timeout (s)	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**.
Request Timeout (s)	Specifies the length of time (in seconds) that a load balancer is willing to wait for a client request to finish. The load balancer terminates the connection if a request takes too long to complete. The request timeout duration ranges from **1** to **300**.
Response Timeout (s)	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. If sticky session is enabled and the load balancer receives no response from the backend server within the response timeout duration, the load balancer returns a 504 Gateway Timeout error to the client directly. The response timeout duration ranges from **1** to **300**.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

Parameter	Description
HTTP Headers	Select HTTP headers as needed. ● Transferring client information – Rewrite X-Forwarded-For-Port to transfer the client port. – Rewrite X-Forwarded-Host to transfer the client domain name. ● Transferring load balancer information – Rewrite X-Forwarded-ELB-IP to transfer the load balancer EIP. – Rewrite X-Forwarded-Port to transfer the listener port. For details, see **HTTP Headers**.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group by referring to **Table 1-40**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-41**.
5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

1.4.3.3 Forwarding Policy

Overview

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

A forwarding policy consists of two parts: forwarding rule and action. For details, see **Table 1-24**.

Table 1-24 Rules and actions supported by a forwarding policy

Policy Type	Forwarding Rules	Actions
Forwarding policy	**Domain name** and **Path**	**Forward to another backend server group** and **Redirect to another listener** (only for HTTP listeners)

Policy Type	Forwarding Rules	Actions
Advanced forwarding policy	**Domain name**, **Path**, **HTTP request method**, **HTTP header**, **Query string**, and **CIDR block**	**Forward to another backend server group**, **Redirect to another listener**, and **Return a specific response body**

NOTE

You can configure an advanced forwarding policy by referring to **Managing an Advanced Forwarding Policy**.

How Requests Are Matched

- After receiving a request, the load balancer attempts to find a matching forwarding policy based on the domain name or path in the request:
  - If a match is found, the request is forwarded to the backend server group you select or create when you add the forwarding policy.
  - If no match is found, the request is forwarded to the default backend server group that is specified when the listener is created.
  - If both a domain name and path are configured for a forwarding policy, the request can match the forwarding policy only when the domain name and path are both met.
- If advanced forwarding is not enabled for a dedicated load balancer, the matching order is determined by the following rules:
  - When a request matches both a domain name-based policy and a path-based policy, the domain named-based policy is matched first. **Table 1-25** shows an example.
  - Forwarding policy priorities are independent of each other regardless of domain names.
  - Path-based forwarding rules are applied in the following order of priority: an exact match rule, a prefix match rule, and a regular expression match rule. For multiple matches of the same type, only the longest path rule will be applied.

Table 1-25 Example forwarding policies

Request	Forwarding Policy	Forwarding Rule	Specified Value
www.elb.com/test	1	Path	/test
	2	Domain name	www.elb.com

NOTE

In this example, although request **www.elb.com/test** matches both forwarding policies, it is routed based on forwarding policy 2 because domain named-based forwarding rules are applied first.

Constraints

- 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.
- When you add a forwarding policy, note the following:
  - The URL in a forwarding rule can contain only a path but cannot contain query strings. For example, if the path is set to **/path/resource?name=value**, the forwarding policy is invalid.
  - Each path must exist on the backend server. If the path does not exist, the backend server will return 404 Not Found.
  - In the regular expression match, the characters are matched sequentially, and the matching ends when any rule is matched. Matching rules cannot overlap with each other.
  - A path cannot be configured for two forwarding policies.
  - A domain name cannot exceed 100 characters.

Adding a Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer you want to add forwarding policy for and click its name.
3. On the **Listeners** tab, add a forwarding policy in either of the following ways:
   - Locate the target listener and click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click the **Forwarding Policies** tab.
4. Click **Add Forwarding Policy**. Configure the parameters based on **Table 1-26**.

Table 1-26 Forwarding policy parameters

Parameter	Type	Description	Example Value
Forwarding rule	Domain name	Specifies the domain name that will be exactly matched against the domain names in requests. You need to specify either a domain name or path.	www.test.com

Parameter	Type	Description	Example Value
	Path	- Description Specifies the path used for forwarding requests. A path can contain letters, digits, and special characters: _~';@^-%#$.*+?,=!:|\/()[]{} - Matching rules – Exact match: The request path is the same as the specified path and must start with a slash (/). – Prefix match: The request path starts with the specified path string and must start with a slash (/). – Regular expression match: The paths are matched using a regular expression.	/login.php
Action	Forward to a backend server group	Specifies the backend server group to which a request is routed if it matches the configured forwarding rule.	N/A
	Redirect to another listener	Specifies the HTTPS listener to which a request is routed if it matches the configured forwarding rule. 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.	N/A

5. Click **Save**.

1.4.3.4 Advanced Forwarding

1.4.3.4.1 Advanced Forwarding

When you use ELB to distribute Layer 7 requests, you may need different forwarding policies to route different client requests. In this case, you can configure advanced forwarding policies to route requests to the right server based on the characteristics of client requests.

Overview

You can configure advanced forwarding policies to forward requests to different backend server groups based on a wide range of forwarding rules and actions.

The following describes how an advanced forwarding policy works:

Step 1 A client sends a request to a load balancer.

Step 2 The load balancer matches the request based on the forwarding policies you configure. If multiple matches are found, the load balancer routes the request based on the **forwarding policy priorities**.

Step 3 The load balancer routes the request to the backend server using the forwarding policy with the highest priority.

Step 4 The load balancer sends a response to the client.

----End

Figure 1-8 How advanced forwarding works

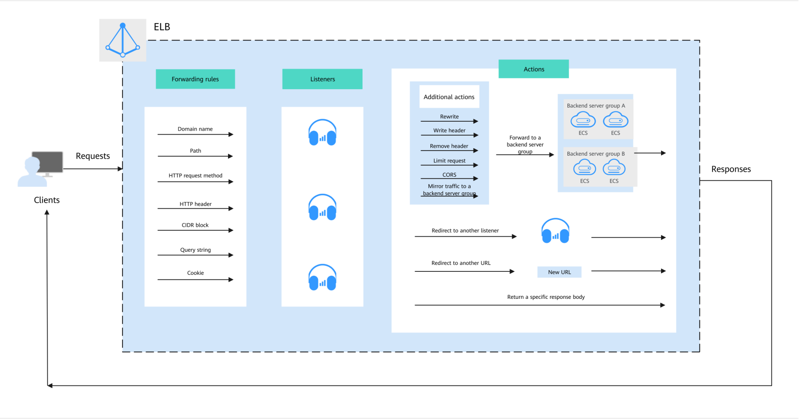


Table 1-27 Rules and actions supported by an advanced forwarding policy

Forwarding Policy	Description
Forwarding rule	The following forwarding rules are supported: domain name, path, HTTP request method, HTTP header, query string, and CIDR block. For details, see **Forwarding Rule**.

How Requests Are Matched

- **Matching rules**: Each client request is matched against **forwarding policies**. Once a match is found, the request is forwarded based on this forwarding policy. If multiple matches are found, the request is forwarded based on the **forwarding policy priorities**. **A smaller forwarding policy number indicates a higher priority and is matched first.**
  - If multiple conditions are configured for a forwarding policy, the request can match this forwarding policy only when all the conditions are met.

  - If the request is matched with any forwarding policy of the listener, it is forwarded based on this forwarding policy.
  - If the request is not matched with any forwarding policy, it is forwarded based on the default forwarding policy.
- **Forwarding policy priority**: **determines** the order in which a client request matches against forwarding policies. If a client request matches multiple forwarding policies, the forwarding policy with the smallest number has the highest priority and is matched first.
- **Default forwarding policy**: After you add a Layer 7 listener to a load balancer, a default forwarding policy is generated. The load balancer then uses this policy to forward requests to the backend server group you specified when adding the listener.
  - If a client request does not match any forwarding policy, it is forwarded according to the default forwarding policy.
  - The default forwarding policy has the lowest priority, which cannot be sorted. You can modify the default backend server group, but cannot delete the default forwarding policy.

Forwarding Rule

Advanced forwarding policies support the following types of forwarding rules: domain name, path, HTTP request method, HTTP header, query string, and CIDR block.

Table 1-28 Forwarding rules

Forwarding Rule	Description
Domain name	Example Request URL: https://**www.example.com**/login.php?locale=en-us=#videos Domain name in the forwarding rule: **www.example.com**
Path	● Description Route requests based on paths. You can configure multiple paths in a forwarding policy. Each path contains 1 to 128 characters, including letters, digits, and special characters: _~';@^-%#$.*+?,=!:|\/()[]{}. ● Matching rules – Exact match: The request path is the same as the specified path and must start with a slash (/). – Prefix match: The request path starts with the specified path string and must start with a slash (/). – Regular expression match: The request path is matched against the specified path using a regular expression. For more information about path matching rules, see **Path Matching**. Example path: Request URL: https://www.example.com/**login.php**?locale=en-us#videos Path in the forwarding rule: **/login.php**

Forwarding Rule	Description
Query string	Route requests based on the query string. A query string consists of a key and one or more values. You need to set the key and values separately. ● The key can contain only letters, digits, and special characters: !$'()*+,./:;=?@^-_' ● Multiple values can be configured for a key. The value can contain letters, digits, and special characters: !$'()*+,./:;=?@^-_'. Asterisks (*) and question marks (?) can be used as wildcard characters. Example Request URL: https://www.example.com/login.php?**locale=en-us**#videos A query string needs to be configured for the forwarding rule: Key: locale Value: en-us
HTTP request method	Route requests based on the HTTP method. ● You can configure multiple request methods in a forwarding policy. ● The following methods are available: GET, POST, PUT, DELETE, PATCH, HEAD, and OPTIONS. Example GET
HTTP header	Route requests based on the HTTP header. An HTTP header consists of a key and one or more values. You need to configure the key and values separately. ● The key can contain only letters, digits, underscores (_), and hyphens (-). **NOTE** The first letter of HTTP request headers User-agent and Connection must be capitalized. ● Multiple values can be configured for a key. The value can contain letters, digits, and special characters: !#$%&'()*+,.\/:;<=>?@[]^-_'{|}~. Asterisks (*) and question marks (?) can be used as wildcard characters. Example Key: Accept-Language Value: en-us
CIDR block	Route requests based on the source IP addresses from where requests originate. Example 192.168.1.0/24 or 2020:50::44/127

Action Types

Advanced forwarding policies support the following actions: forward to a backend server group, redirect to another listener, redirect to another URL, and return a specific response body.

If you set **Action** to **Forward to backend server group** or **Return a specific response body**, you can add additional actions. ELB first matches traffic based on additional actions and then forwards requests to the specified backend server group or returns a specific response body. Among all the additional actions, **Limit request** has the highest priority.

Table 1-29 Actions of an advanced forwarding policy

Action	Description
Forward to a backend server group	Requests are forwarded to the specified backend server group.
Redirect to another listener	Requests are redirected to another listener, which then routes the requests to its associated backend server group. **NOTE** For example, if you configure a redirect for an HTTP listener, HTTP requests to access a web page will be redirected to the HTTPS listener you select and handled by the backend servers associated with the HTTPS listener. As a result, the clients access the web page over HTTPS.

<table>
<tr><th>Action</th><th>Description</th></tr>
<tr><td>Redirect to another URL</td><td>Requests are redirected to the configured URL.
When clients access website A, the load balancer returns 302 or any other 3*xx* status code and automatically redirects the clients to website B. You can customize the redirection URL that will be returned to the clients.
Configure at least one of the following components:
● **Protocol**: ${protocol}, HTTP, or HTTPS **${protocol}**: retains the protocol of the request.
● **Domain Name**: A domain name consists of at least two labels separated by periods (.). Each label can contain only letters, digits, hyphens (-), and periods (.), must start with a letter or digit and cannot end with a hyphen (-). **${host}**: retains the domain name of the request.
● **Port**: ranges from 1 to 65535. **${port}**: retains the port number of the request.
● **Path**: A path can contain letters, digits, and special characters: _~';@^-%#&$.*+?,=!:|\/()[]{} and must start with a slash (/). **${path}**: retains the path of the request.
NOTE
If you select regular expression match, the request path will be overwritten by the variables that match the regular expressions. For details, see **Path Matching Based on Regular Expressions**.
● **Query String**: A query string can contain only letters, digits, and special characters: !$'()*+,./:;=?@&^-_',&. Ampersands (&) can only be used as separators.
● **HTTP Status Code**: 301, 302, 303, 307, or 308
Example
URL for redirection: http://www.example1.com/index.html?locale=en-us#videos
Protocol: HTTP
Domain name: www.example1.com
Port: 8081
Path: /index.html
Query String: locale=en-us
HTTP Status Code: 301</td></tr>
</table>

Action	Description

Return a specific response body	Load balancers return a fixed response to the clients.
	You can custom the status code and response body that load balancers directly return to the clients without the need to route the requests to backend servers.
	Configure the following components:
	- **HTTP Status Code**: By default, 2xx, 4xx, and 5xx status codes are supported.
	- **Content-Type**: text/plain, text/css, text/html, application/javascript, or application/json
	- **Message Body**: This parameter is optional. The value can contain 0 to 1,024 characters.
	Example
	text/plain
	`Sorry, the language is not supported.`
	text/css
	`<head><style type="text/css">div {background-color:red}#div {font-size:15px;color:red}</style></head>`
	text/html
	`<form action="/" method="post" enctype="multipart/form-data"><input type="text" name="description" value="some text"><input type="file" name="myFile"><button type="submit">Submit</button></form>`
	application/javascript
	`String.prototype.trim = function() {var reExtraSpace = /^\s*(.*?)\s+$/;return this.replace(reExtraSpace, "$1")}`
	application/json
	`{ "publicip": { "type": "5_bgp","ip_version": 4},"bandwidth": {"name": "bandwidth123","size": 10,"share_type": "PER"}}`
	NOTE
	Ensure that the response body does not contain carriage return characters. Otherwise, it cannot be saved.

Path Matching

Table 1-30 shows how the five paths configured in the forwarding policies match those in the requests.

Table 1-30 Path matching examples

Request Path	Forwarding Policy	Specified Path	Matching Mode	Forwarding Policy Priority	Destination Backend Server Group
/elb/abc.html	Forwarding policy 01	/elb/abc.html	Prefix match	1	Backend server group 01

Request Path	Forwarding Policy	Specified Path	Matching Mode	Forwarding Policy Priority	Destination Backend Server Group
	Forwarding policy 02	/elb	Prefix match	2	Backend server group 02
/exa/index.html	Forwarding policy 03	/exa[^\s]*	Regular expression match	3	Backend server group 03
	Forwarding policy 04	/exa/index.html	Regular expression match	4	Backend server group 04
/mpl/index.html	Forwarding policy 05	/mpl/index.html	Exact match	5	Backend server group 05

URLs are matched as follows:

- When the request path is /elb/abc.html, it matches both forwarding policy 01 and forwarding policy 02. However, the priority of forwarding policy 01 is higher than that of forwarding policy 02. Forwarding policy 01 is used, and requests are forwarded to backend server group 01.
- When the request path is /exa/index.html, it matches both forwarding policy 03 and forwarding policy 04. However, the priority of forwarding policy 03 is higher than that of forwarding policy 04. Forwarding policy 03 is used, and requests are forwarded to backend server group 03.
- If the request path is /mpl/index.html, it matches forwarding policy 05 exactly, and requests are forwarded to backend server group 05.

Path Matching Based on Regular Expressions

A path can contain letters, digits, and special characters: _~';@^-%#&$.*+?,=!:|\/()[]{} and must start with a slash (/). **${path}** retains the path of the request.

If you select regular expression match, the request path will be overwritten by the variables that match the regular expressions.

How Request Paths Are Overwritten

1. Path matching: The client sends a request, and the request matches a regular expression in the forwarding rule. You can specify one or more regular expressions as the match conditions and set multiple capture groups represented by parentheses **( )** for one regular expression.

2. Extraction and replacement: extracts the content from the capture groups.
3. Destination path: writes them to $1, $2, all the way to $9 configured for the path.

Example

When a client requests to access **/test/ELB/elb/index**, which matches the regular expression **/test/(.*)/(.*)/index**, **$1** will be replaced by **ELB** and **$2** by **elb**, and then the request will be redirected to **/ELB/elb**.

Table 1-31 URL matching based on regular expressions

Matching Step		Description
Forwarding rule: path	Regular expression match	• Matching condition: /test/(.*)/(.*)/index • Request path: /test/ELB/elb/index
Action: rewrite or redirect to another URL	Path	• Path: /$1/$2 • Extracting content $1: ELB $2: elb • Destination path: /ELB/elb

1.4.3.4.2 Managing an Advanced Forwarding Policy

Scenarios

You can configure advanced forwarding policies for HTTP or HTTPS listeners of dedicated load balancers to route requests more specifically.

Each advanced forwarding policy consists of one or more forwarding rules and an action.

- **Supported forwarding rules**: domain name, path, HTTP request method, HTTP header, query string, and CIDR block. For details, see **Forwarding Rule**.
- **Supported actions**: forward to a backend server group, redirect to another listener, redirect to another URL, and return a specific response body. For details, see **Action Types**.
- Multiple forwarding rules can be configured in a single forwarding policy.
- Forwarding policies can be sorted based on their priorities.

Constraints

- Advanced forwarding cannot be disabled once enabled.
- An advanced forwarding policy can contain a maximum of 10 conditions.

Enabling Advanced Forwarding

1. Go to the **load balancer list page**.

2. On the displayed page, locate the load balancer you want to configure forwarding policies for and click its name.
3. On the **Listeners** tab and click the target listener.
4. On the **Summary** tab, click **Enable** next to **Advanced Forwarding**.
5. Click **OK**.

Adding an Advanced Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer you want to configure forwarding policies for and click its name.
3. On the **Listeners** tab, add a forwarding policy in either of the following ways:
   - Locate the target listener and click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click the **Forwarding Policies** tab.
4. Click **Add Forwarding Policy** and configure the parameters based on **Table 1-28** and **Table 1-29**.
5. Click **Save**.

Sorting Forwarding Policies

Each listener can have multiple forwarding policies, which are matched in descending order of priority. A smaller value indicates a higher priority.

You can adjust the priority of custom forwarding policies, but the priority of the default forwarding policy cannot be changed.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer whose forwarding policies you want to modify and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Forwarding Policies** tab, click **Sort**.
5. Drag the forwarding policies to adjust their priorities.
6. Click **Save**.

Modifying a Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer whose forwarding policies you want to modify and click its name.
3. Click the **Listeners** tab, locate the listener, and click its name.
4. On the **Forwarding Policies** tab, select the forwarding policy, and click **Edit**.
5. Modify the parameters and click **Save**.

Deleting a Forwarding Policy

You can delete a forwarding policy if you no longer need it.

Deleted forwarding policies cannot be recovered.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer whose forwarding policies you want to delete and click its name.
3. Click the **Listeners** tab, locate the listener, and click its name.
4. On the **Forwarding Policies** tab, select the forwarding policy and click **Delete** on the top right.
5. In the displayed dialog box, click **OK**.

1.4.3.5 HTTP Headers

HTTP headers are a list of strings sent and received by both the client and server on every Hypertext Transfer Protocol (HTTP) request and response. This section describes HTTP headers supported by HTTP and HTTPS listeners. You can enable these headers as required.

Table 1-32 HTTP headers that can transfer client information

Header	Description
X-Forwarded-For-Port	Rewrites the client port in the X-Forwarded-For-Port header and transfers it to backend servers.
X-Forwarded-Host	Rewrites the client domain name in the X-Forwarded-Host header and transfers it to backend servers.

Table 1-33 HTTP headers that can transfer load balancer information

Header	Description
X-Forwarded-Proto	Rewrites the listener protocol in the X-Forwarded-Proto header and transfers it to backend servers.
X-Forwarded-ELB-IP	Rewrites the EIP used by the load balancer in the X-Forwarded-ELB-IP header and transfers it to backend servers.
X-Forwarded-Port	Rewrites the listener port in the X-Forwarded-Port header and transfers it to backend servers.
X-Forwarded-ELB-ID	Rewrites the load balancer ID in the X-Forwarded-ELB-ID header and transfers it to backend servers.

Adding HTTP Headers

1. Go to the **load balancer list page**.
2. You can add a listener in either of the following ways:
   - On the displayed page, locate the load balancer and click its name. On the **Listeners** tab, click **Add Listener**.

   - On the displayed page, locate the load balancer and click **Add Listener** in the **Operation** column.
3. On the **Add Listener** page, expand **Advanced Settings (Optional)** and select the headers as needed.
4. Configure the listener as prompted.
5. Confirm the configuration and click **Submit**.

Modifying HTTP Headers

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click the **Listeners** tab, locate the target listener and click **Edit** in **Operation** column.
4. On the **Edit** page, expand **Advanced Settings (Optional)** and select the headers as needed.
5. Click **OK**.

1.4.3.6 Enabling HTTP/2 for Faster Communication

What Is HTTP/2?

Hypertext Transfer Protocol 2.0 (HTTP/2) uses a binary format for data transmission. It allows for much faster transmission and multiplexing. To reduce latency and improve efficiency, you can enable HTTP/2 when you add HTTPS listeners.

Constraints

You can enable HTTP/2 only for HTTPS listeners.

Managing HTTP/2

You can enable HTTP/2 when you add an HTTPS listener. You can enable or disable HTTP/2 for an existing HTTPS listener.

Enabling HTTP/2 When Adding a Listener

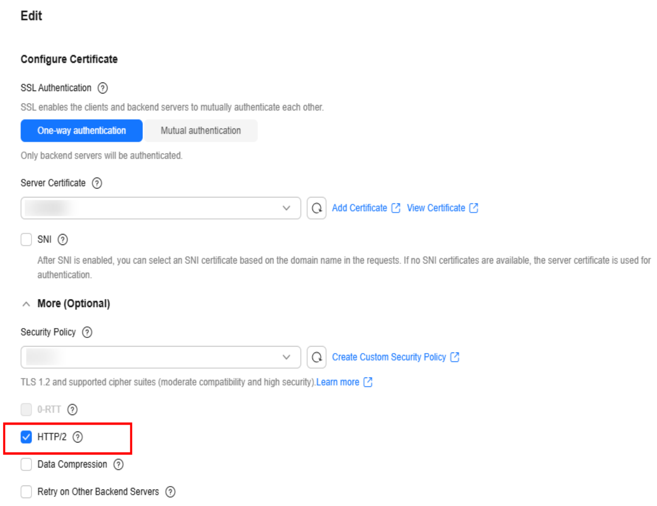
To enable HTTP/2 when adding an HTTPS listener, perform the following operations:

1. Go to the **load balancer list page**.
2. Locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**.
4. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.
5. Expand **Advanced Settings (Optional)** and enable HTTP/2.
6. Confirm the configurations and go to the next step.

Enabling or Disabling HTTP/2 for an Existing Listener

1. Go to the **load balancer list page**.
2. Locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Summary** tab, click **Edit** on the top right.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)** and enable or disable HTTP/2.
6. Click **OK**.

Figure 1-9 Disabling or enabling HTTP/2



Edit

Configure Certificate

SSL Authentication

SSL enables the clients and backend servers to mutually authenticate each other.

One-way authentication | Mutual authentication

Only backend servers will be authenticated.

Server Certificate

Add Certificate View Certificate

□ SNI

After SNI is enabled, you can select an SNI certificate based on the domain name in the requests. If no SNI certificates are available, the server certificate is used for authentication.

More (Optional)

Security Policy

Create Custom Security Policy

TLS 1.2 and supported cipher suites (moderate compatibility and high security).Learn more

□ 0-RTT

☑ HTTP/2

□ Data Compression

□ Retry on Other Backend Servers

1.4.4 Managing a Listener

Scenarios

You can configure modification protection for a listener, modify the settings of a listener, change the backend server group of a listener, and delete a listener.

Prerequisites

- You have created a backend server group by referring to **Creating a Backend Server Group**.
- You have added a listener by referring to **Listener Overview**.

Configuring Modification Protection for a Listener

You can enable modification protection for a listener to prevent it from being modified or deleted by accident.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click the **Listeners** tab, locate the listener, and click its name.
4. On the **Summary** tab, click **Configure** next to **Modification Protection**.

5. In the **Configure Modification Protection** dialog box, enable **Modification Protection**.

NOTE

You need to disable **Modification Protection** if you want to modify or delete a listener.

Modifying Listener Settings

NOTE

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

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Modify the listener in either of the following ways:
   - On the **Listeners** tab, locate the listener, and click **Edit** in the **Operation** column.
   - Click the name of the target listener. On the **Summary** tab, click **Edit** on the right corner.
4. In the **Edit** dialog box, modify parameters, and click **OK**.

Modifying Timeout Durations

You can modify timeout durations (idle timeout, request timeout, and response timeout) for your listeners to meet varied demands. For example, if the size of a request from an HTTP or HTTPS client is large, you can prolong the request timeout duration to ensure that the request can be successfully routed.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click the name of the listener.
4. On the **Summary** tab, click **Edit** on the right corner.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)**.
6. Configure **Idle Timeout (s)**, **Request Timeout (s)**, and **Response Timeout (s)** as you need.
7. Click **OK**.

Changing the Backend Server Group of a Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the target load balancer and click its name.
3. On the **Listeners** tab, locate the target listener and click its name.
4. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
5. 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.

6. Click **OK**.

Deleting Listeners

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
   a. Deleting a listener:
      i. On the **Listeners** tab, locate the listener, and click **Delete** in the **Operation** column.
      ii. In the displayed dialog box, enter **DELETE**.
   b. Batch deleting listeners:
      i. On the **Listeners** tab, select multiple listeners you want to delete.
      ii. Click **Delete** above the listener list.
      iii. In the displayed dialog box, enter **DELETE**.
3. Click **OK**.

1.5 Backend Server Group

1.5.1 Backend Server Group Overview

What Is a Backend Server Group?

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 a cloud server, supplementary network interface, or IP address.

The following table describes how a backend server group forwards traffic.

Table 1-34 Traffic distribution process

Step 1	A client sends a request to your application. The listeners added to your load balancer use the protocols and ports you have configured to forward the request to the associated backend server group.
Step 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.
Step 3	In this way, massive concurrent requests can be processed at the same time, improving the availability of your applications.

For dedicated load balancers, the backend server group type can be **Hybrid** or **IP as a backend server**. You can add cloud servers, supplementary network interfaces, or IP addresses to a hybrid backend server group. If you set the type to **IP as a backend server**, you can only add IP addresses as backend servers.

Figure 1-10 shows the architecture of different types of backend server groups. **Table 1-35** describes different backend server group types.

Figure 1-10 Backend server group architecture

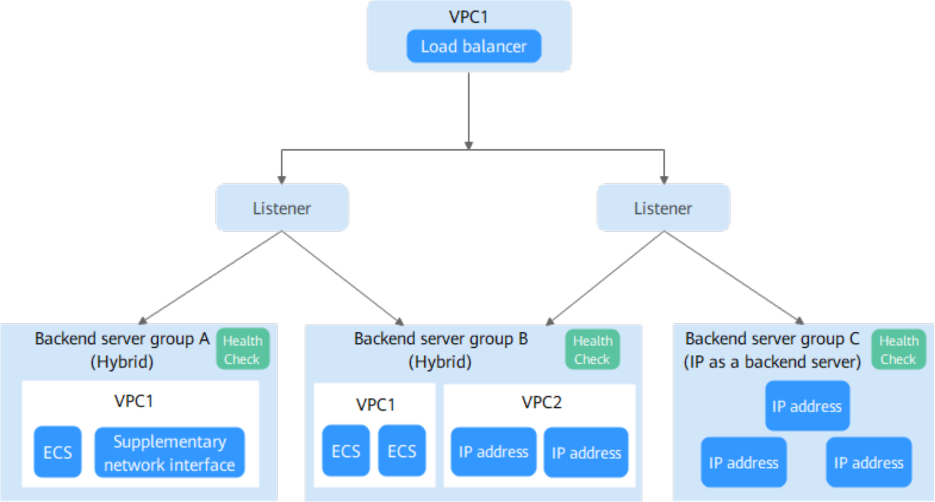

VPC1
Load balancer
Listener
Listener
Backend server group A (Hybrid)
Health Check
VPC1
ECS
Supplementary network interface
Backend server group B (Hybrid)
Health Check
VPC1
ECS
ECS
VPC2
IP address
IP address
Backend server group C (IP as a backend server)
Health Check
IP address
IP address
IP address

Table 1-35 Backend server group types

Backend Server Group Type	Backend Server Type	Example
Hybrid	● Cloud servers and supplementary network interfaces that are in the same VPC as the load balancer ● Cloud servers in other VPCs or on-premises servers if IP as a backend is enabled for the load balancer	As shown in **Figure 1-10**: ● In backend server group A, you can add ECSs or supplementary network interfaces in VPC1. ● In backend server group B, you can add IP addresses in VPC2 as backend servers.
IP as a backend server	IP addresses of cloud or on-premises servers if IP as a backend is enabled for the load balancer	As shown in **Figure 1-10**, IP addresses can be added to backend server group C as backend servers.

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**: The **health check** function ensures traffic is routed only to healthy backend servers in the backend server group.

Controlling Traffic Distribution

You can configure the key functions listed in **Table 1-36** for each backend server group to ensure service stability.

Table 1-36 Key functions

Key Function	Description	Detail
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.	**Enabling Sticky Session to Accelerate Access**
Slow Start	Specifies whether to enable slow start. After you enable it, the load balancer linearly increases the proportion of requests to new backend servers in the backend server group. When the slow start duration elapses, the load balancer sends full share of requests to these backend servers and exits the slow start mode.	**Slow Start**

Backend Server Group and Listener Protocols

You can associate a backend server group with different dedicated load balancers under the same enterprise project or different listeners.

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 1-37**.

Table 1-37 The frontend and backend protocol

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

Helpful Links

- **Creating a Backend Server Group**
- **Controlling Traffic Distribution**
- **Adding Backend Servers in the Same VPC as a Load Balancer**
- **Adding Backend Servers in a Different VPC from a Load Balancer**

1.5.2 Creating a Backend Server Group

Scenario

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

A backend server group can be associated with listeners of different load balancers.

Table 1-38 describes the scenarios for creating a backend server group.

Table 1-38 Scenarios

Scenario	Reference
Creating a backend server group and associating it with a load balancer	**Procedure**
Creating a backend server group when adding a listener	Adding listeners with different protocols by referring to **Listener Overview**
Changing the backend server group associated with a 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 1-39**.

Table 1-39 The frontend and backend protocol

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

Procedure

1. Go to the **backend server group list page**.
2. Click **Create Backend Server Group** in the upper right corner.
3. Configure the routing policy based on **Table 1-40**.

Table 1-40 Parameters required for configuring a routing policy

Parameter	Description
Backend Server Group Name	Specifies the name of the backend server group.
Type	Specifies the type of load balancer that can use the backend server group. Select **Dedicated**.
Load Balancer	Specifies whether to associate a load balancer.
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: The options are HTTP, HTTPS, TCP, and UDP.

Parameter	Description
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. ● **Connection ID**: This algorithm is available when you have selected **QUIC** for **Backend Protocol**. This algorithm allows requests with the same connection ID to be routed to the same backend server. For more information about load balancing algorithms, see **Load Balancing Algorithms**.
Sticky Session	Specifies whether to enable sticky sessions if you have selected **Weighted round robin**, **Connection ID**, 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 **Enabling Sticky Session to Accelerate Access**.

Parameter	Description
Sticky Session Type	Specifies the sticky session type. This parameter is mandatory if **Sticky Session** is enabled. You can select one of the following types: ● **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 ensures 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.
Stickiness Duration (min)	Specifies the minutes that sticky sessions are maintained. This parameter is mandatory if **Sticky Session** is enabled. ● Sticky sessions at Layer 4: **1** to **60** ● Sticky sessions at Layer 7: **1** to **1440**
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**.
Slow Start Duration (s)	Specifies how long the slow start will last, in seconds. This parameter is mandatory if **Slow Start** is enabled.
Description (Optional)	Provides supplementary information about the backend server group.

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

Add cloud servers, supplementary network interfaces, or IP as backend servers to this backend server group. For details, see **Backend Server Overview**.

Configure health check for the backend server group based on **Table 1-41**. For more information about health checks, see **Health Check**.

Table 1-41 Parameters required for configuring a health check

Parameter	Description
Health Check	Specifies whether to enable the health check option. If health check is enabled, click ✎ next to **Advanced Settings (Optional)** 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. ● TCP, HTTP,, and HTTPS are supported. ● If the protocol of the backend server group is UDP and QUIC, the health check protocol is UDP by default and cannot be changed.
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. ● By default, the private IP address of each backend server is used. ● 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 length: 63 characters.
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.
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 (_).

Parameter	Description
Interval (s)	Specifies the maximum time between two consecutive health checks, in seconds. The interval ranges from **1** to **50**.
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The value ranges from **1** to **50**.
Healthy Threshold	Specifies the number of consecutive successful health checks required for declaring a backend server healthy. The value ranges from **1** to **10**.
Unhealthy Threshold	Specifies the number of consecutive failed health checks required for declaring a backend server unhealthy. The value ranges from **1** to **10**.

5. Click **Next**.
6. 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 the ways listed in **Table 1-38**.

1.5.3 Controlling Traffic Distribution

1.5.3.1 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.

The default load balancing algorithm is weighted round robin. You can change it to a different algorithm if needed.

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

Table 1-42 Load balancing algorithms

Load Balancing Algorithm	Description
Weighted round robin	Routes requests to backend servers in sequence based on their weights.

Load Balancing Algorithm	Description
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.

How Load Balancing Algorithms Work

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

Weighted Round Robin

Figure 1-11 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 1-11 Traffic distribution using the weighted round robin algorithm

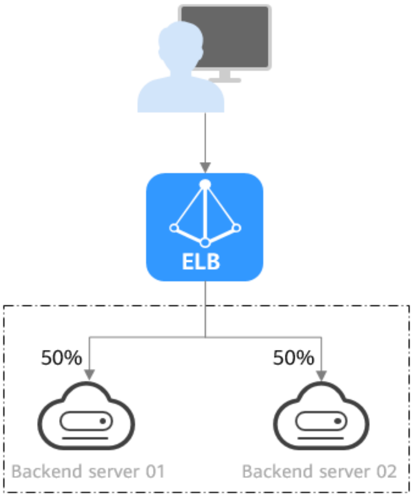


Table 1-43 Weighted round robin

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

Weighted Least Connections

Figure 1-12 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 with backend server 02. New requests are preferentially routed to backend server 02.

Figure 1-12 Traffic distribution using the weighted least connections algorithm

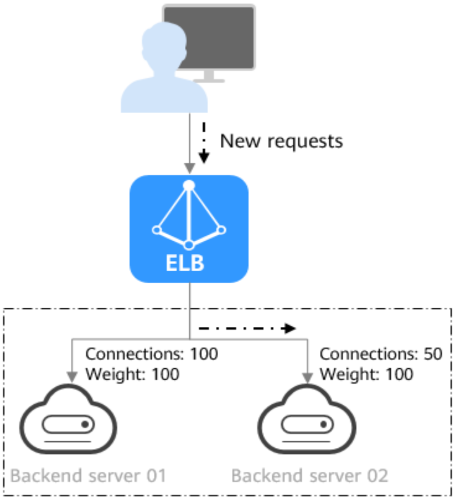


Table 1-44 Weighted least connections

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 load 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 1-13 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 this IP address to backend server 01.

Figure 1-13 Traffic distribution using the source IP hash algorithm

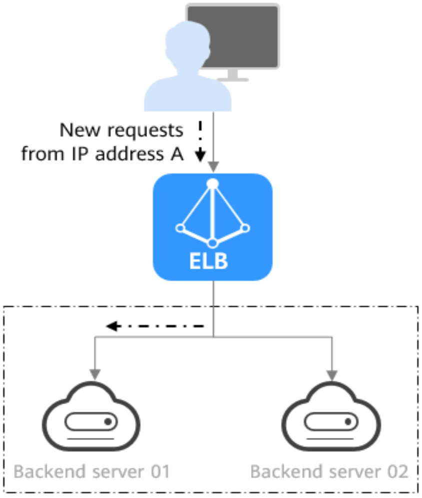


Table 1-45 Source IP hash

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 1-14 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 1-14 Traffic distribution using the connection ID algorithm

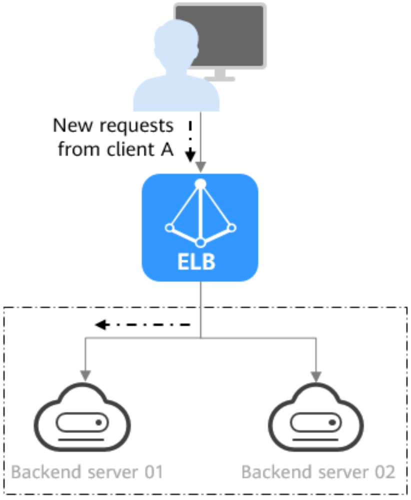


Table 1-46 Connection ID

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

Changing a Load Balancing Algorithm

⚠ CAUTION

The new load balancing algorithm is applied immediately and will be used to route requests over new connections. The previous load balancing algorithm is still be used to route requests over established connections.

1. Go to the **backend server group list page**.
2. In the backend server group list, locate the target backend server group and click **Edit** in the **Operation** column.
3. In the **Modify Backend Server Group** dialog box, change the load balancing algorithm.
4. Click **OK**.

1.5.3.2 Enabling Sticky Session to Accelerate Access

In e-commerce shopping and user login systems, maintaining a sticky session between the client and server is crucial for a seamless user experience. If requests from the same client are distributed to different backend servers, users may need to log in to different servers or operation progress may be interrupted. To address these issues, you can enable sticky session for a backend server group, so that the load balancer can identify the characteristics (such as IP addresses and cookies) of client requests and distribute requests with the same IP address or cookie to the same backend server. This can improve access efficiency and user experience.

Sticky Session

The sticky session types supported by each backend server group vary by protocol and load balancing algorithm. For details, see **Table 1-47**.

Table 1-47 Sticky session types supported by dedicated load balancers

Backend Server Group Protocol	Load Balancing Algorithm	Sticky Session Type
● TCP ● UDP	Weighted round robin	Source IP address
	Weighted least connections	Source IP address
	Source IP hash	Not supported
● HTTP ● HTTPS ● gRPC	Weighted round robin	● Load balancer cookie ● Application cookie
	Weighted least connections	● Load balancer cookie ● Application cookie
	Source IP hash	Not supported

Backend Server Group Protocol	Load Balancing Algorithm	Sticky Session Type
QUIC	Connection ID	Source IP address

Table 1-48 Sticky session types

Sticky Session Type	Description	Stickiness Duration (Minutes)	Scenarios Where Sticky Session Become Invalid
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 requests from a client to a particular server based on the generated key. This allows requests from the same IP address to be forwarded to the same backend server.	● Default: 20 ● Maximum: 60 ● Range: 1–60	● Source IP addresses of the clients have changed. ● The session stickiness duration has been reached.
Load balancer cookie	The load balancer generates a cookie after it receives a request from a client. All the subsequent requests with the same cookie are distributed to the same backend server.	● Default: 20 ● Maximum: 1,440 ● Range: 1–1,440	● Sticky sessions do not take effect when requests sent by the clients do not contain cookies. ● The session stickiness duration has been reached.
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 cookie are routed to the same backend server.		

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

- 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.

Enabling or Disabling Sticky Session

1. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
3. In the **Modify Backend Server Group** dialog box, enable or disable **Sticky Session**.

   If you enable it, select the sticky session type, and set the session stickiness duration.
4. Click **OK**.

1.5.3.3 Slow Start

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.

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

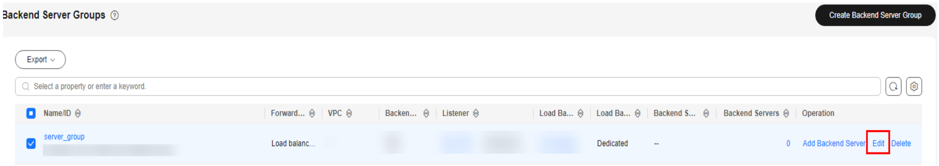
- Slow start is only available for HTTP and HTTPS backend server groups of dedicated load balancers.
- Weighted round robin must be selected as the load balancing algorithm.
- Slow start takes effect only for new backend servers and does not take effect when the first backend server is added to a backend 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.

Enabling or Disabling Slow Start

1. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.

Figure 1-15 Modifying a backend server group


Backend Server Groups
Create Backend Server Group
Export
Select a property or enter a keyword.

Name/ID	Forward...	VPC	Backen...	Listener	Load Ba...	Load Ba...	Backend S...	Backend Servers	Operation
server_group	Load balanc...					Dedicated	--	0	Add Backend Server Edit Delete

3. In the **Modify Backend Server Group** dialog box, enable or disable **Slow Start**.

   If you enable it, you need to set the slow start duration. The duration ranges from 30 to 1200. When the slow start duration elapses, the load balancer sends full share of requests to backend servers and exits slow start.
4. Click **OK**.

1.5.4 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

- 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 1-37**.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the target load balancer and click its name.
3. On the **Listeners** tab, locate the target listener and click its name.
4. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
5. 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.

6. Click **OK**.

1.5.5 Managing a Backend Server Group

You can manage a backend server group as required.

Enabling Modification Protection

You can enable modification protection for a backend server group to prevent the backend servers in it from being modified or deleted by accident.

Enabling modification protection for a backend server group will prohibit any change to both the group and the backend servers in it.

1. Go to the **backend server group list page**.
2. On the displayed page, locate the backend server group and click its name.
3. On the **Summary** tab, click **Configure** next to **Modification Protection**.
4. In the **Configure Modification Protection** dialog box, enable **Modification Protection**.
5. Click **OK**.

NOTE

Disable **Modification Protection** if you want to delete a backend server group or modify its settings.

Enabling Removal Protection for a Backend Server Group

You can enable removal protection for a backend server group to prevent the backend servers in it from being removed by accident.

After removal protection is enabled for a backend server group, you cannot remove backend servers from it.

⚠ CAUTION

If your load balancer is managed by CCE, enabling removal protection for a backend server group may affect the normal running of the cluster.

1. Go to the **backend server group list page**.
2. On the displayed page, locate the backend server group and click its name.
3. On the **Summary** tab, enable **Removal Protection**.

NOTE

Disable **Removal Protection** if you want to remove servers from a backend server group.

Viewing a Backend Server Group

You can view the details of a backend server group.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the backend server group.
3. On the **Summary** tab, view the basic information and health check settings.

Deleting a Backend Server Group

Before deleting 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.

1. Go to the **backend server group list page**.
2. On the backend server group list page, locate the backend server group and click **Delete** in the **Operation** column.
3. In the displayed dialog box, click **OK**.

1.6 Backend Server

1.6.1 Backend Server 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.

Table 1-49 describes the types of backend servers that can be added to a backend server group.

Table 1-49 Backend server types

Backend Server Type	Description	Reference
Cloud server	You can add cloud servers that are in the same VPC as the load balancer.	**Adding Backend Servers in the Same VPC as a Load Balancer**

Backend Server Type	Description	Reference
Supplementary network interface	You can add supplementary network interfaces that are in the same VPC as the load balancer.	**Adding Backend Servers in the Same VPC as a Load Balancer**
IP as backend server	After **IP as a Backend** is enabled, you can add IP addresses as backend servers to process requests. Ensure that these IP address can reach the load balancer.	**Adding Backend Servers in a Different VPC from a 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.
- 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 the load balancer can perform health checks normally, and at least one backend server that is running properly has been added to the load balancer.

Notes and Constraints

- 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 and Network ACL 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 1-50 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.

1.6.2 Security Group and Network ACL Rules

Scenarios

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

- Security group rules of backend servers must allow traffic from the backend subnet where the load balancer is created to the backend servers. (By default, the backend subnet of a load balancer is the same as the subnet where the load balancer works.) 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 subnet where the backend servers are deployed, the rules

must allow traffic from the backend subnet of the load balancer to the subnet of the backend servers. For details about how to configure network ACL rules, see **Configuring Network ACL Rules**.

NOTE

If a dedicated load balancer has Layer 4 listeners and IP as a backend is disabled, security group and network ACL rules will be ignored even you have configured rules to allow traffic.

You can use access control to limit which IP addresses are allowed or denied to access the listener. For details, see **What Is Access Control?**

Constraints

- If health check is enabled for a backend server group, security group rules must allow traffic over the health check port and 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 instances in the security group to communicate with each other but block access from external networks. To ensure that the load balancer can communicate with associated backend servers over both the frontend and health check ports, configure inbound rules for the security group containing these servers.

1. In the ECS list, click the name of the target ECS.

   The ECS details page is displayed.
2. Click the **Security Groups** tab, locate the security group, click its name, and view security group rules.
3. On the **Inbound Rules** tab, click **Add Rule**. Configure inbound rules based on **Table 1-51**.

Table 1-51 Security group rules

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

Backend Protocol	Action	Protocol & Port	Source IP Address
UDP	Allow	**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 to backend servers is still need to be allowed.
- Traffic from the new backend subnet is also need to be allowed to backend servers.

4. 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.

Default network ACL rules deny all inbound and outbound traffic. You can configure inbound rules to allow traffic from the backend subnet of the load balancer over the ports of backend servers.

- If the load balancer is in the same subnet as the backend servers, network ACL rules will not take effect. In this case, the backend servers will be considered healthy and can be accessed by the clients.
- If the load balancer is not in the same subnet as the backend servers, network ACL rules will take effect. In this case, the backend servers will be considered unhealthy and cannot be accessed by the clients.

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 to display **Service List** and choose **Networking** > **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Access Control** > **Network ACLs**.
5. In the network ACL list, locate the target network ACL and click its name.
6. On the **Inbound Rules** or **Outbound Rules** tab, click **Add Rule** to add inbound or outbound rules.
   - **Action**: Select **Allow**.
   - **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 to all IP addresses is permitted.
- **Destination Port Range**: Select a port range.
- (Optional) **Description**: Describe the network ACL rule.

7. Click **OK**.

1.6.3 Adding Backend Servers in the Same VPC as a Load Balancer

When you use ELB to route requests, ensure that at least one backend server is healthy and can process requests routed by the load balancer.

If the incoming traffic increases, you can add more cloud 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.

You can add ECSs and supplementary network interfaces in the VPC where the dedicated load balancer is created.

Constraints

- Cloud servers and supplementary network interfaces can only be added to a hybrid backend server group.
- Only ECSs and supplementary network interfaces in the same VPC as the backend server group can be added.

Procedure

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Click the **Backend Servers** tab and add servers as required.
   a. Cloud servers (ECSs): Locate the **Cloud Servers** area and click **Add** on the right. On the displayed page, search for the cloud servers by keyword and then add the private IP address. If you use private IP addresses for search, you can select the private IP address bound to either the primary or extended network interface.
   b. Supplementary network interfaces: Locate the **Supplementary Network Interfaces** area and click **Add** on the right. On the displayed page, search for the supplementary network interfaces by keyword.
4. Select the servers you want to add and click **Next**.
5. Specify the weights and ports for the servers and click **Finish**.

   You can set ports and weights in batches.

Modifying the Port and Weight of a Backend Server

The server weight ranges from **0** to **100**. If you set the weight of a cloud 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**.

1. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, click the name of the target backend server group.
3. On the **Backend Servers** tab, click **Cloud Servers** or **Supplementary Network Interfaces**.
4. Select the target backend servers and click **Modify Port/Weight** up above the backend server list.
5. In the displayed dialog box, modify ports/weights as you need.
   - **Modifying ports**
     - Modifying the port of a cloud server: Set the port in the **Backend Port** column.
     - Modifying the ports of multiple cloud servers: Set the port next to **Batch Modify Ports** and click **OK**.
   - **Modifying weights**
     - Modifying the weight of a cloud server: Set the weight in the **Weight** column.
     - Modifying the weights of multiple cloud servers: Set the weight next to **Batch Modify Weights** and click **OK**.

     NOTE

     You can set the weights of multiple cloud servers to **0** to block them from receiving requests routed by each load balancer.
6. Click **OK**.

Removing a Cloud Server

If a cloud server is removed, it is disassociated from the load balancer and can still run normally. However, it cannot receive requests from the load balancer. You can add this cloud server to the backend server group again when traffic increases or the reliability needs to be enhanced.

NOTE

If a cloud server is removed, requests are still routed to it. This is because a persistent connection is established between the load balancer and the cloud 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.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Switch to the **Backend Servers** tab and click **Cloud Servers** or **Supplementary Network Interfaces**.
4. Select the backend servers you want to remove and click **Remove** above the backend server list.

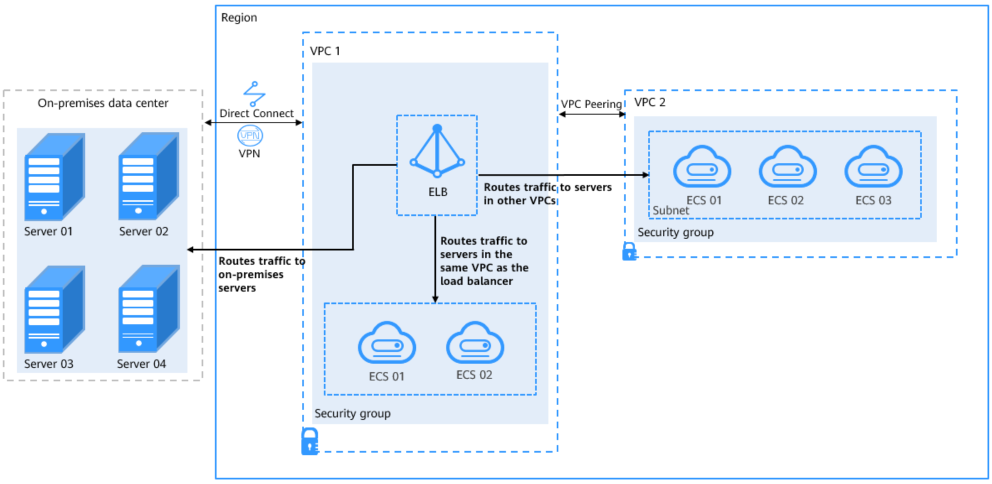
5. In the displayed dialog box, click **OK**.

1.6.4 Adding Backend Servers in a Different VPC from a Load Balancer

Dedicated load balancers can distribute traffic across cloud servers and on-premises servers. You can add cloud servers and supplementary network interfaces in the VPC where the dedicated load balancer is created. After enabling IP as a backend, you can also add the IP addresses of servers in other VPCs or in your on-premises data center.

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

Figure 1-16 Routing requests to cloud and on-premises servers



Constraints

- **IP as a Backend** cannot be disabled after it is enabled.
- Before forwarding requests to servers in other VPCs, ensure that the target VPC can communicate with the VPC where the load balancer is created.
- Only private IPv4 addresses can be added as backend servers.
- A maximum of 100,000 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. Install the **TOA module** to obtain source IP addresses.
- When a UDP listener routes traffic to IP as backend servers in a UDP backend server group over a Direct Connect or VPN connection, the health check result may be unhealthy. In this case, **submit a service ticket**.

Enabling IP as a Backend

1. Go to the **load balancer list page**.
2. On the load balancer list page, click the name of the target load balancer.

3. On the **Summary** tab, click **Enable** next to **IP as a Backend**.
4. Click **OK**.

Adding IP as Backend Servers

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Switch to the **Backend Servers** tab and click **Add** on the **IP as Backend Servers** area.
4. Specify the IP addresses, backend ports, and weights.
5. Click **OK**.

Modifying the Ports/Weights of IP as Backend Servers

The server weight ranges from **0** to **100**. If you set the weight 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**.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Switch to the **Backend Servers** tab and click **IP as Backend Servers**.
4. Select the servers and click **Modify Port/Weight** up the server list.
5. In the displayed dialog box, modify ports/weights as you need.
   - **Modifying ports**
     - Modifying the port of an IP as backend server: Set the port in the **Backend Port** column.
     - Modifying the ports of multiple IP as backend servers: Set the port next to **Batch Modify Ports**, and click OK.
   - **Modifying weights**
     - Modifying the weight of an IP as backend server: Set the weight in the **Weight** column.
     - Modifying the weights of multiple IP as backend servers: Set the weight next to **Batch Modify Weights** and click **OK**.

   NOTE

   You can set the weights of multiple servers to **0** to block them from receiving requests routed by each load balancer.
6. Click **OK**.

Removing IP as Backend Servers

NOTE

If a cloud server is removed, requests are still routed to it. This is because a persistent connection is established between the load balancer and the cloud 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.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Switch to the **Backend Servers** tab and click **IP as Backend Servers**.
4. Select the IP as backend servers to be removed and click **Remove** above the server list.
5. In the displayed dialog box, click **OK**.

1.7 Health Check

1.7.1 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 routing 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 **Configuring a Health Check**.

Select a health check protocol that matches the backend protocol as described in **Table 1-52**.

Table 1-52 Backend and health check protocols (dedicated load balancers)

Backend Protocol	Health Check Protocol
TCP	TCP, HTTP, HTTPS

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

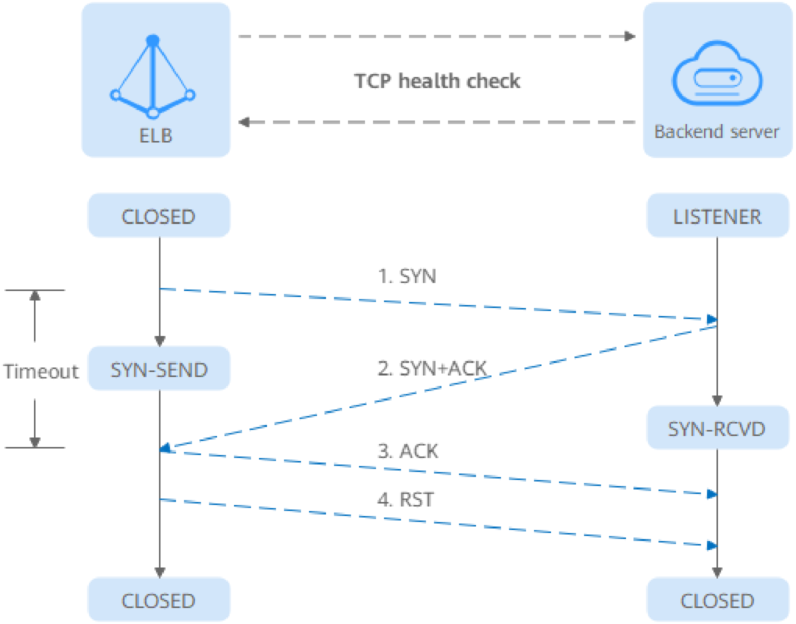
Health Check Source IP Address

A dedicated load balancer uses the IP addresses in its backend subnet to send requests to backend servers and check their health. To perform health checks, you must ensure that the security group rules of the backend servers allow access from the backend subnet where the load balancer works. For details, see **Security Group and Network ACL Rules**.

TCP Health Check

If a backend server group uses TCP, HTTP, or HTTPS as the protocol, you can use TCP to initiate three-way handshakes to check the health of backend servers.

Figure 1-17 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.

⚠ CAUTION

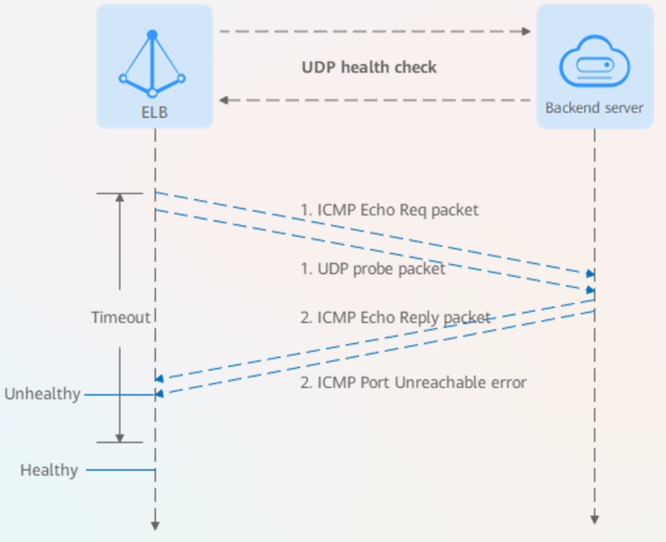
After a successful three-way TCP handshake, an RST packet will be sent to close the TCP connection. The application on the backend server may consider this packet a connection error and reply with a message, for example, "Connection reset by peer". To avoid this issue, take either of the following actions:

- Use **HTTP for health checks**.
- Have the backend server ignore the connection error.

UDP Health Check

If a backend server group uses UDP as the protocol, ELB sends ICMP and UDP probe packets to backend servers to check their health.

Figure 1-18 UDP health check



The UDP health check process is as follows:

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

⚠ CAUTION

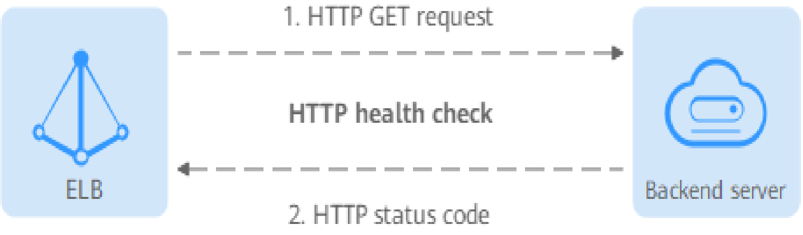
- If there is a large number of concurrent requests, the health check result may be different from the actual health of the backend server.

  If the backend server runs Linux, it may limit the rate of ICMP packets as a defense against ping flood attacks. In this case, even if there is a service exception, ELB will not receive the error message "port XX unreachable", and the server will still be determined healthy. As a result, the health check result is different from the actual health of the backend server.
- The UDP probe packet's payload has no significance and is simply used to fill the packet with data. Typically, the payload is set to "H". Clients should not attempt to interpret its content.

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 1-19** shows how an HTTP health check works.

Figure 1-19 HTTP health check



The HTTP health check process is as follows:

1. The load balancer sends an 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.)
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.

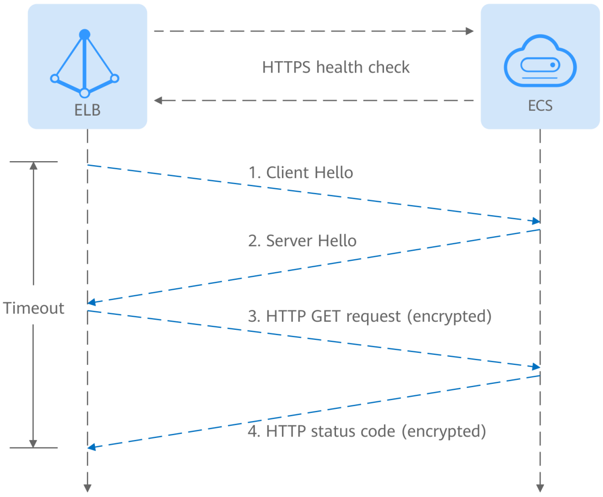
⚠ CAUTION

- If HTTP health check is configured for a TCP listener of a dedicated load balancer, the load balancer uses HTTP/1.0 to send requests to backend servers. HTTP/1.0 is used to establish short-lived connections. This means the load balancer will not translate the HTTP responses until it receives the TCP disconnection packet. Ensure that the backend server disconnects the TCP connection immediately after sending the responses. If the TCP connection is not disconnected, the health check may fail.
- In an HTTP health check, the User-Agent header identifies the requests for health checks. The value of User-Agent may be adjusted based on service requirements. So it is not recommended to rely on this header for verification or judgment.
- If the private IP address of a backend server is used as the domain name for a health check, do not rely on the host header for verification or judgment, as it may be empty.

HTTPS Health Check

If a backend server group uses TCP, HTTP, or HTTPS as the protocol, you can use HTTPS to establish an SSL connection over TLS handshakes to obtain the statuses of backend servers. **Figure 1-20** shows how an HTTPS health check works.

Figure 1-20 HTTPS health check


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

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.

NOTE

- In an HTTPS health check, the User-Agent header identifies that the requests are sent for health checks. The value of User-Agent may be adjusted based on service requirements. So it is not recommended to rely on this header for verification or judgment.
- If the private IP address of a backend server is used as the domain name for a health check, do not rely on the host header for verification or judgment.

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 1-53**.

Table 1-53 Factors affecting the health check time window

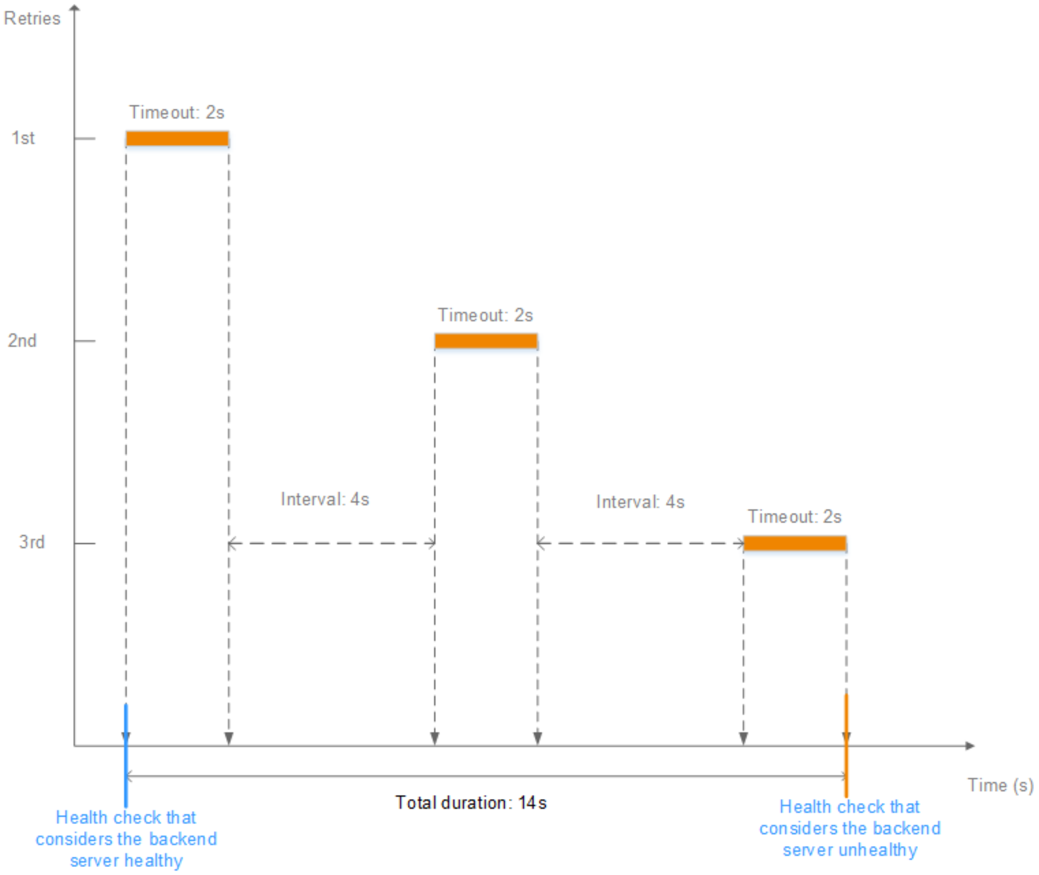
Factor	Description
Interval	How often health checks are performed.
Timeout duration	How long the load balancer waits for the response from the backend server.
Health check threshold	The number of consecutive successful or failed health checks required for determining whether the backend server is healthy 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 × Healthy threshold + Interval × (Healthy threshold – 1)
- Time window for a backend server to be detected unhealthy = Timeout duration × Unhealthy threshold + Interval × (Unhealthy threshold – 1)

In **Figure 1-21**, the health check interval is 4s, timeout duration is 2s, and unhealthy threshold is 3, so the time window for a backend server to be considered unhealthy is calculated as follows: 2 × 3 + 4 × (3 – 1) = 14s.

Figure 1-21 Health check timeout window


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

1.7.2 Configuring a Health Check

Scenarios

This section describes how you can enable or disable the health check option.

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

- 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. For details, see **Security Group and Network ACL 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. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, locate the backend server group and click its name.
3. On the **Summary** page, click **Health Check** on the right.
4. In the **Configure Health Check** dialog box, configure the parameters based on **Table 1-54**.

Table 1-54 Parameters required for configuring health check

Parameter	Description	Example Value
Health Check	Specifies whether to enable the health check option. **NOTE** When the health check is enabled or disabled, the number of healthy or unhealthy backend servers may temporarily fluctuate but will stabilize after a monitoring period.	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 protocol of the backend server group is UDP, the health check protocol is UDP by default. Dedicated load balancers support TCP, HTTP, and HTTPS.	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 length: 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 (/). If the backend server group is associated with a dedicated load balancer, the check path can contain letters, digits, hyphens (-), slashes (/), periods (.), question marks (?), number signs (#), percent signs (%), ampersands (&), and extended character sets _;~!. () *[]@$^:',+	/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 value ranges from **1** to **50**.	3
Healthy Threshold	Specifies the number of consecutive successful health checks required for declaring a backend server healthy. The value ranges from **1** to **10**.	3
Unhealthy Threshold	Specifies the number of consecutive failed health checks required for declaring a backend server unhealthy. The value ranges from **1** to **10**.	3

5. Click **OK**.

Disabling Health Check

1. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, click the name of the target backend server group.
3. On the **Summary** page, click **Health Check** on the right.
4. In the **Configure Health Check** dialog box, disable health check.
5. Click **OK**.

1.8 Security

1.8.1 Using Dedicated Load Balancers to Transfer Client IP Address

Overview

Dedicated load balancers transfer client IP addresses in different ways based on whether they use Layer 4 or Layer 7 listeners to route requests.

- **Transfer Client IP Address** is enabled by default for TCP and UDP listeners of dedicated load balancers. Load balancers communicate with backend servers using client IP addresses. You can check the backend server logs to obtain client IP addresses.
- **Transfer Client IP Address** is enabled by default for HTTP, HTTPS, and QUIC listeners of dedicated load balancers, which means that client IP addresses can be placed in the X-Forwarded-For header and transferred to backend

servers. The first IP address in the X-Forwarded-For header is the client IP address.

Precautions

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 use the same server, the backend server will think the packet from the client is sent by itself and will not return a response packet to the load balancer. As a result, the return traffic will be interrupted.
- Traffic, such as unidirectional data transmission or push traffic, may be interrupted when backend servers are being migrated. After backend servers are migrated, you need to retransmit the packets to restore the traffic.

Transferring Client IP Addresses at Layer 4

In some special cases, **Transfer Client IP Address** does not work. You can obtain client IP addresses by referring to **Table 1-55**.

Table 1-55 Transferring client IP addresses at Layer 4

Listener Protocol	Transfer Client IP Address	When Transfer Client IP Address Fails
TCP	Supported	● TCP listeners communicate with IP as backend servers. ● IPv4/IPv6 translation is enabled for TCP listeners. In this case, client IP addresses are translated.
UDP	Supported	● UDP listeners communicate with IP as backend servers. ● IPv4/IPv6 translation is enabled for UDP listeners. In this case, client IP addresses are translated.

Transferring Client IP Addresses at Layer 7

You can configure the backend servers to ensure that they can correctly parse the X-Forwarded-For header to obtain client IP addresses.

The X-Forwarded-For header is in the following format:

```
X-Forwarded-For: <client-IP-address>, <proxy-server-1-IP-address>, <proxy-server-2-IP-address>, ...
```

The first IP address included in the X-Forwarded-For header is the client IP address.

1.8.2 Configuring TLS Security Policies for Encrypted Communication

HTTPS encryption is commonly used for applications that require secure data transmission, such as banks and finance. ELB allows you to use common TLS security policies to secure data transmission.

When you add HTTPS listeners, you can select the default security policies or create a custom policy by referring to **Creating a Custom Security Policy** to improve security.

A security policy is a combination of TLS protocols of different versions and supported cipher suites.

Default Security Policies

A later TLS version ensures higher HTTPS communication security, but is less compatible with some browsers.

You can use later TLS versions for applications that require enhanced security, and earlier TLS versions for applications that need wider compatibility.

Table 1-56 Default security policies

Security Policy	TLS Versions	Cipher Suites
tls-1-0	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 1.2 TLS 1.1	
tls-1-2	TLS 1.2	

Security Policy	TLS Versions	Cipher Suites
tls-1-0-inherit	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 ● DHE-RSA-AES128-SHA ● ECDHE-RSA-AES256-SHA ● ECDHE-ECDSA-AES256-SHA ● AES128-SHA ● AES256-SHA ● DHE-DSS-AES128-SHA ● CAMELLIA128-SHA ● EDH-RSA-DES-CBC3-SHA ● DES-CBC3-SHA ● ECDHE-RSA-RC4-SHA ● RC4-SHA ● DHE-RSA-AES256-SHA ● DHE-DSS-AES256-SHA ● DHE-RSA-CAMELLIA256-SHA

Security Policy	TLS Versions	Cipher Suites
tls-1-2-strict	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
tls-1-0-with-1-3	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	TLS Versions	Cipher Suites
tls-1-2-fs-with-1-3	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
tls-1-2-fs	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-1-2-strict-no-cbc	TLS 1.2	● ECDHE-ECDSA-AES256-GCM-SHA384 ● ECDHE-ECDSA-AES128-GCM-SHA256 ● ECDHE-RSA-AES256-GCM-SHA384 ● ECDHE-RSA-AES128-GCM-SHA256

NOTE

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

Differences Among Default Security Policies

√ indicates the item is supported, and x indicates the item is not supported.

Table 1-57 Differences between TLS security policies

Security Policy	**tls-1-0**	**tls-1-1**	**tls-1-2**	**tls-1-0-inherit**	**tls-1-2-strict**	**tls-1-0-with-1-3**	**tls-1-2-fs-with-1-3**	**tls-1-2-fs**	**tls-1-2-strict-no-cbc**
Protocol-TLS 1.3	×	×	×	×	×	√	√	√	×
Protocol-TLS 1.2	√	√	√	√	√	√	√	√	√
Protocol-TLS 1.1	√	√	×	√	×	√	×	×	×
Protocol-TLS 1.0	√	×	×	√	×	√	×	×	×

Table 1-58 Differences between TLS security policies (cipher suites)

Security Policy	**tls-1-0**	**tls-1-1**	**tls-1-2**	**tls-1-0-inherit**	**tls-1-2-strict**	**tls-1-0-with-1-3**	**tls-1-2-fs-with-1-3**	**tls-1-2-fs**	**tls-1-2-strict-no-cbc**
ECDHE-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	√	√	√	√	√	√	×	×	×

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-0-inherit	tls-1-2-strict	tls-1-0-with-1-3	tls-1-2-fs-with-1-3	tls-1-2-fs	tls-1-2-strict-no-cbc
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	√	√	√	√	×	√	×	×	×
ECDHE-RSA-AES128-GCM-SHA256	×	×	×	√	×	√	√	√	×
TLS_AES_256_GCM_SHA384	×	×	×	×	×	√	√	√	×

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-0-inherit	tls-1-2-strict	tls-1-0-with-1-3	tls-1-2-fs-with-1-3	tls-1-2-fs	tls-1-2-strict-no-cbc
TLS_CHACHA20_POLY1305_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_GCM_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_CCM_8_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_CCM_SHA256	×	×	×	×	×	√	√	√	×
DHE-RSA-AES128-SHA	×	×	×	√	×	×	×	×	×
DHE-DSS-AES128-SHA	×	×	×	√	×	×	×	×	×
CAMELLIA128-SHA	×	×	×	√	×	×	×	×	×
EDH-RSA-DES-CBC3-SHA	×	×	×	√	×	×	×	×	×
DES-CBC3-SHA	×	×	×	√	×	×	×	×	×
ECDHE-RSA-RC4-SHA	×	×	×	√	×	×	×	×	×
RC4-SHA	×	×	×	√	×	×	×	×	×
DHE-RSA-AES256-SHA	×	×	×	√	×	×	×	×	×
DHE-DSS-AES256-SHA	×	×	×	√	×	×	×	×	×
DHE-RSA-CAMELLIA256-SHA	×	×	×	√	×	×	×	×	×

Table 1-59 Security policies and compatible browsers and clients

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-0-inherit	tls-1-2-strict	tls-1-0-with-1-3	tls-1-2-fs-with-1-3	tls-1-2-fs	tls-1-2-strict-no-cbc
Android 8.0	√	√	√	√	√	√	√	√	√
Android 9.0	√	√	√	√	√	√	√	√	√
Chrome 70 / Win 10	√	√	√	√	√	√	√	√	√
Chrome 80 / Win 10	√	√	√	√	√	√	√	√	√
Firefox 62 / Win 7	√	√	√	√	√	√	√	√	√
Firefox 73 / Win 10	√	√	√	√	√	√	√	√	√
IE 8 / XP	√	√	√	√	×	√	×	×	×
IE 8-10 / Win 7	√	√	√	√	×	√	×	×	×
IE 11 / Win 7	√	√	√	√	√	√	√	√	√
IE 11 / Win 10	√	√	√	√	√	√	√	√	√
Edge 15 / Win 10	√	√	√	√	√	√	√	√	√
Edge 16 / Win 10	√	√	√	√	√	√	√	√	√
Edge 18 / Win 10	√	√	√	√	√	√	√	√	√
Java 8u161	√	√	√	√	√	√	√	√	√
Java 11.0.3	√	√	√	√	√	√	√	√	√
Java 12.0.1	√	√	√	√	√	√	√	√	√
OpenSSL 1.0.2s	√	√	√	√	√	√	√	√	√
OpenSSL 1.1.0k	√	√	√	√	√	√	√	√	√
OpenSSL 1.1.1c	√	√	√	√	√	√	√	√	√
Safari 10 / iOS 10	√	√	√	√	√	√	√	√	√

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-0-inherit	tls-1-2-strict	tls-1-0-with-1-3	tls-1-2-fs-with-1-3	tls-1-2-fs	tls-1-2-strict-no-cbc
Safari 10 / OS X 10.12	√	√	√	√	√	√	√	√	√
Safari 12.1.1 / iOS 12.3.1	√	√	√	√	√	√	√	√	√

Creating a Custom Security Policy

ELB allows you to use common TLS security policies to secure data transmission. If you need to use a certain TLS version and disable some cipher suites, you can create a custom security policy and add it to an HTTPS listener to improve service security.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **TLS Security Policies**.
3. On the displayed page, click **Create Custom Security Policy** in the upper right corner.
4. Configure the parameters based on **Table 1-60**.

Table 1-60 Custom security policy parameters

Parameter	Description
Name	Specifies the name of the custom security policy.
TLS Version	Specifies the TLS version supported by the custom security policy. You can select multiple versions: ● TLS 1.0 ● TLS 1.1 ● TLS 1.2 ● TLS 1.3
Cipher Suite	Specifies the cipher suites that match the selected TLS versions.
Description (Optional)	Provides supplementary information about the custom security policy.

5. Click **OK**.

Managing a Custom Security Policy

After a custom security policy is created, you can modify or delete it.

Modifying a Custom Security Policy

You can modify the name, TLS versions, cipher suites, and description of a custom security policy as required.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **TLS Security Policies**.
3. On displayed page, locate the custom security policy, and click **Modify** in the **Operation** column.
4. In displayed dialog box, modify the custom security policy based on **Table 1-60**.
5. Click **OK**.

Deleting a Custom Security Policy

You can delete a custom security policy as you need.

NOTE

If a custom security policy is used by a listener, it cannot be deleted. Delete the security policy from the listener first.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **TLS Security Policies**.
3. On displayed page, locate the custom security policy, and click **Delete** in the **Operation** column.
4. In the displayed dialog box, click **OK**.

Selecting a Security Policy for an HTTPS Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**.
4. On the **Add Listener** page, set **Frontend Protocol** to **HTTPS**.
5. Expand **Advanced Settings (Optional)** and select a security policy.

   You can select a **default security policy** or custom security policy.

   If there is no custom security policy, you can create one by referring to **Creating a Custom Security Policy**.
6. Confirm the configurations and go to the next step.

Changing a Security Policy for an HTTPS Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, locate the listener, and click its name.

4. On the **Summary** tab, click **Edit** on the top right.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)** and change the security policy.
6. Click **OK**.

1.8.3 Using SNI Certificates for Access Through Multiple Domain Names

Server Name Indication (SNI) is an extension to TLS. It allows clients to specify which domain name of a listener they are trying to connect in the first request. Once receiving the request, the load balancer searches for the certificate based on the domain name.

SNI Overview

Suppose a listener is associated with a server that hosts multiple HTTPS services, each with its own certificate and domain name.

If the HTTPS listener has only one server certificate, it will always present that same certificate to all clients, regardless of the domain name the clients are trying to access. This may make authentication abnormal.

To address this issue, you can enable SNI when you add an HTTPS listener, allowing the listener to select the right certificate for authentication based on the requested domain name. SNI allows clients to specify which domain name they are trying to connect in the initial SSL handshake. Once receiving the request, the load balancer searches for the certificate based on the domain name. If there is no match, the load balancer uses the default server certificate for authentication.

SNI Certificate

- SNI certificates are server certificates used for multi-domain-name authentication. Each certificate must have an SNI domain name. The SNI domain name specified on the ELB console must be the same as the domain name supported by the certificate for authentication.
- A domain name can be used by both an ECC certificate and an RSA certificate. If this happens, ELB selects the ECC certificate first.

Constraints

- Only HTTPS listeners support SNI. After SNI is enabled, you need to configure at least one SNI certificate for the listener. For details about how to add a certificate, see **Adding a Certificate**.
- If a certificate has expired, you need to manually replace or delete it by following the instructions in **Binding or Replacing a Certificate**.
- An HTTPS listener can have up to 30 SNI certificates. All the certificates can have up to 30 domain names.

  NOTE

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

How SNI Certificates and Domain Names Are Matched

- Domain names in an SNI certificate are matched as follows:

  If the domain name of the certificate is *.test.com, a.test.com and b.test.com are supported, but a.b.test.com and c.d.test.com are not supported.

  The domain name with the longest suffix is matched. If a certificate contains both *.b.test.com and *.test.com, a.b.test.com preferentially matches *.b.test.com.

- **cert-default** is the default server certificate bound to the HTTPS listener, and **cert-test01** and **cert-test02** are SNI certificates.

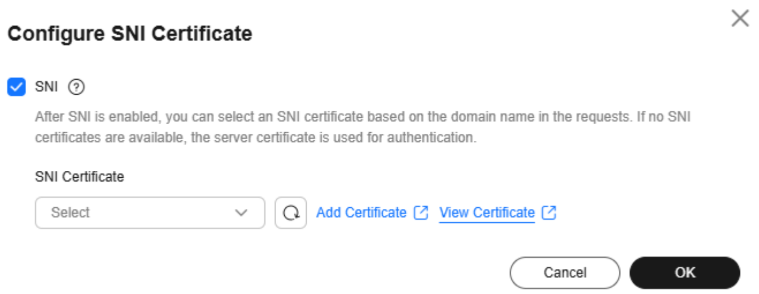
  The domain name of **cert-test01** is **www.test01.com** and that of **cert-test02** is **www.test02.com**.

  If the requested domain name matches either of the domain names, the corresponding SNI certificate will be used for authentication. If no domain name is matched, the default server certificate is used for authentication.

Enabling SNI for an HTTPS Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Summary** tab, click **Configure** on the right of SNI.
5. Enable SNI and select an SNI certificate.

Figure 1-22 Configuring an SNI certificate



Configure SNI Certificate

☑ SNI

After SNI is enabled, you can select an SNI certificate based on the domain name in the requests. If no SNI certificates are available, the server certificate is used for authentication.

SNI Certificate: Select

6. Click **OK**.

Helpful Links

- **Adding a Certificate**
- **Adding an HTTPS Listener**

1.8.4 Certificate

1.8.4.1 Certificate Overview

When you add 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.

Use Cases

When you add an HTTPS or TLS listener to route requests, you need to select **SSL Authentication**. For one-way authentication, you need to configure a server certificate for the listener. For two-way authentication, you need to configure both a server certificate and a CA certificate.

Table 1-61 SSL authentication

One-way Authentication	Only backend servers will be authenticated. You need to bind a server certificate to the listener to authenticate the server.
Mutual Authentication	The clients and the load balancer authenticate each other. Only authenticated clients will be allowed to access the load balancer. You need to bind both a server certificate and a CA certificate to the listener to allow the clients and the load balancer to authenticate each other. You do not need to configure two-way authentication on the backend servers.

ELB supports two types of certificates.

Table 1-62 Certificate types

Server Certificate	Used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
CA Certificate	Also called client CA public key certificate and used to verify the client certificate issuer. If mutual authentication is required, connections can be established only when the client provides a certificate issued by a specific CA.

Precautions

- A certificate can be used by multiple load balancers but only needs to be uploaded to ELB once.
- You must specify a domain name for an SNI certificate. The domain name must be the same as that in the certificate. An SNI certificate can have multiple domain names.
- 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. It is recommended that you use certificates issued by third parties.

- ELB only supports certificates 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.

Certificate Format

You can copy and paste the certificate body to create a certificate or directly upload a 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 requirements as described below.

- The content must start 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-----
Base64–encoded certificate
-----END CERTIFICATE-----
```

Private Key Format

When creating a server 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.

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 contains 64 characters except the last row.

The following is an example:

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

Converting Certificate Formats

ELB only supports certificates 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 and Tomcat servers.

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 servers.

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
```

1.8.4.2 Adding a Certificate

Scenarios

To enable authentication for securing data transmission over HTTPS, ELB allows you to bind the following certificates to HTTPS listeners of a load balancer:

- Server certificate: used for SSL handshake negotiations if an HTTPS listener is used. You can purchase a certificate from Cloud Certificate Manager (CCM) or upload your own certificates.
- CA certificate: a certificate issued by a certificate authority (CA). They are used to verify the client certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can only be established when the client provides a certificate issued by a specific CA. You can only upload your own CA certificates.

NOTE

If you want to use the same certificate in two regions, you need to add a certificate in each region.

Adding a Server Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Click **Add Certificate** on the top right corner and set parameters by referring to **Table 1-63**.

Table 1-63 Server certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **Server certificate**. **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
Source	Specifies the source of a certificate. You can purchase a certificate from CCM or upload your own certificates. ● **SSL Certificate Manager**: server certificates provided by CCM. You need to buy a certificate or upload your own certificates. ● **Your certificate**: You need to upload the certificate content and private key of your own certificate to the ELB console. **NOTE** You are advised to use CCM to manage your certificates.
Certificate	This parameter is only available for certificates managed on the CCM console. You can select a certificate managed by CCM.
Certificate Name	Specifies the name of your certificate. A certificate name can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

Certificate Content	Specifies the content of a certificate. This parameter is only available for your certificates.
	The content must be in PEM format.
	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-----`

Parameter	Description

Private Key	Specifies the private key of a certificate. This parameter is only available for your certificates.
	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 as follows:
	`-----BEGIN PRIVATE KEY-----`
	`[key]`
	`-----END PRIVATE KEY-----`

SNI Domain Name (Optional)	The domain name must be specified if the certificate is intended for SNI. A domain name can contain only letters, digits, and hyphens (-) and consist of multiple labels (max. 63 characters each) separated by periods (.). It cannot start or end with a hyphen (-). You can specify up to 100 domain names, separated by commas (,). A domain name can contain a maximum of 100 characters, and the total length cannot exceed 10,000 characters.
Description	(Optional) Provides supplementary information about the certificate.

4. Click **OK**.

Adding a CA Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Click **Add Certificate** on the top right corner and set parameters by referring to **Table 1-64**.

Table 1-64 CA certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **CA certificate**. **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can only be established when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of the CA certificate.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

Parameter	Description

Certificate Content	Specifies the content of the CA certificate in PEM format.
	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-----`

Description	(Optional) Provides supplementary information about the certificate.

4. Click **OK**.

1.8.4.3 Managing Certificates

Scenarios

You can manage your certificates on the ELB console. If a certificate is no longer needed, you can delete it.

Constraints

A certificate that has been bound to an HTTPS listener cannot be deleted. Disassociate the certificate from the listener first by referring to **Replacing a Certificate**.

Querying Listeners by Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. 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.

Modifying a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Modify** in the **Operation** column.
4. In the **Modify Certificate** dialog box, modify the parameters as required.
5. Confirm the information and click **OK**.

Deleting a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Delete** in the **Operation** column.
4. In the displayed dialog box, click **OK**.

1.8.4.4 Binding or Replacing a Certificate

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.

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 a certificate and private keys does not affect your applications.

Notes and Constraints

- Only HTTPS listeners require certificates.
- If a certificate has expired, you need to manually replace or delete it.
- The new certificate takes effect immediately. The old certificate is used for established connections, and the new one is used for new connections.

Prerequisites

You have added 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. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click the **Listeners** tab, locate the listener, and click **Edit** in **Operation** column.
4. On the displayed dialog box, select a server certificate or CA certificate.
5. Click **OK** in the **Edit** dialog box.

1.8.4.5 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.

Notes and Constraints

- Only HTTPS listeners require certificates.
- The new certificate takes effect immediately. The previous certificate is used for established connections, and the new one is used for new connections.

Modifying a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Modify** in the **Operation** column.
4. Modify the parameters as required.
5. Confirm the information and click **OK**.

1.8.5 Access Control

1.8.5.1 What Is 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.

NOTE

Network ACL rules configured for the frontend subnet of a load balancer do not restrict traffic from clients to the load balancer. Use access control to limit which IP addresses can access the load balancer.

Whitelist and Blacklist

You can set a whitelist or blacklist to control access to a listener.

- **Whitelist**: Only the IP addresses or CIDR blocks specified in the IP address group selected for the whitelist can access the listener.

  Access control policies only take effect for new connections, but not for existing ones. 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, it may be caused by a persistent connection 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.

- **Blacklist**: The IP addresses or CIDR blocks specified in the IP address group selected for the blacklist cannot access the listener.

NOTE

- Access control does not restrict the ping command. You can still ping a load balancer from restricted IP addresses.
- Whitelists and blacklists do not conflict with inbound security group rules. Access control defines the IP addresses or CIDR blocks that are allowed or denied to access listeners, while inbound security group rules control access to backend servers. Requests first match the access control policy then the security group rules before they finally reach backend servers.

Configuring Access Control

⚠ WARNING

Note that modifying an access control policy may interrupt your services or cause network security risks.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. 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**.
4. In the displayed **Configure Access Control** dialog box, configure parameters as described in **Table 1-65**.

Table 1-65 Parameter description

Parameter	Description
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.
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 **What Is an IP Address Group?**

Parameter	Description
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.

5. Click **OK**.

1.8.5.2 IP Address Group

What Is an IP Address Group?

An IP address group allows you to manage a collection of IP addresses that have the same security requirements or whose security requirements change frequently.

If you want to use a whitelist or blacklist for **access control**, you must select an IP address group. For details, see **What Is Access Control?**

- **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 a 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 a 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.

Creating an IP Address Group

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the displayed page, click **Create IP Address Group**.
4. Configure the parameters based on **Table 1-66**.

Table 1-66 IP address group parameters

Parameter	Description	Example Value
Name	Specifies the name of the IP address group.	ipGroup-01

Parameter	Description	Example Value
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed. For details, see the **Enterprise Management User Guide**.	N/A
IP Addresses	Specifies IPv4 or IPv6 IP addresses or CIDR blocks that will be added to the whitelist or blacklist for access control. ● Each line contains a single IP address, a CIDR block, or an IP address range, and ends with a line break. ● You can add remarks at the end of each IP address or CIDR block and separate them with a vertical bar (|). The remarks can be up to 255 characters long. Angle brackets (<>) are not allowed. ● You can add a maximum of 300 entries (including IP addresses and CIDR blocks) to each IP address group.	● Without remarks: 10.168.2.24 ● With remarks: 10.168.16.0/24 | ECS01
Description (Optional)	Provides supplementary information about the IP address group.	N/A

5. Click **OK**.

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**

The IP addresses can be in the formats as described in **Table 1-66**.

Adding IP Addresses

You can add IP addresses to an existing IP address group.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.

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

Changing IP Addresses

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

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. 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. You can modify the name and description of an IP address group, and change all its IP addresses.
      ii. Click **OK**.
   b. Only change IP addresses:
      i. Locate the target IP address group and click its name.
      ii. In the lower part of the displayed page, choose **IP Addresses** tab, click **Change IP Addresses**, and change IP addresses as you need.
      iii. 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. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the target IP address group and click its name.
4. In the IP address list, locate the IP address you want to delete and click **Delete** in the **Operation** column.
5. Confirm the information and click **OK**.

Viewing the Details of an IP Address Group

You can view the details of an IP address group, including:

- Name, ID, and creation time

- IP addresses and CIDR blocks
- Associated listeners

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the target IP address group and click its name.
4. Viewing the basic information about the IP address group.
   a. On the **IP Addresses** tab, view the IP addresses or CIDR blocks.
   b. On the **Associated Listeners** tab, view the listeners associated with the IP address group.

Deleting an IP Address Group

If an IP address group is used for controlling access to a listener, it 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**.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the IP address group and click **Delete** in the **Operation** column.
4. Click **OK**.

1.9 Enabling Access Logging for Your Load Balancer

When you use HTTP/HTTPS listeners of a load balancer to route requests, if there is an unhealthy backend server, it is challenging to quickly locate the root cause by checking the logs of this backend server.

Log Tank Service (LTS) can log Layer 7 requests, of a load balancer including the time when the request was sent, client IP address, request path, server response, and more. If there are service faults or exceptions caused by unhealthy backend servers, you can view logs of requests to load balancers and analyze response status codes to quickly locate unhealthy backend servers.

⚠ WARNING

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

What Is ELB Access Logging?

ELB receives and distributes client access requests. It logs the details of Layer 7 requests. With LTS, you can quickly analyze service requests and locate faults.

The following information is logged:

- Time information: Fields such as **msec** and **time_iso8601** record the time when requests were sent. They are used for analyzing time sequence and locating faults.
- Basic client request information: Fields such as **remote_addr:remote_port**, **request_method scheme://host request_uri server_protocol**, and **http_user_agent** record basic request information, which is used for user analysis and security audit.
- Request response and performance metrics: Fields such as **status**, **bytes_sent**, and **request_time** record response status and processing time, which are used for locating request status and monitoring performance.
- Load balancer information: Fields such as **lb_name**, **listener_id**, **pool_name**, and **eip_address:eip_port** record the details of load balancers that are used to route requests. They are used to identify the resource configuration and improve O&M efficiency.
- Backend server information: Fields such as **upstream_status**, **upstream_connect_time**, and **upstream_addr_priv** record the information returned by backend servers and their configurations. They are used to identify the health status and performance of the backend servers.
- HTTPS information: Fields such as **ssl_protocol**, **sni_domain_name**, and **certificate_id** record the HTTPS information for troubleshooting HTTPS requests.
- Other information: Fields such as **access_log_topic_id**, **log_ver**, and **tenant_id** record the system and log IDs for log management.

Constraints

- Access logging can be configured only for load balancers with HTTP or HTTPS listeners.
- The access logs do not contain requests whose return code is **400 Bad Request**. This is because such requests do not comply with HTTP specification and cannot be processed properly.

Preparations

- Create a load balancer that supports HTTP and HTTPS.
- Enable LTS.
- Create a backend server group, add backend servers to the group, and deploy services on the backend servers. For details, see **Creating a Backend Server Group**.
- Add an HTTP or HTTPS listener to the load balancer.

Configuring Access Logging

1. Go to the **load balancer list page**.

2. On the **Load Balancers** page, 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 have created.

   If there are no log groups and log streams, you can create them on the ELB console.

   Creating a Log Group and Log Stream

   a. Click **Create Log Group and Log Stream**.

   b. In the **Create Log Group and Log Stream** dialog box, configure the log group name, log stream name, and log retention period.

   c. Click **OK**.
5. Click **OK**.

Viewing Access Logs

You can view details about access logs on the:

- ELB console: Go to the **Access Logs** tab of the target load balancer to view access logs.
- (Recommended) LTS console: Locate the target log group and click its name. On the displayed page, locate the target log stream and click **Real-Time Logs** tab.

The log format is as follows, which 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 $request_header_length
$actions_executed $error_reason "$pool_usr_name"
```

The following is a log example:

```
1644819836.370 eb11c5a9-93a7-4c48-80fc-03f61f638595 [2024-02-14T14:23:56+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" "192.168.1.2:8080" "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 "-" 129 waf - "My backend server group"
```

Log analysis:

At 14:23:56 GMT+08:00 on Feb 14, 2024, the load balancer received an HTTP/1.1 POST request from a client whose IP address and port number are 192.168.1.1 and 888, then routed the request to a backend server whose IP address and port number are 100.64.0.129 and 8080, and finally returned 200 OK to the client after receiving the status code from the backend server.

Log analysis result:

The backend server responds to the request normally.

Table 1-67 describes the fields in the log.

Table 1-67 ELB log fields

Parameter	Description	Value Description	Example Value
msec	Time when the log is written, in seconds with a milliseconds resolution.	Floating-point data	1644819836.370
access_log_topic_id	Log stream ID.	uuid	eb11c5a9-93a7-4c48-80fc-03f61f638595
time_iso8601	Local time in the ISO 8601 standard format.	N/A	[2024-02-14T14:23:56+08:00]
log_ver	Log format version.	Fixed value: **elb_01**	elb_01
remote_addr:remote_port	IP address and port number of the client.	Records the IP address and port of the client.	192.168.1.1:888
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 and it does not include the protocol and host name.	"POST https://www.test.com/example/ HTTP/1.1"
request_length	Length of the request received from the client, including the header and body.	Integer	1411
bytes_sent	Number of bytes sent to the client.	Integer	251

Parameter	Description	Value Description	Example Value
body_bytes_sent	Number of bytes sent to the client (excluding the response header).	Integer	3
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	0.011
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"
upstream_connect_time	Time taken to establish a connection with the server, in seconds, with a milliseconds 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	"0.000"

Parameter	Description	Value Description	Example Value
upstream_header_time	Time taken to receive the response header from the server, in seconds, with a milliseconds 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	"0.011"
upstream_response_time	Time taken to receive the response from the server, in seconds, with a milliseconds 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	"0.011"
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 *-*.	IP address and port number	"192.168.1.2:8080"

Parameter	Description	Value Description	Example Value
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.	"okhttp/3.13.1"
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_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	"-"
lb_name	Load balancer name in the format of **loadbalancer**_*load balancer ID*	String	loadbalancer_295a7eee-9999-46ed-9fad-32a62ff0a687
listener_name	Listener name in the format of **listener**_*listener ID*.	String	listener_20679192-8888-4e62-a814-a2f870f62148
listener_id	ID of the listener added to the load balancer.	String	3333fd44fe3b42cbaa1dc2c641994d90
pool_name	Backend server group name in the format of **pool**_*backend server group ID* or **pool**_*backend server group ID***load balancer ID*.	String	pool_89547549-6666-446e-9dbc-e3a551034c46

Parameter	Description	Value Description	Example Value
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	"-"
tenant_id	Tenant ID.	String	f2bc165ad9b4483a9b17762da851bbbb
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.	121.64.212.1:443
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 **-**.	IP address and port number	"-" (Dedicated load balancers)
certificate_id	HTTPS listener: Certificate ID used for establishing an SSL connection. This field is not supported yet.	String	N/A
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	TLSv1.2

Parameter	Description	Value Description	Example Value
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
sni_domain_name	HTTPS listener: SNI domain name provided by the client during SSL handshakes. 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	56704
self_defined_header	This field is reserved. The default value is **-**.	String	"-"
request_header_length	The size of the header in the client request.	Integer	129
actions_executed	The WAF execution result.	String ● **waf**: Requesting WAF succeeded. ● **waf-failed**: Requesting WAF failed. ● **waf-block**: The request is blocked by WAF.	waf

Parameter	Description	Value Description	Example Value
error_reason	Reason why requesting WAF fails.	String ● **WAFUnhandledException**: Internal error. Contact the WAF service personnel to locate the fault. ● **WAFRequestHeaderLengthExceeded**: The client request header length exceeds the upper limit. ● **WAFRequestBodyLengthExceeded**: The client body is too large. ● **WAFRequestHeaderContentLengthEmpty**: The body length of the client's request header is 0. ● **WAFResponseBodyReadError**: Failed to read the body returned by WAF. ● **WAFResponseReadTimeout**: Reading the result returned by WAF times out. ● **WAFConnectionTimeout**: Connecting to WAF times out. ● **WAFConnectionError**: Failed to connect to WAF. ● **WAFNoBackendAvailable**: No	N/A

Parameter	Description	Value Description	Example Value
		backend server is available for WAF. ● **WAFNoBackendOnline**: No backend server is online for WAF.	

Locating an Unhealthy Backend Server

The following is a log that records an exception:

```
1554944564.344 - [2024-04-11T09:02:44+08:00] elb 10.133.251.171:51527 500 "GET http://10.154.73.58/
lrange/guestbook HTTP/1.1" 411 3726 3545 19.028 "500" "0.009" "19.028" "19.028" "172.17.0.82:3000"
"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
73.0.3683.103 Safari/537.36" "http://10.154.73.58:5971/" "-" loadbalancer_ed0f790b-
e194-4657-9f97-53426227099e listener_b21dd0a9-690a-4945-950e-b134095c6bd9
6b6aaf84d72b40fcb2d2b9b28f6a0b83
```

Log analysis

At 09:02:44 GMT+08:00 of April 11, 2024, the load balancer received a GET/ HTTP/1.1 request from the client whose IP address and port number are 10.133.251.171 and 51527 respectively and then routed the request to a backend server that uses 172.17.0.82 and port 3000 to receive requests. The load balancer then received 500 Internal Server Error from the backend server and returned the status code to the client.

Analysis results

The backend server (private IP address: 172.17.0.82; port: 3000) was unhealthy and failed to respond to the request.

1.10 Tags and Quotas

1.10.1 Tag

Scenarios

If you have a large number of cloud resources, you can add different tags to the resources to quickly identify them and use these tags to easily manage your resources.

Adding a Tag to a Load Balancer

You can add a tag to a load balancer in the following methods:

- Add a tag when you create a load balancer.

- Add a tag to an existing load balancer.
  a. Go to the **load balancer list page**.
  b. On the **Load Balancers** page, locate the load balancer and click its name.
  c. Under **Tags**, click **Add Tag**.

     Each tag is a key-value pair, and the tag key is unique.
  d. In the **Add Tag** dialog box, enter a tag key and value and click **OK**.

Modifying a Tag

1. Go to the **load balancer list page**.
2. On the **Load Balancers** page, locate the load balancer and click its name.
3. Click **Tags**, select the tag to be edited, and click **Edit** in the **Operation** column. In the **Edit Tag** dialog box, enter a tag value.

   NOTE

   The tag key cannot be modified.
4. Click **OK**.

Deleting a Tag

1. Go to the **load balancer list page**.
2. On the **Load Balancers** page, locate the load balancer and click its name.
3. Click **Tags**, select the tag to be deleted, and click **Delete** in the **Operation** column.
4. In the displayed dialog box, click **OK**.

1.10.2 Quotas

What Is Quota?

Quotas can limit the number 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. Click 📍 in the upper left corner and select the desired region and project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.
4. 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?

1. Log in to the management console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Quotas** page is displayed.
3. Click **Increase Quota** in the upper right corner of the page.
4. On the **Create Service Ticket** page, configure parameters as required.

   In the **Problem Description** area, fill in the content and reason for adjustment.
5. After all necessary parameters are configured, select **I have read and agree to the Ticket Service Protocol and Privacy Statement** and click **Submit**.

1.11 Cloud Eye Monitoring

1.11.1 Monitoring ELB Resources

Scenarios

Cloud Eye is a multi-dimensional resource monitoring service. You can use Cloud Eye to monitor ELB resources in real time, set alarm rules, identify resource exceptions, and quickly respond to resource changes.

Cloud Eye is enabled automatically after you create a load balancer.

Setting an Alarm Rule

You can set alarm rules on the Cloud Eye console to send you notifications in case of exceptions.

For details about how to set alarm rules, see .

Viewing Monitoring Metrics

You can view the metrics described in **ELB Monitoring Metrics** either on the ELB console or on the Cloud Eye console.

Viewing Monitoring Metrics on the ELB Console

1. Go to the **load balancer list page**.
2. On the load balancer list page, locate the load balancer and click its name.
3. You can view metrics by load balancer, listener, and backend server group.
   a. Load balancer: Click the **Monitoring** tab and select **Load balancer** for **Dimension**.
   b. Listener (two ways):
      i. Click the **Monitoring** tab, select **Listener** for **Dimension**, select the target listener, and view the monitoring metrics.
      ii. Click the **Listeners** tab, locate the target listener, and click its name. Switch to the **Monitoring** tab and view the monitoring metrics.

c. Backend server group: Click the **Monitoring** tab and select **Backend server group** for **Dimension**.

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 & Governance** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Cloud Service Monitoring**. In the displayed page, locate the **Dashboard** column and click **Elastic Load Balance ELB**.
5. On the displayed page, locate the target load balancer and click its name.

   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.

1.11.2 ELB 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.

Namespace

SYS.ELB

Load Balancer Metrics

For dedicated load balancers, you can view the monitoring metrics by load balancer, listener, backend server group, or AZ. You can view only the Layer 7 metrics of a backend server group.

Table 1-68 Metrics supported by each dedicated load balancer

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections established between the monitored object and backend servers. Load balancing at Layer 7: total number of TCP connections established between clients and the monitored object.	≥ 0	Count	N/A	Dedicated load balancer	1 minute
m2_act_conn	Active Connections	The number of active TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Dedicated load balancer	1 minute
m3_inact_conn	Inactive Connections	The number of inactive TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m4_ncps	New Connections	The number of new connections established between clients and the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m5_in_pps	Incoming Packets	The number of packets received by the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m6_out_pps	Outgoing Packets	The number of packets sent from the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m7_in_Bps	Inbound Traffic Rate	How fast the inbound traffic reaches the monitored object.	≥ 0	Byte/s	1000 (SI)	Dedicated load balancer	1 minute
m8_out_Bps	Outbound Traffic Rate	How fast the outbound traffic leaves the monitored object.	≥ 0	Byte/s	1000 (SI)	Dedicated load balancer	1 minute
m9_abnormal_servers	Unhealthy Servers	The number of unhealthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer	1 minute
ma_normal_servers	Healthy Servers	The number of healthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer	1 minute
m22_in_bandwidth	Inbound Bandwidth	The bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m23_out_bandwidth	Outbound Bandwidth	The bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer	1 minute
m26_in_bandwidth_ipv6	IPv6 Inbound Bandwidth	The IPv6 network bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer	1 minute
m27_out_bandwidth_ipv6	IPv6 Outbound Bandwidth	The IPv6 network bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer	1 minute
m1e_server_rps	Reset Packets from Backend Servers	The number of reset packets sent by backend servers to clients per second. These reset packets are generated by the backend servers and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m21_client_rps	Reset Packets from Clients	The number of reset packets sent by clients to backend servers per second. These reset packets are generated by clients and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1f_lvs_rps	Reset Packets from Load Balancer	The number of reset packets generated by the load balancer per second. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
mb_l7_qps	Layer 7 Query Rate	The number of queries the monitored object receives per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
mc_l7_http_2xx	2xx Status Codes (Total)	The number of 2xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
md_l7_http_3xx	3xx Status Codes (Total)	The number of 3xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
me_l7_http_4xx	4xx Status Codes (Total)	The number of 4xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
mf_l7_http_5xx	5xx Status Codes (Total)	The number of 5xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m10_l7_http_other_status	Other Status Codes (Total)	The number of other status codes returned by the monitored object and backend servers per second at Layer 7. Excluded status codes: 2xx, 3xx, 404 Not Found, 499 Client Closed Request, and 502 Bad Gateway Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m11_l7_http_404	404 Not Found (Total)	The number of 404 Not Found status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m12_l7_http_499	499 Client Closed Request (Total)	The number of 499 Client Closed Request status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m13_l7_http_502	502 Bad Gateway (Total)	The number of 502 Bad Gateway status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m14_l7_rt	Average Layer 7 Response Time	Average response time of the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0 ms	ms	1000 (SI)	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m15_l7_upstream_4xx	4xx Status Codes (Backend Servers)	The number of 4xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m16_l7_upstream_5xx	5xx Status Codes (Backend Servers)	The number of 5xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Dedicated load balancer	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1c_l7_rt_max	Maximum Layer 7 Response Time	Maximum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Dedicated load balancer	1 minute
m1d_l7_rt_min	Minimum Layer 7 Response Time	Minimum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Dedicated load balancer	1 minute
l7_con_usage	Layer 7 Concurrent Connection Usage	The percentage of concurrent connections that have been established at Layer 7.	≥ 0	%	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l7_in_bps_usage	Layer 7 Inbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to receive requests from clients at Layer 7. **CAUTION** If the inbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the inbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	Dedicated load balancer	1 minute
l7_out_bps_usage	Layer 7 Outbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to return responses to clients at Layer 7. **CAUTION** If the outbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the outbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l7_ncps_usage	Layer 7 New Connection Usage	The percentage of new connections that have been established at Layer 7.	≥ 0	%	N/A	Dedicated load balancer	1 minute
l7_qps_usage	Layer 7 QPS Usage	The percentage of queries that have been made to the load balancer per second at Layer 7.	≥ 0	%	N/A	Dedicated load balancer	1 minute
l4_con_usage	Layer 4 Concurrent Connection Usage	The percentage of concurrent connections that have been established at Layer 4.	≥ 0	%	N/A	Dedicated load balancer	1 minute
l4_in_bps_usage	Layer 4 Inbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to receive requests from clients at Layer 4. **CAUTION** If the inbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the inbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l4_out_bps_usage	Layer 4 Outbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to return responses to clients at Layer 4. **CAUTION** If the outbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the outbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	Dedicated load balancer	1 minute
l4_ncps_usage	Layer 4 New Connection Usage	The percentage of new connections that have been established at Layer 4.	≥ 0	%	N/A	Dedicated load balancer	1 minute

Listener Metrics

Table 1-69 Metrics supported by each listener

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections established between the monitored object and backend servers. Load balancing at Layer 7: total number of TCP connections established between clients and the monitored object.	≥ 0	Count	N/A	Dedicated load balancer - listener	1 minute
m2_act_conn	Active Connections	The number of active TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m3_inact_conn	Inactive Connections	The number of inactive TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Dedicated load balancer - listener	1 minute
m4_ncps	New Connections	The number of new connections established between clients and the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m5_in_pps	Incoming Packets	The number of packets received by the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m6_out_pps	Outgoing Packets	The number of packets sent from the monitored object per second.	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m7_in_Bps	Inbound Traffic Rate	How fast the inbound traffic reaches the monitored object.	≥ 0	Byte/s	1000 (SI)	Dedicated load balancer - listener	1 minute
m8_out_Bps	Outbound Traffic Rate	How fast the outbound traffic leaves the monitored object.	≥ 0	Byte/s	1000 (SI)	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m9_abnormal_servers	Unhealthy Servers	The number of unhealthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer - listener	1 minute
ma_normal_servers	Healthy Servers	The number of healthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer - listener	1 minute
m22_in_bandwidth	Inbound Bandwidth	The bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer - listener	1 minute
m23_out_bandwidth	Outbound Bandwidth	The bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer - listener	1 minute
m1e_server_rps	Reset Packets from Backend Servers	The number of reset packets sent by backend servers to clients per second. These reset packets are generated by the backend servers and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m21_client_rps	Reset Packets from Clients	The number of reset packets sent by clients to backend servers per second. These reset packets are generated by clients and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m1f_lvs_rps	Reset Packets from Load Balancer	The number of reset packets generated by the load balancer per second. Supported protocols: TCP	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
mb_l7_qps	Layer 7 Query Rate	The number of queries the monitored object receives per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
mc_l7_http_2xx	2xx Status Codes (Total)	The number of 2xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
md_l7_http_3xx	3xx Status Codes (Total)	The number of 3xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
me_l7_http_4xx	4xx Status Codes (Total)	The number of 4xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
mf_l7_http_5xx	5xx Status Codes (Total)	The number of 5xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m10_l7_http_other_status	Other Status Codes (Total)	The number of other status codes returned by the monitored object and backend servers per second at Layer 7. Excluded status codes: 2xx, 3xx, 404 Not Found, 499 Client Closed Request, and 502 Bad Gateway Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m11_l7_http_404	404 Not Found (Total)	The number of 404 Not Found status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m12_l7_http_499	499 Client Closed Request (Total)	The number of 499 Client Closed Request status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m13_l7_http_502	502 Bad Gateway (Total)	The number of 502 Bad Gateway status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m14_l7_rt	Average Layer 7 Response Time	Average response time of the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute
m15_l7_upstream_4xx	4xx Status Codes (Backend Servers)	The number of 4xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute
m16_l7_upstream_5xx	5xx Status Codes (Backend Servers)	The number of 5xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of the backend server associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of the backend server associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1c_l7_rt_max	Maximum Layer 7 Response Time	Maximum response time of the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute
m1d_l7_rt_min	Minimum Layer 7 Response Time	Minimum response time of the monitored object at Layer 7. Supported protocols: HTTP/HTTPS The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients.	≥ 0	ms	1000 (SI)	Dedicated load balancer - listener	1 minute

Backend Server Group Metrics

Table 1-70 Metrics supported by each backend server group

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m9_abnormal_servers	Unhealthy Servers	The number of unhealthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer - backend server group	1 minute
ma_normal_servers	Healthy Servers	The number of healthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Dedicated load balancer - backend server group	1 minute
mb_l7_qps	Layer 7 Query Rate	The number of queries the monitored object receives per second at Layer 7. Supported protocols: HTTP/HTTPS/QUIC/GRPC	≥ 0	Count/s	N/A	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS /GRPC The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of the backend server associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS /GRPC The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server.	≥ 0	ms	1000 (SI)	Dedicated load balancer - backend server group	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of the backend server associated with the monitored object at Layer 7. Supported protocols: HTTP/HTTPS /GRPC The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server.	≥ 0	ms	1000 (SI)	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m15_l7_upstream_4xx	4xx Status Codes (Backend Servers)	The number of 4xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS /GRPC	≥ 0	Count/s	N/A	Dedicated load balancer - backend server group	1 minute
m16_l7_upstream_5xx	5xx Status Codes (Backend Servers)	The number of 5xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS /GRPC	≥ 0	Count/s	N/A	Dedicated load balancer - backend server group	1 minute
m25_l7_resp_Bps	Layer 7 Response Bandwidth	The bandwidth used by the backend servers associated with the monitored object to return responses to clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer - backend server group	1 minute
m24_l7_req_Bps	Layer 7 Request Bandwidth	The bandwidth used by the backend servers associated with the monitored object to receive requests from clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Dedicated load balancer - backend server group	1 minute

AZ Metrics

Table 1-71 AZ metrics

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections established between the monitored object and backend servers. Load balancing at Layer 7: total number of TCP connections established between the clients and the monitored object.	≥ 0	Count	N/A	AZ	1 minute
m2_act_conn	Active Connections	The number of active TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	AZ	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m3_inact_conn	Inactive Connections	The number of inactive TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	AZ	1 minute
m4_ncps	New Connections	The number of new connections established between clients and the monitored object per second.	≥ 0	Count/s	N/A	AZ	1 minute
m5_in_pps	Incoming Packets	The number of packets received by the monitored object per second.	≥ 0	Count/s	N/A	AZ	1 minute
m6_out_pps	Outgoing Packets	The number of packets sent from the monitored object per second.	≥ 0	Count/s	N/A	AZ	1 minute
m7_in_Bps	Inbound Traffic Rate	How fast the inbound traffic reaches the monitored object.	≥ 0	Byte/s	1000 (SI)	AZ	1 minute
m8_out_Bps	Outbound Traffic Rate	How fast the outbound traffic leaves the monitored object.	≥ 0	Byte/s	1000 (SI)	AZ	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m26_in_bandwidth_ipv6	IPv6 Inbound Bandwidth	The IPv6 network bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	AZ	1 minute
m27_out_bandwidth_ipv6	IPv6 Outbound Bandwidth	The IPv6 network bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	AZ	1 minute
m1e_server_rps	Reset Packets from Backend Servers	The number of reset packets sent by backend servers to clients per second. These reset packets are generated by the backend servers and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	AZ	1 minute
m21_client_rps	Reset Packets from Clients	The number of reset packets sent by clients to backend servers per second. These reset packets are generated by clients and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	AZ	1 minute
m1f_lvs_rps	Reset Packets from Load Balancer	The number of reset packets generated by the load balancer per second. Supported protocols: TCP	≥ 0	Count/s	N/A	AZ	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l4_con_usage	Layer 4 Concurrent Connection Usage	The percentage of concurrent connections that have been established at Layer 4.	≥ 0	%	N/A	AZ	1 minute
l4_in_bps_usage	Layer 4 Inbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to receive requests from clients at Layer 4. **CAUTION** If the inbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the inbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	AZ	1 minute
l4_out_bps_usage	Layer 4 Outbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to return responses to clients at Layer 4. **CAUTION** If the outbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the outbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	AZ	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l4_ncps_usage	Layer 4 New Connection Usage	The percentage of new connections that have been established at Layer 4.	≥ 0	%	N/A	AZ	1 minute
l7_in_bps_usage	Layer 7 Inbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to receive requests from clients at Layer 7. **CAUTION** If the inbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the inbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	AZ	1 minute
l7_out_bps_usage	Layer 7 Outbound Bandwidth Usage	The percentage of the bandwidth that the load balancer uses to return responses to clients at Layer 7. **CAUTION** If the outbound bandwidth usage reaches 100%, the load balancer performance has reached the upper limit. If the outbound bandwidth keeps higher than the bandwidth that the load balancer can provide, the service availability cannot be guaranteed.	≥ 0	%	N/A	AZ	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
l7_con_usage	Layer 7 Concurrent Connection Usage	The percentage of concurrent connections that have been established at Layer 7.	≥ 0	%	N/A	AZ	1 minute
l7_ncps_usage	Layer 7 New Connection Usage	The percentage of new connections that have been established at Layer 7.	≥ 0	%	N/A	AZ	1 minute

Dimensions

Key	Value
lbaas_instance_id	ID of a dedicated load balancer.
lbaas_listener_id	ID of a listener added to a dedicated load balancer.
lbaas_pool_id	ID of a backend server group.
available_zone	AZ where a dedicated load balancer works.

1.11.3 Event Monitoring

Overview

You can query system events that are automatically reported to Cloud Eye and custom events reported to Cloud Eye through the API. When there are specified events, you will receive alarm notifications.

Events are key operations on ELB resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific ELB resources.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by ELB to Cloud Eye.

Event monitoring is enabled by default and allows you to view monitoring details of system events and custom events. For operations supported by event monitoring, see **Monitoring Events Supported by ELB**.

Monitoring Events Supported by ELB

Table 1-72 lists the monitoring events supported by dedicated load balancers.

Table 1-72 Monitoring events supported by dedicated load balancers

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
ELB	The backend servers are unhealthy.	healthCheckUnhealthy	Major	Generally, this problem occurs because the backend servers are offline. This event will not be reported after it is reported for several times.	Check whether the backend servers are running properly.	ELB does not forward requests to unhealthy backend servers. If all backend servers in the backend server group are detected unhealthy, services will be interrupted.
	The backend server is detected healthy.	healthCheckRecovery	Minor	The backend server is detected healthy.	No further action is required.	The load balancer routes requests to this backend server.

1.11.4 Viewing Traffic Usage

Scenarios

For livestreaming platforms, traffic often increases suddenly, which makes the services unstable. To address this issue, most of them use ELB to distribute traffic. By working with Cloud Eye, ELB allows you to monitor the traffic usage in real time. You can view the traffic consumed by the EIPs bound to public network load balancers to better balance your application workloads.

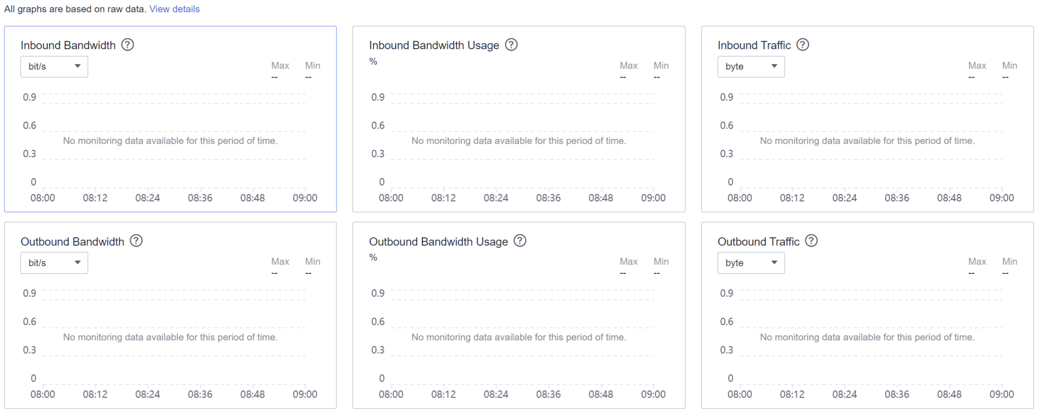
Prerequisites

- Load balancers are running properly.
- If the associated backend server is stopped, faulty, or deleted, its metrics cannot be viewed on Cloud Eye. After such a backend server restarts or recovers, its monitoring data will be displayed on the Cloud Eye console.

Viewing Traffic Usage of the Bound 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. Click ☰ in the upper left corner of the page and choose **Network** > **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Elastic IP and Bandwidth** > **EIPs**.
5. Locate the EIP bound to the load balancer and click its name. On the **Bandwidth** tab, you can view the data for the last 1, 3, 12 hours, last day, or last 7 days.

Figure 1-23 EIP traffic usage



All graphs are based on raw data. View details

No monitoring data available for this period of time.

Table 1-73 EIP and bandwidth metrics

Metric	Meaning	Value Range	Monitored Object	Monitoring Interval (Raw Data)
Outbound Bandwidth (originally named "Upstream Bandwidth")	Network rate of outbound traffic	≥ 0 bits/s	Bandwidth or EIP	1 minute
Inbound Bandwidth (originally named "Downstream Bandwidth")	Network rate of inbound traffic	≥ 0 bits/s	Bandwidth or EIP	1 minute
Outbound Bandwidth Usage	Usage of outbound bandwidth in percentage.	0–100%	Bandwidth or EIP	1 minute
Inbound Bandwidth Usage	Usage of inbound bandwidth in the unit of percent.	0–100%	Bandwidth or EIP	1 minute
Outbound Traffic (originally named "Upstream Traffic")	Network traffic going out of the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute
Inbound Traffic (originally named "Downstream Traffic")	Network traffic going into the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute

Viewing Load Balancer Traffic Metrics

1. Go to the **load balancer list page**.
2. On the load balancer list page, locate the load balancer and click its name.
3. Click the **Monitoring** tab, select load balancer for **Dimension**, and view the graphs of inbound and outbound rates.

   You can view data from the last 1, 3, 12 hours, last day, or the last 7 days.

1.12 CTS Auditing

1.12.1 Key Operations Recorded by CTS

You can use CTS to record operations on ELB for query, auditing, and backtracking.

Table 1-74 lists the operations recorded by CTS.

Table 1-74 ELB operations recorded by CTS

Action	Resource Type	Trace Name
Configuring access logs	logtank	createLogtank
Deleting access logs	logtank	deleteLogtank
Creating a certificate	certificate	createCertificate
Modifying a certificate	certificate	updateCertificate
Deleting a certificate	certificate	deleteCertificate
Creating a health check	healthmonitor	createHealthMonitor
Modifying a health check	healthmonitor	updateHealthMonitor
Deleting a health check	healthmonitor	deleteHealthMonitor
Adding a forwarding policy	l7policy	createL7policy
Modifying a forwarding policy	l7policy	updateL7policy
Deleting a forwarding policy	l7policy	deleteL7policy
Adding a forwarding rule	l7rule	createl7rule
Modifying a forwarding rule	l7rule	updateL7rule
Deleting a forwarding rule	l7rule	deleteL7rule

Action	Resource Type	Trace Name
Adding a listener	listener	createListener
Modifying a listener	listener	updateListener
Deleting a listener	listener	deleteListener
Creating a load balancer	loadbalancer	createLoadbalancer
Modifying a load balancer	loadbalancer	updateLoadbalancer
Deleting a load balancer	loadbalancer	deleteLoadbalancer
Adding a backend server	member	createMember
Modifying a backend server	member	updateMember
Removing a backend server	member	batchUpdateMember
Creating a backend server group	pool	createPool
Modifying a backend server group	pool	updatPool
Deleting a backend server group	pool	deletePool

1.12.2 Viewing Traces

Scenarios

After CTS is enabled, CTS starts recording operations on cloud resources. The CTS management console stores the last seven days of operation records.

This section describes how to query or export the last seven days of operation records on the management console.

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. Under **Management & Governance**, click **Cloud Trace Service**.
4. In the navigation pane on the left, choose **Trace List**.

5. Specify the filters used for querying traces. The following filters are available:
   - **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**

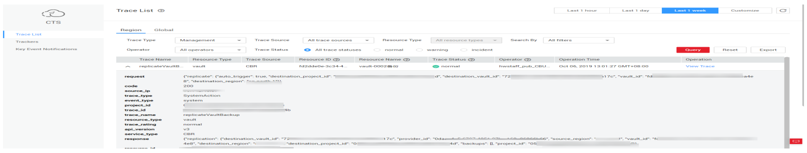
     Select a filter from the drop-down list.

     If you select **Trace name** for **Search By**, you need to select a specific trace name.

     If you select **Resource ID** for **Search By**, select or enter a specific resource ID.

     If you select **Resource name** for **Search By**, select or enter a specific resource name.
   - **Operator**: Select a specific operator (at the user level rather than the tenant level).
   - **Trace Status**: Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.
   - Time range: You can query traces generated at any time range of the last seven days.
6. Click ⌄ on the left of the required trace to expand its details.

Figure 1-24 Expanding trace details



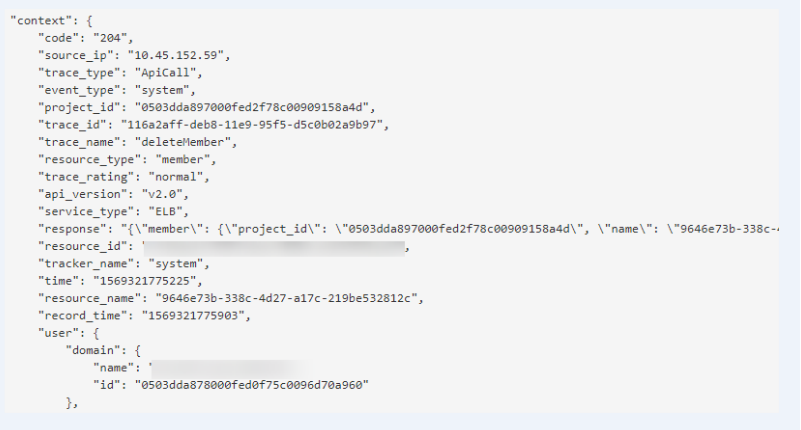
Trace Type: Management; Trace Source: All trace sources; Resource Type: All resource types; Search By: All filters
Operator: All operators; Trace Status: ● All trace statuses ○ normal ○ warning ○ incident

Trace Name	Resource Type	Trace Source	Resource ID	Resource Name	Trace Status	Operator	Operation Time	Operation
replicateVaultB...	vault	CBR	fd2dde0e-3c34-4...	vault-0002备份	normal	hwstaff_pub_CBU...	Oct 06, 2019 13:01:27 GMT+08:00	View Trace

```
request        {"replicate": {"auto_trigger": true, "destination_project_id": "[redacted]d", "destination_vault_id": "72[redacted]17c", "vault_id": "fd[redacted]a4e
               8", "destination_region": "[redacted]"}}
code           200
source_ip      [redacted]
trace_type     SystemAction
event_type     system
project_id     [redacted]
trace_id       [redacted]4b
trace_name     replicateVaultBackup
resource_type  vault
trace_rating   normal
api_version    v3
service_type   CBR
response       {"replication": {"destination_vault_id": "72[redacted]17c", "provider_id": "0da[redacted]", "source_region": "[redacted]", "vault_id": "f[redacted]
               4e8", "destination_region": "[redacted]", "destination_project_id": "0[redacted]4d", "backups": [], "project_id": "05[redacted]"}}
```

7. Click **View Trace** in the **Operation** column to view trace details.

Figure 1-25 View Trace



```
"context": {
    "code": "204",
    "source_ip": "10.45.152.59",
    "trace_type": "ApiCall",
    "event_type": "system",
    "project_id": "0503dda897000fed2f78c00909158a4d",
    "trace_id": "116a2aff-deb8-11e9-95f5-d5c0b02a9b97",
    "trace_name": "deleteMember",
    "resource_type": "member",
    "trace_rating": "normal",
    "api_version": "v2.0",
    "service_type": "ELB",
    "response": "{\"member\": {\"project_id\": \"0503dda897000fed2f78c00909158a4d\", \"name\": \"9646e73b-338c-
    "resource_id": '[redacted],
    "tracker_name": "system",
    "time": "1569321775225",
    "resource_name": "9646e73b-338c-4d27-a17c-219be532812c",
    "record_time": "1569321775903",
    "user": {
        "domain": {
            "name": '[redacted]
            "id": "0503dda878000fed0f75c0096d70a960"
    },
```

For details about key fields in the trace, see the **Cloud Trace Service User Guide**.

Example Traces

- Creating a load balancer

```
request {"loadbalancer":{"name":"elb-test-zcy","description":"","tenant_id":"05041fffa40025702f6dc009cc6f8f33","vip_subnet_id":"ed04fd93-e74b-4794-b63e-e72baa02a2da","admin_state_up":true}}
code 201
```

```
source_ip 124.71.93.36
trace_type ConsoleAction
event_type system
project_id 05041fffa40025702f6dc009cc6f8f33
trace_id b39b21a1-8d49-11ec-b548-2be046112888
trace_name createLoadbalancer
resource_type loadbalancer
trace_rating normal
api_version v2.0
service_type ELB
response {"loadbalancer": {"description": "", "provisioning_status": "ACTIVE", "provider": "vlb",
"project_id": "05041fffa40025702f6dc009cc6f8f33", "vip_address": "172.18.0.205", "pools": [],
"operating_status": "ONLINE", "name": "elb-test-zcy", "created_at": "2022-02-14T03:53:39",
"listeners": [], "id": "7ebe23cd-1d46-4a49-b707-1441c7f0d0d1", "vip_port_id":
"5b36ff96-3773-4736-83cf-38c54abedeea", "updated_at": "2022-02-14T03:53:41", "tags": [],
"admin_state_up": true, "vip_subnet_id": "ed04fd93-e74b-4794-b63e-e72baa02a2da", "tenant_id":
"05041fffa40025702f6dc009cc6f8f33"}}
resource_id 7ebe23cd-1d46-4a49-b707-1441c7f0d0d1
tracker_name system
time 2022/02/14 11:53:42 GMT+08:00
resource_name elb-test-zcy
record_time 2022/02/14 11:53:42 GMT+08:00
request_id
user {"domain": {"name": "CBUInfo", "id": "0503dda87802345ddafed096d70a960"}, "name": "zcy",
"id": "09f106afd2345cdeff5c009c58f5b4a"}
```

- Deleting a load balancer

```
request
code 204
source_ip 124.71.93.36
trace_type ConsoleAction
event_type system
project_id 05041fffa40025702f6dc009cc6f8f33
trace_id 4f838bbf-8d4a-11ec-a1fe-1f93fdaf3bec
trace_name deleteLoadbalancer
resource_type loadbalancer
trace_rating normal
api_version v2.0
service_type ELB
response {"loadbalancer": {"listeners": [], "vip_port_id": "5b36ff96-3773-4736-83cf-38c54abedeea",
"tags": [], "tenant_id": "05041fffa40025702f6dc009cc6f8f33", "admin_state_up": true, "id":
"7ebe23cd-1d46-4a49-b707-1441c7f0d0d1", "operating_status": "ONLINE", "description": "", "pools":
[], "vip_subnet_id": "ed04fd93-e74b-4794-b63e-e72baa02a2da", "project_id":
"05041fffa40025702f6dc009cc6f8f33", "provisioning_status": "ACTIVE", "name": "elb-test-zcy",
"created_at": "2022-02-14T03:53:39", "vip_address": "172.18.0.205", "updated_at":
"2022-02-14T03:53:41", "provider": "vlb"}}
resource_id 7ebe23cd-1d46-4a49-b707-1441c7f0d0d1
tracker_name system
time 2022/02/14 11:58:03 GMT+08:00
resource_name elb-test-zcy
record_time 2022/02/14 11:58:03 GMT+08:00
request_id
user {"domain": {"name": CBUInfo", "id": "0503dda87802345ddafed096d70a960"}, "name": "zcy", "id":
"09f106afd2345cdeff5c009c58f5b4a"}
```

2 User Guide for Shared Load Balancers

2.1 Permissions Management

2.1.1 Creating a User and Granting Permissions

Use **IAM** to implement fine-grained permissions control over your ELB resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing ELB resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust another account or cloud service to perform efficient O&M on your ELB resources.

Skip this section if your account does not need individual IAM users.

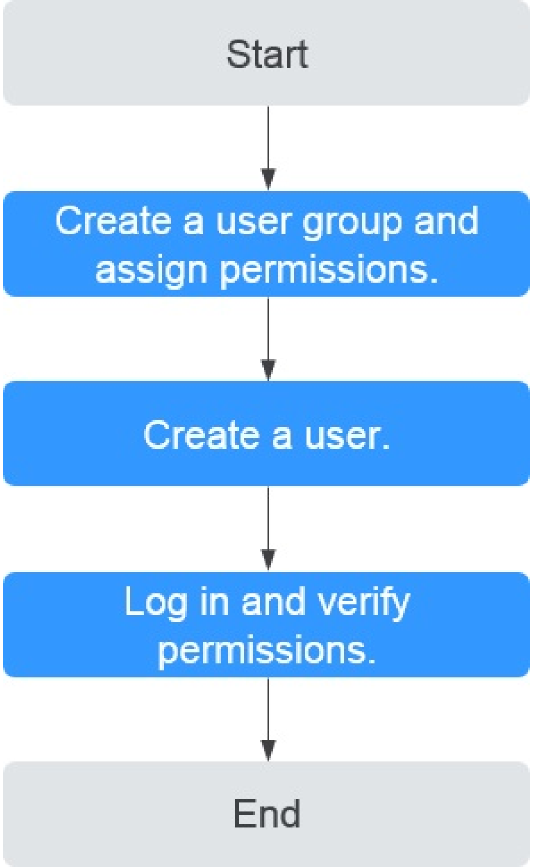
This following describes the procedure for granting permissions.

Prerequisites

You have learned about ELB policies and can select the appropriate policies based on service requirements. Learn about **permissions** supported by ELB. For the permissions of other services, see **System Permissions**.

Process Flow

Figure 2-1 Process for granting ELB permissions


Start
Create a user group and assign permissions.
Create a user.
Log in and verify permissions.
End

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the **ELB ReadOnlyAccess** policy to the group.

2. **Create a user and add it to a user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the ELB console by using the created user, and verify that the user only has read permissions for ELB.

   – Choose **Service List** > **Elastic Load Balance**. Then click **Buy Elastic Load Balancer** on the ELB console. If you cannot create a load balancer, the **ELB ReadOnlyAccess** policy has taken effect.

   – Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **ELB ReadOnlyAccess** policy has already taken effect.

2.1.2 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of ELB. For the actions supported for custom policies, see "Permissions Policies and Supported Actions" in the **Elastic Load Balance API Reference**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details, see **Creating a Custom Policy**. The following section contains examples of common ELB custom policies.

Example Custom Policies

- Example 1: Allowing users to update a load balancer

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elb:loadbalancers:put"
            ]
        }
    ]
}
```

- Example 2: Denying load balancer deletion

  A deny policy must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

  If you grant the system policy **ELB FullAccess** to a user but do not want the user to have the permission to delete load balancers defined in the policy, you can create a custom policy that rejects the deletion of load balancers and grant the **ELB FullAccess** and deny policies to the user, so that the user can perform all operations on ELB except deleting load balancers. The following is an example deny policy:

```
{
      "Version": "1.1",
      "Statement": [
            {
         "Effect": "Deny",
                  "Action": [
                        "elb:loadbalancers:delete"
                  ]
            }
      ]
}
```

- Example 3: Defining permissions for multiple services in a policy

  A custom policy can contain the actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

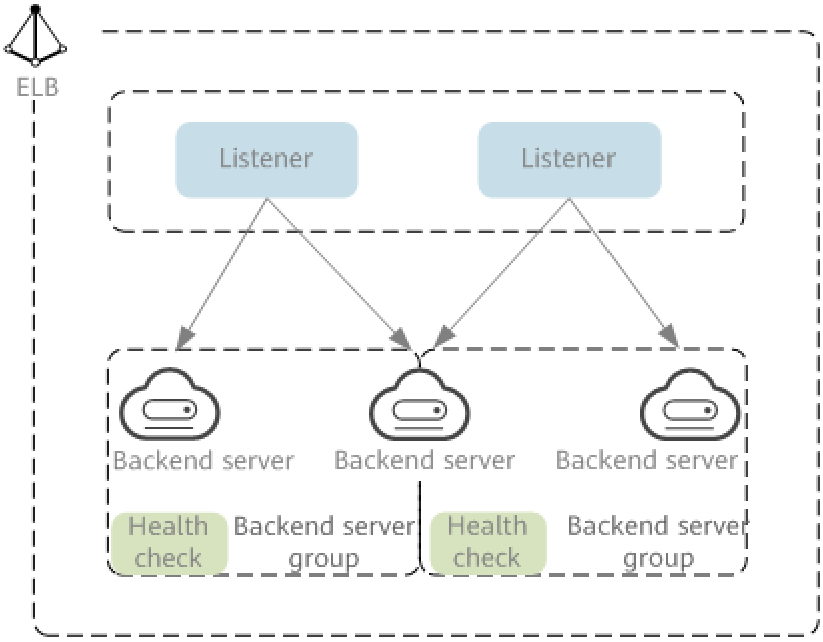
```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elb:loadbalancers:get",
                "elb:loadbalancers:list",
                "elb:loadbalancers:delete",
                "ecs:cloudServers:delete"
            ]
    }
  ]
}
```

2.2 Load Balancer

2.2.1 Shared Load Balancer 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 2-2 ELB components



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.
- Shared load balancers cannot distribute traffic across regions. When creating a load balancer, select the same region as the backend servers.

Network Type

Shared load balancers can work on both public and private networks.

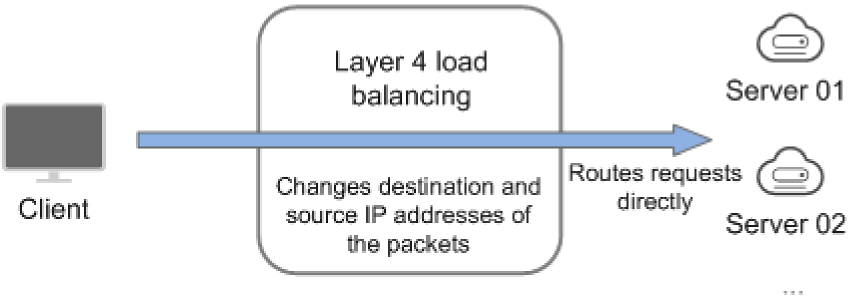
- To distribute requests over the Internet, you need to assign an EIP or bind an existing EIP to a load balancer so that it can route requests from the Internet to backend servers.
- If you want to distribute requests within a VPC, create a private network load balancer. This type of load balancers has only private IP addresses and can be only accessed within a VPC.

Protocol

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

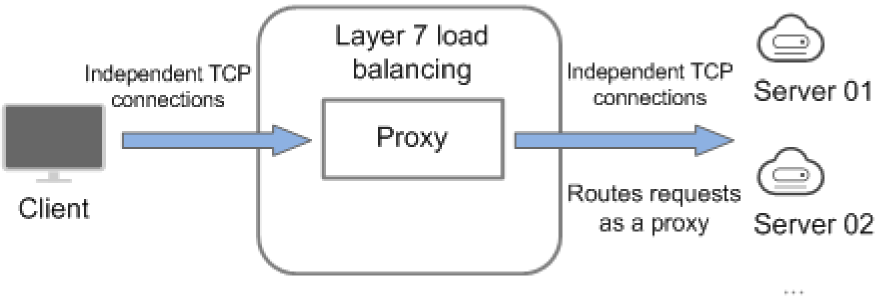
- If you choose TCP or UDP, the load balancer routes requests directly to backend servers. In this process, the destination IP address in a packet is changed to the IP address of the backend server, and the source IP address to the private IP address of the load balancer. A connection is established 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". Once the load balancer receives a request, it works as a proxy for backend servers and initiates a connection (three-way handshake) with the client. It then determines which backend server to route the request to based on the fields in the HTTP/HTTPS request header and the load balancing algorithm you select when you add the listener.

Figure 2-4 Layer-7 load balancing


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

NOTE

ELB establishes persistent connections between the clients and the load balancers to reduce the costs of a large number of short connections. After a persistent connection is established, the client can keep sending HTTP or HTTPS requests to the load balancer until the connection times out.

Backend Server

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 cloud servers, note the following:

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

2.2.2 Creating a Shared Load Balancer

Scenarios

You have prepared everything required for creating a shared load balancer. For details, see **Shared Load Balancer Overview**.

Constraints

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

Procedure

1. Go to the **Buy Elastic Load Balancer** page.
2. On the load balancer list page, click **Buy Elastic Load Balancer**.

   Complete the basic configurations based on **Table 2-1**.

Table 2-1 Parameters for configuring the basic information

Parameter	Description
Type	Specifies the type of the shared load balancer. The type cannot be changed after the load balancer is created. Shared load balancers are suitable for workloads with low traffic, such as small websites and common HA applications.
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.
Name	Specifies the load balancer name. The name can contain: ● 1 to 64 characters. ● Letters, digits, underscores (_), hyphens (-), and periods (.).
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed. For details about creating and managing enterprise projects, see the .

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

Table 2-2 Configuring network parameters

Parameter	Description
Network Type	**Private IPv4 network** is selected by default. The load balancer routes IPv4 requests from clients to backend servers in a VPC. If you want the load balancer to route requests from the Internet, bind an EIP to the load balancer.
VPC	Specifies the VPC where the shared load balancer works. You cannot change the VPC after the load balancer is created. Plan the VPC as required. Select an existing VPC or click **View VPCs** to create a desired one. For more information about VPC, see the ***Virtual Private Cloud User Guide***.
Frontend Subnet	Specifies the frontend subnet from which an IP address will be assigned to the shared load balancer to receive client requests. IP addresses in this subnet will be assigned to your load balancers.
IPv4 Address	Specifies how you want the IPv4 address to be assigned. ● **Automatically assign IP address**: The system assigns an IPv4 address to the load balancer. ● **Manually specify IP address**: You need to manually specify an IPv4 address for the load balancer. **NOTE** Network ACL rules configured for the frontend subnet of a load balancer do not restrict traffic from clients to the load balancer. Use access control to limit which IP addresses can access the load balancer. For details, see **What Is Access Control?**

4. Configure an EIP for the load balancer to enable it to route IPv4 requests over the Internet based on **Table 2-3**.

Table 2-3 Selecting an EIP for the load balancer

Parameter	Description
EIP	Specifies the public IP address that will be bound to the load balancer for receiving and forwarding requests over the Internet. ● **Auto assign**: A new EIP will be assigned to the load balancer. ● **Use existing**: Select an existing EIP. ● **Not required**: You can bind an EIP to the load balancer later.
EIP Type	Specifies the link type (BGP) when a new EIP is used.
Billed By	Specifies how the bandwidth will be billed. You can select one from the following options: ● **Bandwidth**: You specify the maximum bandwidth and pay for the amount of time you use the bandwidth. ● **Traffic**: You specify the maximum bandwidth and pay for the outbound traffic you use. ● **Shared Bandwidth**: Load balancers that have EIPs bound in the same region can share the selected bandwidth, helping you reduce public network bandwidth costs.
Bandwidth (Mbit/s)	Specifies the maximum bandwidth.

5. Configure other parameters for the load balancer as described in **Table 2-4**.

Table 2-4 Configuring other parameters

Parameter	Description
Advanced Settings (Optional) > Description	Click ⌄ to expand the configuration area and set this parameter. Enter a description about the load balancer in the text box as required. Enter up to 255 characters. Angle brackets (<>) are not allowed.

Parameter	Description
Advanced Settings (Optional) > Tag	Click ⌄ to expand the configuration area and set this parameter. Add tags to the load balancers so that they can be easily found. A tag consists of a tag key and a tag value. The tag key marks a tag, and the tag value specifies specific tag content. For details about the naming rules, see **Table 2-5**. You can add a maximum of 20 tags.

Table 2-5 Tag naming rules

Parameter	Rule
Tag key	● Cannot be empty. ● Must be unique for the same load balancer. ● Can contain a maximum of 36 characters. ● Can contain only letters, digits, underscores (_), hyphens (-), at signs (@).
Tag value	● Can contain a maximum of 43 characters. ● Can contain only letters, digits, underscores (_), hyphens (-), at signs (@) are allowed.

6. Click **Buy Now**.

2.2.3 Configuring Modification Protection for Shared Load Balancers

You can enable modification protection or deletion protection for load balancers to prevent them from being modified or deleted by accident.

Enabling or Disabling Deletion Protection

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Switch to the **Summary** tab of the load balancer and enable or disable **Deletion Protection**.

⚠ CAUTION

If your load balancer is managed by CCE, modifying the load balancer will affect the running of the CCE cluster.

4. After deletion protection is enabled, the load balancer cannot be deleted. Other operations are not affected.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Switch to the **Summary** tab and click **Configure** next to **Modification Protection**.
4. In the **Configure Modification Protection** dialog box, enable or disable **Modification Protection**.

   Fill in the reason if needed.

NOTE

You need to disable **Modification Protection** if you want to modify or delete a load balancer.

2.2.4 Changing the Network Configurations of a Shared Load Balancer

You can change the network configurations of a shared load balancer as needed.

Binding or Unbinding an IPv4 EIP

You can bind or unbind an IPv4 EIP to or from a shared load balancer as required.

NOTE

Load balancers without IPv4 EIPs cannot route requests over the public IPv4 network.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
   a. Binding an IPv4 EIP
      i. Click **Bind IPv4 EIP**.
      ii. In the **Bind IPv4 EIP** dialog box, select the EIP you want to bind to the load balancer and click **OK**.
   b. Unbinding an IPv4 EIP
      i. Click **Unbind IPv4 EIP**.
      ii. In the displayed dialog box, confirm the IPv4 EIP that you want to unbind and click **OK**.

Modifying the Bandwidth

If you set the **Network Type** of a load balancer to **Public IPv4 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. When you modify the bandwidth, traffic routing will not be interrupted.

NOTE

The EIP bandwidth defines the limit for clients to access the load balancer.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click **More** in the **Operation** column.
3. Click **Modify IPv4 Bandwidth**.
4. In the **New Configuration** area, modify the bandwidth and click **Next**.

   You can select the bandwidth defined by the system or customize a bandwidth. The bandwidth ranges from 1 Mbit/s to 2,000 Mbit/s.
5. Confirm the new bandwidth and click **Submit**.

   NOTE

   After you change the billing option and bandwidth, the price will be recalculated accordingly.

2.2.5 Exporting a Shared Load Balancer List

Scenarios

You can export the information of all or part of the load balancers in your account as an Excel file to a local directory.

You can export:

- The basic information of all or selected load balancers.
- The details of the selected load balancers.

Basic information includes the name, ID, status, type, and specifications of the load balancers.

Details include the basic information of load balancers and listeners by default. In addition, the forwarding policies, backend server groups, backend servers, and certificate names/IDs can also be exported.

Exporting the Basic Information of Load Balancers

1. Go to the **load balancer list page**.
2. In the upper left corner of the load balancer list, click **Export**.
   a. **Basic information of all resources**: The system automatically exports the basic information of all the load balancers in the current region as an Excel file to a local directory.
   b. **Basic information of selected resources**: The system automatically exports the basic information of the selected load balancers in the current region as an Excel file to a local directory.

2.2.6 Deleting a Shared Load Balancer

Scenarios

You can delete a load balancer if you no longer need it.

⚠ 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.

Deleting a Load Balancer

1. Go to the **load balancer list page**.
2. On the displayed page, locate the target load balancer and choose **More** > **Delete** in the **Operation** column.

   A confirmation dialog box is displayed.
3. In the displayed dialog box, enter **DELETE**.
4. Click **OK**.

2.2.7 Enabling or Disabling a Shared Load Balancer

You can enable or disable a shared load balancer at any time. The load balancer stops receiving and routing traffic after it is disabled.

If some load balancers are not required but cannot be deleted, you can disable them.

Procedure

You can enable or disable a load balancer at any time. The load balancer stops receiving and routing traffic after it is disabled.

1. Go to the **load balancer list page**.
2. Locate the load balancer and choose **More** > **Enable** or **More** > **Disable**.
3. Click **Yes**.
4. Check the status of the target load balancer in the **Status** column on the load balancer list page.

⚠ CAUTION

Disabled load balancers will still be billed.

2.3 Listener

2.3.1 Listener Overview

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

Supported Protocols

ELB provides load balancing at both Layer 4 and Layer 7. You can select TCP or UDP for load balancing at Layer 4 and HTTP or HTTPS for load balancing at Layer 7.

Table 2-6 Protocols supported by ELB

Protocol		Description	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	● Relatively low reliability ● Fast data transfer	Scenarios that require quick response, such as video chat, gaming, and real-time financial news
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
Layer 7	HTTPS	● An extension of HTTP for encrypted data transmission that can prevent unauthorized access ● Encryption and decryption performed on load balancers ● Multiple versions of encryption protocols and cipher suites	Web applications that require encrypted transmission

Frontend Protocols and Ports

Frontend protocols and ports are used by load balancers to receive requests from clients. Load balancers use TCP or UDP at Layer 4, and HTTP or HTTPS at Layer 7. Select a protocol and a port that best suit your requirements.

⚠ CAUTION

The frontend protocol and port cannot be changed once a listener is added. If you want to use a different protocol and port, add another listener.

Table 2-7 Frontend protocols and ports

Frontend Protocol	TCP, UDP, HTTP, and HTTPS
Frontend Port	Listeners using different protocols of a load balancer cannot use the same port. However, UDP listeners can use the same port as listeners that use other protocols. For example, if there is a UDP listener that uses port 88, you can add a TCP listener that also uses port 88. The port number ranges from 1 to 65535. The following are some commonly-used protocols and ports: TCP/80 HTTPS/443

Backend Protocols and Ports

Backend protocols and ports are used by backend servers to receive requests from load balancers. If Windows servers have Internet Information Services (IIS) installed, the default backend protocol and port are HTTP and 80.

Table 2-8 Backend protocols and ports

Backend Protocol	TCP, UDP, and HTTP
Backend Port	Backend servers of a load balancer can use the same port. The port number ranges from 1 to 65535. The following are some commonly-used protocols and ports: TCP/80 HTTP/443

2.3.2 Adding a TCP Listener

Scenarios

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

Constraints

If the front protocol is TCP, the backend protocol defaults to TCP and cannot be changed.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-9**.

Table 2-9 Parameters for configuring a TCP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **TCP**.
Listening 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.
Transfer Client IP Address	Specifies whether to allow the load balancer to communicate with backend servers using client IP addresses. For details, see **Transfer Client IP Address**.

Parameter	Description
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. Access control policies only take effect for new connections, but not for existing ones. 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, it may be caused by a persistent connection 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. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	
Idle Timeout (s)	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**.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-21**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-22**.

5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

2.3.3 Adding a UDP Listener

Scenarios

You can add a UDP listener, if quick response is required but low reliability is acceptable. UDP listeners are suitable for scenarios such as video chat, gaming, and real-time financial news.

Constraints

- UDP listeners do not support fragmentation.
- UDP listeners cannot use port 4789.
- Any UDP packet larger than 1,500 bytes will be discarded. To avoid this, ensure that the MTU value of the network interface is not greater than 1,500 bytes and modify the configuration files of applications based on the MTU value.
- If the listener protocol is UDP, the protocol of the backend server group is UDP by default and cannot be changed.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-10**.

Table 2-10 Parameters for configuring a UDP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **UDP**.
Listening 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.
Name (Optional)	Specifies the listener name.
Transfer Client IP Address	Specifies whether to allow the load balancer to communicate with backend servers using client IP addresses. For details, see **Transfer Client IP Address**.

Parameter	Description
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. Access control policies only take effect for new connections, but not for existing ones. 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, it may be caused by a persistent connection 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. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-21**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-22**.
5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

2.3.4 Adding an HTTP Listener

Scenarios

You can add an HTTP listener if content identification is required. HTTP is a great fit for workloads such as web applications and mobile mini-games.

Constraints

If the listener protocol is HTTP, the backend protocol is HTTP by default and cannot be changed.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-11**.

Table 2-11 Parameters for configuring an HTTP listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTP**.
Listening 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.
Redirect to another listener	Specifies the HTTPS listener to which HTTP requests are redirected to encrypt the communication and improve service security. For example, if you configure an HTTP redirection, HTTP requests to access a web page will be redirected to the HTTPS listener you select and handled by the backend servers associated with the HTTPS listener. As a result, the clients access the web page over HTTPS. Note that the configurations for the HTTP listener will not be applied. Requests will be forwarded to backend servers by the HTTPS listener.
Transfer Client IP Address	**Transfer Client IP Address** is enabled by default for HTTP listeners. When you use an HTTP listener to forward requests, you can use the X-Forwarded-For header to transfer client IP addresses. The first IP address recorded in the X-Forwarded-For header is the client IP address. For details, see **Transfer Client IP Address**.

Parameter	Description
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** **All IP addresses** is selected for access control by default. You can select **Whitelist** or **Blacklist** and choose an IP address group. ● **Whitelist**: Only IP addresses in the whitelist can access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will be forwarded by the listener. Access control policies only take effect for new connections, but not for existing ones. 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, it may be caused by a persistent connection 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. ● **Blacklist**: IP addresses in the blacklist are not allowed to access the listener. Requests from the IP addresses or CIDR blocks specified in the IP address group will not be forwarded by the listener.
More (Optional)	
Idle Timeout (s)	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**.
Request Timeout (s)	Specifies the length of time that a load balancer is willing to wait for a client request to complete. The load balancer terminates the connection if a request takes too long to complete. The request timeout duration ranges from **1** to **300**.

Parameter	Description
Response Timeout (s)	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.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-21**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-22**.
5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

2.3.5 Adding an HTTPS Listener

Scenarios

You can add an HTTPS listener if you require encrypted transmission. Load balancers decrypt HTTPS requests before routing them to backend servers. Once the servers process the requests, they send them back to the load balancers for encryption. Finally, the load balancers send the encrypted requests to the clients.

When you add an HTTPS listener, ensure that the subnet of the load balancer has sufficient IP addresses. If the IP addresses are insufficient, add more subnets on the summary page of the load balancer. After you select a subnet, do not configure network ACL rules for this subnet. If rules are configured, access to the load balancer may be denied.

Constraints

If the listener protocol is HTTPS, the protocol of the backend server group is HTTP by default and cannot be changed.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-12**.

Table 2-12 Parameters for configuring an HTTPS listener

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTPS**.
Listening 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.
Transfer Client IP Address	**Transfer Client IP Address** is enabled by default for HTTPS listeners. When you use an HTTPS listener to forward requests, you can use the X-Forwarded-For header to transfer client IP addresses. The first IP address recorded in the X-Forwarded-For header is the client IP address. For details, see **Transfer Client IP Address**.
Configure Certificate	
SSL Authentication	Specifies whether how you want the clients and backend servers to be authenticated. ● **One-way authentication**: Backend servers will be authenticated by clients. ● **Mutual authentication**: The clients and backend servers will authenticate each other.
Server Certificate	Specifies a server certificate that will be used to authenticate the server when HTTPS is used as the frontend protocol. Both the certificate and private key are required.

Parameter	Description
CA Certificate	Specifies the certificate that will be used to authenticate the client when **SSL Authentication** is set to **Mutual authentication** and the frontend protocol is HTTPS. CA certificates are also called client CA public key certificate. They are used to verify the issuer of a client certificate. HTTPS connections can only be established when the client provides a certificate issued by a specific CA.
SNI	Server Name Indication (SNI) is an extension to TLS. It allows clients to specify which domain name of a listener they are trying to connect in the first request. Once receiving the request, the load balancer searches for the certificate based on the domain name. The client includes the domain name in the initial SSL handshake. Once receiving the request, the load balancer searches for the certificate based on the domain name. If an SNI certificate is found, this certificate will be used for authentication. If no SNI certificates are found, the server certificate is used for authentication. For details, see **Using SNI Certificates for Access Through Multiple Domain Names**.
SNI Certificate	Specifies one or more certificates associated with the domain name when the frontend protocol is HTTPS and SNI is enabled. You can only select the server certificate with SNI domain names.
More (Optional)	
Security Policy	Specifies the security policy you can use if you select HTTPS as the frontend protocol. For more information, see **TLS Security Policy**.
HTTP/2	Specifies whether you want to use HTTP/2 if you select **HTTPS** for **Frontend Protocol**. For details, see **Enabling HTTP/2 for Faster Communication**.
Idle Timeout (s)	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**.

Parameter	Description
Request Timeout (s)	Specifies the length of time that a load balancer is willing to wait for a client request to complete. The load balancer terminates the connection if a request takes too long to complete. The request timeout duration ranges from **1** to **300**.
Response Timeout (s)	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.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.
HTTP Headers	Rewrite X-Forwarded-ELB-IP to transfer the load balancer EIP.

4. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also select **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-21**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure a health check for the backend server group.

         For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-22**.
5. Click **Next: Confirm**.
6. Confirm the configurations and click **Submit**.

2.3.6 Forwarding Policy

Scenarios

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

This is suitable when requests of services such as videos, images, audios, and texts need to be forwarded to different backend servers.

A forwarding policy consists of two parts: forwarding rule and action.

- A forwarding rule can be a domain name or a path.
- HTTP listeners can forward requests to a backend server group or redirect requests to another listener.
- HTTPS listeners can forward requests to a backend server group.

NOTE

Shared load balancers are available in CN-Hong Kong, AP-Singapore, CN East-Shanghai1, LA-Mexico City1, and LA-Sao Paulo1.

How Requests Are Matched

- After receiving a request, the load balancer attempts to find a matching forwarding policy based on the domain name or URL in the request:
  - If a match is found, the request is forwarded to the backend server group you select or create when you add the forwarding policy.
  - If no match is found, the request is forwarded to the default backend server group (that is specified when the listener is created).
  - If both a domain name and path are configured for a forwarding policy, the request can match the forwarding policy only when the domain name and path are both met.
- Matching priority:
  - When a request matches both a domain name-based policy and a path-based policy, the domain named-based policy is matched first. **Table 2-13** shows an example.
  - Forwarding policy priorities are independent of each other regardless of domain names.
  - Path-based forwarding rules are applied in the following order of priority: an exact match rule, a prefix match rule, and a regular expression match rule. For multiple matches of the same type, only the longest path rule will be applied.

Table 2-13 Example forwarding policies

Request	Forwarding Policy	Forwarding Rule	Specified Value
www.elb.com/test	1	Path	/test
	2	Domain name	www.elb.com

NOTE

In this example, although request **www.elb.com/test** matches both forwarding policies, it is routed based on forwarding policy 2 because domain named-based forwarding rules are applied first.

Notes and Constraints

- 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.
- When you add a forwarding policy, note the following:
  - The path in a forwarding rule cannot contain query strings. For example, if the path is set to **/path/resource?name=value**, the forwarding policy is invalid.
  - Each path must exist on the backend server. If the path does not exist, the backend server will return 404 Not Found.
  - In the regular expression match, the characters are matched sequentially, and the matching ends when any rule is matched. Matching rules cannot overlap with each other.
  - A path cannot be configured for two forwarding policies.
  - A domain name cannot exceed 100 characters.

⚠ CAUTION

If you add a forwarding policy that is the same as an existing one by calling an API, there will be a conflict. Even if you delete the existing forwarding policy, the new forwarding policy is still unavailable. Delete the newly-added forwarding policy and add a different one.

Adding a Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer you want to add forwarding policies for and click its name.
3. On the **Listeners** tab, add a forwarding policy in either of the following ways:
   - Locate the target listener and click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click the **Forwarding Policies** tab.
4. Click **Add Forwarding Policy**. Configure the parameters based on **Table 2-14**.
5. After the configuration is complete, click **Save**.

Table 2-14 Forwarding policy parameters

Parameter		Description	Example Value
Forwarding Rule	Domain name	Specifies the domain name that will be exactly matched against the domain names in requests. You need to specify either a domain name or path.	www.test.com
	Path	● Description Specifies the path used for forwarding requests. A path can contain letters, digits, and special characters: _~';@^-%#$.*+?,=!:|\/()[]{} ● Matching rules – Exact match: The request path is the same as the specified path and must start with a slash (/). – Prefix match: The request path starts with the specified path and must start with a slash (/). – Regular expression match: The paths are matched using a regular expression.	/login.php
Action	Forward to a backend server group	Specifies the backend server group to which a request is routed if it matches the configured forwarding rule.	Forward to a backend server group

Parameter		Description	Example Value
	Redirect to another listener	Specifies the HTTPS listener to which a request is routed if it matches the configured forwarding rule. 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. For example, if you configure a redirect for an HTTP listener, HTTP requests to access a web page will be redirected to the HTTPS listener you select and handled by the backend servers associated with the HTTPS listener. As a result, the clients access the web page over HTTPS. The configuration of the HTTP listener will become invalid.	N/A
Backend Server Group		Select a backend server group that will receive requests from the load balancer. This parameter is mandatory when you set **Action** to **Forward to a backend server group**.	N/A
Listener		Select an HTTPS listener that will receive requests redirected from the current HTTP listener. This parameter is mandatory when **Action** is set to **Redirect to another listener**.	N/A

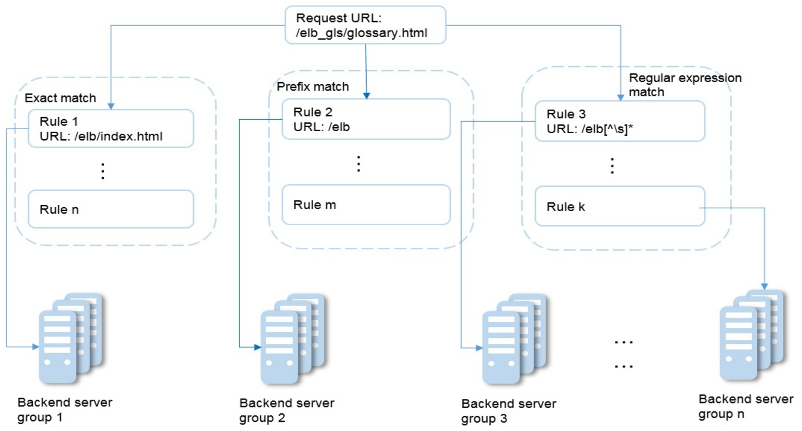
Path Matching Examples

The following table lists how a path is matched, and **Figure 2-5** shows how a request is forwarded to a backend server group.

Table 2-15 Path matching examples

URL Matching Rule	URL in the Request	Specified Path			
N/A	N/A	/elb/index.html	/elb	/elb[^\s]*	/index.html
Exact match	/elb/index.html	√	N/A	N/A	N/A
Prefix match		√	√	N/A	N/A
Regular expression match		√	N/A	√	N/A

Figure 2-5 Request forwarding



In this figure, the system first searches for an exact match of the request URL (/elb_gls/glossary.html). If there is no exact match, the system searches for a prefix match. If a match is found, the request is forwarded to backend server group 2 even if a regular expression match is also found, because the prefix match has a higher priority.

Modifying a Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer whose forwarding policies you want to modify and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Forwarding Policies** tab, select the forwarding policy, and click **Edit**.
5. Modify the parameters and click **Save**.

Deleting a Forwarding Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer whose forwarding policies you want to delete and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Forwarding Policies** tab, select the forwarding policy, and click **Delete** on the top right.
5. In the displayed dialog box, click **OK**.

2.3.7 Enabling HTTP/2 for Faster Communication

What Is HTTP/2?

Hypertext Transfer Protocol 2.0 (HTTP/2) uses a binary format for data transmission. It allows for much faster transmission and multiplexing. To reduce latency and improve efficiency, you can enable HTTP/2 when you add HTTPS listeners.

Constraints

You can enable HTTP/2 only for HTTPS listeners.

Managing HTTP/2

You can enable HTTP/2 when you add an HTTPS listener. You can enable or disable HTTP/2 for an existing HTTPS listener.

Enabling HTTP/2 When Adding a Listener

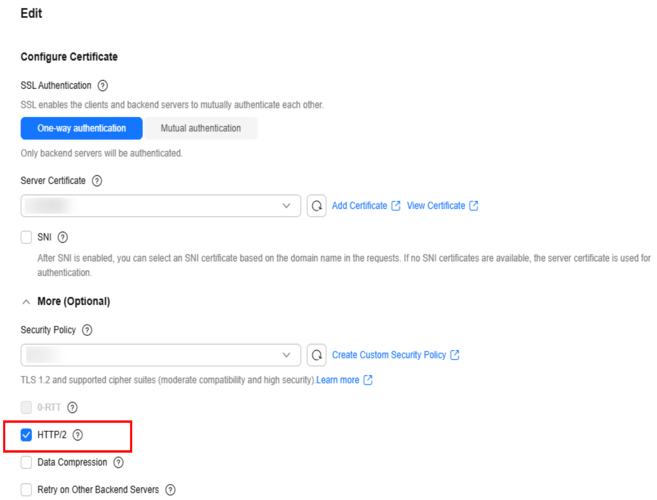
To enable HTTP/2 when adding an HTTPS listener, perform the following operations:

1. Go to the **load balancer list page**.
2. Locate the load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**.
4. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.
5. Expand **Advanced Settings (Optional)** and enable HTTP/2.
6. Confirm the configurations and go to the next step.

Enabling or Disabling HTTP/2 for an Existing Listener

1. Go to the **load balancer list page**.
2. Locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Summary** tab, click **Edit** on the top right.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)** and enable or disable HTTP/2.
6. Click **OK**.

Figure 2-6 Disabling or enabling HTTP/2



Edit

Configure Certificate

SSL Authentication ?

SSL enables the clients and backend servers to mutually authenticate each other.

One-way authentication | Mutual authentication

Only backend servers will be authenticated.

Server Certificate ?

Add Certificate View Certificate

□ SNI ?

After SNI is enabled, you can select an SNI certificate based on the domain name in the requests. If no SNI certificates are available, the server certificate is used for authentication.

More (Optional)

Security Policy ?

Create Custom Security Policy

TLS 1.2 and supported cipher suites (moderate compatibility and high security).Learn more

□ 0-RTT ?

☑ HTTP/2 ?

□ Data Compression ?

□ Retry on Other Backend Servers ?

2.3.8 Managing a Listener

Scenarios

You can configure modification protection for a listener, modify the settings of a listener, and change the backend server group of a listener as needed.

Prerequisites

- You have created a backend server group by referring to **Creating a Backend Server Group**.
- You have added a listener by referring to **Listener Overview**.

Configuring Modification Protection for a Listener

You can enable modification protection for a listener to prevent it from being modified or deleted by accident.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click the **Listeners** tab, locate the listener, and click its name.
4. On the **Summary** tab, click **Configure** next to **Modification Protection**.
5. In the **Configure Modification Protection** dialog box, enable **Modification Protection**.

NOTE

You need to disable **Modification Protection** if you want to modify or delete a listener.

Modifying Listener Settings

NOTE

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

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Modify the listener in either of the following ways:
   - On the **Listeners** tab, locate the listener, and click **Edit** in the **Operation** column.
   - Click the name of the target listener. On the **Summary** tab, click **Edit** on the top right corner.
4. On the **Edit** page, modify parameters, and click **OK**.

Modifying Timeout Durations

You can modify timeout durations (idle timeout, request timeout, and response timeout) for your listeners to meet varied demands. For example, if the size of a request from an HTTP or HTTPS client is large, you can prolong the request timeout duration to ensure that the request can be successfully routed.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click the name of the listener.
4. On the **Summary** tab, click **Edit** on the top right.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)**.
6. Configure **Idle Timeout (s)**, **Request Timeout (s)**, or **Response Timeout (s)** as you need.
7. Click **OK**.

Changing the Backend Server Group of a Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, locate the target listener and click its name.
4. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
5. 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.
6. Click **OK**.

2.3.9 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.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
   a. Deleting a listener:
      i. On the **Listeners** tab, locate the listener, and click **Delete** in the **Operation** column.
      ii. In the displayed dialog box, enter **DELETE**.
   b. Batch deleting listeners:
      i. On the **Listeners** tab, select multiple listeners you want to delete.
      ii. Click **Delete** above the listener list.
      iii. In the displayed dialog box, enter **DELETE**.
3. Click **OK**.

2.4 Backend Server Group

2.4.1 Backend Server Group Overview

What Is a Backend Server Group?

A backend server group is a logical collection of one or more backend servers to receive massive concurrent requests at the same time. Only cloud servers can be added as backend servers.

The following table describes how a backend server group forwards traffic.

Table 2-16 Traffic distribution process

Step 1	A client sends a request to your application. The listeners added to your load balancer use the protocols and ports you have configured to forward the request to the associated backend server group.
Step 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.
Step 3	In this way, massive concurrent requests can be processed at the same time, improving the availability of your applications.

Shared load balancers have only one type of backend server group, where you can only add cloud servers.

Table 2-17 Adding backend servers

Backend Server Type	Description	Reference
Cloud servers	You can add ECSs that are in the same VPC as the load balancer.	**Cloud Servers**

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**: The **health check** function ensures traffic is routed only to healthy backend servers in the backend server group.

Controlling Traffic Distribution

You can configure the key functions listed in **Table 2-18** for each backend server group to ensure service stability.

Table 2-18 Key functions

Key Function	Description	Detail
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.	**Enabling Sticky Session to Accelerate Access**

Backend Server Group and Listener Protocols

A backend server group can be associated with only one shared load balancer and used by only one listener.

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 2-19**.

Table 2-19 The frontend and backend protocol

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

2.4.2 Creating a Backend Server Group

Scenario

To route requests, you need to associate a backend server group to each listener.

You can create a backend server group in the ways listed in **Table 2-20**.

Table 2-20 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 **Listener 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 server group of a shared load balancer can be associated with only one listener.

Procedure

1. Go to the **backend server group list page**.
2. Click **Create Backend Server Group** in the upper right corner.
3. Configure the routing policy based on **Table 2-21**.

Table 2-21 Parameters required for configuring a routing policy

Parameter	Description
Type	Specifies the type of load balancer that can use the backend server group. Select **Shared**.
Load Balancer	Specifies whether to associate a load balancer.
Backend Server Group Name	Specifies the name of the backend server group.
Backend Protocol	Specifies the protocol that backend servers in the backend server group use to receive requests from the listeners. The options are HTTP, TCP, and UDP.
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**.
Sticky Session	Specifies whether to enable sticky sessions. 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 **Enabling Sticky Session to Accelerate Access**.

Parameter	Description
Sticky Session Type	Specifies the type of sticky sessions. After the sticky session is enabled, you need to select a sticky session 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 enables requests from different clients to be routed and ensures that a client is directed to the same server that it was using previously. ● **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. ● **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 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** and **Application cookie** are available when you have selected **HTTP** or **HTTPS** for **Backend Protocol**.
Stickiness Duration (min)	Specifies the time that sticky sessions are maintained, in minutes. ● Sticky sessions at Layer 4: **1** to **60** ● Sticky sessions at Layer 7: **1** to **1440**
Description	Provides supplementary information about the backend server group.

4. Click **Next** to add backend servers and configure health check based on **Table 2-22**. For more information about health checks, see **Health Check**.

Table 2-22 Parameters required for configuring a health check

Parameter	Description
Health Check	Specifies whether to enable the health check option. If the health check option is enabled, click ✎ next to **Advanced Settings (Optional)** to set health check parameters.

Parameter	Description
Health Check Protocol	- The health check protocol can be TCP or HTTP. - If the protocol of the backend server group is UDP, the health check protocol is UDP by default.
Domain Name	Specifies the domain name that will be used for health checks. This parameter is mandatory if the health check protocol is HTTP. By default, the private IP address of each backend server is used. 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 labels with a hyphen. Max total: 100 characters. Max label: 63 characters.
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.
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. 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 (?), number signs (#), percent signs (%), ampersands (&).
Interval (s)	Specifies the maximum time between two consecutive health checks, in seconds. The interval ranges from **1** to **50**.
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The value ranges from **1** to **50**.
Healthy Threshold	Specifies the number of consecutive successful health checks required for declaring a backend server healthy. The value ranges from **1** to **10**.
Unhealthy Threshold	Specifies the number of consecutive failed health checks required for declaring a backend server unhealthy. The value ranges from **1** to **10**.

5. Click **Next**.
6. Confirm the specifications and click **Create Now**.

2.4.3 Controlling Traffic Distribution

2.4.3.1 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.

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

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

Table 2-23 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	Consistent hashing: 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.

How Load Balancing Algorithms Work

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

Weighted Round Robin

Figure 2-7 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 2-7 Traffic distribution using the weighted round robin algorithm

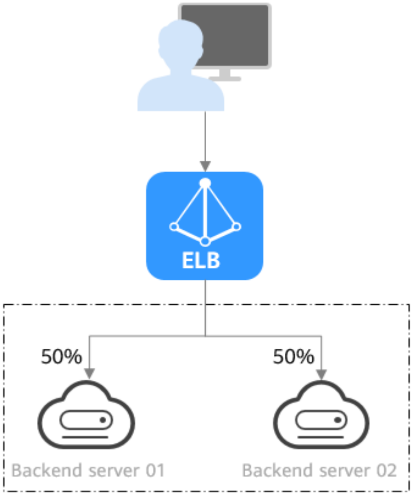


Table 2-24 Weighted round robin

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

Weighted Least Connections

Figure 2-8 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 2-8 Traffic distribution using the weighted least connections algorithm

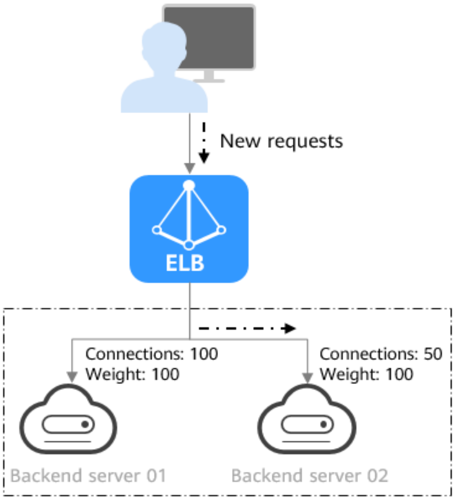


Table 2-25 Weighted least connections

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 many loads on new servers: If existing backend servers have to handle a large number of requests, new requests will be routed to new backend servers. This may deteriorate new servers or even cause them to fail.

Source IP Hash

Figure 2-9 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 2-9 Traffic distribution using the source IP hash algorithm

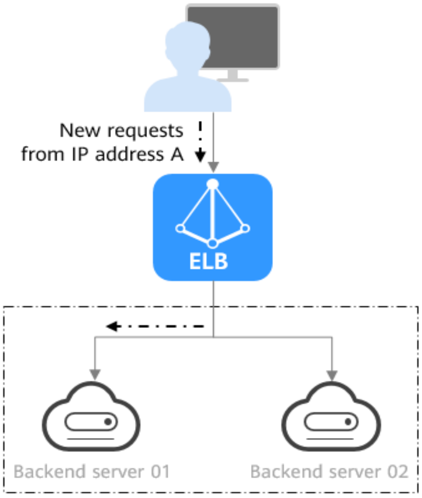


Table 2-26 Source IP hash

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.

Changing a Load Balancing Algorithm

1. Go to the **backend server group list page**.
2. On the backend server group list page, locate the target backend server group and click **Edit** in the **Operation** column.
3. In the **Modify Backend Server Group** dialog box, change the load balancing algorithm.
4. Click **OK**.

NOTE

The change is applied 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.

2.4.3.2 Enabling Sticky Session to Accelerate Access

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 2-27 Sticky session comparison

OSI Layer	Listener Protocol	Sticky Session Type	Stickiness Duration	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 to be forwarded to the same backend server.	● Default: 20 minutes ● Maximum: 60 minutes ● Range: 1 minute to 60 minutes	● Source IP addresses of the clients change. ● The session stickiness duration has been reached.

OSI Layer	Listener Protocol	Sticky Session Type	Stickiness Duration	Scenarios Where Sticky Sessions Become Invalid
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. • **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 cookie are routed to the same backend server.	• Default: 20 minutes • Maximum: 1,440 minutes • Range: 1 minute to 1,440 minutes	• 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**.

Notes and Constraints

- If you use **Direct Connect** or **VPN** to access ELB, you must select **Source IP hash** as the load balancing algorithm and disable sticky session.
- Shared load balancers support three types of sticky session: **Source IP address**, **Load balancer cookie**, and **Application 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.

Enabling or Disabling Sticky Session

1. Go to the **backend server group list page**.
2. On the backend server group list page, locate the backend server group and click **Edit** in the **Operation** column.
3. In the **Modify Backend Server Group** dialog box, enable or disable **Sticky Session**.

   If you enable it, select the sticky session type, and set the session stickiness duration.
4. Click **OK**.

2.4.4 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

- 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 2-19**.
- You can only associate a backend server group that is not used by any listener with a shared load balancer.

Procedure

1. Go to the **load balancer list page**.
2. On the displayed page, locate the target load balancer and click its name.
3. On the **Listeners** tab, locate the target listener and click its name.
4. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
5. 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.

6. Click **OK**.

2.4.5 Managing a Backend Server Group

You can manage a backend server group as required.

Enabling Modification Protection

You can enable the modification protection for a backend server group to prevent the backend servers in it from being modified or deleted by accident.

Enabling modification protection for a backend server group will prohibit any change to both the group and the backend servers in it.

1. Go to the **backend server group list page**.
2. On the backend server group list page, locate the backend server group and click its name.
3. On the **Summary** tab, click **Configure** next to **Modification Protection**.
4. In the **Configure Modification Protection** dialog box, enable **Modification Protection**.
5. Click **OK**.

NOTE

Disable **Modification Protection** if you want to delete a backend server group or modify its settings.

Enabling Removal Protection for a Backend Server Group

You can enable removal protection for a backend server group to prevent the backend servers in it from being removed by accident.

After removal protection is enabled for a backend server group, you cannot remove backend servers from it.

CAUTION

If your load balancer is managed by CCE, enabling removal protection for a backend server group may affect the normal running of the cluster.

1. Go to the **backend server group list page**.
2. On the displayed page, locate the backend server group and click its name.
3. On the **Summary** tab, enable **Removal Protection**.

NOTE

Disable **Removal Protection** if you want to remove servers from the backend server group.

Viewing a Backend Server Group

You can view the details of a backend server group.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the backend server group.
3. Click different tabs to view the required information.
    a. On the **Summary** tab, view the basic information (name, ID, backend protocol) and health check settings.
    b. On the **Backend Servers** tab, view the servers that have been added to the backend server group.

Deleting a Backend Server Group

Before deleting 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.

1. Go to the **backend server group list page**.
2. On the backend server group list page, locate the backend server group and click **Delete** in the **Operation** column.
3. In the displayed dialog box, click **OK**.

2.5 Backend Server

2.5.1 Backend Server 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 eliminating 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.

You can only add servers in the same VPC as the load balancer. For details, see **Cloud Servers**.

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.

Notes and Constraints

- 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 and Network ACL Rules**.

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.

The weight ranges from **0** to **100**. If you set the weight of a cloud 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 described in **Table 2-28**. For more information about load balancing algorithms, see **Load Balancing Algorithms**.

Table 2-28 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.

Load Balancing Algorithm	Weight Setting
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.

2.5.2 Security Group and Network ACL Rules

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

When backend servers receive requests from the load balancer, source IP addresses are translated into those in 100.125.0.0/16.

- Security group rules of backend servers must allow traffic from 100.125.0.0/16 to backend servers. 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 subnet where the backend servers are deployed, the rules must allow traffic from the backend subnet of the load balancer to the subnet of the backend servers. For details about how to configure network ACL rules, see **Configuring Network ACL Rules**.

NOTE

If **Transfer Client IP Address** is enabled for Layer 4 listeners, network ACL and security group 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 **What Is Access Control?**

Constraints

- If health check is enabled for a backend server group, security group rules must allow traffic over the health check port and protocol.
- If UDP is used for health check, there must be a rule that allows ICMP traffic. If there is no such rule, the health of the backend servers cannot be checked.

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 instances in the security group to communicate with each other but block access from external networks. To ensure that the load balancer can communicate with associated backend servers over both the frontend and health check ports, configure inbound rules for the security group containing these servers.

1. Log in to the management console.
2. In the ECS list, click the name of the target ECS.

   The page providing the details about the ECS is displayed.

3. Click the **Security Groups** tab, locate the security group, click its name, and view security group rules.
4. On the **Inbound Rules** tab, click **Add Rule**. Configure inbound rules based on **Table 2-29**.

Table 2-29 Security group rules

Backend Protocol	Action	Protocol & Port	Source IP Address
HTTP	Allow	**Protocol**: TCP **Port**: the port used by the backend server and health check port	100.125.0.0/16
TCP	Allow	**Protocol**: TCP **Port**: health check port	100.125.0.0/16
UDP	Allow	**Protocol**: UDP and ICMP **Port**: health check port	100.125.0.0/16

5. 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. 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 access from 100.125.0.0/16.

ELB translates the public IP addresses into private IP addresses in 100.125.0.0/16 before forwarding traffic to backend servers. So public IP addresses cannot be configured as the source for a network ACL rule to prevent public IP addresses from accessing backend servers.

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 **What Is 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. Click ☰ in the upper left corner to display **Service List** and choose **Networking** > **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Access Control** > **Network ACLs**.
5. In the network ACL list, locate the target network ACL and click its name.
6. On the **Inbound Rules** or **Outbound Rules** tab, click **Add Rule** to add an inbound or outbound rule.
   - **Action**: Select **Allow**.
   - **Protocol**: The protocol must be the same as the backend protocol.
   - **Source**: Set it to **100.125.0.0/16**.
   - **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 to all IP addresses is permitted.
   - **Destination Port Range**: Select a port range.
   - (Optional) **Description**: Describe the network ACL rule if necessary.
7. Click **OK**.

2.5.3 Cloud Servers

When you use ELB to route requests, ensure that at least one backend server is running properly and can receive requests routed by the load balancer.

After a backend server is unbound from a load balancer, the backend server does not receive requests forwarded by the load balancer, but the backend server is disassociated from the load balancer. You can add the backend server to the backend server group again when traffic increases or the reliability needs to be enhanced.

Constraints

- Only servers in the same VPC as the load balancer can be added.

Adding a Cloud Server

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the backend server group.
3. Switch to the **Backend Servers** tab and click **Add** on the right of the **Cloud Servers** area.
4. Search for backend servers using specified keywords.
5. Specify the weights and ports for the backend servers, and click **Finish**.

   Backend server ports can be set in batches.

Modifying Cloud Server Ports/Weights

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. On the **Backend Servers** tab, click **Cloud Servers**.
4. Select the cloud servers and click **Modify Weight** up above the cloud server list.
5. In the displayed dialog box, modify the weights as you need.
   - Changing the weight of a single cloud server: Set the weight in the **Weight** column.
   - Modifying the weights of multiple cloud servers: Select the target cloud servers and set the weight next to **Batch Modify Weights** and click **OK**.

   NOTE

   You can set the weights of multiple cloud servers to **0** to block them from receiving requests routed by each load balancer.
6. Click **OK**.

Removing a Cloud Server

NOTE

If a cloud 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.

1. Go to the **backend server group list page**.
2. On the backend server group list page, click the name of the target backend server group.
3. Switch to the **Backend Servers** tab and click **Cloud Servers**.
4. Select the cloud servers you want to remove and click **Remove** above the cloud server list.
5. In the displayed dialog box, click **OK**.

2.6 Health Check

2.6.1 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 routing 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 **Enabling or Disabling Health Check**.

Select a health check protocol that matches the backend protocol as described in **Table 2-30**.

Table 2-30 Backend and health check protocols (shared load balancers)

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

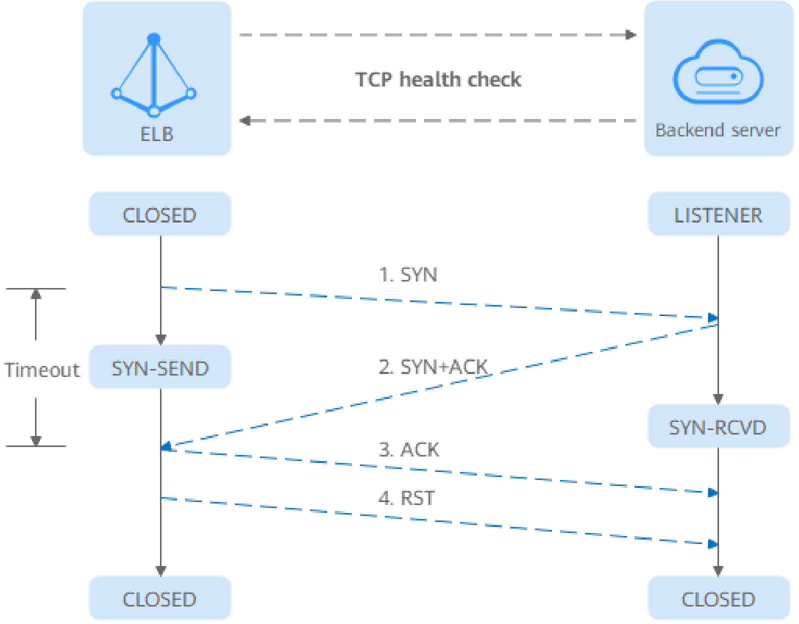
Health Check Source IP Address

A shared load balancer uses an IP address in 100.125.0.0/16 to send requests to backend servers and check their health. To perform health checks, ensure that the security group rules of the backend server allow access from 100.125.0.0/16. For details, see **Security Group and Network ACL Rules**.

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 2-10 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.

⚠ CAUTION

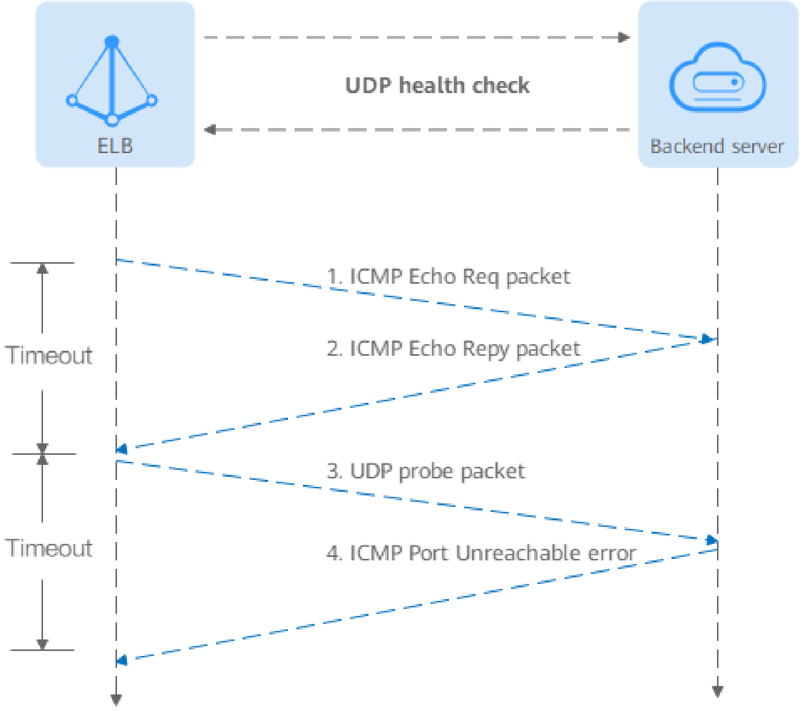
After a successful TCP three-way handshake, an RST packet will be sent to close the TCP connection. The application on the backend server may consider this packet a connection error and reply with a message, for example, "Connection reset by peer". To avoid this issue, take either of the following actions:

- 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 2-11 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.

⚠ CAUTION

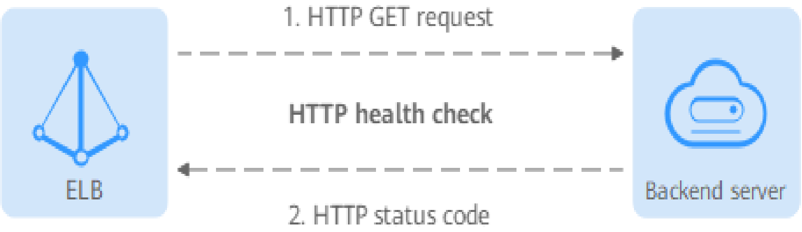
- If there is a large number of concurrent requests, the health check result may be different from the actual health of the backend server.

  If the backend server runs Linux, it may limit the rate of ICMP packets as a defense against ping flood attacks. In this case, even if there is a service exception, ELB will not receive the error message "port XX unreachable", and the server will still be determined healthy. As a result, the health check result is different from the actual health of the backend server.
- The UDP probe packet's payload has no significance and is simply used to fill the packet with data. Typically, the payload is set to "H". Clients should not attempt to interpret its content.

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 2-12** shows how an HTTP health check works.

Figure 2-12 HTTP health check



The HTTP health check process is as follows:

1. The load balancer sends an 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.)
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.

⚠ CAUTION

In an HTTP health check, the User-Agent header identifies that the requests are sent for health checks. The value of User-Agent may be adjusted based on service requirements. So it is not recommended to rely on this header for verification or judgment.

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 2-31**.

Table 2-31 Factors affecting the health check time window

Factor	Description
Check Interval	How often health checks are performed.

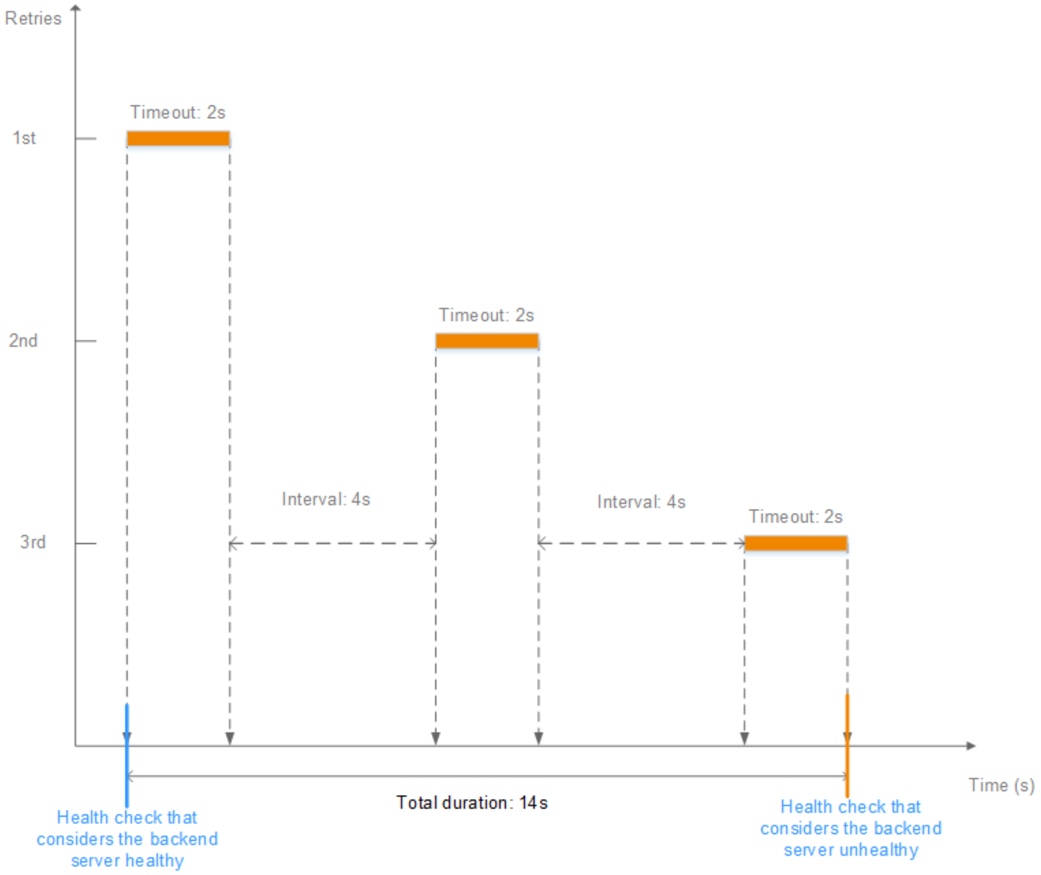
Factor	Description
Timeout Duration	How long the load balancer waits for the response from the backend server.
Health Check Threshold	The number of consecutive successful or failed health checks required for determining whether the backend server is healthy or unhealthy.

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

As shown in **Figure 2-13**, 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 × 3 + 4 × (3 – 1) = 14s.

Figure 2-13 Health check time window


Retries
1st
2nd
3rd
Timeout: 2s
Timeout: 2s
Timeout: 2s
Interval: 4s
Interval: 4s
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?**

2.6.2 Enabling or Disabling Health Check

Scenarios

This section describes how you can enable or disable the health check option.

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

- 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. For details, see **Security Group and Network ACL 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. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, locate the backend server group and click its name.
3. On the **Summary** page, click **Health Check** on the right.
4. In the **Configure Health Check** dialog box, configure the parameters based on **Table 2-32**.

Table 2-32 Parameters required for configuring health check

Parameter	Description
Health Check	Specifies whether to enable the health check option. **NOTE** When the health check is enabled or disabled, the number of healthy or unhealthy backend servers may temporarily fluctuate but will stabilize after a monitoring period.
Health Check Protocol	Specifies the protocol that will be used by the load balancer to check the health of backend servers. If the protocol of the backend server group is UDP, the health check protocol is UDP by default. Shared load balancers support TCP and 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. ● 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.
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.
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. The path can contain 1 to 80 characters and must start with a slash (/). If the backend server group is associated with a shared load balancer, the path can contain letters, digits, hyphens (-), slashes (/), periods (.), question marks (?), percent signs (%), ampersands (&), and underscores (_).
Interval (s)	Specifies the maximum time between two consecutive health checks, in seconds. The interval ranges from **1** to **50**.
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check, in seconds. The value ranges from **1** to **50**.

Parameter	Description
Healthy Threshold	Specifies the number of consecutive successful health checks required for declaring a backend server healthy. The value ranges from **1** to **10**.
Unhealthy Threshold	Specifies the number of consecutive failed health checks required for declaring a backend server unhealthy. The value ranges from **1** to **10**.

5. Click **OK**.

Disabling Health Check

1. Go to the **backend server group list page**.
2. On the **Backend Server Groups** page, click the name of the target backend server group.
3. On the **Summary** page, click **Health Check** on the right.
4. In the **Configure Health Check** dialog box, disable health check.
5. Click **OK**.

2.7 Security

2.7.1 Transfer Client IP Address

Scenarios

Generally, shared load balancers use IP addresses in 100.125.0.0/16 to communicate with backend servers. If you want a load balancer to communicate with backend servers using real IP addresses of the clients, you can enable **Transfer Client IP Address** to pass the IP addresses of the clients to backend servers.

Table 2-33 lists whether you can enable or disable this feature.

Table 2-33 Transfer client IP address support

Listener Type	Enabling Transfer Client IP Address	Disabling Transfer Client IP Address
TCP and UDP	Supported	Supported
HTTP and HTTPS	Enabled by default	Not supported

Notes and Constraints

- When you enable or disable **Transfer Client IP Address**, if the listener has backend servers associated, traffic to this listener will be interrupted for about

10 seconds. The interruption duration is twice the health check interval configured for the backend server group.

- If **Transfer Client IP Address** is enabled, a server cannot serve as both a backend server and a client. This is because backend server will think the packet from the client is sent by itself and will not return a response packet to the load balancer. As a result, the return traffic will be interrupted.
- If a backend server has been associated with the listener and health checks are enabled, enabling this function will check the health of the backend server, and traffic to this server will be interrupted for one or two health check intervals.
- If **Transfer Client IP Address** is enabled, traffic, such as unidirectional download or push traffic, may be interrupted when backend servers are being migrated. After backend servers are migrated, retransmit the packets to restore the traffic.

Enabling Transfer Client IP Address

⚠ CAUTION

After **Transfer Client IP Address** is enabled, configure security groups, network ACLs, and OS and software security policies so that IP addresses of the clients can access these backend servers.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. You can use either of the following methods to enable the feature:
   - On the **Listeners** tab, locate the listener and click **Edit** in the **Operation** column.
   - Click the name of the target listener. On the **Summary** tab, click **Edit** on the top right corner.
4. In the displayed dialog box, enable **Transfer Client IP Address**.
5. Confirm the configurations and click **OK**.

Disabling Transfer Client IP Address

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. You can use either of the following methods to disable the feature:
   - On the **Listeners** tab, locate the listener and click **Edit** in the **Operation** column.
   - Click the name of the target listener. On the **Summary** tab, click **Edit** on the top right corner.
4. In the displayed dialog box, disable **Transfer Client IP Address**.
5. Confirm the configurations and click **OK**.

Alternatives for Obtaining the IP Address of a Client

You can obtain the IP address of a client in the ways listed in **Table 2-34**.

Table 2-34 Alternatives

Listener Type	Alternatives
TCP	**Configuring the TOA Module**
HTTP and HTTPS	**Layer 7 Load Balancing**

2.7.2 SNI Certificate

Server Name Indication (SNI) is an extension to TLS. It allows clients to specify which domain name of a listener they are trying to connect in the first request. Once receiving the request, the load balancer searches for the certificate based on the domain name.

SNI Overview

Suppose a listener is associated with a server that hosts multiple HTTPS services, each with its own certificate and domain name.

If the HTTPS listener has only one server certificate, it will always present that same certificate to all clients, regardless of the domain name the clients are trying to access. This may make authentication abnormal.

To address this issue, you can enable SNI when you add an HTTPS listener, allowing the listener to select the right certificate for authentication based on the requested domain name. SNI allows clients to specify which domain name they are trying to connect in the initial SSL handshake. Once receiving the request, the load balancer searches for the certificate based on the domain name. If there is no match, the load balancer uses the default server certificate for authentication.

SNI Certificate

- SNI certificates are server certificates used for multi-domain-name authentication. Each certificate must have an SNI domain name. The SNI domain name specified on the ELB console must be the same as the domain name supported by the certificate for authentication.
- A domain name can be used by both an ECC certificate and an RSA certificate. If this happens, ELB selects the ECC certificate first.

Constraints

- After SNI is enabled, select an SNI certificate by referring to **Adding a Certificate**.
- SNI can be only enabled for HTTPS listeners.
- An HTTPS listener can have up to 30 SNI certificates. All the certificates can have up to 30 domain names.

How SNI Certificates and Domain Names Are Matched

- Domain names in an SNI certificate are matched as follows:

  If the domain name of the certificate is *.test.com, a.test.com and b.test.com are supported, but a.b.test.com and c.d.test.com are not supported.

  The domain name with the longest suffix is matched. If a certificate contains both *.b.test.com and *.test.com, a.b.test.com preferentially matches *.b.test.com.

- **cert-default** is the default certificate bound to the HTTPS listener, and **cert-test01** and **cert-test02** are SNI certificates.

  The domain name of **cert-test01** is **www.test01.com** and that of **cert-test02** is **www.test02.com**.

  If the domain name accessing the load balancer matches either of the domain names, the corresponding SNI certificate will be used for authentication. If no domain name is matched, the default server certificate is used for authentication.

Enabling SNI for an HTTPS Listener

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click **Listeners**, locate the listener, and click its name.
4. On the **Summary** tab, click **Configure** on the right of SNI.
5. Enable SNI and select an SNI certificate.
6. Click **OK**.

2.7.3 TLS Security Policy

Scenarios

HTTPS encryption is commonly used for applications that require secure data transmission, such as banks and finance. When you add HTTPS listeners, you can select appropriate default security policies to improve security. A security policy is a combination of TLS protocols of different versions and supported cipher suites.

You can only select the default security policies for HTTPS listeners added to a shared load balancer.

Adding a Security Policy

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Under **Listeners**, click **Add Listener**.
4. On the **Add Listener** page, set **Frontend Protocol** to **HTTPS**.
5. Expand **Advanced Settings (Optional)** and select a default security policy.

   Table 2-35 lists the default security policies supported by shared load balancers.

Table 2-35 Default security policies

Name	TLS Versions	Cipher Suites
tls-1-0	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 1.2 TLS 1.1	
tls-1-2	TLS 1.2	
tls-1-2-strict	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

NOTE

- Shared load balancers support TLS 1.2 or earlier versions.
- The above table lists the cipher suites supported by ELB. Generally, clients also support multiple cipher suites. In actual use, the cipher suites supported by both ELB and clients are used, and the cipher suites supported by ELB take precedence.

6. Confirm the configurations and go to the next step.

Differences Between Security Policies

Table 2-36 Differences between the security policies

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-2-strict
TLS version				
Protocol-TLS 1.3	N/A	N/A	N/A	N/A
Protocol-TLS 1.2	Supported	Supported	Supported	Supported
Protocol-TLS 1.1	Supported	Supported	N/A	N/A
Protocol-TLS 1.0	Supported	N/A	N/A	N/A
Cipher suite				
EDHE-RSA-AES128-GCM-SHA256	Supported	Supported	Supported	Supported
ECDHE-RSA-AES256-GCM-SHA384	Supported	Supported	Supported	Supported
ECDHE-RSA-AES128-SHA256	Supported	Supported	Supported	Supported
ECDHE-RSA-AES256-SHA384	Supported	Supported	Supported	Supported
AES128-GCM-SHA256	Supported	Supported	Supported	Supported
AES256-GCM-SHA384	Supported	Supported	Supported	Supported
AES128-SHA256	Supported	Supported	Supported	Supported
AES256-SHA256	Supported	Supported	Supported	Supported
ECDHE-RSA-AES128-SHA	Supported	Supported	Supported	N/A
ECDHE-RSA-AES256-SHA	Supported	Supported	Supported	N/A
AES128-SHA	Supported	Supported	Supported	N/A
AES256-SHA	Supported	Supported	Supported	N/A

Security Policy	tls-1-0	tls-1-1	tls-1-2	tls-1-2-strict
ECDHE-ECDSA-AES128-GCM-SHA256	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES128-SHA256	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES128-SHA	Supported	Supported	Supported	N/A
ECDHE-ECDSA-AES256-GCM-SHA384	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES256-SHA384	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES256-SHA	Supported	Supported	Supported	N/A
ECDHE-RSA-AES128-GCM-SHA256	N/A	N/A	N/A	N/A
TLS_AES_256_GCM_SHA384	N/A	N/A	N/A	N/A
TLS_CHACHA20_POLY1305_SHA256	N/A	N/A	N/A	N/A
TLS_AES_128_GCM_SHA256	N/A	N/A	N/A	N/A
TLS_AES_128_CCM_8_SHA256	N/A	N/A	N/A	N/A
TLS_AES_128_CCM_SHA256	N/A	N/A	N/A	N/A
DHE-RSA-AES128-SHA	N/A	N/A	N/A	N/A
DHE-DSS-AES128-SHA	N/A	N/A	N/A	N/A
CAMELLIA128-SHA	N/A	N/A	N/A	N/A
EDH-RSA-DES-CBC3-SHA	N/A	N/A	N/A	N/A
DES-CBC3-SHA	N/A	N/A	N/A	N/A
ECDHE-RSA-RC4-SHA	N/A	N/A	N/A	N/A
RC4-SHA	N/A	N/A	N/A	N/A
DHE-RSA-AES256-SHA	N/A	N/A	N/A	N/A
DHE-DSS-AES256-SHA	N/A	N/A	N/A	N/A
DHE-RSA-CAMELLIA256-SHA	N/A	N/A	N/A	N/A
ECC-SM4-SM3	N/A	N/A	N/A	N/A
ECDHE-SM4-SM3	N/A	N/A	N/A	N/A

Changing a Security Policy

When you change a security policy, ensure that the security group rules configured for backend servers allow 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, resulting in service interruptions.

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. On the **Listeners** tab, locate the listener, and click its name.
4. On the **Summary** tab, click **Edit** on the top right.
5. In the **Edit** dialog box, expand **Advanced Settings (Optional)** and change the security policy.
6. Click **OK**.

2.7.4 Access Control

2.7.4.1 What Is 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.

Whitelist and Blacklist

You can set a whitelist or blacklist to control access to a listener.

- Once the whitelist is set, only the IP addresses or CIDR blocks specified in the IP address group can access the listener.

  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.
- Once the blacklist is set, the IP addresses or CIDR blocks specified in the blacklist cannot access the listener.

NOTE

- Access control does not restrict the ping command. You can still ping a load balancer from restricted IP addresses.
- To ping the IP address of a shared load balancer, you need to add a listener and associate a backend server to it.
- Whitelists and blacklists do not conflict with inbound security group rules. Access control defines the IP addresses or CIDR blocks that are allowed or denied to access listeners, while inbound security group rules control access to backend servers. Requests first match the whitelists or blacklists then the security group rules before they finally reach backend servers.

Configuring Access Control

1. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. 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**.
4. In the displayed **Configure Access Control** dialog box, configure parameters as described in **Table 2-37**.

Table 2-37 Parameter description

Parameter	Description
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.
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 **What Is an IP Address Group?**
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.

5. Click **OK**.

2.7.4.2 IP Address Group

What Is an IP Address Group?

An IP address group allows you to manage a collection of IP addresses that have the same security requirements or whose security requirements change frequently.

If you want to use a whitelist or blacklist for access control, 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 a 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 a 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.

Creating an IP Address Group

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the displayed page, click **Create IP Address Group**.
4. Configure the parameters based on **Table 2-38**.

Table 2-38 Parameters required for creating an IP address group

Parameter	Description	Example Value
Name	Specifies the name of the IP address group.	ipGroup-01
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed. For details, see the **Enterprise Management User Guide**.	N/A
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. ● You can add remarks at the end of each IP address or CIDR block and separate them with a vertical bar (|). The remarks can be up to 255 characters long. Angle brackets (<>) are not allowed. ● You can add a maximum of 300 entries (including IP addresses and CIDR blocks) to each IP address group.	● Without remarks: 10.168.2.24 ● With remarks: 10.168.16.0/24 | ECS01

Parameter	Description	Example Value
Description	Provides supplementary information about the IP address group.	N/A

5. Click **OK**.

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**

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.
- You can add remarks at the end of each IP address or CIDR block and separate them with a vertical bar (|), for example, 192.168.10.10 | ECS01. The remarks can be up to 255 characters long. Angle brackets (<>) are not allowed.
- You can add a maximum of 300 entries (including IP addresses and CIDR blocks) to each IP address group.

Adding IP Addresses

You can add IP addresses to an existing IP address group.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the target IP address group and click its name.
4. In the lower part of the displayed page, choose the **IP Addresses** tab and click **Add IP Addresses**. On the **Add IP Addresses** page, add IP addresses.
5. Click **OK**.

Changing IP Addresses

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

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. 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. You can modify the name and description of an IP address group, and change all its IP addresses.

   ii. Click **OK**.

b. Only change IP addresses:

   i. Locate the target IP address group and click its name.

   ii. In the lower part of the displayed page, choose **IP Addresses** tab, click **Change IP Addresses**, and change IP addresses as you need.

   iii. 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. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the target IP address group and click its name.
4. In the IP address list, locate the IP address you want to delete and click **Delete** in the **Operation** column.
5. Confirm the information and click **OK**.

Viewing the Details of an IP Address Group

You can view the details of an IP address group, including:

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

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the target IP address group and click its name.
4. Viewing the basic information about the IP address group.

   a. On the **IP Addresses** tab, view the IP addresses or CIDR blocks.

   b. On the **Associated Listeners** tab, view the listeners associated with the IP address group.

Deleting an IP Address Group

If an IP address group is used for controlling access to a listener, it 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**.

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
3. On the **IP Address Groups** page, locate the IP address group and click **Delete** in the **Operation** column.
4. Click **OK**.

2.7.5 Certificate

2.7.5.1 Certificate Overview

When you add 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.

Use Cases

When you add an HTTPS or TLS listener to route requests, you need to select **SSL Authentication**. For one-way authentication, you need to configure a server certificate for the listener. For two-way authentication, you need to configure both a server certificate and a CA certificate.

Table 2-39 SSL authentication

One-way Authentication	Only backend servers will be authenticated. You need to bind a server certificate to the listener to authenticate the server.
Mutual Authentication	The clients and the load balancer authenticate each other. Only authenticated clients will be allowed to access the load balancer. You need to bind both a server certificate and a CA certificate to the listener to allow the clients and the load balancer to authenticate each other. You do not need to configure two-way authentication on the backend servers.

ELB supports two types of certificates.

Table 2-40 Certificate types

Server Certificate	Used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
CA Certificate	Also called client CA public key certificate and used to verify the client certificate issuer. If mutual authentication is required, connections can be established only when the client provides a certificate issued by a specific CA.

Precautions

- A certificate can be used by multiple load balancers but only needs to be uploaded to ELB once.
- You must specify a domain name for an SNI certificate. The domain name must be the same as that in the certificate. An SNI certificate can have multiple domain names.
- 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. It is recommended that you use certificates issued by third parties.
- ELB only supports certificates 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.

Certificate Format

You can copy and paste the certificate body to create a certificate or directly upload a 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 requirements as described below.

- The content must start 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-----
Base64–encoded certificate
-----END CERTIFICATE-----
```

Private Key Format

When creating a server 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.

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 contains 64 characters except the last row.

The following is an example:

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

Converting Certificate Formats

ELB only supports certificates 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 and Tomcat servers.

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 servers.

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
```

2.7.5.2 Adding a Certificate

Scenarios

To enable authentication for securing data transmission over HTTPS, ELB allows you to bind certificates to HTTPS listeners of a load balancer.

- Server certificate: used for SSL handshake negotiations if an HTTPS listener is used. You can purchase a certificate from Cloud Certificate Manager (CCM) or upload your own certificates.
- CA certificate: a certificate issued by a certificate authority (CA). They are used to verify the client certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can only be established when the client provides a certificate issued by a specific CA. You can only upload your own CA certificates.

NOTE

If you want to use the same certificate in two regions, you need to add a certificate in each region.

Adding a Server 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. Click ☰ 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** on the top right corner and set parameters by referring to **Table 2-41**.

Table 2-41 Server certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **Server certificate**. **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.

Parameter	Description
Source	Specifies the source of a certificate. You can purchase a certificate from CCM or upload your own certificates. ● **SSL Certificate Manager**: server certificates provided by CCM. You need to buy a certificate or upload your own certificates. ● **Your certificate**: You need to upload the certificate content and private key of your own certificate to the ELB console. **NOTE** You are advised to use CCM to manage your certificates.
Certificate	This parameter is only available for certificates managed on the CCM console. You can select a certificate managed by CCM.
Certificate Name	Specifies the name of your certificate. A certificate name can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

Certificate Content	Specifies the content of a certificate. This parameter is only available for your certificates.
	The content must be in PEM format.
	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	Specifies the private key of a certificate. This parameter is only available for your certificates.
	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 as follows:
	`-----BEGIN PRIVATE KEY-----`
	`[key]`
	`-----END PRIVATE KEY-----`

Parameter	Description
SNI Domain Name (Optional)	The domain name must be specified if the certificate is intended for SNI. A domain name can contain only letters, digits, and hyphens (-) and consist of multiple labels (max. 63 characters each) separated by periods (.). It cannot start or end with a hyphen (-). You can specify up to 100 domain names, separated by commas (,). A domain name can contain a maximum of 100 characters, and the total length cannot exceed 10,000 characters.
Description	(Optional) Provides supplementary information about the certificate.

Adding a CA 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. Click ☰ 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** on the top right corner and set parameters by referring to **Table 2-42**.

Table 2-42 CA certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **CA certificate**. **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can only be established when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of the CA certificate.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

Parameter	Description

Certificate Content	Specifies the content of the CA certificate in PEM format.
	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-----`

Description	(Optional) Provides supplementary information about the certificate.

6. Click **OK**.

2.7.5.3 Managing Certificates

Scenarios

You can manage your certificates on the ELB console. If a certificate is no longer needed, you can delete it.

Notes and Constraints

A certificate that has been bound to an HTTPS listener cannot be deleted. Disassociate the certificate from the listener first by referring to **Replacing a Certificate**.

Querying Listeners by Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. 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.

Modifying a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Modify** in the **Operation** column.
4. In the **Modify Certificate** dialog box, modify the parameters as required.
5. Confirm the information and click **OK**.

Deleting a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Delete** in the **Operation** column.
4. In the displayed dialog box, click **OK**.

2.7.5.4 Binding or Replacing a Certificate

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.

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 a certificate and private keys does not affect your applications.

Notes and Constraints

- Only HTTPS listeners require certificates.
- If a certificate has expired, you need to manually replace or delete it.
- The new certificate takes effect immediately. The old certificate is used for established connections, and the new one is used for new connections.

Prerequisites

You have added 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. Go to the **load balancer list page**.
2. On the displayed page, locate the load balancer and click its name.
3. Click the **Listeners** tab, locate the listener, and click **Edit** in **Operation** column.
4. On the displayed dialog box, select a server certificate or CA certificate.
5. Click **OK** in the **Edit** dialog box.

2.7.5.5 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.

Notes and Constraints

- Only HTTPS listeners require certificates.
- The new certificate takes effect immediately. The previous certificate is used for established connections, and the new one is used for new connections.

Modifying a Certificate

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, choose **Certificates**.
3. Locate the certificate and click **Modify** in the **Operation** column.
4. Modify the parameters as required.
5. Confirm the information and click **OK**.

2.8 Access Logging

Scenarios

ELB logs HTTP and HTTPS requests received by shared load balancers, including the time when the request was sent, client IP address, request path, and server response.

Log Tank Service (LTS) can log Layer 7 requests, of a load balancer including the time when the request was sent, client IP address, request path, server response, and more. If there are service faults or exceptions caused by unhealthy backend servers, you can view logs of requests to load balancers and analyze response status codes to quickly locate unhealthy backend servers.

WARNING

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

Constraints

- Access logging can be configured only for shared load balancers that have HTTP or HTTPS listeners.

- The access logs do not contain requests whose return code is **400 Bad Request**. This is because such requests do not comply with HTTP specification and cannot be processed properly.

Prerequisites

- You have created an application load balancer.
- You have enabled LTS.
- You have created a backend server group, added backend servers to the group, and deployed services on the backend servers. For details, see **Creating a Backend Server Group**.
- You have added an HTTP or HTTPS listener to the load balancer. For details, see **Adding an HTTP Listener** or **Adding an HTTPS Listener**.

Configuring Access Logging

1. Go to the **load balancer list page**.
2. On the **Load Balancers** page, 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 have created.

   If there are no log groups and log streams, you can create them on the ELB console.

   Creating a Log Group and Log Stream

   a. Click **Create Log Group and Log Stream**.
   b. In the **Create Log Group and Log Stream** dialog box, configure the log group name, log stream name, and log retention period.
   c. Click **OK**.
5. Click **OK**.

Viewing Access Logs

You can view details about access logs on the:

- ELB console: Click the name of the load balancer and click **Access Logs** to view logs.
- (Recommended) LTS console: Locate the target log group and click its name. On the displayed page, locate the target log stream and click **Real-Time Logs** tab.

The log format is as follows, which 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
```

The following is a log example:

```
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 -
```

Table 2-43 describes the fields in the log.

Table 2-43 Parameter description

Parameter	Description	Value Description	Example Value
msec	Time when the log is written, in seconds with a milliseconds resolution.	Floating-point data	1644819836.370
access_log_topic_id	Log stream ID.	uuid	eb11c5a9-93a7-4c48-80fc-03f61f638595
time_iso8601	Local time in the ISO 8601 standard format.	N/A	[2022-02-14T14:23:56+08:00]
log_ver	Log format version.	Fixed value: **elb_01**	elb_01
remote_addr: remote_port	IP address and port number of the client.	Records the IP address and port of the client.	192.168.1.1:888
status	HTTP status code.	Records the request status code.	200

Parameter	Description	Value Description	Example Value
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 and it does not include the protocol and host name.	"POST https://www.test.com/example/ HTTP/1.1"
request_length	Length of the request received from the client, including the header and body.	Integer	1411
bytes_sent	Number of bytes sent to the client.	Integer	251
body_bytes_sent	Number of bytes sent to the client (excluding the response header).	Integer	3
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	0.011

Parameter	Description	Value Description	Example Value
upstream_status	HTTP status code returned by the upstream server. ● When the load balancer attempts to retry a request, there will be multiple HTTP 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"
upstream_connect_time	Time taken to establish a connection with the server, in seconds, with a milliseconds 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	"0.000"

Parameter	Description	Value Description	Example Value
upstream_header_time	Time taken to receive the response header from the server, in seconds, with a milliseconds 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	"0.011"
upstream_response_time	Time taken to receive the response from the server, in seconds, with a milliseconds 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	"0.011"

Parameter	Description	Value Description	Example Value
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 -.	IP address and port number	"100.64.0.129:8080" (used by shared load balancers for internal communications)
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.	"okhttp/3.13.1"
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_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	"-"
lb_name	Load balancer name in the format of **loadbalancer**_*load balancer ID*	String	loadbalancer_295a7eee-9999-46ed-9fad-32a62ff0a687
listener_name	Listener name in the format of **listener**_*listener ID*.	String	listener_20679192-8888-4e62-a814-a2f870f62148

Parameter	Description	Value Description	Example Value
listener_id	Listener ID. This field can be ignored.	String	3333fd44fe3b42cbaa1dc2c641994d90
pool_name	Backend server group name in the format of **pool**_*backend server group ID*	String	pool_89547549-6666-446e-9dbc-e3a551034c46
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	"-"
tenant_id	Tenant ID.	String	f2bc165ad9b4483a9b17762da851bbbb
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.	121.64.212.1:443
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 -.	IP address and port number	"10.1.1.2:8080"

Parameter	Description	Value Description	Example Value
certificate_id	[HTTPS listener] Certificate ID used for establishing an SSL connection. This field is not supported yet.	String	N/A
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	TLSv1.2
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
sni_domain_name	[HTTPS listener] SNI domain name provided by the client during SSL handshakes. 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	56704
self_defined_header	This field is reserved. The default value is -.	String	N/A

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.

2.9 Tags and Quotas

2.9.1 Tag

Scenarios

If you have a large number of cloud resources, you can add different tags to the resources to quickly identify them and use these tags to easily manage your resources.

Adding a Tag to a Load Balancer

You can add a tag to a load balancer in the following methods:

- Add a tag when you create a load balancer.
- Add a tag to an existing load balancer.
  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 to display **Service List** and choose **Network** > **Elastic Load Balance**.
  d. On the **Load Balancers** page, locate the load balancer and click its name.
  e. Under **Tags**, click **Add Tag**.
  f. In the **Add Tag** dialog box, enter a tag key and value and click **OK**.

Adding a Tag to a Listener

To add a tag to an existing listener, perform the following steps:

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 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. Click **Listeners**, locate the listener, and click its name.
6. Under **Tags**, click **Add Tag**.

7. In the **Add Tag** dialog box, enter a tag key and value and click **OK**.

Modifying a Tag of 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. Click ☰ 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. Click **Tags**, select the tag to be edited, and click **Edit** in the **Operation** column. In the **Edit Tag** dialog box, enter a tag value.

   NOTE

   The tag key cannot be changed.
6. In the **Add Tag** dialog box, enter a tag key and value and click **OK**.

The operations for modifying a listener tag are not detailed here. Refer to the operations of modifying a load balancer tag.

Deleting a Tag from 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. Click ☰ 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. Click **Tags**, select the tag to be deleted, and click **Delete** in the **Operation** column.
6. In the displayed dialog box, click **OK**.

The operations for deleting a listener tag are not detailed here. Refer to the operations of deleting a load balancer tag.

2.9.2 Quotas

What Is Quota?

Quotas can limit the number 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. Click 📍 in the upper left corner and select the desired region and project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.
4. 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?

1. Log in to the management console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Quotas** page is displayed.
3. Click **Increase Quota** in the upper right corner of the page.
4. On the **Create Service Ticket** page, configure parameters as required.

   In the **Problem Description** area, fill in the content and reason for adjustment.
5. After all necessary parameters are configured, select **I have read and agree to the Ticket Service Protocol and Privacy Statement** and click **Submit**.

2.10 Cloud Eye Monitoring

2.10.1 Monitoring ELB Resources

Scenarios

Cloud Eye is a multi-dimensional resource monitoring service. You can use Cloud Eye to monitor ELB resources in real time, set alarm rules, identify resource exceptions, and quickly respond to resource changes.

Cloud Eye is enabled automatically after you create a load balancer.

Setting an Alarm Rule

You can set alarm rules on the Cloud Eye console to send you notifications in case of exceptions.

For details about how to set alarm rules, see .

Viewing Monitoring Metrics

You can view the metrics described in **Monitoring Metrics** either on the ELB console or on the Cloud Eye console.

Viewing Monitoring Metrics on the ELB 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 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. View the metrics of each load balancer and listener.
   a. Load balancer: Click the **Monitoring** tab and select **Load balancer** for **Dimension**.
   b. Listener (two ways):
      i. Click the **Monitoring** tab, select **Listener** for **Dimension**, select the target listener, and view the monitoring metrics.
      ii. Click the **Listeners** tab, locate the target listener, and click its name. Switch to the **Monitoring** tab and view the monitoring metrics.

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 & Governance** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Cloud Service Monitoring**. In the displayed page, locate the **Dashboard** column and click **Elastic Load Balance ELB**.
5. On the displayed page, locate the target load balancer and click its name.

   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.

2.10.2 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.

Namespace

SYS.ELB

Load Balancer Metrics

For shared load balancers, you can view the monitoring metrics by load balancer or listener.

Table 2-44 Metrics supported by each shared load balancer

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections established between the monitored object and backend servers. Load balancing at Layer 7: total number of TCP connections established between the clients and the monitored object.	≥ 0	Count	N/A	Shared load balancer	1 minute
m2_act_conn	Active Connections	The number of active TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m3_inact_conn	Inactive Connections	The number of inactive TCP and UDP connections established between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an`	≥ 0	Count	N/A	Shared load balancer	1 minute
m4_ncps	New Connections	The number of new connections established between clients and the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m5_in_pps	Incoming Packets	The number of packets received by the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m6_out_pps	Outgoing Packets	The number of packets sent from the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m7_in_Bps	Inbound Traffic Rate	How fast the inbound traffic reaches the monitored object.	≥ 0	Byte/s	1000 (SI)	Shared load balancer	1 minute
m8_out_Bps	Outbound Traffic Rate	How fast the outbound traffic leaves the monitored object.	≥ 0	Byte/s	1000 (SI)	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m9_abnormal_servers	Unhealthy Servers	The number of unhealthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Shared load balancer	1 minute
ma_normal_servers	Healthy Servers	The number of healthy backend servers associated with the monitored object.	≥ 0	Count	N/A	Shared load balancer	1 minute
m22_in_bandwidth	Inbound Bandwidth	The bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	Shared load balancer	1 minute
m23_out_bandwidth	Outbound Bandwidth	The bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	Shared load balancer	1 minute
m1e_server_rps	Reset Packets from Backend Servers	The number of reset packets sent by backend servers to clients per second. These reset packets are generated by the backend servers and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m21_client_rps	Reset Packets from Clients	The number of reset packets sent by clients to backend servers per second. These reset packets are generated by clients and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m1f_lvs_rps	Reset Packets from Load Balancer	The number of reset packets generated by the load balancer per second. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer	1 minute
mb_l7_qps	Layer 7 Query Rate	The number of queries the monitored object receives per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
mc_l7_http_2xx	2xx Status Codes (Total)	The number of 2xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
md_l7_http_3xx	3xx Status Codes (Total)	The number of 3xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
me_l7_http_4xx	4xx Status Codes (Total)	The number of 4xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
mf_l7_http_5xx	5xx Status Codes (Total)	The number of 5xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m10_l7_http_other_status	Other Status Codes (Total)	The number of other status codes returned by the monitored object and backend servers per second at Layer 7. Excluded status codes: 2xx, 3xx, 404 Not Found, 499 Client Closed Request, and 502 Bad Gateway Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m11_l7_http_404	404 Not Found (Total)	The number of 404 Not Found status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m12_l7_http_499	499 Client Closed Request (Total)	The number of 499 Client Closed Request status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m13_l7_http_502	502 Bad Gateway (Total)	The number of 502 Bad Gateway status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m14_l7_rt	Average Layer 7 Response Time	Average response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m15_l7_upstream_4xx	4xx Status Codes (Backend Servers)	The number of 4xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m16_l7_upstream_5xx	5xx Status Codes (Backend Servers)	The number of 5xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m18_l7_upstream_2xx	2xx Status Codes (Backend Servers)	The number of 2xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute
m19_l7_upstream_3xx	3xx Status Codes (Backend Servers)	The number of 3xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute
m1c_l7_rt_max	Maximum Layer 7 Response Time	Maximum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1d_l7_rt_min	Minimum Layer 7 Response Time	Minimum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer	1 minute
m25_l7_resp_Bps	Layer 7 Response Bandwidth	The bandwidth used by the backend servers associated with the monitored object to return responses to clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Shared load balancer	1 minute
m24_l7_req_Bps	Layer 7 Request Bandwidth	The bandwidth used by the backend servers associated with the monitored object to receive requests from clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Shared load balancer	1 minute

Listener Metrics

Table 2-45 Metrics supported by each listener

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections established between the monitored object and backend servers. Load balancing at Layer 7: total number of TCP connections established between the clients and the monitored object. Unit: Count	≥ 0	Count	N/A	Shared load balancer - listener	1 minute
m2_act_conn	Active Connections	The number of active TCP and UDP connections established 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: Count	≥ 0	Count	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m3_inact_conn	Inactive Connections	The number of inactive TCP and UDP connections established 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: Count	≥ 0	Count	N/A	Shared load balancer - listener	1 minute
m4_ncps	New Connections	The number of new connections established between clients and the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m5_in_pps	Incoming Packets	The number of packets received by the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m6_out_pps	Outgoing Packets	The number of packets sent from the monitored object per second.	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m7_in_Bps	Inbound Traffic Rate	How fast the inbound traffic reaches the monitored object.	≥ 0	Byte/s	1000 (SI)	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m8_out_Bps	Outbound Traffic Rate	How fast the outbound traffic leaves the monitored object.	≥ 0	Byte/s	1000 (SI)	Shared load balancer - listener	1 minute
m22_in_bandwidth	Inbound Bandwidth	The bandwidth used for accessing the monitored object from external networks.	≥ 0	bit/s	1000 (SI)	Shared load balancer - listener	1 minute
m23_out_bandwidth	Outbound Bandwidth	The bandwidth used by the monitored object to access external networks.	≥ 0	bit/s	1000 (SI)	Shared load balancer - listener	1 minute
m1e_server_rps	Reset Packets from Backend Servers	The number of reset packets sent by backend servers to clients per second. These reset packets are generated by the backend servers and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m21_client_rps	Reset Packets from Clients	The number of reset packets sent by clients to backend servers per second. These reset packets are generated by clients and then forwarded by the load balancer. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1f_lvs_rps	Reset Packets from Load Balancer	The number of reset packets generated by the load balancer per second. Supported protocols: TCP	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
mb_l7_qps	Layer 7 Query Rate	The number of queries the monitored object receives per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
mc_l7_http_2xx	2xx Status Codes (Total)	The number of 2xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
md_l7_http_3xx	3xx Status Codes (Total)	The number of 3xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
me_l7_http_4xx	4xx Status Codes (Total)	The number of 4xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
mf_l7_http_5xx	5xx Status Codes (Total)	The number of 5xx status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m10_l7_http_other_status	Other Status Codes (Total)	The number of other status codes returned by the monitored object and backend servers per second at Layer 7. Excluded status codes: 2xx, 3xx, 404 Not Found, 499 Client Closed Request, and 502 Bad Gateway Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m11_l7_http_404	404 Not Found (Total)	The number of 404 Not Found status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m12_l7_http_499	499 Client Closed Request (Total)	The number of 499 Client Closed Request status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m13_l7_http_502	502 Bad Gateway (Total)	The number of 502 Bad Gateway status codes returned by the monitored object and backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m14_l7_rt	Average Layer 7 Response Time	Average response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute
m15_l7_upstream_4xx	4xx Status Codes (Backend Servers)	The number of 4xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m16_l7_upstream_5xx	5xx Status Codes (Backend Servers)	The number of 5xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m18_l7_upstream_2xx	2xx Status Codes (Backend Servers)	The number of 2xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute
m19_l7_upstream_3xx	3xx Status Codes (Backend Servers)	The number of 3xx status codes returned by the backend servers per second at Layer 7. Supported protocols: HTTP/HTTPS	≥ 0	Count/s	N/A	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS **NOTE** The average response time it takes to establish a WebSocket connection may be very long. This metric cannot be used as a reference.	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of the backend server associated with the monitored object at Layer 7. The response time starts when the monitored object routes the requests to the backend server and ends when the monitored object receives a response from the backend server. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m1c_l7_rt_max	Maximum Layer 7 Response Time	Maximum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute
m1d_l7_rt_min	Minimum Layer 7 Response Time	Minimum response time of the monitored object at Layer 7. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Supported protocols: HTTP/HTTPS	≥ 0	ms	1000 (SI)	Shared load balancer - listener	1 minute
m25_l7_resp_Bps	Layer 7 Response Bandwidth	The bandwidth used by the backend servers associated with the monitored object to return responses to clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
m24_l7_req_Bps	Layer 7 Request Bandwidth	The bandwidth used by the backend servers associated with the monitored object to receive requests from clients. **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0	bit/s	1000 (SI)	Shared load balancer - listener	1 minute

Dimensions

Key	Value
lbaas_instance_id	ID of a shared load balancer
lbaas_listener_id	ID of a listener added to a shared load balancer

2.10.3 Viewing Traffic Usage

Scenarios

For livestreaming platforms, traffic often increases suddenly, which makes the services unstable. To address this issue, most of them use ELB to distribute traffic. By working with Cloud Eye, ELB allows you to monitor the traffic usage in real time. You can view the traffic consumed by the EIPs bound to public network load balancers to better balance your application workloads.

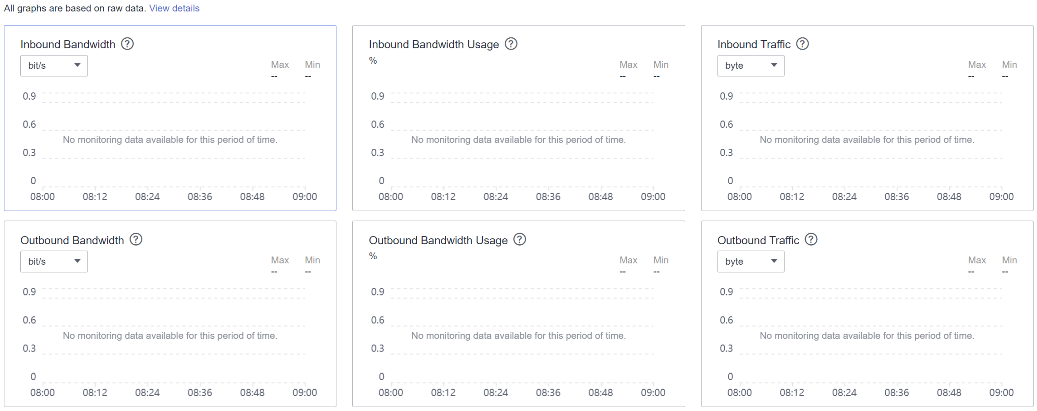
Prerequisites

- Load balancers are running properly.
- If the associated backend server is stopped, faulty, or deleted, its metrics cannot be viewed on Cloud Eye. After such a backend server restarts or recovers, its monitoring data will be displayed on the Cloud Eye console.

Viewing Traffic Usage of the Bound 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. Click ☰ in the upper left corner of the page and choose **Network** > **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Elastic IP and Bandwidth** > **EIPs**.
5. Locate the EIP bound to the load balancer and click its name. On the **Bandwidth** tab, you can view the data for the last 1, 3, 12 hours, last day, or last 7 days.

Figure 2-14 EIP traffic usage



All graphs are based on raw data. View details

Table 2-46 EIP and bandwidth metrics

Metric	Meaning	Value Range	Monitored Object	Monitoring Interval (Raw Data)
Outbound Bandwidth (originally named "Upstream Bandwidth")	Network rate of outbound traffic	≥ 0 bits/s	Bandwidth or EIP	1 minute

Metric	Meaning	Value Range	Monitored Object	Monitoring Interval (Raw Data)
Inbound Bandwidth (originally named "Downstream Bandwidth")	Network rate of inbound traffic	≥ 0 bits/s	Bandwidth or EIP	1 minute
Outbound Bandwidth Usage	Usage of outbound bandwidth in percentage.	0–100%	Bandwidth or EIP	1 minute
Inbound Bandwidth Usage	Usage of inbound bandwidth in the unit of percent.	0–100%	Bandwidth or EIP	1 minute
Outbound Traffic (originally named "Upstream Traffic")	Network traffic going out of the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute
Inbound Traffic (originally named "Downstream Traffic")	Network traffic going into the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute

Viewing Load Balancer Traffic Metrics

1. Go to the **load balancer list page**.
2. On the load balancer list page, locate the load balancer and click its name.
3. Click the **Monitoring** tab, select load balancer for **Dimension**, and view the graphs of inbound and outbound rates.

   You can view data from the last 1, 3, 12 hours, last day, or the last 7 days.

2.11 CTS Auditing

2.11.1 Key Operations Recorded by CTS

You can use CTS to record operations on ELB for query, auditing, and backtracking.

Table 2-47 lists the operations recorded by CTS.

Table 2-47 ELB operations recorded by CTS

Action	Resource Type	Trace Name
Configuring access logs	logtank	createLogtank
Deleting access logs	logtank	deleteLogtank
Creating a certificate	certificate	createCertificate
Modifying a certificate	certificate	updateCertificate
Deleting a certificate	certificate	deleteCertificate
Creating a health check	healthmonitor	createHealthMonitor
Modifying a health check	healthmonitor	updateHealthMonitor
Deleting a health check	healthmonitor	deleteHealthMonitor
Adding a forwarding policy	l7policy	createL7policy
Modifying a forwarding policy	l7policy	updateL7policy
Deleting a forwarding policy	l7policy	deleteL7policy
Adding a forwarding rule	l7rule	createl7rule
Modifying a forwarding rule	l7rule	updateL7rule
Deleting a forwarding rule	l7rule	deleteL7rule
Adding a listener	listener	createListener
Modifying a listener	listener	updateListener
Deleting a listener	listener	deleteListener
Creating a load balancer	loadbalancer	createLoadbalancer
Modifying a load balancer	loadbalancer	updateLoadbalancer
Deleting a load balancer	loadbalancer	deleteLoadbalancer

Action	Resource Type	Trace Name
Adding a backend server	member	createMember
Modifying a backend server	member	updateMember
Removing a backend server	member	batchUpdateMember
Creating a backend server group	pool	createPool
Modifying a backend server group	pool	updatPool
Deleting a backend server group	pool	deletePool

2.11.2 Viewing Traces

Scenarios

After CTS is enabled, CTS starts recording operations on cloud resources. The CTS management console stores the last seven days of operation records.

This section describes how to query or export the last seven days of operation records on the management console.

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. Under **Management & Governance**, click **Cloud Trace Service**.
4. In the navigation pane on the left, choose **Trace List**.
5. Specify the filters used for querying traces. The following filters are available:
   - **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**

     Select a filter from the drop-down list.

     If you select **Trace name** for **Search By**, you need to select a specific trace name.

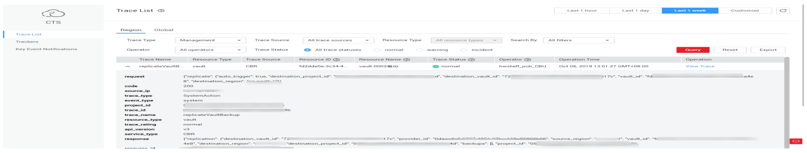
     If you select **Resource ID** for **Search By**, select or enter a specific resource ID.

     If you select **Resource name** for **Search By**, select or enter a specific resource name.
   - **Operator**: Select a specific operator (at the user level rather than the tenant level).

- **Trace Status**: Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.
- Time range: You can query traces generated at any time range of the last seven days.

6. Click ⌄ on the left of the required trace to expand its details.

Figure 2-15 Expanding trace details

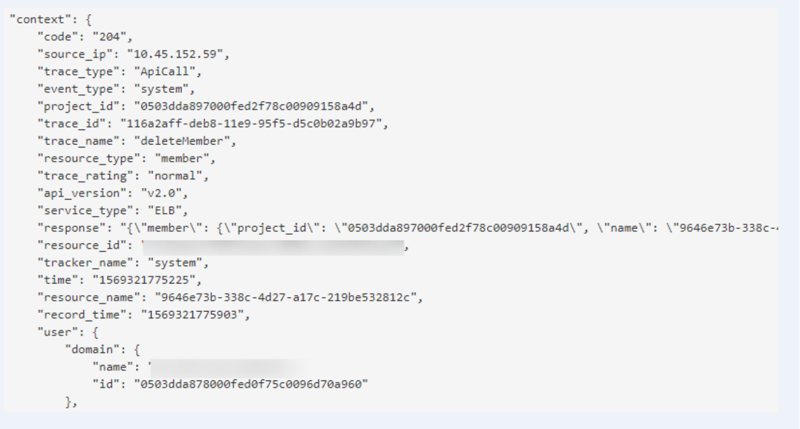


Trace Name	Resource Type	Trace Source	Resource ID	Resource Name	Trace Status	Operator	Operation Time	Operation
replicateVaultB...	vault	CBR	fd2dde0e-3c34-4...	vault-0002备份	normal	hwstaff_pub_CBU...	Oct 06, 2019 13:01:27 GMT+08:00	View Trace

```
request        {"replicate": {"auto_trigger": true, "destination_project_id": "[redacted]d", "destination_vault_id": "72[redacted]17c", "vault_id": "fd[redacted]a4e
               8", "destination_region": "[redacted]
code           200
source_ip      [redacted]
trace_type     SystemAction
event_type     system
project_id     [redacted]
trace_id       [redacted]4b
trace_name     replicateVaultBackup
resource_type  vault
trace_rating   normal
api_version    v3
service_type   CBR
response       {"replication": {"destination_vault_id": "72[redacted]17c", "provider_id": "0da[redacted]", "source_region": "[redacted]", "vault_id": "f[redacted]
               4e8", "destination_region": "[redacted]"destination_project_id": "0[redacted]4d", "backups": [], "project_id": "05[redacted]
```

7. Click **View Trace** in the **Operation** column to view trace details.

Figure 2-16 View Trace



```
"context": {
    "code": "204",
    "source_ip": "10.45.152.59",
    "trace_type": "ApiCall",
    "event_type": "system",
    "project_id": "0503dda897000fed2f78c00909158a4d",
    "trace_id": "116a2aff-deb8-11e9-95f5-d5c0b02a9b97",
    "trace_name": "deleteMember",
    "resource_type": "member",
    "trace_rating": "normal",
    "api_version": "v2.0",
    "service_type": "ELB",
    "response": "{\"member\": {\"project_id\": \"0503dda897000fed2f78c00909158a4d\", \"name\": \"9646e73b-338c-
    "resource_id": '[redacted],
    "tracker_name": "system",
    "time": "1569321775225",
    "resource_name": "9646e73b-338c-4d27-a17c-219be532812c",
    "record_time": "1569321775903",
    "user": {
        "domain": {
            "name": '[redacted]
            "id": "0503dda878000fed0f75c0096d70a960"
        },
```

For details about key fields in the trace, see the **Cloud Trace Service User Guide**.

Example Traces

- Creating a load balancer

```
request {"loadbalancer":{"name":"elb-test-
zcy","description":"","tenant_id":"05041fffa40025702f6dc009cc6f8f33","vip_subnet_id":"ed04fd93-
e74b-4794-b63e-e72baa02a2da","admin_state_up":true}}
code 201
source_ip 124.71.93.36
trace_type ConsoleAction
event_type system
project_id 05041fffa40025702f6dc009cc6f8f33
trace_id b39b21a1-8d49-11ec-b548-2be046112888
trace_name createLoadbalancer
resource_type loadbalancer
trace_rating normal
api_version v2.0
service_type ELB
response {"loadbalancer": {"description": "", "provisioning_status": "ACTIVE", "provider": "vlb",
"project_id": "05041fffa40025702f6dc009cc6f8f33", "vip_address": "172.18.0.205", "pools": [],
"operating_status": "ONLINE", "name": "elb-test-zcy", "created_at": "2022-02-14T03:53:39",
"listeners": [], "id": "7ebe23cd-1d46-4a49-b707-1441c7f0d0d1", "vip_port_id":
"5b36ff96-3773-4736-83cf-38c54abedeea", "updated_at": "2022-02-14T03:53:41", "tags": [],
"admin_state_up": true, "vip_subnet_id": "ed04fd93-e74b-4794-b63e-e72baa02a2da", "tenant_id":
"05041fffa40025702f6dc009cc6f8f33"}}
```

```
resource_id 7ebe23cd-1d46-4a49-b707-1441c7f0d0d1
tracker_name system
time 2022/02/14 11:53:42 GMT+08:00
resource_name elb-test-zcy
record_time 2022/02/14 11:53:42 GMT+08:00
request_id
user {"domain": {"name": "CBUInfo", "id": "0503dda87802345ddafed096d70a960"}, "name": "zcy",
"id": "09f106afd2345cdeff5c009c58f5b4a"}
```

- Deleting a load balancer

```
request
code 204
source_ip 124.71.93.36
trace_type ConsoleAction
event_type system
project_id 05041fffa40025702f6dc009cc6f8f33
trace_id 4f838bbf-8d4a-11ec-a1fe-1f93fdaf3bec
trace_name deleteLoadbalancer
resource_type loadbalancer
trace_rating normal
api_version v2.0
service_type ELB
response {"loadbalancer": {"listeners": [], "vip_port_id": "5b36ff96-3773-4736-83cf-38c54abedeea",
"tags": [], "tenant_id": "05041fffa40025702f6dc009cc6f8f33", "admin_state_up": true, "id":
"7ebe23cd-1d46-4a49-b707-1441c7f0d0d1", "operating_status": "ONLINE", "description": "", "pools":
[], "vip_subnet_id": "ed04fd93-e74b-4794-b63e-e72baa02a2da", "project_id":
"05041fffa40025702f6dc009cc6f8f33", "provisioning_status": "ACTIVE", "name": "elb-test-zcy",
"created_at": "2022-02-14T03:53:39", "vip_address": "172.18.0.205", "updated_at":
"2022-02-14T03:53:41", "provider": "vlb"}}
resource_id 7ebe23cd-1d46-4a49-b707-1441c7f0d0d1
tracker_name system
time 2022/02/14 11:58:03 GMT+08:00
resource_name elb-test-zcy
record_time 2022/02/14 11:58:03 GMT+08:00
request_id
user {"domain": {"name": CBUInfo", "id": "0503dda87802345ddafed096d70a960"}, "name": "zcy", "id":
"09f106afd2345cdeff5c009c58f5b4a"}
```

3 Self-service Troubleshooting

3.1 Overview

ELB self-service troubleshooting helps you detect and fix unhealthy backend servers in a timely manner. It also gets you familiar with billing and service features that you might be curious about. During the troubleshooting process, resource configurations will not be changed and services will work normally.

You may find the answers to the issues listed in **Table 3-1**.

Table 3-1 ELB self-service troubleshooting

Issue	Description
Troubleshooting an Unhealthy Backend Server	● Checks the security group rules. ● Checks the network ACL configurations. ● Checks the health check ports.
ELB Billing	Describes how ELB is billed.
Differences Between Dedicated and Shared Load Balancers	Describes the advantages of each type of load balancer.

3.2 Troubleshooting an Unhealthy Backend Server

Scenarios

This section describes how you can use ELB self-service troubleshooting to detect and fix unhealthy backend servers in a timely manner.

Prerequisites

Before troubleshooting an unhealthy backend server, make sure you have completed the following:

- Creating a Backend Server Group
- Adding a TCP Listener
- Configuring a Health Check

Constraints

- You can only troubleshoot an unhealthy backend server.
- The backend server must be associated with a listener.
- IP as backend servers does not support self-service troubleshooting.

Procedure

1. Go to the **load balancer list page**.
2. In the navigation pane on the left, click **Self-service Troubleshooting**.
3. On the **Elastic Load Balance** tab, click **Unhealthy backend servers**.
4. Select the load balancer that has unhealthy backend servers.
5. Select the unhealthy backend server you want to troubleshoot.
6. Click **Troubleshoot**. On the displayed page, view the troubleshooting progress and details.

   View and rectify the faults in a timely manner as described in **Table 3-2**.

Table 3-2 Health check items

Health Check Category	Health Check Item	Reason	Suggestion
Security group rule configurations	The protocol configured for the inbound rule	The inbound rules of the security group do not allow traffic over the health check protocol.	Change the security group rules by referring to the following: • Security Group and Network ACL Rules • Security Group and Network ACL Rules
	The source configured for the inbound rule	The inbound rules of the security group do not allow traffic from the health check IP address to the backend server.	
	The port configured for the inbound rule	The inbound rules of the security group do not allow traffic over the health check port.	

Health Check Category	Health Check Item	Reason	Suggestion
	The protocol configured for the outbound rule	The outbound rules of the security group do not allow traffic over the health check protocol.	
	The destination configured for the outbound rule	The outbound rules of the security group do not allow traffic from the backend server to the health check IP address.	
	The port configured for the outbound rule	The outbound rules of the security group do not allow traffic over the health check port.	
Network ACL rule configurations	The protocol configured for the inbound rule	The inbound rules of the network ACL do not allow traffic over the health check protocol.	Change the network ACL rules by referring to the following: - **Security Group and Network ACL Rules** - **Security Group and Network ACL Rules**
	The source configured for the inbound rule	The inbound rules of the network ACL do not allow traffic from the health check IP address to the backend server.	
	The source port configured for the inbound rule	The inbound rules of the network ACL do not allow traffic over all source ports.	
	The destination address configured for the inbound rule	The inbound rules of the network ACL do not allow traffic to the destination address.	

Health Check Category	Health Check Item	Reason	Suggestion
	The destination port configured for the inbound rule	The inbound rules of the network ACL do not allow traffic over the destination port.	
	The protocol configured for the outbound rule	The outbound rules of the network ACL do not allow traffic over the health check protocol.	
	The destination configured for the outbound rule	The outbound rules of the network ACL do not allow traffic from the health check IP address to the backend server.	
	The source port configured for the outbound rule	The outbound rules of the network ACL do not allow traffic over the health check port.	
	The destination address configured for the outbound rule	The outbound rules of the network ACL do not allow traffic to the destination address.	
	The destination port configured for the outbound rule	The outbound rules of the network ACL do not allow traffic over all destination ports.	

Health Check Category	Health Check Item	Reason	Suggestion
Health check configurations	The port configured for the health check	The specified health check port is different from that used by the backend server.	Use the backend port as the health check port by referring to **Configuring a Health Check**.

3.3 Other Issues

You can also use ELB self-service troubleshooting to find the answers to the following issues:

- **ELB Billing**
- **Differences Between Dedicated and Shared Load Balancers**

ELB Billing

You can learn more about ELB billing as described in **Table 3-3**.

Table 3-3 ELB billing

Scenario	Reference
Billing rules	- **Billing Items (Dedicated Load Balancers)** - **Billing Items (Shared Load Balancers)**
Specifications	**Modifying Specifications**

Differences Between Dedicated and Shared Load Balancers

Learn more about the advantages of each type of load balancer as described in **Table 3-4**.

Table 3-4 Differences

Scenario	Reference
Feature comparison	**Differences Between Dedicated and Shared Load Balancers**

Scenario	Reference
Creating a backend server group	• **Creating a Backend Server Group** • **Creating a Backend Server Group**
Adding a backend server	• **Backend Server Overview** • **Backend Server Overview**

4 Appendix

4.1 Configuring the TOA Module

Scenarios

ELB provides customized strategies for managing service access. Before these strategies can be customized, the clients' IP addresses contained in the requests are required. To obtain the IP addresses, you can install the TCP Option Address (TOA) kernel module on backend servers.

This section provides detailed operations for you to compile the module in the OS if you use TCP to distribute incoming traffic.

The operations for Linux OSs with kernel version of 2.6.32 are different from those for Linux OSs with kernel version of 3.0 or later.

NOTE

- TOA does not support listeners using the UDP protocol.
- The module can work properly in the following OSs and the methods for installing other kernel versions are similar:
  - CentOS 6.8 (kernel version 2.6.32)
  - SUSE 11 SP3 (kernel version 3.0.76)
  - CentOS 7 and CentOS 7.2 (kernel version 3.10.0)
  - Ubuntu 16.04.3 (kernel version 4.4.0)
  - Ubuntu 18.04 (kernel version 4.15.0)
  - Ubuntu 20.04 (Kernel version 5.4.0)
  - OpenSUSE 42.2 (kernel version 4.4.36)
  - Debian 8.2.0 (kernel version 3.16.0)

Prerequisites

- The development environment for compiling the module must be the same as that of the current kernel. For example, if the kernel version is kernel-3.10.0-693.11.1.el7, the kernel development package version must be kernel-devel-3.10.0-693.11.1.el7.

- Servers can access OS repositories.
- Users other than **root** must have sudo permissions.

Procedure

- In the following operations, the Linux kernel version is 3.0 or later.

1. Prepare the compilation environment.

   NOTE

   - During the installation, download the required module development package from the Internet if it cannot be found in the source.
   - If the kernel development package (kernel-devel) cannot be obtained, contact the image provider.

   The following are operations for compiling the module in different Linux OSs. Perform appropriate operations.

   – CentOS

     i. Run the following command to install the GCC:

        sudo yum install gcc

     ii. Run the following command to install the make tool:

        sudo yum install make

     iii. Run the following command to install the module development package (the package header and module library must have the same version as the kernel):

        sudo yum install kernel-devel-`uname -r`

        NOTE

        - During the installation, download the required module development package from the following address if it cannot be found in the source:

          https://mirror.netcologne.de/oracle-linux-repos/ol7_latest/getPackage/

          For example, to install 3.10.0-693.11.1.el7.x86_64, run the following command:

          rpm -ivh kernel-devel-3.10.0-693.11.1.el7.x86_64.rpm
        - If the kernel development package (kernel-devel) cannot be obtained, contact the image provider.

   – Ubuntu and Debian

     i. Run the following command to install the GCC:

        sudo apt-get install gcc

     ii. Run the following command to install the make tool:

        sudo apt-get install make

     iii. Run the following command to install the module development package (the package header and module library must have the same version as the kernel):

        sudo apt-get install linux-headers-`uname -r`

   – SUSE

     i. Run the following command to install the GCC:

        sudo zypper install gcc

ii. Run the following command to install the make tool:

sudo zypper install make

iii. Run the following command to install the module development package (the package header and module library must have the same version as the kernel):

sudo zypper install kernel-default-devel

2. Compile the module.

a. Use the git tool and run the following command to download the module source code:

git clone https://github.com/Huawei/TCP_option_address.git

NOTE

If the git tool is not installed, download the module source code from the following link:

https://github.com/Huawei/TCP_option_address

b. Run the following commands to enter the source code directory and compile the module:

cd src

make

If no warning or error code is prompted, the compilation was successful. Verify that the **toa.ko** file was generated in the current directory.

NOTE

- If error message "config_retpoline=y but not supported by the compiler, Compiler update recommended" is displayed, the GCC version is outdated. Upgrade the GCC to a later version.
- If the kernel version has been manually upgraded in the standard Linux distribution and the TOA module fails to be compiled, you are advised to upgrade the GCC to a later version.

3. Load the module.

a. Run the following command to load the module:

sudo insmod toa.ko

b. Run the following command to check the module loading and to view the kernel output information:

dmesg | grep TOA

If **TOA: toa loaded** is displayed in the command output, the module has been loaded.

NOTE

After compiling the CoreOS module in the container, copy it to the host system and then load it. The container for compiling the module shares the **/lib/modules** directory with the host system, so you can copy the module in the container to this directory, allowing the host system to use it.

4. Set the script to enable it to automatically load the module.

To make the module take effect when the system starts, add the command for loading the module to your startup script.

You can use either of the following methods to automatically load the module:

  – Add the command for loading the module to a customized startup script as required.

  – Perform the following operations to configure a startup script:

    i. Create the **toa.modules** file in the **/etc/sysconfig/modules/** directory. This file contains the module loading script.

       The following is an example of the content in the **toa.modules** file.

       ```
       #!/bin/sh
       /sbin/modinfo -F filename /root/toa/toa.ko > /dev/null 2>&1
       if [ $? -eq 0 ]; then
       /sbin/insmod /root/toa/toa.ko
       fi
       ```

       /root/toa/toa.ko is the path of the module file. You need to replace it with their actual path.

    ii. Run the following command to add execution permissions for the **toa.modules** startup script:

       sudo chmod +x /etc/sysconfig/modules/toa.modules

       NOTE

       If the kernel is upgraded, the current module will no longer match. Compile the module again.

5. Install the module on multiple servers.

   To load the module in the same OS, copy the **toa.ko** file to servers where the module is to be loaded and then perform the operations in **3**.

   After the module is successfully loaded, applications can obtain the real IP address contained in the request.

   NOTE

   The OS of the server must have the same version as the kernel.

6. Verify the module.

   After the module is successfully installed, the source address can be directly obtained. The following provides an example for verification.

   Run the following command to start SimpleHTTPServer on the backend server where Python is installed:

   python -m SimpleHTTPServer *port*

   The value of **port** must be the same as the port configured for the backend server, and the default value is **80**.

   Access the IP address of the load balancer from a client. Access logs on the server are as follows:

   ```
   192.168.0.90 - - [06/Aug/2020 14:24:21] "GET / HTTP/1.1" 200 –
   ```

   NOTE

   192.168.0.90 indicates the client IP address that is obtained by the backend server.

- **In the following operations, the Linux kernel version is 2.6.32.**

NOTE

The TOA plug-in supports the OSs (CentOS 6.8 image) with a kernel of 2.6.32-xx. Perform the following steps to configure the module:

1. Obtain the kernel source code package **Linux-2.6.32-220.23.1.el6.x86_64.rs.src.tar.gz** containing the module from the following link:

   http://kb.linuxvirtualserver.org/images/3/34/Linux-2.6.32-220.23.1.el6.x86_64.rs.src.tar.gz

2. Decompress the kernel source code package.
3. Modify compilation parameters.
   a. Open the **linux-2.6.32-220.23.1.el6.x86_64.rs** folder.
   b. Edit the **net/toa/toa.h** file.

      Change the value of **#define TCPOPT_TOA200** to **#define TCPOPT_TOA254**.

   c. On the shell page, run the following commands:

      sed -i 's/CONFIG_IPV6=m/CONFIG_IPV6=y/g' .config

      echo -e '\n# toa\nCONFIG_TOA=m' >> .config

      After the configuration, the IPv6 module is compiled into the kernel. TOA is compiled into a separate module and can be independently started and stopped.

   d. Edit **Makefile**.

      You can add a description to the end of **EXTRAVERSION =**. This description will be displayed in **uname -r**, for example, **-toa**.

4. Run the following command to compile the software package:

   make -j *n*

   NOTE

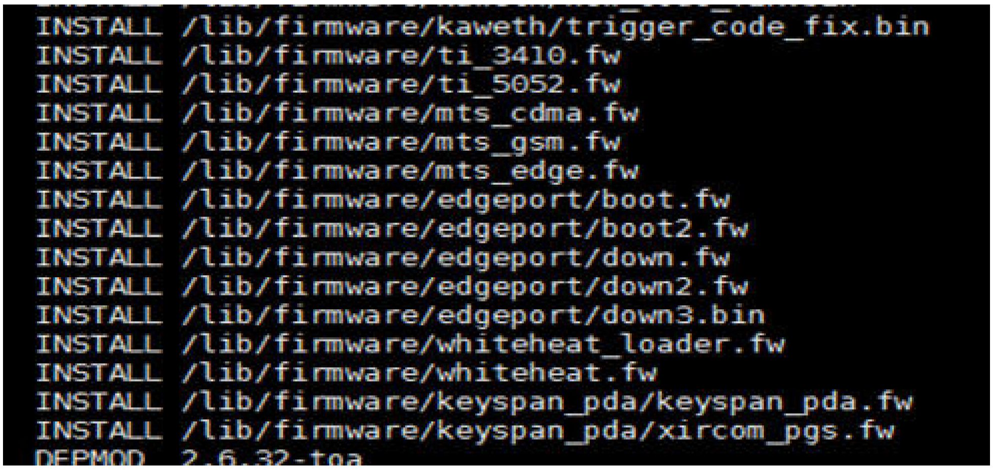
   n indicates the number of vCPUs. For example, if there are four vCPUs, *n* must be set to *4*.

5. Run the following command to install the module:

   make modules_install

   The following information is displayed.

Figure 4-1 Installing the module



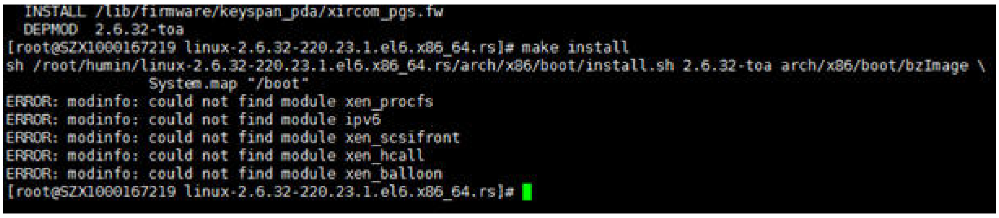
```
  INSTALL /lib/firmware/kaweth/trigger_code_fix.bin
  INSTALL /lib/firmware/ti_3410.fw
  INSTALL /lib/firmware/ti_5052.fw
  INSTALL /lib/firmware/mts_cdma.fw
  INSTALL /lib/firmware/mts_gsm.fw
  INSTALL /lib/firmware/mts_edge.fw
  INSTALL /lib/firmware/edgeport/boot.fw
  INSTALL /lib/firmware/edgeport/boot2.fw
  INSTALL /lib/firmware/edgeport/down.fw
  INSTALL /lib/firmware/edgeport/down2.fw
  INSTALL /lib/firmware/edgeport/down3.bin
  INSTALL /lib/firmware/whiteheat_loader.fw
  INSTALL /lib/firmware/whiteheat.fw
  INSTALL /lib/firmware/keyspan_pda/keyspan_pda.fw
  INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
  DEPMOD  2.6.32-toa
```

6. Run the following command to install the kernel:

   make install

   The following information is displayed.

   Figure 4-2 Installing the kernel

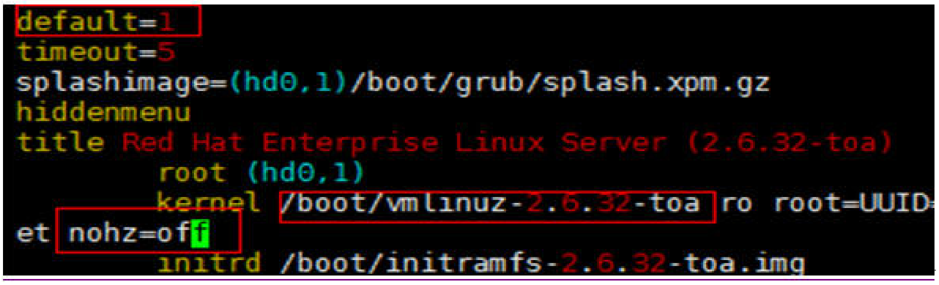
```
  INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
  DEPMOD  2.6.32-toa
[root@SZX1000167219 linux-2.6.32-220.23.1.el6.x86_64.rs]# make install
sh /root/humin/linux-2.6.32-220.23.1.el6.x86_64.rs/arch/x86/boot/install.sh 2.6.32-toa arch/x86/boot/bzImage \
                System.map "/boot"
ERROR: modinfo: could not find module xen_procfs
ERROR: modinfo: could not find module ipv6
ERROR: modinfo: could not find module xen_scsifront
ERROR: modinfo: could not find module xen_hcall
ERROR: modinfo: could not find module xen_balloon
[root@SZX1000167219 linux-2.6.32-220.23.1.el6.x86_64.rs]#
```

7. Open the **/boot/grub/grub.conf** file and configure the kernel to start up when the system starts.

   a. Change the default startup kernel from the first kernel to the zeroth kernel by changing **default=1** to **default=0**.

   b. Add the **nohz=off** parameter to the end of the line containing the **vmlinuz-2.6.32-toa** kernel. If **nohz** is not disabled, the CPU0 utilization may be high and overload the kernel.

   Figure 4-3 Configuration file

```
default=1
timeout=5
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-toa)
        root (hd0,1)
        kernel /boot/vmlinuz-2.6.32-toa ro root=UUID
et nohz=off
        initrd /boot/initramfs-2.6.32-toa.img
```

   c. Save the modification and exit. Restart the OS.

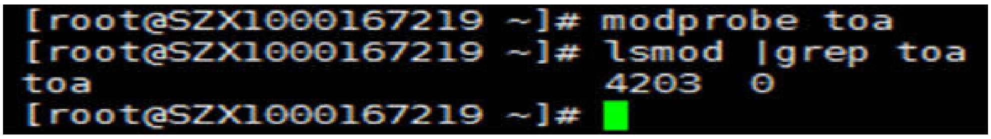
   During the restart, the system will load the **vmlinuz-2.6.32-toa** kernel.

8. After the restart, run the following command to load the module:

   modprobe toa

   Add the **modprobe toa** command to both the startup script and the system scheduled monitoring script.

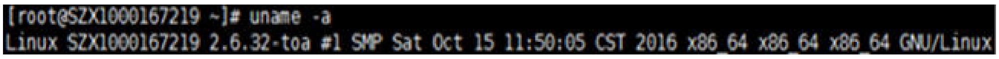
   Figure 4-4 Adding the **modprobe toa** command

```
[root@SZX1000167219 ~]# modprobe toa
[root@SZX1000167219 ~]# lsmod |grep toa
toa                     4203  0
[root@SZX1000167219 ~]#
```

   After the module is loaded, query the kernel information.

   Figure 4-5 Querying the kernel

```
[root@SZX1000167219 ~]# uname -a
Linux SZX1000167219 2.6.32-toa #1 SMP Sat Oct 15 11:50:05 CST 2016 x86_64 x86_64 x86_64 GNU/Linux
```

9. Verify the module.

   After the module is installed, the source IP address can be directly obtained. The following provides an example for verification.

   Run the following command to start SimpleHTTPServer on the backend server where Python is installed:

   python -m SimpleHTTPServer *port*

   The value of **port** must be the same as the port configured for the backend server, and the default value is **80**.

   Access the IP address of the load balancer from a client. Access logs on the server are as follows:

   ```
   192.168.0.90 - - [06/Aug/2020 14:24:21] "GET / HTTP/1.1" 200 –
   ```

   NOTE

   192.168.0.90 indicates the client's source IP address that is obtained by the backend server.